Exploiting structure defined by data in machine learning: some new analyses

by

Guy Lever

Department of Computer Science
University College London
Gower Street
United Kingdom

Email: g.lever@cs.ucl.ac.uk

THESIS SUBMITTED FOR THE DEGREE OF PHD IN COMPUTER SCIENCE

Statement of originality

I, Guy Lever, confirm that the work presented here is my own. When results have been derived from other sources this is indicated in the text.

Acknowledgements

I would like to thank Mark Herbster for being an excellent and dedicated supervisor for the last three years. Mark has been extremely generous with his time and energy. Throughout at least the first year of my PhD I received significantly more contact with Mark than is typical for a PhD student, which provided me with invaluable insights into the research process, much of it by sacrificing his own free time. It is extremely rare to have such a generous and enthusiastic supervisor. It is largely thanks to this great supervision that I quickly became able to produce this research. Mark is also a really creative researcher and working with him has been inspirational and a lot of fun.

I would also like to thank John Shawe-Taylor and Massimiliano Pontil who have both acted as my second supervisor. Both have done more than is typical for that role and I have benefited immensely from having such a lot of contact with great researchers. I have been encouraged by all of my supervisors to have fun and I have certainly done that.

I would also like to thank everyone else that I have worked with over the past 3 years especially François Laviolette, Matthew Higgs, Steffen Grünewälder, Pascal Germain, Zakria Hussain, Tom Diethe, Andreas Argyriou, Malcom Reynolds and Jean Morales. Its been a privilege to be able to work with the group I thank everyone else at CSML, especially all of the staff, for making it such a stimulating research group.

I'd like to thank people who've organised workshops from which I have benefited: Kristiaan Pelckmans, Thomas Gärtner, François Laviolette, Matthew Higgs, Steffen Grünewälder and Jean-Yves Audibert.

It was my good fortune to sit next to Zakria Hussain and Tom Diethe who have been my first port of call for many questions and who have always been generous with their time in answering them.

Finally I have been fully supported throughout this PhD by an EPSRC grant without which none of this would have been possible.

To my parents and dimension stories

Abstract

This thesis offers some new analyses and presents some new methods for learning in the context of exploiting structure defined by data – for example, when a data distribution has a submanifold support, exhibits cluster structure or exists as an object such as a graph.

1. We present a new PAC-Bayes analysis of learning in this context, which is sharp and in some ways presents a better solution than uniform convergence methods. The PAC-Bayes prior over a hypothesis class is defined in terms of the unknown true risk and smoothness of hypotheses w.r.t. the unknown data-generating distribution. The analysis is "localized" in the sense that complexity of the model enters not as the complexity of an entire hypothesis class, but focused on functions of ultimate interest. Such bounds are derived for various algorithms including SVMs.

2. We consider an idea similar to the p-norm Perceptron for building classifiers on graphs. We define p-norms on the space of functions over graph vertices and consider interpolation using the p-norm as a smoothness measure. The method exploits cluster structure and attains a mistake bound logarithmic in the diameter, compared to a linear lower bound for standard methods.

3. Rademacher complexity is related to cluster structure in data, quantifying the notion that when data clusters we can learn well with fewer examples. In particular we relate transductive learning to cluster structure in the empirical resistance metric.

4. Typical methods for learning over a graph do not scale well in the number of data points – often a graph Laplacian must be inverted which becomes computationally intractable for large data sets. We present online algorithms which, by simplifying the graph in principled way, are able to exploit the structure while remaining computationally tractable for large datasets. We prove state-of-the-art performance guarantees.

Contents

List of Tables

List of Figures

Introduction and Motivation

This thesis is about understanding structure and geometry defined by data and its role in supervised learning processes. It is a recent development, particularly in the domain of semi-supervised learning, that the learning process should ideally exploit the structure defined by the learning problem that is revealed in a data sample.

In a learning process data is typically represented in a (high dimensional) "ambient" metric space. The structure of this space can be arbitrary and inappropriate; given a set of images each represented as a vector of pixel values, for example, it is unlikely that the Euclidean metric on these vectors defines an appropriate distance between two images. The data distribution might have support a low dimensional submanifold or some other highly structured geometry such as a collection of clusters and a more appropriate metric space for the data is often defined by the data distribution – the geodesic distance on a submanifold support, for instance. This *intrinsic* structure defined by data is often very different to that captured by the geometry of the ambient space.

A second (and, as we will see, sometimes related) example of the need to understand structure defined by data arises wherever data naturally inhabit objects such as graphs, strings or networks, the structure of which is perhaps poorly understood from a learning theory perspective. This situation is increasingly common in practical applications of machine learning. For example, biological and chemical data such as gene networks or drug molecules, web data or social network data are typically naturally represented as a graph.

A working hypothesis of recent research, and this thesis, is that the ease with which a task can be learnt is dependent on the intrinsic structure defined by the data, and learning methods should be tuned to operate with and exploit this structure. This assumption appears to hold in reality as evidenced by the practical success of semi-supervised machine learning methods in particular and, after all, biological learners routinely demonstrate that it is possible to learn extremely effectively and efficiently (i.e. with few examples) in a setting whose intrinsic structure is embedded in an ambient space of seemingly intractable dimensionality.

This thesis offers some new analyses and presents some new methods for learning in this context of exploiting structure defined by data. The aim is to improve upon classical analyses, enhance the understanding of learning in this context and motivate improved learning methods. Specific contributions

are outlined in Section 1.4.2.

The goal of learning theory is to describe the learning process, but there still exists a large gap between the performance demonstrated by modern machine learning algorithms and the ability of a mathematical theory to explain this performance. Understanding the role of data-defined structure perhaps plays an important role in understanding the learning process and exploiting this structure is perhaps part of the key to learning well.

Chapter 1

Preliminaries

1.1 Learning theory background

1.1.1 Typical settings and analytical frameworks

The problem of inferring a function from finite samples is fundamental to learning. Learning theory is the mathematical theory which explains such a learning process. A common setting (and focus of this thesis) is that of *supervised learning*: a learner is given access to a *sample* of labelled *examples* $\mathcal{S} := \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), ...(\boldsymbol{x}_m, y_m)\}$ from a product space $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, and must infer from this sample a function $f : \mathcal{X} \to \mathcal{Y}$ which explains the data and can be used to make future predictions given new unlabelled instances from $\mathcal{X}$. The classical tasks of regression and classification fall within this setting, and this thesis will be concerned mainly with the task of binary classification, where $\mathcal{Y} = \{-1, 1\}$. Given some class of *hypotheses* $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, if we view a *learning algorithm* as a function $A : \cup_{m \in \mathbb{N}} \mathcal{Z}^m \to \mathcal{H}$ which takes a training sample (of size m) and outputs a hypothesis, then a goal of learning theory is to explain properties of A. We are particularly interested in providing certain formal guarantees on the performance of the hypothesis $A(\mathcal{S})$ produced by A. Such analyses provide some explanation of the learning process and such insights are used to motivate new learning methodologies, ultimately used to build learning machines.

The statistical learning theory framework

An analytical framework that has reached substantial maturity is *statistical learning theory*, pioneered by Vapnik and Chervonenkis (Vapnik and Chervonenkis, 1971; Vapnik, 1982). Here, it is typically assumed that data are drawn identically and independently according to a joint distribution D over the space $\mathcal{X} \times \mathcal{Y}$ of labelled inputs. For a class of hypotheses $\mathcal{H} \subseteq \mathcal{D}^{\mathcal{X}}$, where the *decision space* $\mathcal{D}$ may or may not correspond to $\mathcal{Y}$, we consider a *loss* function $\ell : \mathcal{D} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$, which, for any $h \in \mathcal{H}$, captures the degree of mismatch between $h(\boldsymbol{x})$ and y on any labelled instance $(\boldsymbol{x}, y) \in \mathcal{Z}$. To any function $h : \mathcal{X} \to \mathcal{D}$ we can then assign a measure of its performance on a randomly chosen labelled instance

drawn from D, called the *generalization ability* or *risk* of h,

$$\mathrm{risk}^{\ell}(h) \quad := \mathbb{E}_{(X,Y)\sim D}\left[\ell(h(X),Y)\right]. \tag{1.1}$$

In particular when ℓ is the $0-1$ loss of binary classification problem,

$$\ell_{0-1}(y,y') := \begin{cases} 0 \text{ if } y = y' \\ 1 \text{ if } y \neq y' \end{cases}, \tag{1.2}$$

then we denote the associated risk by risk, omitting the superscript ℓ.

A problem is then to learn, from $\mathcal{S}$, a function $h : \mathcal{X} \to \mathcal{D}$ with low risk. The function with least possible risk is called the *Bayes function* $f^* \in \mathrm{argmin}_{f\in\mathcal{Y}^{\mathcal{X}}} \mathrm{risk}^{\ell}(f)$, and the corresponding smallest possible risk the *Bayes risk*. We say that an algorithm A is consistent[1] if $\mathrm{risk}^{\ell}(A(\mathcal{S})) \to \mathrm{risk}^{\ell}(f^*)$ as $m \to \infty$, with various modes of convergence determining various modes of consistency[2].

In the statistical learning theory setting we seek often to provide a *risk bound*, which is an upper bound on the risk which holds with high probability over the i.i.d. draw of the training sample $\mathcal{S}$ from D^m,

$$\mathbb{P}_{\mathcal{S}}\left(\mathrm{risk}^{\ell}(A(\mathcal{S})) \leq F\left(A,\mathcal{H},\mathcal{S},\delta\right)\right) \geq 1-\delta. \tag{1.3}$$

for some function F. We can define the empirical counterpart to (1.1) on a labelled sample $\mathcal{S}$,

$$\widehat{\mathrm{risk}}^{\ell}_{\mathcal{S}}(h) := \frac{1}{|\mathcal{S}|}\sum_{(X,Y)\in\mathcal{S}} \ell(h(X),Y), \tag{1.4}$$

and a common example of a risk bound is a deviation inequality, bounding the deviation between the true risk and the empirical risk observed on the sample such as,

$$\mathbb{P}_{\mathcal{S}}\left(\mathrm{risk}^{\ell}(A(\mathcal{S})) \leq \widehat{\mathrm{risk}}^{\ell}_{\mathcal{S}}(A(\mathcal{S})) + F'\left(A,\mathcal{H},\mathcal{S},\delta\right)\right) \geq 1-\delta. \tag{1.5}$$

A key questions is, for example, establishing the (optimal) rate at which the quantity $F'\left(A,\mathcal{H},\mathcal{S},\delta\right)$ decays in $|\mathcal{S}|$.

Since the hypothesis $A(\mathcal{S})$ is a-priori unknown, one route to obtain a bound such as (1.5) is to establish convergence uniformly for all functions in an a-priori fixed hypothesis class[3] $\mathcal{H}$ (Vapnik and Chervonenkis, 1971). In such settings certain measures of *complexity*, or the expressive power, of a function class emerge as key quantities required to quantify the learning process, the most foundational of which is the *VC dimension*[4] leading to VC theory. In the uniform convergence approach much work has

[1] This word is also used to describe several related concepts.

[2] It is a surprising fact that universally consistent learning rules exist, that is, algorithms which are consistent for any distribution D over $\mathcal{X}\times\mathcal{Y}$ (Devroye et al., 1996).

[3] A class $\mathcal{F} \subset \mathbb{R}^{\mathcal{Z}}$ for which

$$\sup_{f\in\mathcal{F}} |\mathbb{E}_{Z\sim D} f(Z) - \frac{1}{m}\sum_{i=1}^{m} f(\boldsymbol{z}_i)| \to 0 \tag{1.6}$$

D^m-almost surely and such that convergence is uniform over all probability measures D on $\mathcal{Z}$ is called uniformly Glivenko-Cantelli.

[4] The VC dimension of a class $\mathcal{H} \subset \{-1,1\}^{\mathcal{X}}$ is the cardinality of the largest set *shattered* by $\mathcal{H}$, where a set $\{x_1,,...x_n\}$ of size n is shattered by $\mathcal{H}$ if for each $\boldsymbol{y} \in \{-1,1\}^n$ there exists $h \in \mathcal{H}$ such that $h(x_i) = y_i$.

been completed on establishing conditions under which the D^m-almost sure convergence of $\mathrm{risk}(A(\mathcal{S}))$ to $\min_{h\in\mathcal{H}} \mathrm{risk}(h)$ as $m \to \infty$ is attained for certain learning rules in terms of general abstract properties of $\mathcal{H}$ leading to a simple and powerful theory.

The related field of PAC (probably approximately correct) learning (Valiant, 1984) has similar goals and is further concerned with the efficiency of the learning method (or algorithm). These theories are a mature field of research and we refer to reader to Bousquet et al. (2003a); Vapnik (1998); Devroye et al. (1996); Boucheron et al. (2005); Cucker and Smale (2002); Anthony and Bartlett (1999) for a comprehensive overview of the subject and recent research.

The online learning framework

An alternative learning framework that receives significant attention is the adversarial *online learning* setting, in which learning proceeds sequentially (Littlestone, 1988; Vovk, 1990; Cesa-Bianchi and Lugosi, 2006). This setting provides a game-theoretical foundation for learning, with fewer assumptions than is typical in statistical analyses, providing a different setting in which to compare learning methods. In this setting the training sample $\mathcal{S} := \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), ...(\boldsymbol{x}_m, y_m)\}$ is revealed sequentially, and at each trial t a learner is provided with $\boldsymbol{x}_t$ and must make a prediction $h_t(\boldsymbol{x}_t)$ for the label y_t, after which the true label is revealed, and the learner modifies its hypothesis for the next trial. The goal is typically to minimise the cumulative loss

$$M = \sum_{t\leq m} \ell(h_t(\boldsymbol{x}_t), y_t),$$

(which corresponds to the number of mistakes if ℓ is the $0-1$ loss of the binary classification problem). Note that no assumptions are made on the distribution of examples and in particular nature can be viewed as an adversary, so that any mistake bound performance guarantees hold for every conceivable realization of trial sequence. We refer the reader to Cesa-Bianchi and Lugosi (2006) for an overview of this learning model.

1.2 Methods

The no free lunch theorem of Wolpert (1996) establishes the fact that if all problems are equally likely then the performance (measured by their true risk, given the problem and training set) of all learning algorithms are equal in expectation. Thus given any learning algorithm there is a data distribution on which it performs badly (attains high true risk in general) and there is no universally optimal learning method. In order to learn well assumptions (or prior knowledge) about the nature of the data distribution (*bias*) must be introduced, usually in the form of a preference for "simple" functions, e.g. such that there exists a simple relation between inputs and outputs or such that the function has a "small description length" for instance (simplicity is not a universal notion). This is usually realized by placing restrictions on the class of functions to be learnt. We now discuss some key learning principles relevant to this thesis.

1.2.1 Empirical risk minimization

A foundational strategy for obtaining a hypothesis from data is that of empirical risk minimisation (ERM): find the hypothesis that minimizes (1.4) over the training sample. If we allow $\mathcal{H}$ to be the set of all functions (or, e.g., all continuous functions) mapping $\mathcal{X} \to \mathcal{Y}$ then the ERM problem $\operatorname{argmin}_{h\in\mathcal{H}} \widehat{\operatorname{risk}}^{\ell}_{\mathcal{S}}(h)$ is generally ill-posed[5] since there is no unique solution. The ERM hypothesis is unstable (in the sense that small changes in training sample can cause large changes in the learned hypothesis) due to its tendency to significantly overfit the training data. Thus, following Tikhonov and Arsenin (1977), the class $\mathcal{H}$ is generally restricted in some way and additional regularization terms are added to the minimization problem in order to obtain a well-posed problem and improve the stability of the ERM solution,

$$h^* := \operatorname*{argmin}_{\{h\in\mathcal{H}\}} \widehat{\operatorname{risk}}^{\ell}_{\mathcal{S}}(h) + \operatorname{reg}(h),$$

where $\operatorname{reg} : \mathcal{H} \to \mathbb{R}$ is a regularization term. Solving the ERM problem with the binary classification risk is generally NP-hard even for simple hypothesis classes, so convex surrogates (along with convex regularization) are often used in practice.

Both capacity control and regularization are generally realised by applying a *smoothness assumption*, so that $||x - x'|| \approx 0 \implies f(x) \approx f(x')$, for example. This is achieved by, for example, penalising large derivatives of f. A principled way of controlling the capacity of function classes is that of structural risk minimization (e.g. Vapnik, 1998).

1.2.2 Kernel methods and the RKHS formalism

A framework which is now ubiquitous in the machine learning community is that of kernel methods (e.g. Shawe-Taylor and Cristianini, 2004; Schölkopf and Smola, 2002). This refers to choosing as a hypothesis class a *reproducing kernel Hilbert space* (RKHS) of functions, which possess some extremely desirable qualities (Aronszajn, 1950).

Given any symmetric positive-definite kernel[6] K on a set $\mathcal{X}$, consider the pre-Hilbert space $\widehat{\mathcal{H}}_K = \operatorname{span}\{K(\boldsymbol{x}, \cdot) : \boldsymbol{x} \in \mathcal{X}\}$ of functions mapping $\mathcal{X}$ into $\mathbb{R}$, consisting of all finite linear combinations of the *features* $\{K(\boldsymbol{x}, \cdot)\}_{\boldsymbol{x}\in\mathcal{X}}$. The inner product in $\widehat{\mathcal{H}}_K$ is defined by $\langle K(\boldsymbol{x}, \cdot), K(\boldsymbol{x}', \cdot)\rangle_K := K(\boldsymbol{x}, \boldsymbol{x}')$, for all $\boldsymbol{x}$, $\boldsymbol{x}' \in \mathcal{X}$. The *reproducing property* $h(\boldsymbol{x}) = \langle h, K(\boldsymbol{x}, \cdot)\rangle_K$ is immediate from the definition, and provides the means of evaluating hypotheses on sample points (the kernel K is called the *representer of evaluation*). The RKHS $\mathcal{H}_K$ is formed by completing $\widehat{\mathcal{H}}_K$ with respect to the norm $||\cdot||_K$, $\mathcal{H}_K = \overline{\operatorname{span}\{K(\boldsymbol{x}, \cdot) : \boldsymbol{x} \in \mathcal{X}\}}$. It can be easily seen that the topological operation of completion defines a space of functions over $\mathcal{X}$ since, by virtue of the reproducing property, the limits of Cauchy sequences

[5] A problem is well-posed, in the sense of Hadamard, If a unique solution exists and it depends continuously on the data.

[6] A positive-definite kernel on $\mathcal{X}$ is a symmetric continuous function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that, for any finite collection of points $\{\boldsymbol{x}_i\}_{i=1}^n$ and any constants $\{c_i\}_{i=1}^n$, $\sum_{i,j=1}^n c_i c_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) > 0$.

in $\widehat{\mathcal{H}}_K$ correspond to elements which are well-defined functions pointwise,

$$\begin{aligned} |h_i(x) - h_j(x)| &= \langle h_i - h_j, K(x, \cdot)\rangle_K \\ &\leq ||h_i - h_j||_K K(x, x), \end{aligned} \tag{1.7}$$

thus if $\{h_i\}$ is Cauchy in $\widehat{\mathcal{H}}_K$, the point evaluations are also Cauchy in $\mathbb{R}$ and have a well-defined limit.

The RKHS is so useful in machine learning because of the so-called *kernel trick* first realized in Aizerman et al. (1964). The RKHS is typically of very high (often infinite) dimensionality whose dimensions correspond to correlations between the dimensions of the original input space. Thus, by first mapping inputs into "feature space", $\boldsymbol{x} \to K(\boldsymbol{x}, \cdot)$, and learning a linear function in feature space, $f(\boldsymbol{x}) := \langle h, K(\boldsymbol{x}, \cdot)\rangle_K$ for some $h \in \mathcal{H}_K$, the learned classifier generally corresponds to a highly non-linear function in the original input space, vastly increasing the discriminative power of the functions attainable by the linear algorithm. The key observation is the fact that to produce predictions on new inputs many classical algorithms (such as the Support Vector Machine (e.g. Cristianini and Shawe-Taylor, 2000) or Perceptron (Rosenblatt, 1958; Minsky and Papert, 1969; Novikoff, 1963)) require only inner products between examples to be calculated and, since the RKHS inner product between two features is equal to the kernel evaluation at the corresponding points, inner products can be evaluated without paying the computational cost of operating directly in a potentially infinite-dimensional space. Thus many linear algorithms are therefore "kernelizable" enabling the learning of highly non-linear functions with the computation ease of linear methods. The various *representer theorems* (e.g. Wahba, 1990) imply that, given a set of points $\mathcal{S} = \{\boldsymbol{x}_1, ...\boldsymbol{x}_m\}$ drawn from $\mathcal{X}$, the solution to many (kernelized) classical algorithms (the SVM for example) is a hypothesis which has an expansion in the features of the training sample $h = \sum_{i=1}^m \alpha_i K(\boldsymbol{x}_i, \cdot)$, with $\boldsymbol{\alpha} \in \mathbb{R}^m$.

The RKHS norm $||h||_K$ often has an interpretation in terms of a measure of complexity such as of how variable the function h is (in the case of the Gaussian kernel the RKHS norm captures the smoothness of all derivatives of a function, for example). The following sheds some light on this relationship.

Kernels, regularization and smoothness

Consider the function space $\mathcal{L}^2(\mathcal{X}, \Sigma, \nu)$ of square integrable functions on the measure space $(\mathcal{X}, \Sigma, \nu)$ equipped with the inner product $\langle f, g\rangle_{\mathcal{L}^2} := \int_{\mathcal{X}} f(x)g(x)\nu(\mathrm{d}x)$. Associated with a positive-definite kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is the linear integral operator $A_K : \mathcal{L}^2(\mathcal{X}, \Sigma, \nu) \to \mathcal{L}^2(\mathcal{X}, \Sigma, \nu)$ defined by,

$$A_K f(x) := \int_{\mathcal{X}} K(x', x) f(x')\nu(\mathrm{d}x'). \tag{1.8}$$

The function A_K is a compact self-adjoint operator on $\mathcal{L}^2(\mathcal{X}, \Sigma, \nu)$ and, by the spectral theorem for compact self-adjoint operators on a separable Hilbert space, it provides a countable basis $\{\phi_i\}$ for $\mathcal{L}^2(\mathcal{X}, \Sigma, \nu)$ consisting of the orthonormal eigenfunctions of A_K, i.e. such that,

$$A_K \phi_i := \lambda_i \phi_i,$$

for corresponding eigenvalues $\{\lambda_i\}$. Mercer's theorem (Mercer, 1909) then provides the expansion,

$$K(x, x') = \sum_{i=1}^{\infty} \lambda_i \phi_i(x) \phi_i(x'). \tag{1.9}$$

Further, when $(\mathcal{X}, \Sigma, \nu)$ is a finite measure space, the operator A_K is of trace class, $\sum_i \lambda_i < \infty$, which follows from the continuity of the kernel.

In Appendix A we briefly sketch a theory (presented in Smola et al. (1998)) which unifies many key paradigms in learning theory, by showing that the apparently distinct approaches of kernel methods, certain regularization paradigms and encouraging smoothness over data are all aspects of a single framework: there is a natural correspondence between regularizing in $\mathcal{L}^2(\mathcal{X}, \Sigma, \nu)$ and controlling capacity by using the RKHS norm – the kernel corresponding to the *Green's function* of the $\mathcal{L}^2$ regularizer. It often turns out that common kernels are the Green's function of intuitively useful regularizers, and we will see examples of this duality later.

Wahba (1990) provides the following insightful further characterization of the RKHS norm:

Lemma 1.2.1. *(Wahba, 1990, Lemma 1.1.1) For any $h \in \mathcal{L}^2(\mathcal{X}, \Sigma, \nu)$ with $h = \sum_{i=1}^{\infty} h_i \phi_i$,*

$$h \in \mathcal{H}_K \Leftrightarrow \sum_{i=1}^{\infty} \frac{1}{\lambda_i} h_i^2 < \infty,$$

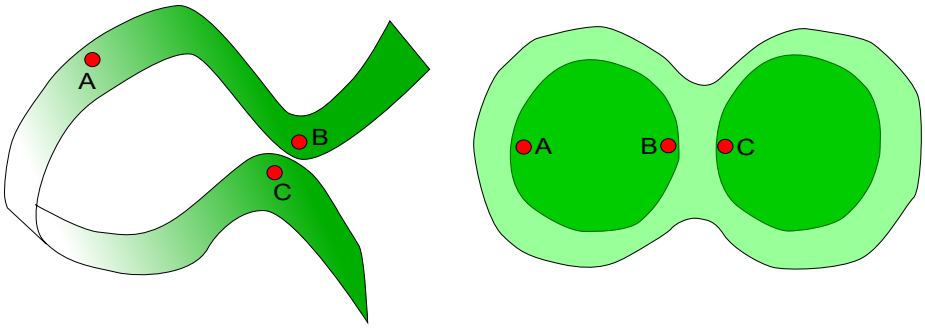
(where we define $\frac{0}{0} = 0$). Whenever $h \in \mathcal{H}_K$,

$$||h||_K^2 = \sum_{i=1}^{\infty} \frac{1}{\lambda_i} h_i^2.$$

1.3 Exploiting structure defined by data

As mentioned in the introduction the intrinsic geometry of data is often very different to that captured by the geometry of the ambient space, and we now present an informal overview of these ideas. We want to highlight the difference between two possible approaches to using geometry when learning from data; the first uses an ambient geometry of the representation space, and the second attempts to learn an intrinsic geometry defined by the data-generating distribution. If the ambient representation space is equipped with, for example, a metric we highlight that it will not in general provide an accurate means to measure similarity between data points. Figure 1.1 illustrates two possible data densities in ambient space ($\mathbb{R}^3$ and $\mathbb{R}^2$ respectively, with the Euclidean inner product) highlighting the possible mismatch between intrinsic and extrinsic geometry.

In Figure 1.1(a) data inhabits a submanifold of the ambient space, and the manifold geodesics are clearly not captured by the ambient Euclidean metric in $\mathbb{R}^3$. In Figure 1.1(b) a useful analogy is to view the data distribution as two dense blobs of some conductive medium (conducting, for example electricity or heat) separated by a high resistance bridge. In this analogy, if we consider the ease with which heat or electricity flows between points as a measure of similarity we see that the ambient geometry fails to provide a satisfactory measure of similarity. In both cases, if we want an intrinsic metric $d_I(\cdot, \cdot)$ to



(a) Support on a submanifold (b) Two clusters

Figure 1.1: Mismatch between intrinsic and extrinsic geometry

capture similarity between points then we intuitively want $d_I(A, B) \ll d_I(B, C)$. However in both cases a typical extrinsic metric $d_E(\cdot, \cdot)$ would satisfy $d_E(A, B) > d_E(B, C)$ and, in particular, a hyperplane classifier which separates points B and C would have a narrow margin w.r.t. the extrinsic geometry.

It is reasonable to conjecture (and a working hypothesis of this research) that the intrinsic structure of a task plays a key role in the learning process (simple structures are easier to learn and if we observe simple structure, and can see that a good classifier respects that structure then we should be able to be more confident in our analyses) and that therefore an accurate explanation of the learning process should relate to that structure and learning methods should seek to exploit it. A way of achieving this is to attempt to learn the intrinsic geometry of the data generating distribution from the sample and to exploit that structure, using something like an assumption that good hypotheses will be smooth with respect to the data-defined geometry, as has become standard in settings of semi-supervised and transductive learning.

As mentioned in the introduction, a further setting in which the need to understand and exploit data-defined structure is in the increasingly common applications of machine learning to domains where data naturally inhabit a structure such as a graph, as is typical in bioinformatics (Sharan and Ideker, 2006), chemoinformatics (Bonchev and Rouvray, 1991), social network analysis (Kumar et al., 2006), web data analysis (Washio and Motoda, 2003), as well as areas in which a graph is often used to model data such as computer vision (Harchaoui, 2007) and natural language processing (Collins and Duffy, 2001). A survey of some results pertaining to the structure of such graphs arising from data of the "information age" is presented in Chung and Lu (2006). Such objects are perhaps poorly understood from a learning theory perspective, and it is clear that both methods and analyses of learning in such domains should be tuned to the structure of such objects.

1.3.1 The role of graph theoretical methods in capturing data geometry

Principal definitions

We first outline the graph-theoretic notions that will be fundamental throughout this thesis. An *(undirected) graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a set of *vertices* $\mathcal{V} = \{v_1, \ldots, v_n\}$ and a set of *edges* $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ consisting of unordered pairs of vertex indices, so that $(i, j) \in \mathcal{E}$ is viewed as connecting vertices v_i and v_j. We denote $i \sim j$ whenever v_i and v_j are connected by an edge. Associated with each edge $(i, j) \in \mathcal{E}$ is a weight $A_{ij} > 0$ and $A_{ij} = 0$ if $(i, j) \notin \mathcal{E}$, so that $\boldsymbol{A}$ is the (weighted) symmetric *adjacency matrix*. We say that $\mathcal{G}$ is *unweighted* if $\boldsymbol{A} \in \{0, 1\}^{n \times n}$. Given any n-vertex graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, any function $f : \mathcal{V} \to \mathbb{R}$ can be identified with a vector in $\boldsymbol{f} \in \mathbb{R}^n$ whereby $f_i = f(v_i)$, hence we can identify the class of real-valued functions defined on $\mathcal{V}$, $\mathcal{F} = \mathbb{R}^{\mathcal{V}}$, with $\mathbb{R}^n$.

The *(combinatorial) Laplacian* $\boldsymbol{L}$ of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is the $n \times n$ matrix $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$, where $\boldsymbol{D}$ is the diagonal *degree matrix* such that $D_{ii} = \sum_j A_{ij}$. For any graph, $\boldsymbol{L}$ is positive semi-definite and therefore we can define a semi-inner product on $\mathbb{R}^n$; $\langle \boldsymbol{f}, \boldsymbol{g} \rangle_{\boldsymbol{L}} := \boldsymbol{f}^\top \boldsymbol{L} \boldsymbol{g}$. Note the following key identity,

$$\begin{aligned} \boldsymbol{f}^\top \boldsymbol{L} \boldsymbol{g} &= \sum_{ij} f_i (D_{ij} - A_{ij}) g_j \\ &= \sum_i f_i D_{ii} g_i - \sum_{ij} f_i A_{ij} g_j \\ &= \sum_{ij} f_i A_{ij} (g_i - g_j) \\ &= \sum_{(i,j) \in \mathcal{E}} (f_i - f_j)(g_i - g_j) A_{ij}. \end{aligned}$$

Given any real-valued function $\boldsymbol{f} : \mathcal{V} \to R$ this semi-inner product defines a natural *smoothness functional* $S_{\mathcal{G}}(\boldsymbol{f}) := \langle \boldsymbol{f}, \boldsymbol{f} \rangle_{\boldsymbol{L}}$ (Zhu et al., 2003a; Belkin et al., 2004) and note that,

$$\begin{aligned} S_{\mathcal{G}}(\boldsymbol{f}) &= \boldsymbol{f}^\top \boldsymbol{L} \boldsymbol{f} \\ &= \sum_{(i,j) \in \mathcal{E}} (f_i - f_j)^2 A_{ij}. \end{aligned} \tag{1.10}$$

Since $\langle \boldsymbol{f}, \boldsymbol{f} \rangle_{\boldsymbol{L}}$ is large if many adjacent vertices are labelled differently the smoothness functional indeed measures the smoothness of real-valued functions on $\mathcal{V}$, and is therefore a basic measure how well a function respects the geometry of the graph. When $\boldsymbol{f} \in \{-1, 1\}^n$, we say that a cut occurs on edge (i, j) whenever $f_i \neq f_j$ and (1.10) therefore measures the number of cuts.

A similar object, the *normalised Laplacian* is defined to be $\tilde{\boldsymbol{L}} = \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{L} \boldsymbol{D}^{-\frac{1}{2}} = \boldsymbol{I} - \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{A} \boldsymbol{D}^{-\frac{1}{2}}$ and has similar properties but this thesis will focus on use of the combinatorial Laplacian.

The graph Laplacian is essentially a discrete counterpart to the Laplace-Beltrami operator defined on a Riemannian manifold. A formalisation of these ideas which provides insight is presented in Zhou and Schölkopf (2005) which we here recall to provide clarity. Let $\mathcal{H}(\mathcal{V})$ denote the Hilbert space of real-valued functions on $\mathcal{V}$ equipped with the inner product $\langle f, g \rangle_{\mathcal{V}} := \sum_{v_i \in \mathcal{V}} f(v_i) g(v_i)$ and $\mathcal{H}(\mathcal{E})$ denote the Hilbert space of real-valued functions on $\mathcal{E}$ equipped with the inner product $\langle f', g' \rangle_{\mathcal{E}} :=$

$\sum_{(i,j)\in\mathcal{E}} f'(i,j)g'(i,j)$.

Definition The *gradient operator* on $\mathcal{G}$ is a map $\nabla_{\mathcal{G}} : \mathcal{H}(\mathcal{V}) \to \mathcal{H}(\mathcal{E})$ defined[7] by,

$$(\nabla_{\mathcal{G}} f)(i,j) := \sqrt{A_{ij}}(f(v_i) - f(v_j)).$$

We can further define the gradient at v_i by $\nabla_{\mathcal{G}} f(i) := \{(\nabla_{\mathcal{G}} f)(i,j) : j \sim i\}$. These definitions are consistent with the notion of a gradient as a vector field quantifying the rate of change of a function in each possible direction.

Definition The *divergence* on $\mathcal{G}$ is the operator $\mathrm{div}_{\mathcal{G}} : \mathcal{H}(\mathcal{E}) \to \mathcal{H}(\mathcal{V})$ adjoint to $\nabla_{\mathcal{G}}$ i.e. such that

$$\langle f', \nabla_{\mathcal{G}} g\rangle_{\mathcal{E}} = \langle \mathrm{div}_{\mathcal{G}} f', g\rangle_{\mathcal{V}}. \tag{1.11}$$

In analogy to the definition of the Laplace-Beltrami operator on a Riemannian manifold, we can now define a Laplace operator on $\mathcal{G}$:

Definition The *Laplace operator*, $L_{\mathcal{G}} : \mathcal{H}(\mathcal{V}) \to \mathcal{H}(\mathcal{V})$, on $\mathcal{G}$ is defined,

$$L_{\mathcal{G}} := \mathrm{div}_{\mathcal{G}} \nabla_{\mathcal{G}}.$$

The operator $L_{\mathcal{G}}$ is linear, since $\mathrm{div}_{\mathcal{G}}$ and $\nabla_{\mathcal{G}}$ are, and we have,

$$\begin{aligned} \langle f, L_{\mathcal{G}} g\rangle_{\mathcal{V}} &= \langle \nabla_{\mathcal{G}} f, \nabla_{\mathcal{G}} g\rangle_{\mathcal{E}} \\ &= \sum_{(i,j)\in\mathcal{E}} (f(v_i) - f(v_j))(g(v_i) - g(v_j))A_{ij}, \end{aligned} \tag{1.12}$$

and so by identifying $\mathcal{H}(\mathcal{V})$ with $\mathbb{R}^n$ equipped with the Euclidean inner product we can identify $L_{\mathcal{G}}$ precisely with the combinatorial Laplacian matrix $\boldsymbol{L}$ defined above. In particular we see that $\boldsymbol{f}^\top \boldsymbol{L}\boldsymbol{f} = ||\nabla_{\mathcal{G}} f||^2_{\mathcal{E}} = \sum_{v_i\in\mathcal{V}} ||\nabla_{\mathcal{G}} f(i)||^2$ (with the final norm denoting the standard Euclidean norm of $\mathbb{R}^{d(i)}$ where $d(i)$ is the degree of vertex v_i) so that the smoothness functional $S_{\mathcal{G}}(\cdot)$ measures smoothness on $\mathcal{H}(\mathcal{V})$ in a way analogous to the Dirichlet energy functional, a basic measure of the variability of a function, which is defined on functions over $\mathbb{R}^n$ by,

$$E(f) := \int_{\mathbb{R}^n} ||\boldsymbol{\nabla} f(\boldsymbol{x})||^2 \mathrm{d}\boldsymbol{x}. \tag{1.13}$$

This analogy will provide insights into the techniques used throughout this thesis.

We refer the reader to Diestel (2005); Bollobas (1998) for introductions to graph theory and to Chung (1997) for a focus on the graph Laplacian and the properties of its spectrum in particular.

Capturing the geometry defined by data with a graph

A graph is used to model data by representing objects as vertices and capturing similarity between objects with edges between vertices (and their associated weights). As mentioned above, in the contexts of

[7] The ordering (or orientation) which must be imposed on the edges in order to make this well-defined is any arbitrary ordering.

bioinformatics, chemoinformatics, social networks, web data-mining and many more application areas data are naturally represented as a graph. We now detail how a graph can be used to estimate, from a random sample a data distribution D with support in an arbitrary metric space $\mathcal{X}$. Given a sample $\mathcal{S} := \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), ...(\boldsymbol{x}_n, y_n)\}$ drawn from D^n we can form an n-vertex graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in which we associate each point $\boldsymbol{x}_i$ with a vertex v_i. We form an edge (i, j) whenever v_i and v_j satisfy some criterion of closeness with respect to a metric $d_{\mathcal{X}}(\cdot, \cdot)$ on the ambient space $\mathcal{X}$. A typical criterion would be that v_i, v_j are a pair of k-nearest neighbours or are both located in an ϵ ball w.r.t. $d_{\mathcal{X}}(\cdot, \cdot)$. Such a graph might be unweighted or we might set edge weights according to the distance in ambient space, such as $A_{ij} = e^{-d^2_{\mathcal{X}}(\boldsymbol{x}_i, \boldsymbol{x}_j)}$. This method is closely related to the field of kernel density estimation. The structure of such a graph will, in certain ways, be a discrete approximation to the structure of the underlying data distribution. For example, if the data has support on a submanifold of $\mathcal{X}$ a series of results (Hein et al., 2007; Hein, 2006) demonstrate that certain graph Laplacians converge to a generalized (distribution dependent) Laplace-Beltrami operator on the manifold support, and that the smoothness functional converges to a very natural and desirable measure of the smoothness of functions with respect to the data-generating distribution.

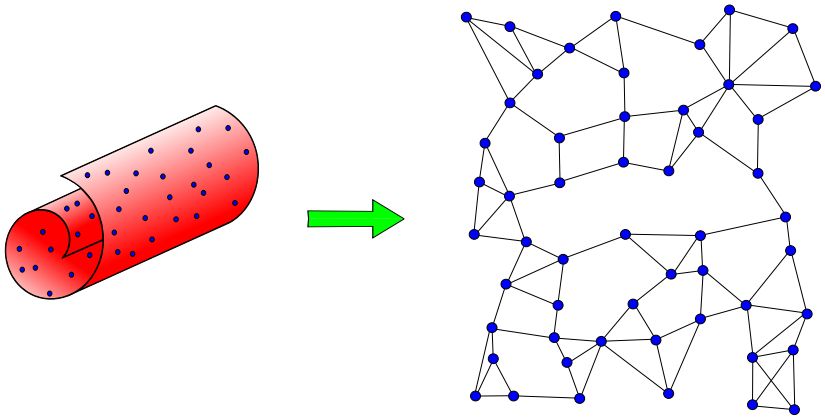


Figure 1.2: Vertices on the graph represent points sampled from the data distribution. Informally, distances between vertices capture the structure of the manifold support.

Semi-supervised learning

Semi-supervised learning refers to the setting in which the training sample consists of labelled and unlabelled data, $\mathcal{S} = \mathcal{S}_{\text{labelled}} \cup \mathcal{S}_{\text{unlabelled}}$, where $\mathcal{S}_{\text{labelled}} = \{(\boldsymbol{X}_{s_1}, Y_{s_1}), ...(\boldsymbol{X}_{s_m}, Y_{s_m})\}$ and $\mathcal{S}_{\text{unlabelled}} = \{\boldsymbol{X}_{s_{m+1}}, ...\boldsymbol{X}_{s_{m+u}}\}$, so that $n = m + u$ is the total amount of labelled and unlabelled points. The setting is common in practice since the labelling of data (labelling a scan of a patients liver

as 'healthy' or 'unhealthy', or the contents of an email as 'malicious' or 'harmless', for example) can be expensive and time consuming and require expert input whereas unlabelled data (the scans or emails) might be readily available in great number or essentially almost "free". Semi-supervised techniques attempt to exploit the additional information provided in this setting by the unlabelled data. Because of the potential of such approaches the setting has received significant attention for some time (Ratsaby and Venkatesh, 1995; Castelli and Cover, 1995, 1996; Blum and Mitchell, 1998; Nigam et al., 1998; Zhang and Oles, 2000; Chapelle et al., 2006).

Transductive learning

Transduction refers to the learning setting in which the unlabeled instances from the test set are available at the start of the learning process, and it is assumed that they are drawn from the same underlying distribution, so that there is no bias in the labeling[8]. For analytical purposes the setting is equivalently posed as follows: denote by $\mathcal{X}$ a finite input space and $\mathcal{Y}$ the corresponding label space so that $\mathcal{Z} = \mathcal{X} \times \mathcal{Y} = \{(\boldsymbol{x}_1, y_1), ...(\boldsymbol{x}_n, y_n)\}$ is the joint space of labeled inputs. From $\mathcal{Z}$ is drawn *uniformly without replacement* a *training sample* of labeled examples $\mathcal{S} = \{(\boldsymbol{X}_{s_1}, Y_{s_1}), ...(\boldsymbol{X}_{s_m}, Y_{s_m})\} \subseteq \mathcal{Z}$, leaving the remaining *test set* $\mathcal{T} = \{(\boldsymbol{X}_{t_1}, Y_{t_1}), ...(\boldsymbol{X}_{t_u}, Y_{t_u})\} = \mathcal{Z} \backslash \mathcal{S}$. The training sample together with all unlabeled instances from the test set $\{\boldsymbol{X}_{t_1}, ...\boldsymbol{X}_{t_u}\}$ is available to the learner and each unlabeled data point must be labeled. For a given loss function $\ell : \mathcal{D} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ a notion of risk suitable for a binary classifier $h : \mathcal{X} \to \mathcal{D}$ in this transductive setting is the average loss incurred on the test set,

$$\text{risk}_{\mathcal{T}}^{\ell}(h) := \frac{1}{u} \sum_{i=1}^{u} \ell(h(x_{t_i}), y_{t_i}), \tag{1.14}$$

which is sometimes called the *transductive risk*.

Analysis of transduction is often slightly different to that of inductive settings, the difference being that the labelled sample is picked uniformly without replacement from a finite set and so the empirical risk of a hypothesis follows the hypergeometric, rather than the binomial distribution, the former having shallower tails. Vapnik (1998) provides bounds as do Blum and Langford (2003). For example we have the following simple bound,

Theorem 1.3.1. *Derbeko et al. (2004) Let P be any (prior) probability distribution over a class $\mathcal{H}$ of functions on the finite input space $\mathcal{X}$, and ℓ any bounded loss function, $\ell(h(x), y) \in [0, \beta]$, then for any $\delta \in (0, 1]$,*

$$\mathbb{P}_{\mathcal{S}} \left(\forall h \in \mathcal{H} \ : \ \text{risk}_{\mathcal{T}}(h) \leq \text{risk}_{\mathcal{S}}(h) + \beta \sqrt{\frac{m+u}{u} \frac{u+1}{u} \frac{\ln \frac{1}{P(h)} + \ln \frac{1}{\delta}}{2m}} \right) \geq 1 - \delta. \tag{1.15}$$

Analyses of semi-supervised learning and transduction

As discussed in Ben-David et al. (2008), there are two approaches to the analysis of semi-supervised learning and transduction. The first simply attempts to prove better sample complexity bounds for the

[8]This is termed "Setting 1" of the transductive framework in (Vapnik, 1998, Chapter 8) (see also Derbeko et al. (2004) for a discussion of the various alternatives.)

settings using additional information provided by the unlabelled data. For example, in Benedek and Itai (1988) the knowledge of the marginal probability distribution $D_{\mathcal{X}}$ over $\mathcal{X}$ is used to construct an ϵ-cover (with respect to the $\mathcal{L}^1(D_{\mathcal{X}})$ metric) of the hypothesis class thus reducing the size of the search space by discarding many functions that differ only on regions of low density so that better sample complexity bounds can be obtained. Similarly in El-Yaniv and Pechyony (2007) it is observed that in transduction the hypothesis class can be chosen to depend upon all instances $\{\boldsymbol{x}_1, , ...\boldsymbol{x}_n\} \subset \mathcal{X}$ in the labelled and unlabelled sample so that, again, the hypothesis class can be reduced in a similar way. The second (more common) use of knowledge (provided by unlabelled data) of the distribution $D_{\mathcal{X}}$ over instances is closer to the aims of this thesis, where it it is used in conjunction with an assumption on the conditional distribution $D_{\mathcal{Y}|\mathcal{X}}$ so that further assumptions are made about how good classifiers are likely to interact with the instances. We next review some common semi-supervised methods of realising such an assumption using unlabelled data.

Foundational semi-supervised methods

A range of methods for semi-supervised learning and transduction exist which encode assumptions about how good classifiers are likely to interact with the (marginal) distribution of instances from $\mathcal{X}$. Since this objective forms a large part of the motivation for this thesis, we give a brief overview of key methods. Because of the fundamental role of the graph in representing data in these settings many semi-supervised and, in particular, transductive techniques, utilise a graph to capture data geometry and often reduce, in essence, to the problem labelling the vertices of a partially labelled graph.

(i) **Harmonic energy minimization** (Zhu et al., 2003a): with reference to Section 1.3.1 the fundamental assumption that a good classifier is likely to respect the geometry defined by the data translates into the foundational transductive technique of minimising, over real-valued labellings, the smoothness functional (1.10), derived from a graph defined on the data, subject to constraints imposed by the labelled data,

$$\boldsymbol{h}^* := \underset{\boldsymbol{h}\in\mathbb{R}^n}{\operatorname{argmin}} \left\{ \boldsymbol{h}^\top \boldsymbol{L}\boldsymbol{h} \ : \ h_1 = y_1, ...h_m = y_m \right\},$$

which provides a binary classifier by thresholding. Functions minimising a Dirichlet energy functional (1.13), subject to constraints, are *harmonic* (which means in particular, in this case, $\boldsymbol{L}\boldsymbol{h}^* = \boldsymbol{0}$) leading to their many pleasant properties (e.g. Doyle and Snell, 2000). Such functions arise as the solution to many constrained physical systems and the process, called *harmonic energy minimisation* occurs in nature (for example, as we will see in Chapter 5, the voltage induced in an electric circuit with potential constraints is a harmonic function): in some situations, the harmonic energy minimization principle is that which nature prefers to "label" (i.e. extrapolate values from those imposed upon a constraint set to a medium) points given certain constraints..

(ii) **Graph mincuts** (Blum and Chawla, 2001; Blum et al., 2004): here again the idea is to minimise the smoothness functional but over binary-valued labellings, subject to constraints imposed by the

labelled data,

$$\boldsymbol{h}^* := \underset{\boldsymbol{h}\in\{-1,1\}^n}{\operatorname{argmin}} \{\boldsymbol{h}^\top \boldsymbol{L}\boldsymbol{h} \ : \ h_1 = y_1, ...h_m = y_m\}.$$

which can be solved efficiently with a max-flow algorithm, since the smoothness functional equates to the graph cut over binary labellings. Minimising the graph cut cost was first proposed for clustering and image segmentation (Wu and Leahy, 1993). The solution has been criticised because it leads to unbalanced cuts: for example, if only few data points are labelled the optimal solution may just disconnect a few labelled points from the rest of the graph (e.g. Joachims, 2003) leading to degenerate solutions – see point (vii) below for a potential solution to this problem.

(iii) **Graph Regularization** (Belkin et al., 2004) : the harmonic energy minimisation principle is extended to allow for noise, by generalising to a regularization scheme,

$$\boldsymbol{h}^* := \underset{\boldsymbol{h}\in\mathbb{R}^n, \boldsymbol{h}^\top \mathbf{1}=0}{\operatorname{argmin}} \frac{1}{m}\sum_{i=1}^{m}(h_i - y_i)^2 + \gamma \boldsymbol{h}^\top \boldsymbol{L}\boldsymbol{h}. \tag{1.16}$$

Risk bounds relative to the Fiedler vector – the smallest non-trivial eigenvalue of the Laplacian and a basic measure of 'algebraic connectivity' of the graph (Fiedler, 1973; Chung, 1997) – are provided via a stability analysis.

(iv) **Local and global consistency** (Zhou et al., 2003): this presents a regularization method similar to (1.16), using the normalized Laplacian and extending the method to the multi class problem.

(v) **Laplacian support vector machines (LapSVM)** (Belkin et al., 2006): the above methods can be extended to define semi-supervised algorithms providing classifiers valid out of sample on new unseen instances. For an arbitrary kernel K defining an RKHS $\mathcal{H}_K$ the generic regularization problem,

$$h^* := \underset{h\in\mathcal{H}_K}{\operatorname{argmin}} \frac{1}{m}\sum_{i=1}^{m}\ell(h(x_i), y_i) + \gamma_A||h||_K^2 + \frac{1}{m+u}\gamma_I \boldsymbol{h}^\top \boldsymbol{L}\boldsymbol{h}, \tag{1.17}$$

is solved, where $\boldsymbol{h} \in \mathbb{R}^n$ is the vector of point evaluations of h on the labelled and unlabelled data, $\boldsymbol{h} := (h(\boldsymbol{x}_i))$, and γ_A, γ_I control the relative weight attached to ("ambient") regualrization in the RKHS and with respect to the intrinsic geometry respectively. By specializing to the hinge loss (1.17) defines the LapSVM solution.

(vi) **Transductive and semi-supervised support vector machines**: It is possible to apply the large margin principle to the transductive and semi-supervised settings; the classifier chosen is essentially that which maximises the margin over the full set of labelled (training) and unlabelled (test) data, rather than over the training data alone, as in the inductive case. This idea was first proposed in Vapnik (1998) and implemented as the "S^3VM" (Bennett and Demiriz, 1998) and "transductive SVM" (TSVM) (Joachims, 1999). Given a sample $\mathcal{S} = \{(\boldsymbol{X}_{s_1}, Y_{s_1}), ...(\boldsymbol{X}_{s_m}, Y_{s_m})\} \cup \{\boldsymbol{X}_{s_{m+1}}, ...\boldsymbol{X}_{s_{m+u}}\}$ let $\{\boldsymbol{\xi}(\boldsymbol{x}_i)\}_{i=1}^n$ be the data inputs mapped into some feature space, and write

for brevity $\boldsymbol{\xi}_i = \boldsymbol{\xi}(\boldsymbol{x}_i)$. The approach is to maximise (over hyperplane classifiers $(\boldsymbol{w}, b)$, and inferred labellings $\boldsymbol{y}^* \in \{-1, 1\}^u$) the margin

$$\min_{i=1,...n} \frac{\hat{y}_i(\boldsymbol{w}^\top \boldsymbol{\xi}_i + b)}{||\boldsymbol{w}||}, \tag{1.18}$$

where $\hat{y}_i \begin{cases} y_i \text{ if } i \leq m \\ y_i^* \text{ if } i > m \end{cases}$ denotes the (true or inferred, as appropriate) label of $\boldsymbol{x}_i$. The optimisation is thus, for the linearly separable case[9],

$$\begin{aligned} & \underset{\boldsymbol{y}^* \in \{-1,1\}^u, \boldsymbol{w}, b}{\operatorname{argmin}} \frac{1}{2}||\boldsymbol{w}||^2 \\ \text{such that :} \quad & y_i(\boldsymbol{w}^\top \boldsymbol{\xi}_i + b) \geq 1 \qquad i = 1, ...m \\ & y_i^*(\boldsymbol{w}^\top \boldsymbol{\xi}_{m+i} + b) \geq 1 \qquad i = 1, ...u \end{aligned} \tag{1.19}$$

This combinatorial optimisation problem is more difficult than the inductive SVM due to the possible assignment of $\boldsymbol{y}^*$ to any combination of two classes. It is in fact intractable for even modest test sets. The optimisation is solvable approximately, but unfortunately this tends to require a complicated suite of heuristics which can give bad results and have often been criticised (Chapelle and Zien, 2005; Belkin et al., 2006). The inner optimisation in (1.19) over $(\boldsymbol{w}, b)$ is simply that of an inductive SVM and can be readily performed in the dual form. The key idea of the TSVM, therefore, is to calculate a labelling of the test data with an inductive SVM, and then swap labels of the test examples so that the objective function decreases.

Alternatively the problem (1.19) can be reformulated as,

$$\begin{aligned} & \underset{\boldsymbol{w}, b}{\operatorname{argmin}} \frac{1}{2}||\boldsymbol{w}||^2 \\ \text{such that :} \quad & y_i(\boldsymbol{w}^\top \boldsymbol{\xi}_i + b) \geq 1 \qquad i = 1, ...m \\ & |\boldsymbol{w}^\top \boldsymbol{\xi}_{m+i} + b| \geq 1 \qquad i = 1, ...u, \end{aligned}$$

but the final (transductive) set of constraints are not convex, which gives another insight into the difficulty of solving the problem. An approach to solving this non-convex problem via a gradient descent method is presented in Chapelle and Zien (2005).

(vii) **The spectral graph transducer and normalized complexities** : as discussed above, minimising the graph cut of a binary labelling of a graph tends to induce unbalanced, degenerate cuts. Following Shi and Malik (2000) a potential solution to this problem is to modify the objective to capture the ratio of the cut to the size of the partitions produced resulting in normalized measures of complexity: suppose a binary labelling $\boldsymbol{h} \in \{-1, 1\}^n$ of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ partitions $\mathcal{V}$ into subsets $\mathcal{V}^+ = \{v_i \ : \ h_i = 1\}$ and $\mathcal{V}^- = \{v_i \ : \ h_i = -1\}$. Let $d(v_i) = |\{j \ : \ (i, j) \in \mathcal{E}\}|$ denote the degree of vertex v_i, and $d(\mathcal{U}) = \sum_{v_i \in \mathcal{U}} d(v_i)$.

[9] In the paper Joachims (1999) the theory is developed for the SVM with soft constraints, thus accommodating misclassifications, but for simplicity of presentation of the principle the simpler case of hard constraints is given.

Definition The *sparsity* of $\boldsymbol{h}$ is

$$s(\boldsymbol{h}) = \frac{\boldsymbol{h}^\top \boldsymbol{L}\boldsymbol{h}}{\min(|\mathcal{V}^+|, |\mathcal{V}^-|)} \tag{1.20}$$

Definition The *ratio cut* of $\boldsymbol{h}$ is

$$r(\boldsymbol{h}) = \frac{\boldsymbol{h}^\top \boldsymbol{L}\boldsymbol{h}}{|\mathcal{V}^+||\mathcal{V}^-|} \tag{1.21}$$

Definition The *conductance* of $\boldsymbol{h}$ is

$$\gamma(\boldsymbol{h}) = \frac{\boldsymbol{h}^\top \boldsymbol{L}\boldsymbol{h}}{d(\mathcal{V}^+)d(\mathcal{V}^-)} \tag{1.22}$$

The conductance of $\mathcal{G}$ is $\Gamma(\mathcal{G}) = \min_{\boldsymbol{h}\in\{-1,1\}^n} \gamma(\boldsymbol{h})$. These normalised measures of complexity have been considered as measures of the complexity of a function defined over the vertices of a graph, and are a particular focus in the computer vision and spectral clustering communities but minimizing such measures subject to constraints exactly is generally NP-hard.

In Joachims (2003) the unsupervised problem of finding the minimum ratio cut in a graph is extended to include vertex label constraints via a regularisation as follows.

$$\begin{aligned}
\min_{\boldsymbol{h}\in\{-1,1\}^n} r(\boldsymbol{h}) &= \min_{\boldsymbol{h}\in\{-1,1\}^n} \frac{\boldsymbol{h}^\top \boldsymbol{L}\boldsymbol{h}}{n\boldsymbol{h}^\top\boldsymbol{h} - (\boldsymbol{h}^\top\mathbf{1})^2} \\
&= \min_{\boldsymbol{h}\in\{\gamma^-,\gamma^+\}^n} \frac{\boldsymbol{h}^\top \boldsymbol{L}\boldsymbol{h}}{n\boldsymbol{h}^\top\boldsymbol{h}} \\
&= \min_{\boldsymbol{h}\in\{\gamma^-,\gamma^+\}^n} \frac{\boldsymbol{h}^\top \boldsymbol{L}\boldsymbol{h}}{n^2}
\end{aligned}$$

where $\gamma^- = -\frac{|\mathcal{V}^+|}{|\mathcal{V}^-|}$ and $\gamma^+ = \frac{|\mathcal{V}^-|}{|\mathcal{V}^+|}$, and $\mathcal{V}^+$, $\mathcal{V}^-$ are the positive and negative vertex sets. Noting that the constraints $\boldsymbol{h} \in \{\gamma^-, \gamma^+\}^n$ imply that $\boldsymbol{h}^\top\mathbf{1} = 0$ and $\boldsymbol{h}^\top\boldsymbol{h} = n$, so this is relaxed to the minimisation problem

$$\operatorname*{argmin}_{\boldsymbol{h}\in\mathbb{R}^n, \boldsymbol{h}^\top\mathbf{1}=0, \boldsymbol{h}^\top\boldsymbol{h}=n} \boldsymbol{h}^\top \boldsymbol{L}\boldsymbol{h},$$

and constraints are included by formulating the problem as a regularisation,

$$\operatorname*{argmin}_{\boldsymbol{h}\in\mathbb{R}^n, \boldsymbol{h}^\top\mathbf{1}=0, \boldsymbol{h}^\top\boldsymbol{h}=n} \boldsymbol{h}^\top \boldsymbol{L}\boldsymbol{h} + c(\boldsymbol{h}-\boldsymbol{\gamma})^\top \boldsymbol{C}(\boldsymbol{h}-\boldsymbol{\gamma}), \tag{1.23}$$

where $\gamma_i = \gamma^+$ ($\gamma_i = \gamma^-$) if v_i has a positive (negative) constraint and is zero otherwise. Since γ^+ is unknown it is estimated from the proportion of positive and negative examples in the training set (which is valid if the training sample is not biased toward including examples from either class). A closed form for the solution to (1.23) is presented.

Similar methodologies derived from various optimisations involving such normalised complexity measures are considered in Bie et al. (2004); Eriksson et al. (2007); Bie and Cristianini (2003). The combinatorial problems are NP-hard, and each approach tends to present a particular relaxation. Simple relaxations do not reduce the problem to a convex optimisation (the constraints are generally non-convex), but the non-convex problems can be solvable. Solutions typically involve a lot of heuristics.

(viii) **Kernels derived from the graph Laplacian**: recalling Section 1.2.2 the discrete Green's function corresponding to the graph Laplacian $\boldsymbol{L}$ (of a connected graph) operating on the space $\mathcal{H}^{\perp} := \{\boldsymbol{h} \in \mathbb{R}^n : \boldsymbol{h} \perp \mathbf{1}\}$ is precisely the kernel defined by the pseudoinverse of the graph Laplacian, $\boldsymbol{L}^{+}$. Thus the regularization operator $\mathrm{reg}(\boldsymbol{h}) := \sqrt{\frac{1}{2}\boldsymbol{h}^{\top}\boldsymbol{L}\boldsymbol{h}}$ over $\boldsymbol{h} \in \mathcal{H}^{\perp}$ is the natural RKHS norm for the RKHS generated by the kernel $\boldsymbol{L}^{+}$, as pointed out in Smola and Kondor (2003); Herbster and Pontil (2007). Thus $\boldsymbol{L}^{+}$ is a kernel whose RKHS norm measures the smoothness of functions on the graph formed on data. Developing upon this fact in Chapelle et al. (2002); Smola and Kondor (2003); Zhu et al. (2004) a variety of kernels derived from the graph Laplacian, essentially by transforming the spectrum in some way, are presented which can be used as empirically-defined kernels in any kernelizable learning algorithm in the transductive setting. In particular Herbster and Pontil (2007) consider learning using the kernel Perceptron with the kernel $\boldsymbol{L}^{+}$.

(ix) **Gaussian processes over functions defined on a graph** (Zhu et al., 2003b): The method of harmonic energy minimisation and can essentially be seen as choosing the MAP hypothesis from the corresponding Markov random field, over the space $\mathbb{R}^n$ of real-valued graph labellings, defined by the density,

$$p(\boldsymbol{h}) := \frac{1}{Z} e^{-\frac{1}{2}\boldsymbol{h}^{\top}\boldsymbol{L}\boldsymbol{h}}, \tag{1.24}$$

condition on observed data (which is just a finite dimensional Gaussian distribution). Likewise graph mincut method can be seen as the corresponding discrete Markov random field. A Bayesian approach is to use the density (1.24) to define a prior for a Gaussian process and perform Bayesian inference given the observed labels.

Other applications of graph theoretical methods - clustering and dimensionality reduction

Because of its ability to capture the geometry defined by data the graph has become a fundamental object in areas of machine learning in which understanding the structure of the data-generating distribution is a particular focus or where exploiting the geometry should be particularly effective. In particular the field of *spectral clustering* in which, typically, the Laplacian spectrum is used to determine appropriate partitions of data is an active one; in essence lower eigenvectors of the Laplacian are smoother functions over the vertices. Likewise, applications of graph-theoretical methods to *non-linear dimensionality reduction* are celebrated (Tenenbaum et al., 2000; Roweis and Saul, 2000; Belkin and Niyogi, 2003; Weinberger and Saul, 2004); a methodology, for example, is to find a low dimensional representation of data using the lowest elements of (some modification of) the Laplacian eigensystem.

Some theoretical limitations of semi-supervised methods

The above methods have all demonstrated practical success on real and artificial data and in practical applications (see for example Chapelle et al. (2006) for a comparison of many methods). However it is also worth pointing out some theoretical limitations of the semi-supervised methods as discussed above.

Even when assumptions about the data-generating distribution are ideally met – for example when the cluster assumption holds strongly and data is generated by 2 unimodal distributions (Gaussians) labelled homogeneously – standard methods such as "low density separation" methods will hinder the learning process while ERM which ignores the unlabelled data will learn the optimal classifier quickly (Ben-David et al., 2008).

1.4 Some limitations of current analyses and the contribution of this thesis

1.4.1 Limitations of classical analyses of statistical learning theory

Let us briefly mention some limitations of the statistical analyses introduced in Section 1.1.1. One foundational example of a risk bound as suggested in (1.5) is the VC bound on generalization (e.g. Bousquet et al., 2003a),

$$\mathbb{P}_{\mathcal{S}}\left(\sup_{h\in\mathcal{H}}(\mathrm{risk}(h)-\widehat{\mathrm{risk}}_{\mathcal{S}}(h))\leq 2\sqrt{\frac{2\mathrm{VC}(\mathcal{H})\ln\frac{2me}{\mathrm{VC}(\mathcal{H})}+\ln\left(\frac{2}{\delta}\right)}{m}}\right)\geq 1-\delta,$$

relating uniform convergence of empirical risk to true risk to the VC dimension of the hypothesis class. The VC dimension is independent of the data-generating distribution and so the complexity term is identical for all possible distributions and all samples, and since it holds for the worst possible distribution the bound is substantially pessimistic in general. Data-dependent measures of complexity such as Rademacher complexity (Koltchinskii and Panchenko, 2000; Bartlett and Mendelson, 2002) attempt to offer a more refined analysis, but such terms are often upper bounded by quantities only weakly dependent on the data sample, such as the trace of a kernel gram matrix on the data. It will be the focus of Chapter 3 to offer a more detailed understanding of the relationship between such data-dependent complexity measures and structure (observed) in data.

A drawback of uniform convergence analyses in general is the problem that bounding the supremum of the deviation between true and empirical risk over a hypothesis class lacks what is called "localization". This refers to the fact that the deviation which is bounded may be much smaller at the elements of the hypothesis class which are ultimately the objects of interest (our chosen hypotheses), for example the variance of the empirical risk of a regularized empirical risk minimizer might be expected to be significantly smaller than the largest variance in the class, thus uniform convergence bounds tend to be overly pessimistic and in fact many of the performance guarantees so derived are vacuous. Attempts to refine this type of analysis include the "local Rademacher complexities" (Bartlett et al., 2002) and (less explicitly) other frameworks such a some PAC-Bayes analyses which will be introduced in Chapter 2.

Another problem and a main focus of this research is that classical analyses are not suited to exploit the role of geometry defined by data in the learning process. This is intimately related to the fact that central to the uniform convergence framework is the notion of a *sample-independent hypothesis class*; the VC and Rademacher bounds are not valid whenever the hypothesis class $\mathcal{H}$ is informed by the

data sample. This is a significant restriction on the ability to capture how we expect a good hypothesis to interact with the structure defined by the data-generating distribution, if what we know about that structure is learned from the data sample as outlined in Section 1.3. For example, if our chosen hypothesis class is an RKHS $\mathcal{H}_K$ the kernel K must not be informed by the data sample, and therefore cannot in general be a kernel tuned to the specific geometry defined by the problem, such as those derived from a graph Laplacian discussed in Section 1.3.1 (viii).

One way of achieving something along these lines within the structural risk minimization framework (e.g. Vapnik, 1998) is the so-called "luckiness framework" (Shawe-Taylor et al., 1998) in which the hypothesis class is structured by its (Hilbert space) norm which, together with the restriction that $|h(\boldsymbol{x})| \geq 1$ on labelled data, is tantamount to preferring hypotheses which achieve large margin on the data sample. Thus, though structured according to data-independent quantities (the data-independent RKHS norm for instance) the hypothesis class is implicitly structured according to the margin achieved on data. Nonetheless such an analysis does not immediately extend to more sophisticated notions of how classifiers interact with the data-distribution (such as smoothness or Dirichlet energy functionals discussed above) and, for example, the distance to a separating hyperplane must be measured with respect to a metric of the ambient space rather than a more appropriate metric informed by the data, such as geodesic distance on a manifold defined by the data distribution or empirical "resistance" distances which take account of the density of the data-generating distribution and will be introduced in Chapter 5.

One means of overcoming these restrictions is to work with an *unknown* hypothesis class $\mathcal{H}$ defined by the unknown data-generating distribution. This would generally present a problem for an algorithm which must pick a function from $\mathcal{H}$ as its chosen hypothesis, but during learning it is likely possible to find a hypothesis that is with high probability in the unknown $\mathcal{H}$ and so attain a valid risk bound. This is essentially the idea underlying the semi-supervised analysis of Balcan and Blum (2005) which uses a notion of function compatibility[10] with the unknown data-generating distribution to define an unknown hypothesis class. However, this adds an additional layer of convergence to the risk analysis – that of the hypotheses' compatibility with the data sample to their compatibility with the underlying distribution. Therefore, in order to quantify this convergence and pick a hypothesis that is with high probability in the unknown $\mathcal{H}$ one must work in an a-priori known hypothesis class; again, one must ultimately rely upon an a-priori known hypothesis class not informed by the data-generating distribution or a sample from it (we will encounter such a problem in Section 3.5 and so refer the reader to that section for a concrete example). So the problem of analysing methods using using only data-defined hypothesis classes is not satisfactorily solved at all. Further if the compatibility between classifier and data is sophisticated then the required convergence can be quite difficult to establish resulting in a significant deterioration in the bound. Indeed, the compatibilities considered in Balcan and Blum (2005) are simple first order interactions so that standard results for the concentration sums of i.i.d. random variables can be used to establish the required convergence. This is restrictive since, for example, the smoothness of a hypothesis

[10] Compatible functions are typically those satisfying a certain level of "smoothness" over the unknown data-generating distribution for instance.

on a graph formed on data (1.10) is a second order compatibility and convergence of the smoothness functional to its expectation uniformly over all hypotheses in a class requires convergence of a second-order U-process and is so not immediately captured by the framework.

Only in the transductive setting is learning with an empirically defined hypothesis class currently possible without a degradation in convergence analysis of empirical to true risk, since in the transductive setting the geometry of the input distribution is entirely known at the start of the learning process, and we can build an empirically-defined hypothesis class, or, similarly, work in empirically-defined distance metrics. This means that the transductive setting is an interesting playground for ideas on how to exploit data-defined structure.

1.4.2 Contributions of this thesis

Now equipped with these preliminaries we outline the contributions of this thesis.

Contributions of Chapter 2

We present a new statistical analysis of learning in the context of exploiting structure defined by data. The analysis uses PAC-Bayes theory which provides some of the sharpest risk analyses available. By defining the PAC-Bayes prior over a hypothesis class in terms of the unknown true risk and a notion of "smoothness" of hypotheses, the analysis is "localized" in the sense discussed in Section 1.4.1, so that complexity of the model enters not as the complexity of an entire hypothesis class, but around the functions of ultimate interest. As well as providing a sharp risk analysis for several learning methods including SVMs the framework developed is flexible enough to permit defining the unknown hypothesis class in terms of quite sophisticated interactions between the hypotheses and the unknown data-generating distribution – and with apparently little degradation compared to classical attempts in the uniform convergence framework. The research opens potentially interesting new notion of hypothesis class complexity. Part of this chapter was published as Lever et al. (2010) which was joint work with John Shawe-Taylor and Francois Laviolette.

Contributions of Chapter 3

We relate the Rademacher complexity of a function class to cluster structure in data. In particular this quantifies the intuitive notion that when data clusters we can learn well with fewer examples, under typical smoothness assumptions, and be more confident in our analysis. In particular we relate learning to cluster structure in the empirical resistance metric considered in Chapter 5 and derive a bound on the complexity of functions defined over the vertices of a graph. This potentially facilitates algorithms whose use of regularization is determined by the observed cluster structure in data. Part of this chapter was published as Lever (2010).

Contributions of Chapter 4

Typical methods for learning over a graph do not scale well in the number of data points – often a graph Laplacian must be inverted which becomes computationally intractable for large data sets. Here we present some online algorithms which, by simplifying the structure of the data in principled way, are able to exploit the structure while remaining computationally tractable for large datasets. We prove close to state-of-the-art performance guarantees for these methods. Part of this chapter was published as Herbster et al. (2008) which was joint work with Mark Herbster and Massimiliano Pontil.

Contributions of Chapter 5

We present a new class of methods for learning over a graph which we study in the online framework. Inspired by the p-norm Perceptron's ability to learn sparse concepts in $\mathbb{R}^n$, with a mistake bound logarithmic in n, we consider a similar idea for building classifiers on graphs. We introduce a family of p-seminorms defined on the labellings of a graph which include the smoothness functional of Belkin et al. (2004); Zhu et al. (2003a) and the label-consistent graph cut (Blum and Chawla, 2001) as limiting cases. We present an online algorithm for learning concepts defined on graphs based upon minimum p-seminorm interpolation and derive a mistake bound in which the graph cut of a labelling is the measure of the complexity of the learning task. The dual seminorm gives rise to a generalisation of the notion of electrical resistance between graph vertices which we term p-resistance and show that it is a natural measure of similarity between graph vertices. We give a brief survey of its fundamental properties by extending a well-known analogy with resistive networks. Cluster structure in the graph w.r.t. the p-resistance distance (captured via covering number of the vertex set) features as the "structural" term in our mistake bound. Expressing the bound in this way helps to demonstrate that our algorithm exploits connectivity and cluster structure in data and we show that a learner can choose the parameter p (using only information available a-priori to the learner) to ensure a performance guarantee which is logarithmic with regard to graph diameter, whereas some foundational methods have a linear lower bound. Part of this chapter was published as Herbster and Lever (2009) which was joint work with Mark Herbster.

Chapter 2

Distribution-dependent PAC-Bayes priors

Abstract

We further develop the idea that the PAC-Bayes prior can be informed by the data-generating distribution. We prove sharp bounds for an existing framework of Gibbs algorithms, and develop insights into function class complexity in this model. In particular we consider controlling capacity with respect to the *unknown* geometry of the data-generating distribution. We finally extend the localized PAC-Bayes analysis to more practical learning methods, in particular RKHS regularization schemes such as SVMs.

2.1 Introduction

This research takes its inspiration from Catoni (2007), who developed localised PAC-Bayes analysis by using a prior defined in terms of the data generating distribution. At first sight this might appear to be 'cheating', since we must define the prior before seeing the data. However, by defining in terms of the distribution we avoid this difficulty since the distribution itself can be considered as fixed before the sample is generated. PAC-Bayes bounds are one of the sharpest analyses of the learning process. A weakness in the standard PAC-Bayes approach is that analysis is constrained by the choice of prior distribution, since the divergence between prior and posterior forms part of the bound. This choice of prior tends to be rather generic; typically not tailored to the particular problem, so that, in particular, good classifiers do not generally receive large prior weight. Thus the divergence term in the PAC-Bayes analysis can typically be large. By tuning the prior to the distribution Catoni is able to remove the Kullback-Leibler (KL) term from the bound hence significantly reducing the complexity penalty.

We begin by investigating the 'Gibbs algorithms' in which the predictive posterior is a Boltzman distribution over hypotheses. We use Catoni's definition of the prior involving a Boltzmann distribution, but prove a new sharp bound (Theorem 2.3.2) using a new lemma (Lemma 2.2.4) and the re-use of the PAC-Bayes bound to remove the KL term (Lemma 2.3.1). The resulting bound suggests a new complexity parameter γ that enters as a $\gamma/m^{3/2}$ term (where m is the sample size). This opens a potential new direction in the generalization analysis of learning machines.

In our context this suggests the need to regularize in this learning method. The flexibility of the framework we develop is that it allows us to encode our prior meta-assumptions about how we anticipate a good classifier will interact with the data; we can control capacity, for example, with respect to the smoothness of a classifier over the *unknown* data generating distribution thus giving high weight to classifiers that are, for example, smooth over a manifold defined by the support of the data distribution. The analysis is achieved with a novel PAC-Bayes bound on U-statistics estimation.

Finally we cover a third main theme, which is the extension of the data distribution dependent priors to the Gaussian prior and posterior PAC-Bayes bounds for RKHS regularization algorithms, for example, SVMs (Langford and Shawe-taylor, 2002). In Theorem 2.4.5 we present a new localised PAC-Bayes bound for this setting. Here we are able to remove the KL term again leaving a term that only involves a similar complexity parameter γ appearing as $O(\gamma/(\eta^2 m^2))$, where η is the regularization parameter, in contrast to the usual $O(\|\mathbf{w}\|^2/m)$. This again suggests a new measure of complexity for SVM classifiers with the possibility of using the bound to optimise the regularization parameter. We go on to extend this method to the case where the data is used to define the kernel in an SVM, deriving in Theorem 2.4.10 a localized PAC-Bayes bound for algorithms such as LapSVM.

We now review the relation of our approach to earlier work. The luckiness framework explored the possibility that we could learn the hierarchy of classes of hypotheses from the data as part of the learning process giving rise to so-called data-dependent structural risk minimization (Shawe-Taylor et al., 1998). The most successful example of this approach was large margin classification such as support vector machines. However, although we cannot measure a margin without seeing the data, by moving to real-valued functions, we can equate large margin with small norm when we constrain $y_i f(\mathbf{x}_i) \geq 1$ on the training data, $i = 1, \ldots, m$, resulting in a fixed prior. Nonetheless this is equivalent to placing a prior over the classifiers in terms of the data generating distribution, that is we favour hyperplanes that have low input density in the slab defined by shifting the decision boundary parallel to itself by $\pm\gamma$.

Further research in this direction has been developed by Balcan and Blum (2010) with their notion of compatibility, which is used to restrict the hypotheses considered in the learning process to those satisfying a given level of compatibility *estimated from the training data*, hence reducing the effective complexity of the class. Perhaps less well-known is work by Catoni (2007) where he introduces 'localised' PAC-Bayes analysis effectively defining the prior in terms of the data-generating distribution in a PAC-Bayes bound on generalization.

We should finally distinguish between distribution defined priors and using part of the data to learn a prior and the rest to learn the function (Ambroladze et al., 2006).

2.2 Preliminaries

We consider the general setting in which we are given a sample of labelled and unlabelled points $\mathcal{S} = \{(X_1, Y_1), ...(X_m, Y_m)\} \cup \{X_{m+1}, ...X_n\} = \mathcal{S}_{\text{labelled}} \cup \mathcal{S}_{\text{unlabelled}}$ drawn i.i.d. according to a

probability measure ν over $\mathcal{Z} := \mathcal{X} \times \mathcal{Y}$, the product space of labelled inputs (or its marginalization to $\mathcal{X}$). We suppose throughout that $(\mathcal{Z}, \Sigma, \nu)$ is a probability measure space. We initially consider supervised learning setting in which $\mathcal{S}_{\text{unlabelled}} = \emptyset$ but our analysis will later include the semi-supervised learning setting in which $\mathcal{S}_{\text{unlabelled}} \neq \emptyset$.

We are interested in the case where $\mathcal{Y} = \{-1, +1\}$, and study binary classification. We are interested in the notion of *risk* of a hypothesis $h \in \mathcal{H}$,

$$\text{risk}^{\ell}(h) \quad := \mathbb{E}_{(X,Y)\sim D}\ell(h(X), Y),$$

and its empirical counterpart on a labelled sample $\mathcal{S}$,

$$\widehat{\text{risk}}^{\ell}_{\mathcal{S}}(h) := \frac{1}{|\mathcal{S}|}\sum_{(X,Y)\in\mathcal{S}} \ell(h(X), Y),$$

where $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is some loss function. When $\ell(\cdot,\cdot)$ is the $0-1$ loss of binary classification, $\ell_{0-1}(y, y') := \begin{cases} 0 \text{ if } y = y' \\ 1 \text{ if } y \neq y' \end{cases}$, then for simplicity we denote the corresponding binary classification risk and its empirical counterpart by $\text{risk}(\cdot)$ and $\widehat{\text{risk}}_{\mathcal{S}}(\cdot)$ respectively. Our objective is to obtain a probabilistic guarantee on the true binary classification risk of a classifier by relating it to its empirical counterpart.

The following quantities feature in the PAC-Bayes analysis: the Kullback-Leibler divergence between distributions Q and P, and its specialization to Bernouilli distributions,

$$\text{KL}(Q||P) := \mathbb{E}_{h\sim Q} \ln \frac{dQ}{dP}(h), \qquad \text{kl}(q, p) := q \ln \frac{q}{p} + (1-q)\ln\frac{1-q}{1-p} \quad q, p \in (0, 1),$$

and we define

$$\xi(m) := \sum_{k=0}^{m} \binom{m}{k} \left(\frac{k}{m}\right)^k \left(1 - \frac{k}{m}\right)^{m-k} \in [\sqrt{m}, 2\sqrt{m}],$$

where the inequalities follow from (Maurer, 2004, Equations (1) and (2)), after noticing that $\xi(m) = \mathbb{E}[e^{m\text{kl}(\frac{1}{m}\sum_{i=1}^{m} W_i, \zeta)}]$, where W_i are i.i.d. random variables with mean ζ (see, e.g. Germain et al., 2009, for a derivation). The PAC-Bayes bounds apply to a stochastic Gibbs classifier G_Q drawn from a *posterior distribution* Q over a hypothesis class $\mathcal{H}$, this distribution will typically depend upon the data sample. This is in contrast to the *prior* distribution, denoted throughout by P, which is used for analysis and must not be defined in terms of the sample. We denote $\text{risk}(G_Q) := \mathbb{E}_{h\sim Q}\text{risk}(h)$ and $\widehat{\text{risk}}^{\ell}_{\mathcal{S}}(G_Q) := \mathbb{E}_{h\sim Q}\widehat{\text{risk}}^{\ell}_{\mathcal{S}}(h)$.

The following is a generalization of (Germain et al., 2009, Th 2.1) and is proved using the same sequence of arguments.

Theorem 2.2.1. *For any functions $A(h)$, $B(h)$ over $\mathcal{H}$, either of which may be a statistic of the sample $\mathcal{S}$, any distributions P over $\mathcal{H}$, any $\delta \in (0, 1]$, any $t > 0$, and a convex function $\mathcal{D} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ we have with probability at least $1 - \delta$ over the draw of $\mathcal{S}$,*

$$\forall\, Q \text{ on } \mathcal{H} : \mathcal{D}(\mathbb{E}_{h\sim Q}A(h), \mathbb{E}_{h\sim Q}B(h)) \leq \frac{1}{t}\left(\text{KL}(Q||P) + \ln\left[\frac{\mathcal{L}_P}{\delta}\right]\right),$$

where $\mathcal{L}_P := \mathbb{E}_{\mathcal{S}}\mathbb{E}_{h\sim P}\left[e^{t\mathcal{D}(A(h),B(h))}\right]$.

Note that $\mathcal{L}_P$ is the moment generating function of $\mathcal{D}(A(h), B(h))$.

Proof. Since $\mathbb{E}_{h\sim P}e^{t\mathcal{D}(A(h),B(h))}$ is a non-negative random variable, Markov's inequality gives

$$\mathbb{P}_{\mathcal{S}}\left(\forall\, Q \text{ on } \mathcal{H}:\ \ \mathbb{E}_{h\sim P}\, e^{t\,\mathcal{D}(A(h),B(h))} \leq \frac{\mathcal{L}_P}{\delta}\right) \geq 1-\delta\,.$$

Hence, by taking the logarithm on each side of the innermost inequality and by transforming the expectation over P into an expectation over Q, we obtain

$$\mathbb{P}_{\mathcal{S}}\left(\forall Q : \ln\left[\mathbb{E}_{h\sim Q}\frac{\mathrm{d}P}{\mathrm{d}Q}(h)e^{t\,\mathcal{D}(A(h),B(h))}\right] \leq \ln\left[\frac{\mathcal{L}_P}{\delta}\right]\right) \geq 1-\delta\,.$$

Since $\ln(x)$ is concave, Jensen's inequality then gives

$$\ln\left[\mathbb{E}_{h\sim Q}\frac{\mathrm{d}P}{\mathrm{d}Q}(h)e^{t\mathcal{D}(A(h),B(h))}\right] \geq -\mathrm{KL}(Q\|P) + t\mathbb{E}_{h\sim Q}\mathcal{D}(A(h),B(h))\,,$$

and the theorem follows from another application of Jensen's inequality to the convex function $\mathcal{D}(\cdot,\cdot)$, *i.e.*,

$$\mathbb{E}_{h\sim Q}\,\mathcal{D}(A(h),B(h)) \geq \mathcal{D}(\mathbb{E}_{h\sim Q}A(h),\,\mathbb{E}_{h\sim Q}B(h)).$$

□

Theorem 2.2.1 is a recipe for generating a variety of PAC-Bayes bounds, by specializing to a choice for $\mathcal{D}(\cdot,\cdot)$, t, $A(\cdot)$ and $B(\cdot)$, and choosing P to be a "prior" (i.e. not sample-dependent) so that the order of expectation in the r.h.s. can be exchanged and evaluated. For example, by choosing $t = m$, $A(h) = \widehat{\mathrm{risk}}_{\mathcal{S}}(h)$, $B(h) = \mathrm{risk}(h)$, and $\mathcal{D}(q,p) = \mathrm{kl}(q,p)$, one can derive Seeger's bound (Seeger, 2002; Langford, 2005). By choosing $\mathcal{D}(q,p) = \mathcal{F}(p) - C\cdot q$ for some positive constant C and where $\mathcal{F}(p) = \ln\frac{1}{(1-p\,[1-e^{-C}])} = -\frac{1}{m}\ln(M_X(-C))$ where $M_X(t) = 1-p+pe^t$ is the moment-generating function of a binomial random variable with parameters (m,p), one will obtain Catoni's PAC-Bayes bound (Catoni, 2007). To derive these bounds from Theorem 2.2.1, in the first case, one simply has to show that $\mathcal{L}_P = \xi(m)$, and in the second case that $\mathcal{L}_P = 1$. These equalities are obtained by straightforward calculations. The following theorem gives Seeger's bound, and a slightly relaxed version of Catoni's bound; these results will be needed later on.

Theorem 2.2.2. *Seeger (2002); Langford (2005); Catoni (2007) For any (unknown) distribution D, any set $\mathcal{H}$ of classifiers, any distribution P of support $\mathcal{H}$, any $\delta \in (0,1]$, and any positive constant C, we have, where $C^\star := \frac{C}{1-e^{-C}}$,*

$$\mathbb{P}_{\mathcal{S}}\left(\forall\, Q \,\mathrm{on}\,\mathcal{H} : \mathrm{kl}(\widehat{\mathrm{risk}}_{\mathcal{S}}(G_Q), \mathrm{risk}(G_Q)) \leq \tfrac{1}{m}\left(\mathrm{KL}(Q||P) + \ln\tfrac{\xi(m)}{\delta}\right)\right) \geq 1-\delta \qquad \textit{(Seeger's bound)}$$

$$\mathbb{P}_{\mathcal{S}}\left(\forall\, Q \,\mathrm{on}\,\mathcal{H} : \mathrm{risk}(G_Q) \leq C^\star\left(\widehat{\mathrm{risk}}_{\mathcal{S}}(G_Q) + \tfrac{1}{C\cdot m}\left(\mathrm{KL}(Q||P) + \ln\tfrac{1}{\delta}\right)\right)\right) \geq 1-\delta \quad \textit{(Catoni's bound)}.$$

Note that the PAC-Bayes bound proposed by McAllester in his pioneer work on the subject (McAllester, 1999) can be retrieved from Seeger's bound using the inequality

$$2(q-p)^2 \leq \mathrm{kl}(q,p). \tag{2.1}$$

Hence, the rate of convergence given by Seeger's bound for fixed Q and P is at least in $\mathcal{O}(\frac{\ln\sqrt{m}}{\sqrt{m}})$. On another hand, the Catoni's bound guarantees a rate of convergence of $\mathcal{O}(\frac{1}{m})$ "up to" some predefined multiplicative factor. Of course, because $C^\star \to 1$ only when $C \to 0$, the constants involved in this $\mathcal{O}(\frac{1}{m})$ rate degrade as the multiplicative factor approaches 1.

PAC-Bayes bounds are among the sharpest in learning theory (Langford, 2005). Typically the KL term is the dominant quantity in the bound and analysis is constrained by the need to choose Q such that $\mathrm{KL}(Q||P)$ is not large. Note then that the $\mathrm{KL}(Q||P)$ term can significantly deteriorate these bounds if classifiers with small empirical risk receive low probability from the prior P, i.e. if the prior has been "badly" chosen. The data distribution-defined priors we investigate are specifically constructed to give large weight to classifiers with low true risk, and the KL-divergence between Q and P decays with the sample size.

2.2.1 Choosing a distribution-dependent prior

Suppose an algorithm takes as input a training sample $\mathcal{S}$ from the distribution ν^m over $\mathcal{Z}^m$ and outputs a posterior distribution $Q_\mathcal{S}$ over $\mathcal{H}$. We consider the problem of choosing a prior for $Q_\mathcal{S}$ which attains a sharp PAC-Bayes bound. In this section, we assume that there exists a reference measure μ on $\mathcal{H}$ (when $\mathcal{H}$ is of finite dimensionality this would typically be a uniform measure such as Lebesgue measure) and denote in lower case the density of a measure w.r.t. μ, e.g. $q_\mathcal{S}(h) = \frac{\mathrm{d}Q_\mathcal{S}}{\mathrm{d}\mu}(h)$.

Let $\mathcal{P}_\mathcal{H}$ be the set of probability distributions over $\mathcal{H}$, and in the interest of obtaining a good PAC-Bayes bound for $Q_\mathcal{S}$, consider the minimizer of $\mathrm{KL}(Q_\mathcal{S}||P)$ *in expectation*:

Lemma 2.2.3.

$$\operatorname*{argmin}_{P\in\mathcal{P}_\mathcal{H}} \mathbb{E}_\mathcal{S}[\mathrm{KL}(Q_\mathcal{S}||P)] = \mathbb{E}_\mathcal{S}[Q_\mathcal{S}]. \tag{2.2}$$

Proof.

$$\begin{aligned}
\mathbb{E}_\mathcal{S}[\mathrm{KL}(Q_\mathcal{S}||P)] &= \mathbb{E}_\mathcal{S}\mathbb{E}_{h\sim Q_\mathcal{S}}\left[\ln\frac{q_\mathcal{S}(h)}{p(h)}\right] \\
&= \mathbb{E}_\mathcal{S}\mathbb{E}_{h\sim Q_\mathcal{S}}\left[\ln q_\mathcal{S}(h) + \ln\frac{1}{p(h)}\right] \\
&= \mathbb{E}_\mathcal{S}\mathbb{E}_{h\sim Q_\mathcal{S}}\left[\ln q_\mathcal{S}(h)\right] + \mathbb{E}_\mathcal{S}\left[\int_\mathcal{H} q_\mathcal{S}(h)\ln\frac{1}{p(h)}\mathrm{d}\mu\right] \\
&= \mathbb{E}_\mathcal{S}\mathbb{E}_{h\sim Q_\mathcal{S}}\left[\ln q_\mathcal{S}(h)\right] + \int_\mathcal{H} \mathbb{E}_\mathcal{S}\left[q_\mathcal{S}(h)\right]\ln\frac{1}{p(h)}\mathrm{d}\mu.
\end{aligned}$$

The quantity $\int_\mathcal{H} \mathbb{E}_\mathcal{S}[q_\mathcal{S}(h)]\ln\frac{1}{p(h)}\mathrm{d}\mu$ is the cross entropy between $\mathbb{E}_\mathcal{S}[Q_\mathcal{S}]$ and P and is minimized when $P = \mathbb{E}_\mathcal{S}[Q_\mathcal{S}]$ (Cover and Thomas, 1991). □

This result is noted in this context in Catoni (2007) as is the immediate fact that the resulting expected divergence is equal to the mutual information, $I(h;\mathcal{S})$, between sample and classifier, where a pair $(h,\mathcal{S})$ is viewed as drawn from the joint distribution $\widehat{Q}$ over $\mathcal{H}\times\mathcal{Z}^m$ defined by its density with respect to the product measure $\mu\times\nu^m$, $\widehat{q}(h,\mathcal{S}) := \frac{\mathrm{d}\widehat{Q}}{\mathrm{d}(\mu\times\nu^m)}(h,\mathcal{S}) := q_{\mathcal{S}}(h)$ so that marginalization of $\widehat{Q}$ to $\mathcal{Z}^m$ is simply ν^m,

$$\begin{aligned}\mathbb{E}_{\mathcal{S}}[\mathrm{KL}(Q_{\mathcal{S}}||\mathbb{E}_{\mathcal{S}}[Q_{\mathcal{S}}])] &= \int_{\mathcal{Z}^m}\int_{\mathcal{H}} q_{\mathcal{S}}(h)\ln\left(\frac{q_{\mathcal{S}}(h)}{\mathbb{E}_{\mathcal{S}}[q_{\mathcal{S}}(h)]}\right)\mathrm{d}\mu\mathrm{d}\nu^m\\ &= \int_{\mathcal{H}\times\mathcal{Z}^m}\widehat{q}(h,\mathcal{S})\ln\left(\frac{\widehat{q}(h,\mathcal{S})}{\mathbb{E}_{\mathcal{S}}[q_{\mathcal{S}}(h)]}\right)\mathrm{d}(\mu\times\nu^m)\\ &= I(h;\mathcal{S}),\end{aligned}$$

where the fact that this is a mutual information follows because $\mathbb{E}_{\mathcal{S}}[q_{\mathcal{S}}(h)]$ is simply the marginal densitiy of $\widehat{Q}$ (w.r.t μ) after marginalizing to $\mathcal{H}$ and the constant 1 is the marginal density of $\widehat{Q}$ (w.r.t ν^m) after marginalizing to $\mathcal{Z}^m$. In a sense, implicitly, we want to learn the marginal density $\int_{\mathcal{Z}^m}\widehat{q}(h,\mathcal{S})\mathrm{d}\nu^m = \mathbb{E}_{\mathcal{S}}[q_{\mathcal{S}}(h)]$ and approximate it with the random quantity $q_{\mathcal{S}}(h)$, the sample-based estimate.

In the following for notational convenience we refer to the posterior distribution as Q ommitting the dependence upon $\mathcal{S}$, but it should always be understood to be implicit that Q is a random variable dependent on $\mathcal{S}$. Given the above we could define, for a given posterior Q, the *'optimal' prior*, $P_{\mathrm{opt}}(h) := \mathbb{E}_{\mathcal{S}}[Q]$. We note that PAC-Bayes analysis using this prior appears to be quite difficult since the prior can be difficult to manipulate. As suggested by Catoni we study other more flexible choices of prior which enable us to obtain very sharp PAC-Bayes bounds. We consider the case when the posterior and prior are of the following form,

$$q(h) := \frac{dQ}{d\mu}(h) := \frac{1}{Z}e^{-F_Q(h)} \qquad p(h) := \frac{dP}{d\mu}(h) := \frac{1}{Z'}e^{-F_P(h)}, \tag{2.3}$$

where F_Q, F_P are "energy functions", to be chosen, and $Z = \int_{\mathcal{H}} e^{-F_Q(h)}\mathrm{d}\mu$, $Z' = \int_{\mathcal{H}} e^{-F_P(h)}\mathrm{d}\mu$. We note the following upper bound on the KL divergence, which reduces obtaining a bound on the KL divergence to establishing a PAC-Bayesian concentration result for the energy functions.

Lemma 2.2.4. *For Q and P as defined by (2.3),*

$$\mathrm{KL}(Q||P) \leq \left(\mathbb{E}_{h\sim Q} - \mathbb{E}_{h\sim P}\right)\left[F_P(h) - F_Q(h)\right]. \tag{2.4}$$

Proof.

$$\begin{aligned}\mathrm{KL}(Q||P) &= \mathbb{E}_{h\sim Q}\ln\frac{Z'e^{-F_Q(h)}}{Ze^{-F_P(h)}}\\ &= \mathbb{E}_{h\sim Q}\left[F_P(h) - F_Q(h)\right] - \ln\frac{\int e^{-F_Q(h)}\mathrm{d}\mu}{Z'}\\ &= \mathbb{E}_{h\sim Q}\left[F_P(h) - F_Q(h)\right] - \ln\int p(h)e^{F_P(h)-F_Q(h)}\mathrm{d}\mu\\ &\leq \left(\mathbb{E}_{h\sim Q} - \mathbb{E}_{h\sim P}\right)\left[F_P(h) - F_Q(h)\right],\end{aligned} \tag{2.5}$$

where the final line follows from the convexity of $-\ln(\cdot)$. □

Note that the r.h.s. of (2.5) is precisely the type of quantity that PAC-Bayes theory provides a bound for. In particular, the choice $F_P = \mathbb{E}_{\mathcal{S}}[F_Q]$ seems natural and we remark that (2.5) is then reduced to obtaining a concentration inequality for F_Q to its expectation. Thus whereas P_{opt} appears rather difficult to manipulate, the prior and posterior defined by choosing $H_P = \mathbb{E}_{\mathcal{S}}[H_Q]$ seems amenable to analysis and a good compromise.

2.3 Prediction by Gibbs algorithms

We first consider posterior and prior densities, w.r.t. μ, over $\mathcal{H}$ of the following forms:

$$q(h) = \frac{1}{Z} e^{-\left(\gamma \widehat{\text{risk}}_{\mathcal{S}}(h) + \eta F_Q(h)\right)} \tag{2.6}$$

$$p(h) = \frac{1}{Z'} e^{-(\gamma \text{risk}(h) + \eta F_P(h))}. \tag{2.7}$$

where $F_Q : \mathcal{H} \to \mathbb{R}$, $F_P : \mathcal{H} \to \mathbb{R}$ are regularization functions, and Z a normalization constant. The unregularized case corresponds to "Gibbs algorithms", (e.g. Catoni, 2007) and is a type of stochastic empirical risk minimization-type prediction. $F_Q(\cdot)$ and $F_P(\cdot)$ may be different and in particular we will consider the special case where $F_Q(\cdot)$ is a sample statistic, allowing us to perform data-dependent regularization.

We note that Lemma 2.2.4 implies the following upper bound on the KL divergence

$$\text{KL}(Q||P) \quad \leq \quad (\mathbb{E}_{h\sim Q} - \mathbb{E}_{h\sim P}) \left[\gamma \text{risk}(h) - \gamma \widehat{\text{risk}}_{\mathcal{S}}(h) + \eta F_P(h) - \eta F_Q(h)\right]. \tag{2.8}$$

As we will see later, for suitable choices of parameters γ and η, this divergence decays with the sample. We now consider several choices of $F_Q(\cdot)$ and $F_P(\cdot)$ and give PAC-Bayes bounds for the resulting Gibbs classifiers.

2.3.1 The non-regularized case : $\eta = 0$

We recall that the distribution D over $\mathcal{X} \times \mathcal{Y}$ is unknown, hence so is the prior distribution given by (2.7). To obtain a bound, we need to bound the KL divergence $\text{KL}(Q||P)$. With reference to (2.8), in the situation where $\eta = 0$ such an upper bound can be obtained given an upper bound for $\text{risk}(G_Q) - \widehat{\text{risk}}_{\mathcal{S}}(G_Q)$ and a lower bound for $\text{risk}(G_P) - \widehat{\text{risk}}_{\mathcal{S}}(G_P)$, and such bounds can obtained via Theorem 2.2.2.

Lemma 2.3.1. *Let P and Q be defined as in (2.6) and (2.7) with $\eta = 0$ then with probability at least $1 - \delta$, the following hold simultaneously,*

$$\forall\, Q \text{ on } \mathcal{H} : \text{kl}(\widehat{\text{risk}}_{\mathcal{S}}(G_Q), \text{risk}(G_Q)) \leq \frac{1}{m}\left(\text{KL}(Q||P) + \ln\frac{\xi(m)}{\delta}\right) \tag{2.9}$$

$$\text{KL}(Q||P) \leq \gamma\sqrt{\frac{2}{m}}\sqrt{\ln\frac{\xi(m)}{\delta}} + \frac{\gamma^2}{2m}. \tag{2.10}$$

Proof. Equation (2.9) is just the Seeger bound of Theorem 2.2.2. Then from (2.9), applied for the choices

$Q = Q$ and $Q = P$, and from (2.1) we obtain that, simultaneously,

$$\begin{aligned} \mathrm{risk}(G_Q) - \widehat{\mathrm{risk}}_{\mathcal{S}}(G_Q) &\leq \frac{1}{\sqrt{2m}} \sqrt{\mathrm{KL}(Q||P) + \ln \frac{\xi(m)}{\delta}}\,, \\ -\left(\mathrm{risk}(G_P) - \widehat{\mathrm{risk}}_{\mathcal{S}}(G_P)\right) &\leq \frac{1}{\sqrt{2m}} \sqrt{\ln \frac{\xi(m)}{\delta}}\,. \end{aligned}$$

Together with (2.8) the last inequalities give,

$$\begin{aligned} \mathrm{KL}(Q||P) &\leq \gamma\left(\mathrm{risk}(G_Q) - \widehat{\mathrm{risk}}_{\mathcal{S}}(G_Q)\right) - \gamma\left(\mathrm{risk}(G_P) - \widehat{\mathrm{risk}}_{\mathcal{S}}(G_P)\right) \\ &\leq \frac{\gamma}{\sqrt{2m}} \sqrt{\mathrm{KL}(Q||P) + \ln \frac{\xi(m)}{\delta}} + \frac{\gamma}{\sqrt{2m}} \sqrt{\ln \frac{\xi(m)}{\delta}}. \end{aligned}$$

If $\mathrm{KL}(Q||P) \leq \frac{\gamma}{\sqrt{2m}} \sqrt{\ln \frac{\xi(m)}{\delta}}$, we are done. Otherwise, by straightforward algebraic manipulations we then obtain the following inequality, which, together with the fact that $\mathrm{KL}(Q||P) \geq 0$, directly implies the result.

$$(\mathrm{KL}(Q||P))^2 - \frac{2\gamma}{\sqrt{2m}} \sqrt{\ln \frac{\xi(m)}{\delta}}\, \mathrm{KL}(Q||P) + \frac{\gamma^2}{2m} \ln \frac{\xi(m)}{\delta} \leq \frac{\gamma^2}{2m}\, \mathrm{KL}(Q||P) + \frac{\gamma^2}{2m} \ln \frac{\xi(m)}{\delta}.$$

□

Thus, Theorem 2.2.2 can be specialized to the following bound.

Theorem 2.3.2. *Let P and Q be defined as in (2.6) and (2.7) with $\eta = 0$, then*

$$\begin{aligned} \mathbb{P}_{\mathcal{S}}\left(\mathrm{kl}(\widehat{\mathrm{risk}}_{\mathcal{S}}(G_Q), \mathrm{risk}(G_Q)) \leq \frac{1}{m}\left(\frac{\gamma}{\sqrt{m}} \sqrt{\ln \frac{\xi(m)}{\delta}} + \frac{\gamma^2}{4m} + \ln \frac{\xi(m)}{\delta}\right)\right) &\geq 1 - \delta, \\ \mathbb{P}_{\mathcal{S}}\left(\mathrm{risk}(G_Q) \leq C^\star\, \widehat{\mathrm{risk}}_{\mathcal{S}}(G_Q) + \frac{C^\star}{C \cdot m}\left(\frac{\gamma}{\sqrt{m}} \sqrt{\ln \frac{2\xi(m)}{\delta}} + \frac{\gamma^2}{4m} + \ln \frac{2}{\delta}\right)\right) &\geq 1 - \delta \end{aligned}$$

Proof. The first result is obtained by combining the two components of Lemma 2.3.1. The second result is obtained by applying the union bound to (2.10) and Catoni's bound of Theorem 2.2.2. □

Observe that for a large value of γ, the posterior Gibbs classifier G_Q will be concentrated on the classifiers of $\mathcal{H}$ with smallest empirical risk. Hence the two bounds of Theorem 2.3.2 are risk bounds for a type of stochastic empirical risk minimization algorithm. Since the KL-divergence term has been evaluated and is small, it appears that there is no component of the bound that depends on the complexity of the learning problem or the class of classifiers. In fact the parameter that controls the effective complexity is the "inverse temperature", γ (or γ^2 if we view it in the role of a VC dimension). If the problem is 'easy' in the sense that the measure of the set of classifiers with low empirical risk is not too small then a low value of γ will deliver low empirical risk for the Gibbs classifier. If, however, the measure of the classifiers that have low empirical risk is very small (as would be likely if the function class itself is large) then we require a larger value of γ before the Gibbs risk is controlled. The complexity that γ measures is related to the fit between input distribution and function class in that it will depend on the measure of the distribution Q on the low empirical risk functions.

In practice γ would need to be chosen from a grid Γ of values in response to the particular training problem. Hence, in order to apply the bound we would need to use the union bound over the $|\Gamma|$ applications of the bound resulting in an extra $\log(|\Gamma|)$ term in the right hand side brackets. Another possibility would be to make use the generalized union bound known as Occam's hammer (Blanchard and Fleuret, 2007).

2.3.2 Regularization with $F_Q(\cdot) = F_P(\cdot)$

Given the above argument it appears necessary to control function class capacity in this model in order to deliver low empirical Gibbs risk. We therefore consider the presence of regularization terms in (2.6), (2.7) which encode a preference for classifiers which satisfy some notion of simplicity. The flexibility of this model is such that, with reference to (2.8), when $F_Q(\cdot) = F_P(\cdot)$, the bounds of Theorem 2.3.2 hold for this case. We can therefore apply arbitrary (non data-dependent) regularization and attain the same bound of Theorem 2.3.2, and there are many natural possibilities. For example, if $\mathcal{H}$ is equipped with a norm $||\cdot||_{\mathcal{H}}$ we can choose $F_Q(\cdot) = F_P(\cdot) = ||\cdot||_{\mathcal{H}}$. This should permit learning with smaller γ.

2.3.3 Regularization in the intrinsic data geometry

The flexibility of this model further permits, in a straightforward way, regularization w.r.t. the geometry defined by the *unknown* data-generating distribution, and we detail one way of achieving this. The regularization methods considered in Section 2.3.2 utilise a geometry which is extrinsic to the data, that is, determined by the ambient representation space rather than the intrinsic geometry of data. For example, if the data generating distribution has support on some submanifold of the ambient space, then encouraging smoothness on the manifold ought to be more suitable for learning (since if the structure of data is a key factor in the learnability of a task, it is the intrinsic geometry which will capture this relevant structure most accurately). In general, when using a regularizer informed by the intrinsic geometry of the data-generating distribution the prior and posterior regularizers must be different since the posterior regularizer will be an empirical quantity (here, chosen to be an estimate, based on the sample, of the prior regularizer).

Given a sample $\mathcal{S} = \{(X_1, Y_1), ...(X_m, Y_m)\} \cup \{X_{m+1}, ...X_n\}$, we consider regularizing via the following *smoothness functional*, typical in semi-supervised learning (e.g. Belkin et al., 2004; Zhu et al., 2003a), over functions from some function class $\mathcal{H}$:

$$\widehat{U}_{\mathcal{S}}(h) := \frac{1}{n(n-1)} \sum_{ij} (h(X_i) - h(X_j))^2 W(X_i, X_j) \tag{2.11}$$

where the symmetric $W : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ captures similarity or "weight" between data points, for example $W(\boldsymbol{x}, \boldsymbol{x}') = \begin{cases} 1 \text{ if } ||\boldsymbol{x} - \boldsymbol{x}'|| \leq \epsilon \\ 0 \text{ otherwise} \end{cases}$ or $W(\boldsymbol{x}, \boldsymbol{x}') = e^{-||\boldsymbol{x}-\boldsymbol{x}'||^2}$ for some norm $||\cdot||$ over $\mathcal{X}$. Note that $\widehat{U}_{\mathcal{S}}(h) = \frac{2}{n(n-1)} \boldsymbol{h}^\top \boldsymbol{L} \boldsymbol{h}$ where $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$ is the graph Laplacian of a graph $\mathcal{G}$ whose vertices are the sample instances and whose *edge weights* are controlled by W, and $D_{ij} = \delta_{ij} \sum_k W_{ik}$ and where $\boldsymbol{h} \in \mathbb{R}^n$ is the "point evaluation" of h on the sample, $h_i := h(\boldsymbol{x}_i)$. Minimizing (2.11) encourages

functions to be smooth over the sample $\mathcal{S}$. Note that $\widehat{U}_{\mathcal{S}}(h)$ is a U-statistic of order 2 with *kernel* $f_h(X_i, X_j) := (h(X_i) - h(X_j))^2 W(X_i, X_j)$ indexed by $\mathcal{H}$. A family of U-statistics indexed by a function space is often called a U-process. We suppose that the weights are bounded, $|W(\boldsymbol{x}, \boldsymbol{x}')| \leq w$, for example if $W(\boldsymbol{x}, \boldsymbol{x}') = e^{-||\boldsymbol{x}-\boldsymbol{x}'||^2}$ we have $w = 1$, and that $\sup_{h\in\mathcal{H}, \boldsymbol{x}\in\mathcal{X}} |h(\boldsymbol{x})| = b$.

A series of results (Bousquet et al., 2003b; Hein et al., 2007) demonstrate that under certain conditions on the distribution of instances, certain constructions of graph Laplacian converge to a generalized Laplace operator on the support of the data generating distribution and the smoothness functional converges to a natural distribution-dependent dirichlet energy functional over functions defined over the data.

We choose $F_Q(\cdot) = \widehat{U}_{\mathcal{S}}(\cdot)$ so that,

$$q(h) = \frac{1}{Z} e^{-\left(\gamma \widehat{\text{risk}}_{\mathcal{S}}(h) + \eta \widehat{U}_{\mathcal{S}}(h)\right)}. \tag{2.12}$$

The exponent seeks to minimize empirical risk plus the smoothness on the graph formed on the sample, as is a typical methodology in semi-supervised learning (Belkin et al., 2006, 2004).

We further choose $F_P(h) = U(h) := \mathbb{E}_{X,X'}[(h(X) - h(X'))^2 W(X, X')] = \mathbb{E}_{\mathcal{S}}[\widehat{U}_{\mathcal{S}}(h)]$, giving the prior $p(h) = \frac{1}{Z'} e^{-(\gamma \text{risk}(h) + \eta U(h))}$.

Convergence of the smoothness functional

We consider PAC-Bayes convergence of the U-process (see Ralaivola et al. (2010) for an alternative PAC-Bayes analysis of U-statistics). Let $\mathcal{S} = \{X_1, ...X_n\}$ be an i.i.d. sample. For any second-order U-statistic $\widehat{U}_{\mathcal{S}}(h) = \frac{1}{n(n-1)} \sum_{i\neq j} f_h(X_i, X_j)$ with expectation $U(h)$, and with kernel $f_h(x, x')$ indexed by $\mathcal{H}$ and bounded, $a \leq f_h(x, x') \leq b$, we have the following.

Theorem 2.3.3. *For all t, any prior P and simultaneously for all posteriors Q over $\mathcal{H}$,*

$$\mathbb{P}_{\mathcal{S}}\left(\mathbb{E}_{h\sim Q}[\widehat{U}_{\mathcal{S}}(h) - U(h)] \leq \frac{1}{t}\left(\text{KL}(Q||P) + \frac{t^2(b-a)^2}{2n} + \ln\left(\frac{1}{\delta}\right)\right)\right) \geq 1 - \delta \tag{2.13}$$

$$\mathbb{P}_{\mathcal{S}}\left(\mathbb{E}_{h\sim Q}[U(h) - \widehat{U}_{\mathcal{S}}(h)] \leq \frac{1}{t}\left(\text{KL}(Q||P) + \frac{t^2(b-a)^2}{2n} + \ln\left(\frac{1}{\delta}\right)\right)\right) \geq 1 - \delta. \tag{2.14}$$

In particular, choosing $t = \sqrt{n}$ gives $\mathcal{O}(\frac{1}{\sqrt{n}})$ convergence.

Proof. We note that Theorem 2.2.1 implies that with probability at least $1 - \delta$, $\forall Q$ on $\mathcal{H}$:

$$\mathbb{E}_{h\sim Q}[\widehat{U}_{\mathcal{S}}(h) - U(h)] \leq \frac{1}{t}\left(\text{KL}(Q||P) + \ln\left(\frac{1}{\delta}\mathbb{E}_{h\sim P}\mathbb{E}_{\mathcal{S}}\left[e^{t(\widehat{U}_{\mathcal{S}}(h) - U(h))}\right]\right)\right),$$

so we simply need to bound $\mathbb{E}_{\mathcal{S}}\left[e^{t(\widehat{U}_{\mathcal{S}}(h) - U(h))}\right]$. Employing Hoeffding's canonical decomposition of U-statistics into forward martingales (e.g. Serfling, 1980), let,

$$V_k := \sum_{i=1}^{k} \left(\mathbb{E}[f_h(X_i, X) \mid X_i] - U(h)\right)$$

$$W_k := \sum_{j=1}^{k} \sum_{i=1}^{j-1} \left(f_h(X_i, X_j) + U(h) - \mathbb{E}[f_h(X_i, X) \mid X_i] - \mathbb{E}[f_h(X, X_j) \mid X_j]\right),$$

so that, $\widehat{U}_{\mathcal{S}}(h) - U(h) = \frac{2}{n}V_n + \frac{2}{n(n-1)}W_n$. We then have that,

$$V_k - V_{k-1} = \mathbb{E}[f_h(X_k, X) \mid X_k] - U(h)$$
$$W_k - W_{k-1} = \sum_{i=1}^{k-1} \left(f_h(X_i, X_k) + U(h) - \mathbb{E}[f_h(X_i, X) \mid X_i] - \mathbb{E}[f_h(X_k, X) \mid X_k]\right),$$

and note the martingale structure $\mathbb{E}_{X_k}[V_k - V_{k-1}] = \mathbb{E}_{X_k}[W_k - W_{k-1}] = 0$. Note further that,

$$V_k - V_{k-1} + \frac{1}{n-1}(W_k - W_{k-1}) = \frac{n-k}{n-1}(\mathbb{E}[f_h(X_k, X) \mid X_k] - U(h)) + \frac{1}{n-1}\sum_{i=1}^{k-1} f_h(X_i, X_k) - \mathbb{E}[f_h(X_i, X) \mid X_i],$$

so that,

$$|V_k - V_{k-1} + \frac{1}{n-1}(W_k - W_{k-1})| \leq (b-a)\frac{n-k}{n-1} + (b-a)\frac{k-1}{n-1} = b - a. \tag{2.15}$$

Now,

$$\begin{aligned}
&\mathbb{E}_{\mathcal{S}}\left[e^{t(\widehat{U}_{\mathcal{S}}(h) - U(h))}\right] = \mathbb{E}_{\mathcal{S}}\left[e^{\frac{2t}{n}\sum_{i=1}^{n} V_i - V_{i-1} + \frac{1}{n-1}(W_i - W_{i-1})}\right] \\
&= \mathbb{E}_{X_1,\ldots X_{n-1}}\left[\mathbb{E}_{X_n}\left[e^{\frac{2t}{n}\sum_{i=1}^{n} V_i - V_{i-1} + \frac{1}{n-1}(W_i - W_{i-1})} \mid X_1, \ldots X_{n-1}\right]\right] \\
&= \mathbb{E}_{X_1,\ldots X_{n-1}}\left[e^{\frac{2t}{n}\sum_{i=1}^{n-1} V_i - V_{i-1} + \frac{1}{n-1}(W_i - W_{i-1})}\mathbb{E}_{X_n}\left[e^{\frac{2t}{n}\left(V_n - V_{n-1} + \frac{1}{n-1}(W_n - W_{n-1})\right)}\right]\right] \\
&\leq \mathbb{E}_{X_1,\ldots X_{n-1}}\left[e^{\frac{2t}{n}\sum_{i=1}^{n-1} V_i - V_{i-1} + \frac{1}{n-1}(W_i - W_{i-1})}\right] e^{\frac{t^2(b-a)^2}{2n^2}} \\
&\vdots \\
&\leq \prod_{i=1}^{n} e^{\frac{t^2(b-a)^2}{2n^2}} \;=\; e^{\frac{t^2(b-a)^2}{2n}},
\end{aligned}$$

where in the final lines we used Hoeffding's lemma, Lemma B.0.4, combined with (2.15) recursively. This proves (2.13), and (2.14) follows by a symmetrical argument. □

We can now give the following bound for the classification risk of the Gibbs classifier G_Q drawn from the distribution (2.12) over $\mathcal{H}$:

Theorem 2.3.4. *For $\eta < \sqrt{n}$,*

$$\mathbb{P}_{\mathcal{S}}\left(\mathrm{kl}(\widehat{\mathrm{risk}}_{\mathcal{S}}(G_Q), \mathrm{risk}(G_Q)) \leq \frac{1}{m}\left(A^2 + B + A\sqrt{2B + A^2} + \ln\frac{\xi(m)}{\delta}\right)\right) \geq 1 - \delta,$$

where

$$\begin{aligned}
A &:= \frac{\gamma\sqrt{n}}{2\sqrt{m}(\sqrt{n} - \eta)} = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right) \\
B &:= \frac{\sqrt{n}}{\sqrt{n} - \eta}\left(\gamma\sqrt{\frac{2}{m}\ln\frac{2\xi(m)}{\delta}} + \frac{2\eta}{\sqrt{n}}\left(32b^4w^2 + \ln\frac{4}{\delta}\right)\right) = \mathcal{O}\left(\sqrt{\frac{\ln m}{m}}\right).
\end{aligned}$$

Proof. From (2.8) we have

$$\begin{aligned}
\mathrm{KL}(Q||P) \;\leq\; &\gamma(\mathrm{risk}(G_Q) - \widehat{\mathrm{risk}}_{\mathcal{S}}(G_Q)) + \gamma(\widehat{\mathrm{risk}}_{\mathcal{S}}(G_P) - \mathrm{risk}(G_P)) \\
&+\eta\mathbb{E}_{h\sim Q}\left[U(h) - \widehat{U}_{\mathcal{S}}(h)\right] + \eta\mathbb{E}_{h\sim P}\left[\widehat{U}_{\mathcal{S}}(h) - U(h)\right].
\end{aligned} \tag{2.16}$$

With probability at least $1 - \frac{\delta}{2}$, we have from Theorem 2.2.2,

$$\forall\, Q \,\text{on}\, \mathcal{H} : \mathrm{kl}(\widehat{\mathrm{risk}}_{\mathcal{S}}(G_Q), \mathrm{risk}(G_Q)) \leq \frac{1}{m}\left(\mathrm{KL}(Q||P) + \ln\frac{2\xi(m)}{\delta}\right). \tag{2.17}$$

Then from (2.17), applied for the choices $Q = Q$ and $Q = P$, and from (2.1) we obtain that, with probability at least $1 - \frac{\delta}{2}$, simultaneously,

$$\begin{aligned}
\mathrm{risk}(G_Q) - \widehat{\mathrm{risk}}_{\mathcal{S}}(G_Q) &\leq \frac{1}{\sqrt{2m}}\sqrt{\mathrm{KL}(Q||P) + \ln\frac{2\xi(m)}{\delta}}\,, \\
-\left(\mathrm{risk}(G_P) - \widehat{\mathrm{risk}}_{\mathcal{S}}(G_P)\right) &\leq \frac{1}{\sqrt{2m}}\sqrt{\ln\frac{2\xi(m)}{\delta}}\,.
\end{aligned}$$

And now noting that, because $|h(\boldsymbol{x})| \leq b$, $W(\boldsymbol{x}, \boldsymbol{x}') \leq w$, the kernel satisfies $|f_h(\boldsymbol{x}, \boldsymbol{x}')| \leq 4b^2 w$, Theorem 2.3.3 applied to the final terms in (2.16), for the choices $Q = Q$ and $Q = P$, together with the union bound gives that, with probability at least $1 - \frac{\delta}{2}$, simultaneously,

$$\begin{aligned}
\eta\mathbb{E}_{h\sim Q}\left[U(h) - \widehat{U}_{\mathcal{S}}(h)\right] &\leq \frac{\eta}{\sqrt{n}}\left(\mathrm{KL}(Q||P) + 32b^4w^2 + \ln\frac{4}{\delta}\right) \\
\eta\mathbb{E}_{h\sim P}\left[\widehat{U}_{\mathcal{S}}(h) - U(h)\right] &\leq \frac{\eta}{\sqrt{n}}\left(32b^4w^2 + \ln\frac{4}{\delta}\right).
\end{aligned}$$

The union bound then implies that with probability at least $1 - \delta$ over the draw of $\mathcal{S}$,

$$\begin{aligned}
\mathrm{KL}(Q||P) &\leq \gamma\sqrt{\frac{1}{2m}\left(\mathrm{KL}(Q||P) + \ln\frac{2\xi(m)}{\delta}\right)} + \gamma\sqrt{\frac{1}{2m}\ln\frac{2\xi(m)}{\delta}} \\
&\quad + \frac{\eta}{\sqrt{n}}\left(\mathrm{KL}(Q||P) + 32b^4w^2 + \ln\frac{4}{\delta}\right) + \frac{\eta}{\sqrt{n}}\left(32b^4w^2 + \ln\frac{4}{\delta}\right) \\
&\leq \gamma\sqrt{\frac{1}{2m}\mathrm{KL}(Q||P)} + \frac{\eta}{\sqrt{n}}\mathrm{KL}(Q||P) + \gamma\sqrt{\frac{2}{m}\ln\frac{2\xi(m)}{\delta}} + \frac{2\eta}{\sqrt{n}}\left(32b^4w^2 + \ln\frac{4}{\delta}\right) \\
\left(\sqrt{\mathrm{KL}(Q||P)} - \frac{1}{\sqrt{2}}A\right)^2 &\leq B + \frac{A^2}{2} \\
\mathrm{KL}(Q||P) &\leq A^2 + B + A\sqrt{2B + A^2},
\end{aligned}$$

which we plug into (2.17). □

We remark that the ease with which we can obtain this bound for regularization w.r.t. the geometry defined by the unknown data-generating distribution, with apparently little deterioration in the bound, is unusual and that in classical frameworks this type of structuring of a function class usually results in significant deterioration in the bound.

2.4 Prediction by RKHS regularization

We now extend the localization framework to the more practical setting of predicting with a Gaussian process whose mean is the solution to an empirical risk minimization with RKHS regularization, such as an SVM solution. We consider a separable[1] RKHS $\mathcal{H}_K = \overline{\mathrm{span}\{K(\boldsymbol{x}, \cdot) : \boldsymbol{x} \in \mathcal{X}\}}$, for some positive-definite kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, of real-valued functions on $\mathcal{X}$ with inner product $\langle \cdot, \cdot \rangle_K$

[1] This is a mild condition, an RKHS $\mathcal{H}_K$ is separable if $\mathcal{X}$ is and if the kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is continuous (Krein, 1963).

defined by $\langle K(\boldsymbol{x},\cdot), K(\boldsymbol{x}',\cdot)\rangle_K := K(\boldsymbol{x},\boldsymbol{x}')$. The class $\mathcal{H}_K$ can be identified as binary classifiers via $h_{\text{class}}(\boldsymbol{x}) = \text{sgn}(h(\boldsymbol{x})) = \text{sgn}(\langle h, K(\boldsymbol{x},\cdot)\rangle_K)$. For simplicity we suppose that $\mathcal{X}$ is a compact Hausdorff space.

For any chosen loss function $\ell : \mathcal{Y}\times\mathcal{Y}\to\mathbb{R}$, we are interested in the classifiers,

$$h^*_{\mathcal{S}} := \underset{h\in\mathcal{H}_K}{\text{argmin}}\{\widehat{\text{risk}}^{\ell}_{\mathcal{S}}(h) + \eta||h||^2_K\} \qquad \text{and} \qquad h^* := \mathbb{E}_{\mathcal{S}}[h^*_{\mathcal{S}}],$$

where η is a regularization parameter and expectation is taken with respect to samples $\mathcal{S}$ with m labelled points. For our intended applications, typically $\widehat{\text{risk}}^{\ell}_{\mathcal{S}}(\cdot)$ will be convex so that $h^*_{\mathcal{S}}$ is unique and h^* well-defined[2].

2.4.1 Prior and posterior distributions

Our posterior Q and prior P will be Gaussian processes over $\mathcal{X}$ with mean $h^*_{\mathcal{S}}$ and h^* respectively and covariance $\frac{1}{\gamma}K(\boldsymbol{x},\boldsymbol{x}')$, where γ is a parameter which controls the variance of these processes. To define this we actually define a distribution over the Hilbert space $\mathcal{L}^2(\mathcal{X},\Sigma,\nu)$ of all square integrable real-valued functions on $\mathcal{X}$ with inner product $\langle h, g\rangle_{\mathcal{L}^2} := \int_{\mathcal{X}} h(\boldsymbol{x})g(\boldsymbol{x})\nu(\mathrm{d}\boldsymbol{x})$. Consider the countable orthonormal basis $\{\phi_i\}$ for $\mathcal{L}^2(\mathcal{X},\Sigma,\nu)$ provided by the eigenfunctions of the integral operator A_K defined by $(A_K h)(\boldsymbol{x}) := \int K(\boldsymbol{x},\boldsymbol{x}')h(\boldsymbol{x}')\nu(\mathrm{d}\boldsymbol{x}')$, i.e. such that $A_K(\phi_i) = \lambda_i\phi_i$, for eigenvalues $\{\lambda_i\}$ and $\langle\phi_i,\phi_j\rangle_{\mathcal{L}^2} = \int_{\mathcal{X}}\phi_i(\boldsymbol{x})\phi_j(\boldsymbol{x})\nu(\mathrm{d}\boldsymbol{x}) = \delta_{ij}$. Denote $h_i := \langle h,\phi_i\rangle_{\mathcal{L}^2}$ and consider the isomorphism $I : \mathcal{L}^2(\mathcal{X},\Sigma,\nu)\to\ell^2$ given by $I(h) = (h_i)$ identifying $\mathcal{L}^2(\mathcal{X},\Sigma,\nu)$ with the space of square summable real-valued sequences. Denote by N_{a,σ^2} the one-dimensional Gaussian measure on (the Borel σ-algebra on) $\mathbb{R}$ with mean a and variance σ^2. We define,

$$Q_i := N_{h^*_{\mathcal{S},i},\frac{1}{\gamma}\lambda_i} \quad \text{and } P_i := N_{h^*_i,\frac{1}{\gamma}\lambda_i}, \tag{2.18}$$

where $h^*_{\mathcal{S},i} = \langle h^*_{\mathcal{S}},\phi_i\rangle_{\mathcal{L}^2}$, $h^*_i = \langle h^*,\phi_i\rangle_{\mathcal{L}^2}$, as above. We then define the product measures,

$$Q := \prod_{i=1}^{\infty} Q_i = \prod_{i=1}^{\infty} N_{h^*_{\mathcal{S},i},\frac{1}{\gamma}\lambda_i} \quad \text{and } P := \prod_{i=1}^{\infty} P_i = \prod_{i=1}^{\infty} N_{h^*_i,\frac{1}{\gamma}\lambda_i}. \tag{2.19}$$

The following result is the subject of (Da Prato, 2006, Chapter 1) and are outlined in Appendix C: Q and P define probability measures on the space $\mathbb{R}^\infty$ of all real-valued sequences whenever the operator A_K is of trace class, that is $\sum_i\lambda_i < \infty$. A_K is of trace class whenever $(\mathcal{X},\Sigma,\nu)$ is a finite measure space and $K(\cdot,\cdot)$ is bounded – this follows by applying Mercer's theorem,

$$\begin{aligned}\sum_i \lambda_i &= \int_{\mathcal{X}}\sum_i \lambda_i\phi_i(x)\phi_i(x)\nu(\mathrm{d}x)\\ &= \int_{\mathcal{X}} K(x,x)\nu(\mathrm{d}x) < \infty,\end{aligned}$$

[2]One may wonder whether $\mathcal{S}\to h^*_{\mathcal{S}}$ is a measurable function (i.e. whether $h^*_{\mathcal{S}}$ is a valid random variable). In essence in typical situations this will follow from the continuity of the map $\mathcal{S}\to h^*_{\mathcal{S}}$ since we will usually be able to work with a natural topology on $\mathcal{Z}^m$ such that D^m is a Borel measure. For example if $\mathcal{X} = \mathbb{R}^n$ and $\mathcal{Y}\subseteq\mathbb{R}$ and D^m is Borel on $\mathcal{Z}^m = (\mathcal{X}\times\mathcal{Y})^m$ then the continuity will follow from a general result on the continuity of the argmin of a strictly convex objective along the lines of Theorem B.0.3, and measurability follows immediately. For results on the measurability of the SVM function see (Steinwart and Christmann, 2008, Chapter 5).

the first equality following because Mercer's theorem ensures absolute and uniform convergence of $K(x,x) = \sum_i \lambda_i \phi_i(x)\phi_i(x)$ so that integration and summation commute. Further, Q and P are defined on $\mathbb{R}^\infty$ but their support is precisely ℓ^2, i.e. $\mathcal{L}^2(\mathcal{X}, \Sigma, \nu)$ under the above isomorphism. Thus we refer to Q and P as both measures on ℓ^2 and $\mathcal{L}^2(\mathcal{X}, \Sigma, \nu)$ by isomorphism[3].

For practical prediction purposes we are interested in the following result: when $(\mathcal{X}, \Sigma, \nu)$ is a finite measure space and $\mathcal{X}$ is compact, prediction with the Gibbs classifier drawn from the posterior (2.19) is equivalent to predicting with a Gaussian process $\{G_{\boldsymbol{x}}\}_{\boldsymbol{x}\in\mathcal{X}}$ on $\mathcal{X}$ with mean $\mathbb{E}[G_{\boldsymbol{x}}] = h^*_{\mathcal{S}}(\boldsymbol{x})$ and covariance $\mathbb{E}[(G_{\boldsymbol{x}} - \mathbb{E}[G_{\boldsymbol{x}}])(G_{\boldsymbol{x}'} - \mathbb{E}[G_{\boldsymbol{x}'}])] = \frac{1}{\gamma}K(\boldsymbol{x}, \boldsymbol{x}')$. This equivalence is a special case of the Karhunen-Loève theorem outlined in Appendix D.

We recall the following identity:

Lemma 2.4.1. *(Wahba, 1990, Lemma 1.1.1) For any $h \in \mathcal{L}^2(\mathcal{X}, \Sigma, \nu)$,*

$$h \in \mathcal{H}_K \Leftrightarrow \sum_{i=1}^{\infty} \frac{1}{\lambda_i} h_i^2 < \infty,$$

(where we define $\frac{0}{0} = 0$). Whenever $h \in \mathcal{H}_K$,

$$||h||_K^2 = \sum_{i=1}^{\infty} \frac{1}{\lambda_i} h_i^2.$$

2.4.2 Deriving a PAC-Bayes bound for Q

To obtain a PAC-Bayes bound for the Gibbs classifier drawn from Q, we need to evaluate the relative entropy between the Gaussian measures Q and P. In the finite dimensional setting this would be straightforward and follow from a well-known result. For a Gaussian measure on an infinite dimensional Hilbert space we need to take more care, and the following lemma essentially states that the well-known formula for the relative entropy between finite dimensional Gaussian distributions extends naturally to our case.

Lemma 2.4.2. $\mathrm{KL}(Q||P) = \frac{\gamma}{2}||h^*_{\mathcal{S}} - h^*||_K^2$.

Proof. We define by $A_K^{\frac{1}{2}}$ the unique self-adjoint positive definite operator on $\mathcal{L}^2(\mathcal{X}, \Sigma, \nu)$ such that $A_K^{\frac{1}{2}}\phi_i = \lambda_i^{\frac{1}{2}}\phi_i$ and by $A_K^{-\frac{1}{2}}$, A_K^{-1} the operators defined on $\mathrm{span}\{\phi_i \ : \ \lambda_i \neq 0\}$ such that $A_K^{-\frac{1}{2}}\phi_i = \lambda_i^{-\frac{1}{2}}\phi_i$ and $A_K^{-1}\phi_i = \lambda_i^{-1}\phi_i$. For any $h \in \mathcal{H}_K$ if we define $f = \sum_{i=1}^{\infty} f_i\phi_i$ by $f_i = \frac{h_i}{\sqrt{\lambda_i}}$, so that

[3]To build intuition, when $\mathcal{H}_K$ is of finite dimensionality (i.e. only a finite number of the λ_i are non-zero) then the measures are just finite-dimensional Gaussian distributions on the subspace spanned by the eigenfunctions corresponding to non-zero eigenvalues and have (Gaussian) density (w.r.t. Lebesgue measure under the above isomorphism),

$$q(h) = \frac{1}{Z} e^{-\frac{\gamma}{2}||h - h^*_{\mathcal{S}}||_K^2} \text{ and } \qquad p(h) = \frac{1}{Z'} e^{-\frac{\gamma}{2}||h - h^*||_K^2}, \tag{2.20}$$

where, Z, Z' enforce normalization. In the general (possibly infinite-dimensional) case we are building the corresponding distributions but note that the densities (2.20) no longer make sense and in fact there is no analogue of Lebesgue measure on an infinite dimensional vector space.

$h = A_K^{\frac{1}{2}} f$, then,

$$\sum_{i=1}^{\infty} f_i^2 = \sum_{i=1}^{\infty} \frac{1}{\lambda_i} h_i^2$$
$$< \infty,$$

and $f \in \mathcal{L}^2(\mathcal{X}, \Sigma, \nu)$ and $h \in A_K^{\frac{1}{2}}(\mathcal{L}^2(\mathcal{X}, \Sigma, \nu))$. Thus by Theorem C.0.10 (b) Q and P are both equivalent to $\prod_{i=1}^{\infty} N_{0, \frac{1}{\gamma}\lambda_i}$ and so are equivalent to each other[4] . Further by translation we have that,

$$\frac{\mathrm{d}Q}{\mathrm{d}P}(h) = \frac{\mathrm{d} \prod_{i=1}^{\infty} N_{(h_{\mathcal{S}}^* - h^*)_i, \frac{1}{\gamma}\lambda_i}}{\mathrm{d} \prod_{i=1}^{\infty} N_{0, \frac{1}{\gamma}\lambda_i}}(h),$$

and thus by Theorem C.0.10 (c),

$$\begin{aligned}
\frac{\mathrm{d}Q}{\mathrm{d}P}(h) &= \exp\left(\langle h - h^*, \left(\frac{1}{\gamma} A_K\right)^{-1} (h_{\mathcal{S}}^* - h^*) \rangle_{\mathcal{L}^2} - \frac{1}{2} || \left(\frac{1}{\gamma} A_K\right)^{-\frac{1}{2}} (h_{\mathcal{S}}^* - h^*) ||_{\mathcal{L}^2}^2 \right) \qquad P - a.e. \\
&= \exp\left(\frac{\gamma}{2} \lim_{n \to \infty} \sum_{i=1}^{n} \frac{1}{\lambda_i} (h_i^* - h_{\mathcal{S},i}^*)(h_i^* + h_{\mathcal{S},i}^* - 2h_i) \right) \\
\ln \frac{\mathrm{d}Q}{\mathrm{d}P}(h) &= \frac{\gamma}{2} \lim_{n \to \infty} \sum_{i=1}^{n} \frac{1}{\lambda_i} (h_i^* - h_{\mathcal{S},i}^*)(h_i^* + h_{\mathcal{S},i}^* - 2h_i) \qquad P - a.e. \\
\ln \frac{\mathrm{d}Q}{\mathrm{d}P}(h) &= \frac{\gamma}{2} \lim_{n \to \infty} \sum_{i=1}^{n} \frac{1}{\lambda_i} (h_i^* - h_{\mathcal{S},i}^*)(h_i^* + h_{\mathcal{S},i}^* - 2h_i) \qquad Q - a.e.,
\end{aligned}$$

the final line following since any set with positive Q-measure is a set of positive P-measure. Thus,

$$\begin{aligned}
KL(Q||P) &= \mathbb{E}_{h \sim Q} \left[\ln \frac{\mathrm{d}Q}{\mathrm{d}P}(h) \right] \\
&= \mathbb{E}_{h \sim Q} \left[\frac{\gamma}{2} \lim_{n \to \infty} \sum_{i=1}^{n} \frac{1}{\lambda_i} (h_i^* - h_{\mathcal{S},i}^*)(h_i^* + h_{\mathcal{S},i}^* - 2h_i) \right] \\
&= \mathbb{E}_{h_1 \sim Q_1} \mathbb{E}_{h_2 \sim Q_2} ... \mathbb{E}_{h_j \sim Q_j} ... \left[\frac{\gamma}{2} \lim_{n \to \infty} \sum_{i=1}^{n} \frac{1}{\lambda_i} (h_i^* - h_{\mathcal{S},i}^*)(h_i^* + h_{\mathcal{S},i}^* - 2h_i) \right] \qquad (2.21) \\
&= \frac{\gamma}{2} \lim_{n \to \infty} \mathbb{E}_{h \sim Q} \left[\sum_{i=1}^{n} \frac{1}{\lambda_i} (h_i^* - h_{\mathcal{S},i}^*)(h_i^* + h_{\mathcal{S},i}^* - 2h_i) \right] \\
&= \frac{\gamma}{2} \sum_{i=1}^{\infty} \frac{1}{\lambda_i} (h_i^* - h_{\mathcal{S},i}^*)^2 \\
&= \frac{\gamma}{2} ||h_{\mathcal{S}}^* - h^*||_K^2.
\end{aligned}$$

Each expectation commutes with the limit in (2.21) since there is only one term in the summation in each h_i. □

We remark that we do in fact need some conditions on $h_{\mathcal{S}}^*$ and h^* in order for the above lemma to hold and the fact that $h_{\mathcal{S}}^* - h^* \in \mathcal{H}_K$ is sufficient in our case, but it is not true in general.

We now proceed to upper bound the divergence via a method of bounded differences. For any Mercer kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, we denote

[4] Recall that this means that each is absolutely continuous w.r.t. to the other, i.e. any set is P-null iff it is Q-null.

$$\kappa(\boldsymbol{x}) := \sup_{h \in \mathcal{H}_K} \frac{|h(\boldsymbol{x})|}{||h||_K} = \sqrt{K(\boldsymbol{x}, \boldsymbol{x})} \qquad \text{and} \qquad \kappa := \sup_{\boldsymbol{x} \in \mathcal{X}} \kappa(\boldsymbol{x}),$$

and define the distance $d_K(\boldsymbol{x}, \boldsymbol{x}') := ||K(\boldsymbol{x}, \cdot) - K(\boldsymbol{x}', \cdot)||_K$. Note that $d_K(\boldsymbol{x}, \boldsymbol{x}') \leq 2\kappa$. Our analyses will make use of the following property of a loss function:

Definition (Bousquet and Elisseeff, 2002, Definition 19) $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is α-admissible with respect to $\mathcal{H}_K$ if it is convex in its first argument and for all $y \in \mathcal{Y}$,

$$|\ell(y_1, y) - \ell(y_2, y)| \leq \alpha |y_1 - y_2|,$$

for all y_1, y_2 in the domain of the functions from $\mathcal{H}_K$.

The hinge loss and absolute loss are thus 1-admissible. We recall the following definition of Bregman divergence[5] on a Hilbert space $\mathcal{H}$: for differentiable[6] convex $\Phi : \mathcal{H} \to \mathbb{R}$,

$$D_\Phi(u, v) := \Phi(u) - \Phi(v) - \langle \boldsymbol{\nabla} \Phi(v), u - v \rangle_{\mathcal{H}}. \tag{2.22}$$

Consider a sample $\mathcal{S}$ and its "perturbation" $\mathcal{S}^{(i)}$,

$$\mathcal{S} := \{(X_1, Y_1), ...(X_m, Y_m)\} \tag{2.23}$$

$$\mathcal{S}^{(i)} := \{(X_1, Y_1), ...(X_{i-1}, Y_{i-1}), (X_i', Y_i'), (X_{i+1}, Y_{i+1}), ...(X_m, Y_m)\}. \tag{2.24}$$

Lemma 2.4.3. *If $\ell(\cdot, \cdot)$ is α-admissible and differentiable[7] then*

$$||h^*_{\mathcal{S}^{(i)}} - h^*_{\mathcal{S}}||_K \leq \frac{\alpha}{2\eta m} (\kappa(X_i) + \kappa(X_i')). \tag{2.25}$$

Proof. The method of proof is a stability argument which follows (Bousquet and Elisseeff, 2002, Theorem 22). Denote the "objectives"

$$\Omega(h) := \widehat{\text{risk}}^{\ell}_{\mathcal{S}}(h) + \eta ||h||^2_K,$$
$$\Omega^{(i)}(h) := \widehat{\text{risk}}^{\ell}_{\mathcal{S}^{(i)}}(h) + \eta ||h||^2_K.$$

Since $\boldsymbol{\nabla} \Omega(h^*_{\mathcal{S}}) = \boldsymbol{\nabla} \Omega^{(i)}(h^*_{\mathcal{S}^{(i)}}) = 0$, we have,

$$\begin{aligned} D_\Omega(h^*_{\mathcal{S}^{(i)}}, h^*_{\mathcal{S}}) + D_{\Omega^{(i)}}(h^*_{\mathcal{S}}, h^*_{\mathcal{S}^{(i)}}) &= \Omega(h^*_{\mathcal{S}^{(i)}}) - \Omega(h^*_{\mathcal{S}}) + \Omega^{(i)}(h^*_{\mathcal{S}}) - \Omega^{(i)}(h^*_{\mathcal{S}^{(i)}}) \\ &= \frac{1}{m} (\ell(h^*_{\mathcal{S}^{(i)}}(X_i), Y_i) - \ell(h^*_{\mathcal{S}^{(i)}}(X_i'), Y_i') \\ &\quad + \ell(h^*_{\mathcal{S}}(X_i'), Y_i') - \ell(h^*_{\mathcal{S}}(X_i), Y_i)). \end{aligned}$$

[5] See, for example, Frigyik et al. (2008) for an overview of Bregman divergence on function spaces.

[6] By which we mean that the Fréchet derivative $D_\Phi(v)$ of Φ at v exists everywhere. As $D_\Phi(v) : \mathcal{H} \to \mathbb{R}$ is a bounded (and therefore continuous) linear operator the Reisz representation theorem guarantees the existence of a single element of $\mathcal{H}$, which we denote $\boldsymbol{\nabla} \Phi(v)$, such that $(D_\Phi(v))(u) = \langle \boldsymbol{\nabla} \Phi(v), u \rangle_{\mathcal{H}}$ for all $u \in \mathcal{H}$.

[7] We note that for the case of the hinge loss or absolute loss this condition can be relaxed – we can define the derivative to be zero at the point at which they are non-differentiable. For general subdifferentiable convex loss functions we recover the results if we define the gradient to be zero at the minimum.

Noting the additivity, $D_{\Phi+\Psi} = D_\Phi + D_\Psi$, and non-negativity of Bregman divergences and that $D_{\eta||\cdot||_K^2}(h,g) = \eta||h-g||_K^2$ we have,

$$\begin{aligned}
2\eta||h^*_{\mathcal{S}} - h^*_{\mathcal{S}^{(i)}}||_K^2 &= D_{\eta||\cdot||_K^2}(h^*_{\mathcal{S}^{(i)}}, h^*_{\mathcal{S}}) + D_{\eta||\cdot||_K^2}(h^*_{\mathcal{S}}, h^*_{\mathcal{S}^{(i)}}) \\
&\leq D_\Omega(h^*_{\mathcal{S}^{(i)}}, h^*_{\mathcal{S}}) + D_{\Omega^{(i)}}(h^*_{\mathcal{S}}, h^*_{\mathcal{S}^{(i)}}) \\
&= \frac{1}{m}\big(\ell(h^*_{\mathcal{S}^{(i)}}(X_i), Y_i) - \ell(h^*_{\mathcal{S}^{(i)}}(X'_i), Y'_i) + \ell(h^*_{\mathcal{S}}(X'_i), Y'_i) - \ell(h^*_{\mathcal{S}}(X_i), Y_i)\big) \\
&\leq \frac{\alpha}{m}(|h^*_{\mathcal{S}}(X_i) - h^*_{\mathcal{S}^{(i)}}(X_i)| + |h^*_{\mathcal{S}}(X'_i) - h^*_{\mathcal{S}^{(i)}}(X'_i)|) \\
&\leq \frac{\alpha}{m}(||h^*_{\mathcal{S}} - h^*_{\mathcal{S}^{(i)}}||_K(\kappa(X_i) + \kappa(X'_i))).
\end{aligned}$$

□

Lemma 2.4.4. *If $\ell(\cdot,\cdot)$ is α-admissible, differentiable[7] and $\mathcal{H}_K$ is separable then*

$$\mathbb{P}_{\mathcal{S}}\left(||h^*_{\mathcal{S}} - h^*||_K \leq \frac{2\alpha\kappa}{\eta}\sqrt{\frac{1}{m}\ln\frac{4}{\delta}}\right) \geq 1-\delta. \tag{2.26}$$

Proof. Define the Doob martingale,

$$V_i = \mathbb{E}[h^*_{\mathcal{S}} - h^* \mid (X_1,Y_1), ...(X_i,Y_i)],$$

and note that $V_0 = 0$, $V_m = h^*_{\mathcal{S}} - h^*$, and that

$$\begin{aligned}
\mathbb{E}[V_i \mid (X_1,Y_1), ...(X_{i-1},Y_{i-1})] &= \mathbb{E}[h^*_{\mathcal{S}} - h^* \mid (X_1,Y_1), ...(X_{i-1},Y_{i-1})] \\
&= V_{i-1}.
\end{aligned}$$

Thus $\{V_i\}_{i=1}^m$ is a martingale and we have further, if we denote $\mathcal{S}$ and $\mathcal{S}^{(i)}$ as in (2.23) and (2.24), by Lemma 2.4.3 and the convexity of $||\cdot||_K$ that,

$$\begin{aligned}
||V_i - V_{i-1}||_K &= ||\mathbb{E}\left[h^*_{\mathcal{S}} \mid (X_1,Y_1), ...(X_i,Y_i)\right] - \mathbb{E}\left[h^*_{\mathcal{S}} \mid (X_1,Y_1), ...(X_{i-1},Y_{i-1})\right]||_K \\
&= ||\mathbb{E}_{(X'_i,Y'_i),(X_{i+1},Y_{i+1}),...(X_m,Y_m)}\left[h^*_{\mathcal{S}} - h^*_{\mathcal{S}^{(i)}} \mid (X_1,Y_1), ...(X_i,Y_i)\right]||_K \\
&\leq \mathbb{E}_{(X'_i,Y'_i),(X_{i+1},Y_{i+1}),...(X_m,Y_m)}\left[||h^*_{\mathcal{S}} - h^*_{\mathcal{S}^{(i)}}||_K \mid (X_1,Y_1), ...(X_i,Y_i)\right] \\
&\leq \frac{\kappa\alpha}{\eta m}.
\end{aligned}$$

Since $\mathcal{H}_K$ is separable it has a countable basis and so is isomorphic to either $\ell^2(\mathbb{R})$ or $\mathbb{R}^d$ and the result follows from the result of (Kallenberg and Sztencel, 1991, Theorem 3.1) (which gives a version of Azuma's inequality for ℓ^2-valued martingales, see the details in Theorem B.0.5 and Corollary B.0.6 of the Appendix). □

We can now give the PAC-Bayes bound for the classification risk of the Gibbs classifier, G_Q, drawn from $\mathcal{L}^2(\mathcal{X},\Sigma,\nu)$ according to the distribution Q defined by (2.19).

Theorem 2.4.5. *If $\ell(\cdot,\cdot)$ is α-admissible, differentiable[7] and $\mathcal{H}_K$ is separable then,*

$$\mathbb{P}_{\mathcal{S}}\left(\mathrm{kl}(\widehat{\mathrm{risk}}_{\mathcal{S}}(G_Q), \mathrm{risk}(G_Q)) \leq \frac{1}{m}\left(\frac{2\gamma\alpha^2\kappa^2}{\eta^2 m}\ln\frac{8}{\delta} + \ln\frac{2\xi(m)}{\delta}\right)\right) \geq 1-\delta.$$

Proof. Lemma 2.4.2 and Lemma 2.4.4 immediately imply that,

$$\mathbb{P}_{\mathcal{S}}\left(KL(Q||P) \leq \frac{2\gamma\alpha^2\kappa^2}{\eta^2 m}\ln\frac{8}{\delta}\right) \geq 1 - \frac{\delta}{2},$$

which we combine with Theorem 2.2.2 using the union bound. □

Note that the PAC-Bayes bounds for Gibbs classifiers presented here will provide sharp bounds on the mean classifier (which, with suitable choices for parameters, could be various types of SVM), with an additional factor of $1 + \epsilon$, under a margin assumption, by standard techniques (Langford and Shawe-taylor, 2002).

2.4.3 Data-dependent regularization in a "warped" RKHS

We now consider RKHS regularization algorithms in which the RKHS $\mathcal{H}_K$ is defined using the data in an attempt to make the structure of the RKHS – specifically the norm – reflect the smoothness of functions $h \in \mathcal{H}_K$ with respect to the data sample. Specifically we analyse methods related to LapSVM (Belkin et al., 2006) in which the RKHS norm is mixed with an empirical norm defined using the Laplacian of a graph formed on the data sample so that regularizing in this "warped" RKHS encourages the solution to be smooth over the data sample.

Given a RKHS $\mathcal{H}_K$ with kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and a sample $\mathcal{I} := \{X_1, ...X_t\}$ of instances from the input space we define the following empirical semi-inner product over $\mathcal{H}_K$,

$$\langle h, g\rangle_{\boldsymbol{L}} := \frac{2}{t(t-1)}\boldsymbol{h}^\top \boldsymbol{L}\boldsymbol{g},$$

where, recalling Section 2.3.3, $\boldsymbol{L}$ is the Laplacian of a graph formed on the instances $\mathcal{I}$ and $\boldsymbol{h} := (h(X_i)) \in \mathbb{R}^n$, $\boldsymbol{g} := (g(X_i)) \in \mathbb{R}^n$ are the point evaluations of h and g on $\mathcal{I}$. We consider the "warped" RKHS (Sindhwani et al., 2005) $\widetilde{\mathcal{H}}_K$ of functions from $\mathcal{H}_K$ with modified inner product,

$$\langle h, g\rangle_{\widetilde{\mathcal{H}}_K} := \langle h, g\rangle_K + \tau\langle h, g\rangle_{\boldsymbol{L}},$$

where τ controls the relative weight given to the inner product in $\mathcal{H}_K$ and the empirical inner product. The motivation here is that we are using the data to construct an empirically defined RKHS whose inner product captures the intrinsic geometry of the data; recalling Section 2.3.3, functions which have a small Hilbert space norm are smooth on the data. This intrinsic geometry can be quite different from that captured by the ambient geometry. According to arguments in Sindhwani et al. (2005) $\widetilde{\mathcal{H}}_K$ is a RKHS with kernel $\widetilde{K} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ given by

$$\widetilde{K}(\boldsymbol{x}, \boldsymbol{x}') = K(\boldsymbol{x}, \boldsymbol{x}') - \frac{2\tau}{t(t-1)}\boldsymbol{k}_{\boldsymbol{x}}^\top\left(\boldsymbol{I} + \frac{2\tau}{t(t-1)}\boldsymbol{L}\boldsymbol{K}\right)^{-1}\boldsymbol{L}\boldsymbol{k}_{\boldsymbol{x}'},$$

where $\boldsymbol{k}_{\boldsymbol{x}} = (K(\boldsymbol{x}_1, \boldsymbol{x}), ...K(\boldsymbol{x}_t, \boldsymbol{x}))^\top$, and $\boldsymbol{K}$ is the $t \times t$ Gram matrix $K_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ for $i, j \leq t$. Thus we can identify $\widetilde{\mathcal{H}}_K = \mathcal{H}_{\widetilde{K}}$ and $\langle\cdot,\cdot\rangle_{\widetilde{\mathcal{H}}_K} = \langle\cdot,\cdot\rangle_{\widetilde{K}}$.

Note that $\boldsymbol{L}$, $\widetilde{K}$ and $\mathcal{H}_{\widetilde{K}}$ are all empirical quantities which depend upon $\mathcal{I}$ but for ease of notation the dependence upon $\mathcal{I}$ will only be implicit. Recalling Section 2.3.3 and (2.11) note that,

$$||h||^2_{\widetilde{K}} = ||h||^2_K + \tau\widetilde{U}_{\mathcal{I}}(h).$$

In the following we denote,

$$\begin{aligned}\widetilde{\kappa}(\boldsymbol{x}) &:= \sup_{h\in\mathcal{H}_K} \frac{|\langle h, \widetilde{K}(\boldsymbol{x},\cdot)\rangle_{\widetilde{K}}|}{||h||_{\widetilde{K}}} = ||\widetilde{K}(\boldsymbol{x},\cdot)||_{\widetilde{K}} = \sqrt{\widetilde{K}(\boldsymbol{x},\boldsymbol{x})} \\ \widetilde{\kappa} &:= \sup_{\boldsymbol{x}\in\mathcal{X}} \widetilde{\kappa}(\boldsymbol{x}).\end{aligned}$$

Using only unlabelled data to define the RKHS

In the presence of a reasonable quantity of unlabelled data, so that we have a sample $\mathcal{S} := \{(X_1,Y_1),...(X_m,Y_m)\} \cup \{X_{m+1},...X_n\}$ of labelled and unlabelled points drawn from $\mathcal{X}\times\mathcal{Y}$, we can take $\mathcal{I} = \{X_{m+1},...X_n\}$ and form the empirical kernel $\widetilde{K}$ accordingly. We can then perform standard supervised classification using this kernel by training on the labelled part of the sample exactly as described in Section 2.4.1. Because $\widetilde{K}$ is defined using only the unlabelled component of the sample this reduces to the case already studied and we simply note that the bound of Theorem 2.4.5 holds in this case (with κ replaced by $\widetilde{\kappa}$):

Theorem 2.4.6. *If $\ell(\cdot,\cdot)$ is α-admissible, differentiable*[7] *and $\mathcal{H}$ is separable then*

$$\mathbb{P}_{\mathcal{S}}\left(\mathrm{kl}(\widehat{\mathrm{risk}}_{\mathcal{S}}(G_Q), \mathrm{risk}(G_Q)) \leq \frac{1}{m}\left(\frac{2\gamma\alpha^2\widetilde{\kappa}^2}{\eta^2 m}\log\frac{8}{\delta} + \ln\frac{2\xi(m)}{\delta}\right)\right) \geq 1-\delta.$$

Using all data to define the RKHS

The analysis of the previous section is adequate for the semi-supervised setting with plenty of unlabelled data, but, ideally, we would like obtain a classifier by regularizing with respect to the empirically-defined RKHS whose geometry captures the data structure defined by all labelled and unlabelled data. In particular, when we have access to little or no unlabelled data we would like to use the labelled sample to inform this construction, and still obtain a risk bound in the vein of Theorem 2.4.5. The following analysis provides a bound for algorithms such as LapSVM (Belkin et al., 2006) when the empirically-defined RKHS is informed by the whole data sample.

Again we suppose that we have a sample $\mathcal{S} := \{(X_1,Y_1),...(X_m,Y_m)\} \cup \{X_{m+1},...X_n\}$ of labelled and unlabelled[8] points drawn from $\mathcal{X}\times\mathcal{Y}$ and now take $\mathcal{I} = \{X_1,...X_n\}$ and form the RKHS $\mathcal{H}_{\widetilde{K}}$ with kernel $\widetilde{K}$ described in Section 2.4.3. We are interested in this case in the (semi-supervised) hypotheses,

$$h^*_{\mathcal{S}} := \operatorname*{argmin}_{h\in\mathcal{H}} \{\widehat{\mathrm{risk}}^{\ell}_{\mathcal{S}}(h) + \eta||h||^2_{\widetilde{K}}\} \tag{2.27}$$

$$h^* := \mathbb{E}_{\mathcal{S}}[h^*_{\mathcal{S}}], \tag{2.28}$$

where, as before, $\ell(\cdot,\cdot)$ is some admissible loss function and expectation is over the draw of the sample $\mathcal{S}$ with m labelled instances and $n-m$ unlabelled instances. Recalling Section 2.4.1 we then form distributions over $\mathcal{L}^2(\mathcal{X},\Sigma,\nu)$ via isomorphism with ℓ^2 which are identical to those defined by (2.19),

[8] The unlabelled set can be small or empty.

but with different means defined in this case by the hypotheses (2.27) and (2.28),

$$\begin{aligned} Q_i &:= N_{h^*_{\mathcal{S},i},\frac{1}{\gamma}\lambda_i} \qquad \text{and} \qquad P_i := N_{h^*_i,\frac{1}{\gamma}\lambda_i}, \\ Q &:= \prod_{i=1}^{\infty} Q_i \qquad\qquad\qquad\quad P := \prod_{i=1}^{\infty} P_i. \end{aligned} \tag{2.29}$$

Note that the $\{\lambda_i\}$ correspond to the kernel K and not $\widetilde{K}$. We remark that using the empirically-defined warped RKHS to obtain not just the mean of the Gaussian process (as is done here) but also to define the covariance structure seems to require a much more involved analysis.

Lemma 2.4.7.

$$\mathrm{KL}(Q||P) = \frac{\gamma}{2}||h^* - h^*_{\mathcal{S}}||^2_{\mathcal{H}}. \tag{2.30}$$

Proof. This follows analogously to Lemma 2.4.2. □

We bound this divergence using arguments analogous to Lemma 2.4.3 and Lemma 2.4.4 for the non-empirical case. Consider a sample $\mathcal{S}$ and its perturbation $\mathcal{S}^{(i)}$,

$$\begin{aligned} \mathcal{S} &:= \{(X_1,Y_1),...(X_m,Y_m)\} \cup \{X_{m+1},...X_n\} \\ \mathcal{S}^{(i)} &:= \{(X_1,Y_1),...(X_{i-1},Y_{i-1}),(X'_i,Y'_i),(X_{i+1},Y_{i+1}),...(X_m,Y_m)\} \cup \{X_{m+1},...X_n\} \quad i \le m \\ \mathcal{S}^{(i)} &:= \{(X_1,Y_1),...(X_m,Y_m)\} \cup \{X_{m+1},...X_{i-1},X'_i,X_{i+1},...X_n\} \quad i > m \end{aligned}$$

Lemma 2.4.8. *If $\ell(\cdot,\cdot)$ is α-admissible then for $i \le m$,*

$$||h^*_{\mathcal{S}} - h^*_{\mathcal{S}^{(i)}}||_K \le \frac{\alpha\kappa}{m\eta} + \frac{16\bar{y}\kappa^2\tau w}{n\sqrt{\eta}},$$

and for $m < i \le n$,

$$||h^*_{\mathcal{S}} - h^*_{\mathcal{S}^{(i)}}||_K \le \frac{16\bar{y}\kappa^2\tau w}{n\sqrt{\eta}},$$

where $\bar{y} := \sup_{y\in\mathcal{Y}} \ell(0,y)$ denotes the maximum loss incurred by the zero function[9]

Proof. Denote by $\widetilde{K}$ the empirical kernel formed on the sample $\mathcal{S}$, and by $\widetilde{K}^{(i)}$ the empirical kernel formed on the sample $\mathcal{S}^{(i)}$. Denote the "objectives",

$$\begin{aligned} \Omega(h) &:= \widehat{\mathrm{risk}}^{\ell}_{\mathcal{S}}(h) + \eta||h||^2_{\widetilde{K}} \\ \Omega^{(i)}(h) &:= \widehat{\mathrm{risk}}^{\ell}_{\mathcal{S}^{(i)}}(h) + \eta||h||^2_{\widetilde{K}^{(i)}}, \end{aligned}$$

and associated Bregman divergences,

$$\begin{aligned} D_{\Omega}(h,g) &:= \Omega(h) - \Omega(g) - \langle \boldsymbol{\nabla}\Omega(g), h-g\rangle_K \\ D_{\Omega^{(i)}}(h,g) &:= \Omega^{(i)}(h) - \Omega^{(i)}(g) - \langle \boldsymbol{\nabla}\Omega^{(i)}(g), h-g\rangle_K \ . \end{aligned}$$

[9] For the hinge loss and absolute loss $\bar{y} = 1$ when $\mathcal{Y} = \{-1,1\}$.

Since $\nabla\Omega(h^*_{\mathcal{S}}) = \nabla\Omega^{(i)}(h^*_{\mathcal{S}^{(i)}}) = 0$ we have,

$$\begin{aligned} D_\Omega(h^*_{\mathcal{S}^{(i)}}, h^*_{\mathcal{S}}) + D_{\Omega^{(i)}}(h^*_{\mathcal{S}}, h^*_{\mathcal{S}^{(i)}}) &= \Omega(h^*_{\mathcal{S}^{(i)}}) - \Omega(h^*_{\mathcal{S}}) + \Omega^{(i)}(h^*_{\mathcal{S}}) - \Omega^{(i)}(h^*_{\mathcal{S}^{(i)}}) \\ &= \frac{1}{m}(\ell(h^*_{\mathcal{S}^{(i)}}(X_i), Y_i) - \ell(h^*_{\mathcal{S}^{(i)}}(X'_i), Y'_i) \\ &\quad +\ell(h^*_{\mathcal{S}}(X'_i), Y'_i) - \ell(h^*_{\mathcal{S}}(X_i), Y_i)) \\ &\quad +\eta\tau(\widehat{U}_{\mathcal{S}}(h^*_{\mathcal{S}^{(i)}}) - \widehat{U}_{\mathcal{S}}(h^*_{\mathcal{S}}) + \widehat{U}_{\mathcal{S}^{(i)}}(h^*_{\mathcal{S}}) - \widehat{U}_{\mathcal{S}^{(i)}}(h^*_{\mathcal{S}^{(i)}})). \end{aligned}$$

Noting the additivity, $D_{\Phi+\Psi} = D_\Phi + D_\Psi$, and non-negativity of Bregman divergences and that $D_{\eta||\cdot||^2_{\widetilde{K}}}(h, g) = \eta||h - g||^2_{\widetilde{K}}$ we have,

$$\begin{aligned} 2\eta||h^*_{\mathcal{S}} - h^*_{\mathcal{S}^{(i)}}||^2_K &\leq \eta||h^*_{\mathcal{S}} - h^*_{\mathcal{S}^{(i)}}||^2_{\widetilde{K}} + \eta||h^*_{\mathcal{S}} - h^*_{\mathcal{S}^{(i)}}||^2_{\widetilde{K}^{(i)}} \\ &= D_{\eta||\cdot||^2_{\widetilde{K}}}(h^*_{\mathcal{S}^{(i)}}, h^*_{\mathcal{S}}) + D_{\eta||\cdot||^2_{\widetilde{K}^{(i)}}}(h^*_{\mathcal{S}}, h^*_{\mathcal{S}^{(i)}}) \\ &\leq D_\Omega(h^*_{\mathcal{S}^{(i)}}, h^*_{\mathcal{S}}) + D_{\Omega^{(i)}}(h^*_{\mathcal{S}}, h^*_{\mathcal{S}^{(i)}}) \\ &= \frac{1}{m}(\ell(h^*_{\mathcal{S}^{(i)}}(X_i), Y_i) - \ell(h^*_{\mathcal{S}^{(i)}}(X'_i), Y'_i) \\ &\quad +\ell(h^*_{\mathcal{S}}(X'_i), Y'_i) - \ell(h^*_{\mathcal{S}}(X_i), Y_i)) \\ &\quad +\eta\tau(\widehat{U}_{\mathcal{S}}(h^*_{\mathcal{S}^{(i)}}) - \widehat{U}_{\mathcal{S}}(h^*_{\mathcal{S}}) + \widehat{U}_{\mathcal{S}^{(i)}}(h^*_{\mathcal{S}}) - \widehat{U}_{\mathcal{S}^{(i)}}(h^*_{\mathcal{S}^{(i)}})). \end{aligned}$$

Now by noting that,

$$\widehat{U}_{\mathcal{S}}(h) - \widehat{U}_{\mathcal{S}^{(i)}}(h) = \frac{2}{n(n-1)} \sum_{j:j\neq i} \left((h(X_i) - h(X_j))^2 W(X_i, X_j) - (h(X'_i) - h(X_j))^2 W(X'_i, X_j)\right),$$

and by the α-admissibility assumption,

$$\begin{aligned} 2\eta||h^*_{\mathcal{S}} - h^*_{\mathcal{S}^{(i)}}||^2_K &\leq \frac{\alpha}{m}(|h^*_{\mathcal{S}}(X_i) - h^*_{\mathcal{S}^{(i)}}(X_i)| + |h^*_{\mathcal{S}}(X'_i) - h^*_{\mathcal{S}^{(i)}}(X'_i)|) \\ &\quad +\frac{2\eta\tau}{n(n-1)} \sum_{j:j\neq i} W(X_i, X_j) \left((h^*_{\mathcal{S}^{(i)}}(X_i) - h^*_{\mathcal{S}^{(i)}}(X_j))^2 - (h^*_{\mathcal{S}}(X_i) - h^*_{\mathcal{S}}(X_j))^2\right) \\ &\quad +W(X'_i, X_j) \left((h^*_{\mathcal{S}}(X'_i) - h^*_{\mathcal{S}}(X_j))^2 - (h^*_{\mathcal{S}^{(i)}}(X'_i) - h^*_{\mathcal{S}^{(i)}}(X_j))^2\right) \\ &\leq \frac{\alpha}{m}\left(||h^*_{\mathcal{S}} - h^*_{\mathcal{S}^{(i)}}||_K\kappa(X_i) + ||h^*_{\mathcal{S}} - h^*_{\mathcal{S}^{(i)}}||_K\kappa(X'_i)\right) \\ &\quad +\frac{2\eta\tau w}{n(n-1)} \sum_{j:j\neq i} \Big(\left|h^*_{\mathcal{S}^{(i)}}(X_i) - h^*_{\mathcal{S}^{(i)}}(X_j) - h^*_{\mathcal{S}}(X_i) + h^*_{\mathcal{S}}(X_j)\right| \\ &\quad \times \left|h^*_{\mathcal{S}^{(i)}}(X_i) - h^*_{\mathcal{S}^{(i)}}(X_j) + h^*_{\mathcal{S}}(X_i) - h^*_{\mathcal{S}}(X_j)\right| \\ &\quad + \left|h^*_{\mathcal{S}}(X'_i) - h^*_{\mathcal{S}}(X_j) - h^*_{\mathcal{S}^{(i)}}(X'_i) + h^*_{\mathcal{S}^{(i)}}(X_j)\right| \left|h^*_{\mathcal{S}}(X'_i) - h^*_{\mathcal{S}}(X_j) + h^*_{\mathcal{S}^{(i)}}(X'_i) - h^*_{\mathcal{S}^{(i)}}(X_j)\right| \Big) \\ &\leq \frac{2\alpha\kappa}{m}||h^*_{\mathcal{S}} - h^*_{\mathcal{S}^{(i)}}||_K \\ &\quad +\frac{2\eta\tau w}{n(n-1)} \sum_{j:j\neq i} \Big(\left|\langle h^*_{\mathcal{S}^{(i)}} - h^*_{\mathcal{S}}, K(X_i, \cdot) - K(X_j, \cdot)\rangle_K\right| \\ &\quad \times \left|\langle h^*_{\mathcal{S}^{(i)}}, K(X_i, \cdot) - K(X_j, \cdot)\rangle_K + \langle h^*_{\mathcal{S}}, K(X_i, \cdot) - K(X_j, \cdot)\rangle_K\right| \\ &\quad + \left|\langle h^*_{\mathcal{S}} - h^*_{\mathcal{S}^{(i)}}, K(X'_i, \cdot) - K(X_j, \cdot)\rangle_K\right| \left|\langle h^*_{\mathcal{S}}, K(X'_i, \cdot) - K(X_j, \cdot)\rangle_K + \langle h^*_{\mathcal{S}^{(i)}}, K(X'_i, \cdot) - K(X_j, \cdot)\rangle_K\right| \Big) \\ &\leq \frac{2\alpha\kappa}{m}||h^*_{\mathcal{S}} - h^*_{\mathcal{S}^{(i)}}||_K + \frac{4\eta\tau w}{n}||h^*_{\mathcal{S}^{(i)}} - h^*_{\mathcal{S}}||_K \sup_{x,x'\in\mathcal{X}}\{d^2_K(x, x')\}(||h^*_{\mathcal{S}^{(i)}}||_K + ||h^*_{\mathcal{S}}||_K) \\ &\leq \frac{2\alpha\kappa}{m}||h^*_{\mathcal{S}} - h^*_{\mathcal{S}^{(i)}}||_K + \frac{16\kappa^2\eta\tau w}{n}||h^*_{\mathcal{S}^{(i)}} - h^*_{\mathcal{S}}||_K(||h^*_{\mathcal{S}^{(i)}}||_K + ||h^*_{\mathcal{S}}||_K) \\ &\leq \frac{2\alpha\kappa}{m}||h^*_{\mathcal{S}} - h^*_{\mathcal{S}^{(i)}}||_K + \frac{32\bar{y}\kappa^2\sqrt{\eta}\tau w}{n}||h^*_{\mathcal{S}^{(i)}} - h^*_{\mathcal{S}}||_K. \end{aligned}$$

where the final line follows since $\eta||h^*_{\mathcal{S}}||^2_K \leq \bar{y}$ and $\eta||h^*_{\mathcal{S}^{(i)}}||^2_K \leq \bar{y}$, otherwise the zero function contradicts the Ω-minimality of $h^*_{\mathcal{S}}$ and the $\Omega^{(i)}$-minimality of $h^*_{\mathcal{S}^{(i)}}$.

This proves the first inequality. The second follows in the same way but noting that, for $i \geq m$, $\widehat{\text{risk}}^{\ell}_{\mathcal{S}^{(i)}}(h) = \widehat{\text{risk}}^{\ell}_{\mathcal{S}}(h)$ and so all terms related to the risk cancel. □

We now bound $||h^* - h^*_{\mathcal{S}}||_K$ w.h.p. as in the previous sections.

Lemma 2.4.9. *Under the conditions of Lemma 2.4.8 and for separable $\mathcal{H}_K$ we have*

$$\mathbb{P}_{\mathcal{S}}\left(||h^*_{\mathcal{S}} - h^*||_K \leq 2\frac{\kappa}{\eta}\sqrt{\left(\frac{\alpha^2}{m} + \frac{256\bar{y}^2\kappa^2\eta\tau^2w^2}{n} + \frac{32\bar{y}\kappa\sqrt{\eta}\tau w\alpha}{n}\right)\log\frac{4}{\delta}}\right) \geq 1-\delta.$$

Proof. This follows analogously to Lemma 2.4.4 – we create the same martingale and use Corollary B.0.6 noting that Lemma 2.4.8 implies that for $i = 1, ...m$

$$c_i^2 \leq \frac{\kappa^2}{\eta^2}\left(\frac{\alpha^2}{m^2} + \frac{256\bar{y}^2\kappa^2\eta\tau^2w^2}{n^2} + \frac{32\bar{y}\kappa\sqrt{\eta}\tau w\alpha}{mn}\right),$$

and for $i = m+1, ...n$

$$c_i^2 \leq \frac{\kappa^2}{\eta^2}\left(\frac{256\bar{y}^2\kappa^2\eta\tau^2w^2}{n^2}\right).$$

□

We can now give the PAC-Bayes bound for the classification risk of the Gibbs classifier, G_Q, drawn from $\mathcal{L}^2(\mathcal{X}, \Sigma, \nu)$ according to the distribution Q defined by (2.29).

Theorem 2.4.10. *If $\ell(\cdot,\cdot)$ is α-admissible, differentiable[7] and $\mathcal{H}_K$ is separable then with probability at least $1-\delta$ over the draw of $\mathcal{S}$,*

$$\text{kl}(\widehat{\text{risk}}_{\mathcal{S}}(G_Q), \text{risk}(G_Q)) \leq \frac{1}{m}\left(\frac{2\gamma\kappa^2}{\eta^2}\left(\frac{\alpha^2}{m} + \frac{256\bar{y}^2\kappa^2\eta\tau^2w^2}{n} + \frac{32\bar{y}\kappa\sqrt{\eta}\tau w\alpha}{n}\right)\log\frac{8}{\delta} + \ln\frac{2\xi(m)}{\delta}\right),$$

Proof. Claim 2.4.7 and Lemma 2.4.9 immediately implies that,

$$\mathbb{P}_{\mathcal{S}}\left(KL(Q||P) \leq \frac{2\gamma\kappa^2}{\eta^2}\left(\frac{\alpha^2}{m} + \frac{256\bar{y}^2\kappa^2\eta\tau^2w^2}{n} + \frac{32\bar{y}\kappa\sqrt{\eta}\tau w\alpha}{n}\right)\log\frac{8}{\delta}\right) \geq 1 - \frac{\delta}{2},$$

which we combine with Theorem 2.2.2 using the union bound. □

Note that Theorem 2.4.5 is a special case of Theorem 2.4.10 obtained by setting $\tau = 0$ (or $w = 0$).

Remark We recall a few notes on the quantities in Theorem 2.4.10: α, $\bar{y}$, κ, w could all reasonably be typically approximately 1. For example, for the hinge and absolute loss $\alpha = 1$ and $\bar{y} = 1$, for the exponential kernel $\kappa = 1$, and it is common to build a graph such that $w = 1$, for example by choosing $0/1$ weights or weights determined by the Gaussian kernel. This leaves only parameters of the algorithm in Theorem 2.4.10; η and τ which control how much we regularize and γ which controls the variance of our Gaussian process.

Chapter 3

Relating function class complexity and cluster structure with applications to transduction

Abstract

We relate function class complexity to cluster structure in the function domain. This facilitates risk analysis relative to cluster structure in the input space which is particularly effective in semi-supervised learning. In particular we quantify the complexity of function classes defined over a graph in terms of the graph structure.

3.1 Introduction

We relate the learning process to cluster structure in the data which the learner is attempting to classify. It is well-known that data-dependent measures of function class complexity can lead to sharper risk bounds than those which do not capture the data distribution. We elaborate this principle by demonstrating a relationship between the richness of a function class and structural features in data drawn from the underlying input space $\mathcal{X}$ on which it acts. Specifically, a typical assumption in machine learning is that data are clustered and we refine a recent upper bound on Rademacher complexity of a function class, by relating it to cluster structure in the domain.

The intended application of these ideas is in the settings of transductive and semi-supervised learning. In Chapelle and Zien (2005) it is argued that virtually all successful semi-supervised learning techniques exploit the cluster assumption. In these frameworks we typically work with *empirically defined hypothesis classes* and it is natural to relate the learning process to the data which informs their construction. In such frameworks, an empirical metric on $\mathcal{X}$ which captures the intrinsic geometry of the data, can be constructed giving an opportunity to relate learning to the *intrinsic* structure of data. A typical empirical metric, equivalent to electrical resistance distance, is particularly sensitive to clustering, thus relating function class complexity to the cluster structure of $\mathcal{X}$ is effective in this case.

A key object in these settings is a graph formed using the available data and, as pointed out in Hanneke (2006) it is important to reach an understanding of which properties of a graph are relevant to the performance of an algorithm which predicts the labeling of the graph, and we provide a further step in that direction: in the spirit of the work of Herbster (2008) in the online setting, we present risk bounds (and suggest a prototype regularization scheme) derived from the cluster structure of the graph in the resistance metric. In particular we bound the richness of a class of functions with bounded cut defined over the vertices of a graph. When a graph exhibits good k-means clustering, in the resistance metric, this cluster structure seems to serve as a sharp practical measure of the richness of classifiers over a graph when learning under the typical "smoothness" assumption of a small graph cut; this is intuitive and is established using a duality theory.

We finally give a semi-supervised risk bound in which the complexity terms are related to the cluster structure of the (labeled and unlabeled) data instances.

3.2 Preliminaries

We denote by $\mathcal{H}$ a class of real-valued functions (*hypotheses*) mapping a domain $\mathcal{X}$ to a decision space $\mathcal{D}$ and refer to $h(\boldsymbol{x}) \in \mathcal{D}$ as the (*soft*) *classification* of $\boldsymbol{x}$ by $h \in \mathcal{H}$. It is typical to assign a measure of complexity $F : \mathcal{H} \to \mathbb{R}_{\geq 0}$ over functions in $\mathcal{H}$. This generally captures a prior belief that the hypothesis most likely to explain the relationship between data and their classification is simple, or that the true classifier respects the structure of the input space. Given $F : \mathcal{H} \to \mathbb{R}_{\geq 0}$ we denote

$$\mathcal{H}_\alpha := \{h \in \mathcal{H} \ : \ F(h) \leq \alpha\}.$$

We consider only function classes consisting of linear functions (in some, possibly kernelized, space) so that (soft) classification is $h(\boldsymbol{x}) = \langle h, \boldsymbol{x} \rangle$.

Given a distribution P_{XY} over the labeled input space $\mathcal{X} \times \mathcal{Y}$, and a loss function $\ell : \mathcal{D} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ we denote the true risk of $h \in \mathcal{H}$ by $\mathrm{risk}^\ell(h) := \mathbb{E}_{(X,Y)\sim P_{XY}} \ell(h(X), Y)$, and the risk on a specific set $\mathcal{T}$ by $\mathrm{risk}^\ell_{\mathcal{T}}(h) := \frac{1}{|\mathcal{T}|} \sum_{(X,Y)\in\mathcal{T}} \ell(h(X), Y)$ and, in particular, the empirical risk on a labeled training sample $\mathcal{S}$ by $\widehat{\mathrm{risk}}^\ell_{\mathcal{S}}(h) := \frac{1}{|\mathcal{S}|} \sum_{(X,Y)\in\mathcal{S}} \ell(h(X), Y)$. When $\ell(\cdot, \cdot)$ is the $0-1$ loss of binary classification, $\ell_{0-1}(y, y') := \begin{cases} 0 \text{ if } y = y' \\ 1 \text{ if } y \neq y' \end{cases}$, then, for simplicity, we denote the corresponding binary classification risk and its empirical counterpart by $\mathrm{risk}(\cdot)$ and $\widehat{\mathrm{risk}}_{\mathcal{S}}(\cdot)$ respectively.

Definition The *empirical Rademacher complexity* of a function class $\mathcal{H}$, on a sample $\mathcal{S} = \{\boldsymbol{x}_1, ...\boldsymbol{x}_m\}$ is defined,

$$\widehat{\mathcal{R}}_{\mathcal{S}}(\mathcal{H}) := \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{h\in\mathcal{H}} \left(\frac{1}{m} \sum_{i=1}^{m} h(\boldsymbol{x}_i)\sigma_i \right) \right]$$

where the σ_i are Rademacher random variables, $\mathbb{P}(\sigma_i = \pm 1) = 1/2$.

Definition Given a probability distribution over the draw of training samples from $\mathcal{X}$, the *Rademacher complexity* of a function class $\mathcal{H}$, w.r.t. samples of size m, is defined $\mathcal{R}_m(\mathcal{H}) := \mathbb{E}_{\mathcal{S}}(\widehat{\mathcal{R}}_{\mathcal{S}}(\mathcal{H}))$.

Interest in the Rademacher complexity of function classes stems from the fact that it can provide generalization bounds which are typically sharper than VC bounds, since it captures the distribution of the data under consideration. For example, it is known that $\mathcal{R}_m(\mathcal{H}) = \mathcal{O}\left(\sqrt{\frac{\mathrm{VC}(\mathcal{H})}{m}}\right)$ and we have:

Theorem 3.2.1. *(Bartlett and Mendelson, 2002)*[1] *Assume a loss function $\ell : \mathcal{D} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ is K-Lipschitz in its first argument and bounded by C, then for any $\delta > 0$, we have, with probability at least $1 - \delta$ over the draw of a training sample $\mathcal{S}$ of size m, that*

$$\sup_{h \in \mathcal{H}} \left(\mathrm{risk}^{\ell}(h) - \widehat{\mathrm{risk}}^{\ell}_{\mathcal{S}}(h) \right) \leq 2K\mathcal{R}_m(\mathcal{H}) + C\sqrt{\frac{\log \frac{1}{\delta}}{2m}}.$$

3.3 Relating function class complexity to structure in the function domain

Definition Given a set $\mathcal{S}$ of points drawn from a vector space $\mathcal{X}$ a *clustering* of $\mathcal{S}$ is any partition $\mathcal{C} = \{\mathcal{C}_1, ...\mathcal{C}_N\}$ of $\mathcal{S}$. Given a metric $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$, for each k we define the *center* of $\mathcal{C}_k$ by $\boldsymbol{c}_k := \mathrm{argmin}_{\boldsymbol{x} \in \mathcal{X}} \sum_{\boldsymbol{x}' \in \mathcal{C}_k} d^2(\boldsymbol{x}', \boldsymbol{x})$ and note that if $d(\cdot, \cdot)$ arises from the Euclidean inner product, $d^2(\boldsymbol{x}, \boldsymbol{x}') = \langle \boldsymbol{x} - \boldsymbol{x}', \boldsymbol{x} - \boldsymbol{x}' \rangle$, then this is identical to the *centroid* $\boldsymbol{c}_k = \frac{1}{|\mathcal{C}_k|} \sum_{\boldsymbol{x} \in \mathcal{C}_k} \boldsymbol{x}$. For each $\boldsymbol{x} \in \mathcal{S}$ we denote its corresponding center by $c(\boldsymbol{x}) := \boldsymbol{c}_k$ where k is such that $\boldsymbol{x} \in \mathcal{C}_k$.

3.3.1 A "duality" of complexity on $\mathcal{H}$ and distance on $\mathcal{X}$

Given a class of linear functions $\mathcal{H} : \mathcal{X} \to \mathbb{R}$, any norm $|| \cdot ||$ on $\mathcal{H}$ (which would generally capture complexity in $\mathcal{H}$) gives rise to a specific metric $d(\cdot, \cdot) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ defined, via the dual norm $|| \cdot ||^*$, by

$$\begin{aligned} d(\boldsymbol{x}_i, \boldsymbol{x}_j) : &= ||\boldsymbol{x}_i - \boldsymbol{x}_j||^* \\ &= \sup_{h \in \mathcal{H}, ||h|| \neq 0} \frac{|h(\boldsymbol{x}_i) - h(\boldsymbol{x}_j)|}{||h||}. \end{aligned}$$

Call such a metric the *implied metric*. Intuitively, if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ can be classified differently by some simple hypothesis in h they are distant in $d(\cdot, \cdot)$, and conversely if they are distinctly classified only by complex hypotheses then they are close. Given a norm on $\mathcal{H}$, it is this implied metric which we use to quantify cluster structure in $\mathcal{X}$.

Examples

1. **Linear classification in an arbitrary RKHS.** Given any kernel K on a space $\mathcal{X}$, consider the reproducing kernel Hilbert space $\mathcal{H}_K = \overline{\mathrm{span}\{K(\boldsymbol{x}, \cdot) : \boldsymbol{x} \in \mathcal{X}\}}$, consisting of all linear combinations of the *features* $\{K(\boldsymbol{x}, \cdot)\}_{\boldsymbol{x} \in \mathcal{X}}$. The inner product in $\mathcal{H}$ is defined by $\langle K(\boldsymbol{x}, \cdot), K(\boldsymbol{x}', \cdot) \rangle_K :=$

[1] This is actually a sharper result than that in the cited reference, obtained using the sharper contraction inequality of (Meir and Zhang, 2003, Theorem 7) than that provided by (Ledoux and Talagrand, 1991, Theorem 4.12).

$K(\boldsymbol{x}, \boldsymbol{x}')$, for all $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$ (Aronszajn, 1950). Given a set of points $\mathcal{S} = \{\boldsymbol{x}_1, ...\boldsymbol{x}_m\}$ drawn from $\mathcal{X}$, we consider classifiers of the form $h = \sum_{i=1}^m \alpha_i K(\boldsymbol{x}_i, \cdot)$, with $\boldsymbol{\alpha} \in \mathbb{R}^m$, such that a given point $\boldsymbol{x} \in \mathcal{X}$ receives the (soft) classification $h(\boldsymbol{x}) = \langle h, K(\boldsymbol{x}, \cdot)\rangle_K = \sum_{i=1}^m \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x})$. Kernel methods often amount to seeking a classifier by minimizing, or regularizing in $\mathcal{H}$ w.r.t., the norm $||h||_K = \sqrt{\langle h, h\rangle_K}$, whose dual, by the arguments above, defines an implied metric on the feature space (and by extension on $\mathcal{X}$),

$$\begin{aligned} d_K(\boldsymbol{x}, \boldsymbol{x}') :&= d(K(\boldsymbol{x}, \cdot), K(\boldsymbol{x}', \cdot)) \\ &= ||K(\boldsymbol{x}, \cdot) - K(\boldsymbol{x}', \cdot)||_K^* \\ &= \sup_{||h||_K \neq 0} \left\{ \frac{|\langle h, K(\boldsymbol{x}, \cdot) - K(\boldsymbol{x}', \cdot)\rangle_K|}{||h||_K} \right\} \\ &= \sqrt{K(\boldsymbol{x}, \boldsymbol{x}) + K(\boldsymbol{x}', \boldsymbol{x}') - 2K(\boldsymbol{x}, \boldsymbol{x}')}. \end{aligned}$$

2. **Transductive classification on a graph.** Given an n-vertex connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with (weighted) adjacency $\boldsymbol{A}$, we seek a classifier $\boldsymbol{h} \in \mathbb{R}^n$ which classifies the vertices $\mathcal{V} = \{v_1, ...v_n\}$ according to $\boldsymbol{h}(v_i) := \operatorname{sgn}(\boldsymbol{h}^\top \boldsymbol{e}_i) = \operatorname{sgn}(h_i)$, where we have identified each vertex v_i with the corresponding standard basis vector $\boldsymbol{e}_i$ in $\mathbb{R}^n$. A typical scheme is to minimize a *smoothness functional*

$$\begin{aligned} F_{\boldsymbol{L}}(h) :&= \frac{1}{2}||\boldsymbol{h}||_{\boldsymbol{L}}^2 := \frac{1}{2}\boldsymbol{h}^\top \boldsymbol{L}\boldsymbol{h} \\ &= \frac{1}{2}\sum_{(i,j)\in\mathcal{E}} (h_i - h_j)^2 A_{ij} \end{aligned}$$

induced by the graph Laplacian $\boldsymbol{L}$, subject to label constraints (Zhu et al., 2003a; Belkin et al., 2004). By following the above procedure, the dual of the semi-norm $||\boldsymbol{h}||_{\boldsymbol{L}}$, again implies a metric $d_{\boldsymbol{L}}(\cdot, \cdot)$ on $\mathcal{V}$ as follows,

$$\begin{aligned} d_{\boldsymbol{L}}(v_i, v_j) :&= ||\boldsymbol{e}_i - \boldsymbol{e}_j||_{\boldsymbol{L}}^* \\ &= \sup_{\boldsymbol{h}\in\mathbb{R}^n, ||\boldsymbol{h}||_{\boldsymbol{L}}\neq 0} \left\{ \frac{|\boldsymbol{h}^\top(\boldsymbol{e}_i - \boldsymbol{e}_j)|}{||\boldsymbol{h}||_{\boldsymbol{L}}} \right\} \\ &= \sup_{\boldsymbol{h}\in\mathbb{R}^n, ||\boldsymbol{h}||_{\boldsymbol{L}}\neq 0} \left\{ \frac{|(\boldsymbol{L}\boldsymbol{h})^\top \boldsymbol{L}^+(\boldsymbol{e}_i - \boldsymbol{e}_j)|}{\sqrt{(\boldsymbol{L}\boldsymbol{h})^\top \boldsymbol{L}^+(\boldsymbol{L}\boldsymbol{h})}} \right\} \\ &= \sup_{\boldsymbol{w}\in\operatorname{col}(\boldsymbol{L}), \boldsymbol{w}^\top \boldsymbol{L}^+\boldsymbol{w}\neq 0} \left\{ \frac{|\boldsymbol{w}^\top \boldsymbol{L}^+(\boldsymbol{e}_i - \boldsymbol{e}_j)|}{\sqrt{\boldsymbol{w}^\top \boldsymbol{L}^+\boldsymbol{w}}} \right\} \\ &= \sqrt{(\boldsymbol{e}_i - \boldsymbol{e}_j)^\top \boldsymbol{L}^+(\boldsymbol{e}_i - \boldsymbol{e}_j)}, \end{aligned}$$

where $\boldsymbol{L}^+$ is the pseudoinverse of the graph Laplacian. This metric is equal to the square root of the electrical resistance between vertices on $\mathcal{G}$ (Klein and Randić, 1993), which arises by viewing the graph as an electrical network in which each edge corresponds to a resistor with conductance equal to the edge weight[2]. Note the connection to the previous example: the space of balanced[3]

[2]This captures the geometry of a finite transductive input space particularly effectively, measuring the ease with which current flows through the body defined by the data which is more appropriate than a generic distance in an ambient space.

[3]i.e. vectors in $\mathbb{R}^n$ perpendicular to the all ones vector.

classifiers on a graph, with the inner product given by the quadratic form induced graph Laplacian, can be viewed as a RKHS whose Gram matrix is $\boldsymbol{L}^+$ (Herbster and Pontil, 2007).

3. **Semi-supervised classification.** The previous transductive example can be extended "out of sample". Suppose wish to build a classifier $h : \mathcal{X} \to \mathbb{R}$ and are given a sample of data points $\mathcal{S} = \{\boldsymbol{x}_1, ...\boldsymbol{x}_n\}$ from $\mathcal{X}$, but the true distribution of data from $\mathcal{X}$ is otherwise unknown. Given a kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ which defines a RKHS of functions $\mathcal{H}_K$ over $\mathcal{X}$ with inner product $\langle \cdot, \cdot \rangle_K$, we may consider the space $\widetilde{\mathcal{H}}$ of functions from $\mathcal{H}_K$ with modified inner product,

$$\langle h, g \rangle_{\widetilde{\mathcal{H}}} := \gamma_{\mathcal{H}} \langle h, g \rangle_K + \gamma_{\mathcal{S}} \langle Sh, Sg \rangle_{\mathcal{S}},$$

where $S(\cdot)$ is the (linear) *point evaluation function* on $\mathcal{S}$, $Sh = (h(\boldsymbol{x}_1), ...h(\boldsymbol{x}_n))^\top$, and $\langle \cdot, \cdot \rangle_{\mathcal{S}}$ is an inner product over the space of functions over $\mathcal{S}$, and $\gamma_{\mathcal{H}}$, $\gamma_{\mathcal{S}}$ control the relative weight given to the inner product in $\mathcal{H}_K$ and the empirical inner product. If $\langle Sh, Sg \rangle_{\mathcal{S}} = (Sh)^\top \boldsymbol{M}(Sg)$, where $\boldsymbol{M}$ is a positive semi-definite matrix measuring smoothness on a graph $\mathcal{G}$ formed on $\mathcal{S}$, such as the graph Laplacian, according to arguments in Sindhwani et al. (2005) $\widetilde{\mathcal{H}}$ is a RKHS $\mathcal{H}_{\widetilde{K}}$ with kernel $\widetilde{K} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ given by

$$\widetilde{K}(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{\gamma_{\mathcal{H}}} K(\boldsymbol{x}, \boldsymbol{x}') - \frac{\gamma_{\mathcal{S}}}{\gamma_{\mathcal{H}}} \boldsymbol{k}_{\boldsymbol{x}}^\top (\gamma_{\mathcal{H}} \boldsymbol{I} + \gamma_{\mathcal{S}} \boldsymbol{M}\boldsymbol{K})^{-1} \boldsymbol{M} \boldsymbol{k}_{\boldsymbol{x}'}, \tag{3.1}$$

where $\boldsymbol{k}_{\boldsymbol{x}} = (K(\boldsymbol{x}_1, \boldsymbol{x}), ...K(\boldsymbol{x}_n, \boldsymbol{x}))^\top$, and $\boldsymbol{K}$ is the $n \times n$ Gram matrix $K_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ for i, $j \leq n$.

By similar arguments to those above, seeking a classifier $h \in \mathcal{H}_{\widetilde{K}}$ by minimizing the norm $||h||_{\widetilde{K}} := ||h||_{\widetilde{\mathcal{H}}} := \sqrt{\langle h, h \rangle_{\mathcal{H}_{\widetilde{K}}}}$ implies a metric on $\mathcal{X}$ given by

$$d_{\widetilde{K}}(\boldsymbol{x}, \boldsymbol{x}') = \sqrt{\widetilde{K}(\boldsymbol{x}, \boldsymbol{x}) + \widetilde{K}(\boldsymbol{x}', \boldsymbol{x}') - 2\widetilde{K}(\boldsymbol{x}, \boldsymbol{x}')}.$$

Thus, (an approximation to) the resistance distance (or another such empirical distance) can be extended to the whole of $\mathcal{X}$.

3.3.2 Bounding Rademacher complexity

With reference to Appendix E we require the notion of convex conjugate, strong convexity and smoothness. Note that any positive semi-definite quadratic form $\frac{1}{2}\boldsymbol{h}^\top \boldsymbol{M}\boldsymbol{h}$ is 1-strongly convex w.r.t. the (semi-)norm $||\boldsymbol{h}||_{\boldsymbol{M}} = \sqrt{\boldsymbol{h}^\top \boldsymbol{M}\boldsymbol{h}}$. We require the following lemma, which is a straightforward generalization of (Kakade et al., 2008, Lemma 4).

Lemma 3.3.1. *Let $S \in \mathcal{W}$ be a closed convex set and $F : S \to \mathbb{R}_{\geq 0}$ be κ-strongly convex w.r.t. a norm $||\cdot||$ over S. Let $\{Z_i\}_{i=1}^m$ be conditionally zero mean random variables (i.e. $\mathbb{E}[Z_i | Z_1, ..., Z_{i-1}] = 0$) with values in $\mathcal{W}^*$ such that $\mathbb{E}[(||Z_i||^*)^2] \leq r_i^2$. Then $\mathbb{E}[F^\star(\sum_{i=1}^m Z_i)] \leq \frac{1}{2\kappa} \sum_{i=1}^m r_i^2$, where $F^\star$ denotes the Legendre-Fenchel conjugate of F.*

Proof. Let $S_k := \sum_{i=1}^{k} Z_i$. F is κ-strongly convex w.r.t. $||\cdot||$ and so, by Theorem E.0.12, $F^\star$ is $\frac{1}{\kappa}$-strongly smooth w.r.t. $||\cdot||^*$, this means,

$$F^\star\left(S_{m-1}+Z_m\right) \leq F^\star\left(S_{m-1}\right) + \left\langle \boldsymbol{\nabla} F^\star\left(S_{m-1}\right), Z_m\right\rangle + \frac{1}{2\kappa}(||Z_m||^*)^2.$$

Denoting $\mathbb{E}_{k-1}(\cdot) := \mathbb{E}_{Z_k}(\cdot \mid Z_1, \ldots Z_{k-1})$ and taking conditional expectation gives,

$$\mathbb{E}_{m-1}[F^\star\left(S_m\right)] \leq F^\star\left(S_{m-1}\right) + \frac{1}{2\kappa}\mathbb{E}_{m-1}[\left(||Z_m||^*\right)^2],$$

and since $F^\star(0) = \sup_{\boldsymbol{z}}(-F(\boldsymbol{z})) \leq 0$ the result follows by iterated use of the tower rule. □

We now refine a result of (Kakade et al., 2008, Theorem 3) (which uses convex duality to bound Rademacher complexity, but does not account for detailed structure such as cluster structure in the input space) by demonstrating the dependence of the Rademacher complexity of a function class $\mathcal{H}$ on the cluster structure of the data drawn from the domain $\mathcal{X}$ on which it acts.

Theorem 3.3.2. *For a class $\mathcal{H}$ of bounded linear functions on a set $\mathcal{X}$, if $F : \mathcal{H} \to \mathbb{R}_{\geq 0}$ is κ-strongly convex w.r.t. a norm $||\cdot||_F$ on $\mathcal{H}$, then for any sample $\mathcal{S} = \{\boldsymbol{x}_1, \ldots \boldsymbol{x}_m\}$ of points from $\mathcal{X}$ and all clusterings $\mathcal{C}$ of $\mathcal{S}$ we have, for all $\alpha > 0$,*

$$\widehat{\mathcal{R}}_{\mathcal{S}}(\mathcal{H}_\alpha) \leq B\sqrt{\frac{|\mathcal{C}|}{m}} + \sqrt{\frac{2\alpha\rho_{\mathcal{S}}}{m\kappa}}, \tag{3.2}$$

where $\rho_{\mathcal{S}} := \frac{1}{m}\sum_{i=1}^{m} d_F^2(\boldsymbol{x}_i, c(\boldsymbol{x}_i))$, $d_F(\cdot,\cdot)$ is the implied metric on $\mathcal{X}$ and $B := \sup_{h\in\mathcal{H}_\alpha, \boldsymbol{x}\in\mathcal{X}} |h(\boldsymbol{x})|$. Further, for all clusterings $\mathcal{C}$ of $\mathcal{X}$ we have,

$$\mathcal{R}_m(\mathcal{H}_\alpha) \leq B\mathbb{E}_{\mathcal{S}}\left[\sqrt{\frac{|\mathcal{C}_{\mathcal{S}}|}{m}}\right] + \sqrt{\frac{2\alpha}{m\kappa}}\mathbb{E}_{\mathcal{S}}[\sqrt{\rho_{\mathcal{S}}}], \tag{3.3}$$

where expectation is over the draw of a random sample $\mathcal{S} = \{\boldsymbol{X}_1, \ldots \boldsymbol{X}_m\}$ from $\mathcal{X}$ and $\mathcal{C}_{\mathcal{S}} := \{\mathcal{C}_k \in \mathcal{C} : \mathcal{S} \cap \mathcal{C}_k \neq \emptyset\}$ is the clustering restricted to the sample $\mathcal{S}$.

Proof. Let $\mathcal{C} = \{\mathcal{C}_1, \ldots \mathcal{C}_N\}$ be an arbitrary clustering of $\mathcal{S}$, and denote $m_j := |\mathcal{C}_j|$.

$$\begin{aligned}
\widehat{\mathcal{R}}_{\mathcal{S}}(\mathcal{H}_\alpha) &= \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}_\alpha} \langle h, \frac{1}{m}\sum_{i=1}^{m}\sigma_i\boldsymbol{x}_i\rangle\right] \\
&= \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}_\alpha}\left(\langle h, \frac{1}{m}\sum_{i=1}^{m}\sigma_i c(\boldsymbol{x}_i)\rangle + \langle h, \frac{1}{m}\sum_{i=1}^{m}\sigma_i(\boldsymbol{x}_i - c(\boldsymbol{x}_i))\rangle\right)\right] \\
&\leq \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}_\alpha} \langle h, \frac{1}{m}\sum_{j=1}^{N}\sum_{i:\boldsymbol{x}_i\in\mathcal{C}_j}\sigma_i\boldsymbol{c}_j\rangle\right] + \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}_\alpha} \langle h, \frac{1}{m}\sum_{i=1}^{m}\sigma_i(\boldsymbol{x}_i - c(\boldsymbol{x}_i))\rangle\right]
\end{aligned} \tag{3.4}$$

We take these two terms in turn.

$$\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}_\alpha}\langle h,\frac{1}{m}\sum_{j=1}^{N}\sum_{i:\boldsymbol{x}_i\in\mathcal{C}_j}\sigma_i\boldsymbol{c}_j\rangle\right]\leq\frac{1}{m}\sum_{j=1}^{N}\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{h_j\in\mathcal{H}_\alpha}\left(\left(\sum_{i:\boldsymbol{x}_i\in\mathcal{C}_j}\sigma_i\right)\langle h_j,\boldsymbol{c}_j\rangle\right)\right]$$
$$\leq\frac{B}{m}\sum_{j=1}^{N}\mathbb{E}_{\boldsymbol{\sigma}}\left[\left|\sum_{i:\boldsymbol{x}_i\in\mathcal{C}_j}\sigma_i\right|\right]$$
$$\leq\frac{B}{m}\sum_{j=1}^{N}\sqrt{m_j}\leq B\sqrt{\frac{N}{m}}. \tag{3.5}$$

The final lines hold by the concavity of the square root and since $\sum_{j=1}^{N}m_j=m$. For the second term we follow the procedure in Kakade et al. (2008): denote, $\boldsymbol{\theta}:=\frac{1}{m}\sum_{i=1}^{m}\sigma_i(\boldsymbol{x}_i-c(\boldsymbol{x}_i))$. By Fenchel's inequality we have, for any $\lambda>0$, $\langle h,\lambda\boldsymbol{\theta}\rangle\leq F(h)+F^\star(\lambda\boldsymbol{\theta})$, so,

$$\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}_\alpha}\langle h,\boldsymbol{\theta}\rangle\right]\leq\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}_\alpha}\left(\frac{F(h)}{\lambda}\right)+\frac{F^\star(\lambda\boldsymbol{\theta})}{\lambda}\right]$$
$$\leq\frac{\alpha}{\lambda}+\frac{1}{\lambda}\mathbb{E}_{\boldsymbol{\sigma}}\left[F^\star(\lambda\boldsymbol{\theta})\right] \tag{3.6}$$

We have that $||\frac{\lambda}{m}\sigma_i(\boldsymbol{x}_i-c(\boldsymbol{x}_i))||_F^*=\frac{\lambda d_F(\boldsymbol{x}_i,c(\boldsymbol{x}_i))}{m}$ and so by Lemma 3.3.1, $\mathbb{E}_{\boldsymbol{\sigma}}\left[F^\star(\lambda\boldsymbol{\theta})\right]\leq\frac{\lambda^2}{2\kappa m^2}\sum_{i=1}^{m}(d_F(\boldsymbol{x}_i,c(\boldsymbol{x}_i)))^2=\frac{\lambda^2\rho_{\mathcal{S}}}{2\kappa m}$. Therefore by picking $\lambda=\sqrt{\frac{2\alpha m\kappa}{\rho_{\mathcal{S}}}}$ in (3.6), we have,

$$\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}_\alpha}\langle h,\boldsymbol{\theta}\rangle\right]\leq\sqrt{\frac{2\alpha\rho_{\mathcal{S}}}{m\kappa}}. \tag{3.7}$$

Combining (3.4), (3.5) and (3.7) gives the result. □

Note that these bounds are optimized by the best k-means clustering for some k. In line with intuition, if the data distribution clusters and a good classifier respects this structure (i.e. has a small complexity) we can learn well with few examples and if the training sample reveals this structure we can be more confident in our risk analysis. In Appendix G we suggest a possible means of deriving a cluster structure-dependent risk analysis and regularization scheme from this result.

3.4 Application to transduction

Statistical analyses of induction typically require that the hypothesis class is not informed by available data instances, thus, being necessarily inherited from the geometry of the ambient representation space of the data, the metric in which structure is quantified in our theory is unlikely to ideally capture the *intrinsic* geometry of the data distribution. In the settings of transduction and semi-supervised learning the learner is more informed about the true nature of the data distribution, effectively reducing an element of uncertainty, and typically uses this information to choose a data-dependent hypothesis class implying a metric on the input space which captures the intrinsic geometry of the data. Furthermore, we will see that the empirically-defined metric implied on the input space by learning under typical "smoothness" assumptions is very sensitive to the clustering of data – much more so than any non-empirical metric can be – so the ideas above should be effective in this case. We recall the definitions relevant to transduction in the discussion of the subject in Section 1.3.1.

3.4.1 Transductive Rademacher complexity

Recalling Section 3.2, for clarity we henceforth denote the *transductive Rademacher complexity* by $\mathcal{R}_m^{\mathrm{trs}}(\cdot)$ when the draw of a sample is uniform without replacement from a finite set and $\mathcal{R}_m^{\mathrm{ind}}(\cdot)$ the standard inductive Rademacher complexity[4]. We specialize the bound provided by (3.3) to the transductive setting.

Corollary 3.4.1. *For a class $\mathcal{H}$ of bounded functions on a finite set $\mathcal{X}$, if $F : \mathcal{H} \to \mathbb{R}$ is κ-strongly convex w.r.t. a norm $||\cdot||_F$ on $\mathcal{H}$, then for all clusterings $\mathcal{C}$ of $\mathcal{X}$, for all $\alpha > 0$,*

$$\mathcal{R}_m^{\mathrm{trs}}(\mathcal{H}_\alpha) \leq B\mathbb{E}_{\mathcal{S}}\left[\sqrt{\frac{|\mathcal{C}_{\mathcal{S}}|}{m}}\right] + \sqrt{\frac{2\alpha\rho}{m\kappa}}, \tag{3.8}$$

where $\rho := \frac{1}{n}\sum_{i=1}^n d_F^2(\boldsymbol{x}_i, c(\boldsymbol{x}_i))$, $d_F(\cdot,\cdot)$ denotes the implied metric on $\mathcal{X}$, $B := \sup_{\boldsymbol{h}\in\mathcal{H}_\alpha, \boldsymbol{x}\in\mathcal{X}} |h(\boldsymbol{x})|$, expectation is w.r.t. the (uniform without replacement) draw of a sample $\mathcal{S} = \{\boldsymbol{X}_{s_1}, ...\boldsymbol{X}_{s_m}\}$ from $\mathcal{X}$ and $\mathcal{C}_{\mathcal{S}} := \{\mathcal{C}_k \in \mathcal{C} \ : \ \mathcal{C}_k \cap \mathcal{S} \neq \emptyset\}$ is the clustering restricted to the sample $\mathcal{S}$.

Note that the expectation can be evaluated with ease since the distribution of training samples is known.

Proof. In (3.3) we exploit the concavity of $\sqrt{\cdot}$ and then we evaluate the expectation. □

Binary Classifiers With Bounded Graph Cut

Transduction is typically posed as predicting the labeling of a partially labeled n-vertex graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. By representing each $v_i \in \mathcal{V}$ by the standard basis element $\boldsymbol{e}_i \in \mathbb{R}^n$ we seek a classifier $\boldsymbol{h} \in \mathcal{H}$, such that $\boldsymbol{h}(v_i) := h_i = \boldsymbol{h}^\top \boldsymbol{e}_i$ is the (soft) classification of vertex v_i. As discussed in Section 3.3.1 one principle involves minimizing the smoothness functional $F_{\boldsymbol{L}}(\boldsymbol{h}) := \frac{1}{2}\boldsymbol{h}^\top \boldsymbol{L}\boldsymbol{h}$, derived from the graph Laplacian[5]. Note that for $\boldsymbol{h} \in \{-1, 1\}^n$, $\frac{1}{4}\boldsymbol{h}^\top \boldsymbol{L}\boldsymbol{h} = \mathrm{cut}(\boldsymbol{h})$, the weighted sum of all edges connecting differently labeled vertices. This is 1-strongly convex w.r.t. $||\boldsymbol{h}||_{\boldsymbol{L}} := \sqrt{\boldsymbol{h}^\top \boldsymbol{L}\boldsymbol{h}}$ and the implied metric on $\mathcal{V}$ in this case is given by $d_{\boldsymbol{L}}(v_i, v_j) = \sqrt{(\boldsymbol{e}_i - \boldsymbol{e}_j)^\top \boldsymbol{L}^+(\boldsymbol{e}_i - \boldsymbol{e}_j)}$, the square root of the electrical resistance on the graph. The above result therefore bounds the Rademacher complexity of the class

$$\mathcal{H}_\phi := \{\boldsymbol{h} \in \{-1, 1\}^n \ : \ \boldsymbol{h}^\top \boldsymbol{L}\boldsymbol{h} \leq \phi\}$$

of binary classifiers with bounded cut:

Corollary 3.4.2. *Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, for any clustering $\mathcal{C}$ of $\mathcal{V}$, for all $\phi > 0$,*

$$\mathcal{R}_m^{\mathrm{trs}}(\mathcal{H}_\phi) \leq \mathbb{E}_{\mathcal{S}}\left[\sqrt{\frac{|\mathcal{C}_{\mathcal{S}}|}{m}}\right] + \sqrt{\frac{\phi\rho}{m}}. \tag{3.9}$$

where $\rho := \frac{1}{n}\sum_{i=1}^n d_{\boldsymbol{L}}^2(v_i, c(v_i))$ and $\mathcal{C}_{\mathcal{S}} := \{\mathcal{C}_k \in \mathcal{C} \ : \ \mathcal{S} \cap \mathcal{C}_k \neq \emptyset\}$ is the clustering restricted to the sample $\mathcal{S}$.

[4] Another form of transductive Rademacher complexity is studied in El-Yaniv and Pechyony (2007).

[5] There is a technical point here; because of the shift from single points to pairs of points arising in the cluster analysis, all duality inequalities that we want to hold do hold, which is not the case otherwise. In particular we do not need to restrict the function class to functions perpendicular to the null space of the Laplacian, as some analyses do.

Note that each centroid $c(v_i)$ is not a point on the graph but is represented in $\mathbb{R}^n$ by $\frac{1}{|\mathcal{C}_k|}\sum_{\{j:v_j\in\mathcal{C}_k\}}\boldsymbol{e}_j$ where k is such that $v_i \in \mathcal{C}_k$. Thus if $\mathcal{G}$ exhibits good k-means clustering in the (square root of the) resistance metric then the class of binary classifiers $\mathcal{H}_\phi$ is small. Because of the strong convexity framework we can also extend this analysis to the "p-resistances" of Chapter 5, a generalization of p-norms to graphs[6]: Lemma 5.3.5 establishes the $(p-1)$-strong convexity of the complexity $\frac{1}{2}\|\cdot\|^2_{\boldsymbol{\Psi},p}$.

Analysis for prototypical clusters

The prototypical example of a cluster is a clique, we consider the (unweighted) graph $\mathcal{K}$, a collection of N cliques $\mathcal{K}_1, ...\mathcal{K}_N$, such that $|\mathcal{K}_i| = k_i$, connected arbitrarily with edges (see Figure 3.1).

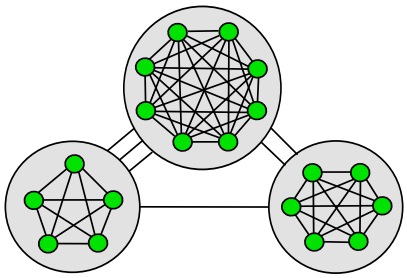


Figure 3.1: A Collection Of Cliques

By standard rules for resistors in series and parallel, the electrical resistance between any two distinct vertices in an k-clique is $\frac{2}{k}$, and, by Rayleigh's monotonicity principle, the intra-clique distances in a k-clique on $\mathcal{K}$ satisfy $d^2_{\boldsymbol{L}}(v_i, v_j) \leq \frac{2}{k}$. Now, for any set of n vertices $\mathcal{V}'$ we have

$$\begin{aligned}
\frac{1}{n}\sum_{i:v_i\in\mathcal{V}'} d^2_{\boldsymbol{L}}(v_i, c(v_i)) &= \frac{1}{n}\sum_{i:v_i\in\mathcal{V}'}\left(\boldsymbol{e}_i - \frac{1}{n}\sum_{j:v_j\in\mathcal{V}'}\boldsymbol{e}_j\right)^\top \boldsymbol{L}^+\left(\boldsymbol{e}_i - \frac{1}{n}\sum_{k:v_k\in\mathcal{V}'}\boldsymbol{e}_k\right)\\
&= \frac{1}{n}\sum_{i:v_i\in\mathcal{V}'}\left(\boldsymbol{e}_i - \frac{1}{n}\sum_{j:v_j\in\mathcal{V}'}\boldsymbol{e}_j\right)^\top \boldsymbol{L}^+\boldsymbol{e}_i\\
&= \frac{1}{2n^2}\sum_{i,j:v_i,v_j\in\mathcal{V}'}(\boldsymbol{e}_i - \boldsymbol{e}_j)^\top \boldsymbol{L}^+(\boldsymbol{e}_i - \boldsymbol{e}_j)\\
&= \frac{1}{2n^2}\sum_{i,j:v_i,v_j\in\mathcal{V}'} d^2_{\boldsymbol{L}}(v_i, v_j)\\
&\leq \frac{1}{2}\frac{1}{n}\sum_{i:v_i\in\mathcal{V}'}\frac{1}{n-1}\sum_{j:v_j\in\mathcal{V}', j\neq i} d^2_{\boldsymbol{L}}(v_i, v_j),
\end{aligned}$$

so, on $\mathcal{K}$, the resistance distance from any vertex v_i to the centroid of its clique $\mathcal{K}_j$ satisfies

[6]Further, this analysis is easily generalized to certain quadratic forms $F_{\boldsymbol{M}}(h) := \frac{1}{2}\boldsymbol{h}^\top\boldsymbol{M}\boldsymbol{h}$ where $\boldsymbol{M}$ is a p.s.d. matrix derived from the graph Laplacian. Sensible choices might include the "canonical regularizers" derived from the Laplacian in Smola and Kondor (2003), for example $\boldsymbol{L}^2$ or the heat kernel, or norms whose implied metric would be the diffusion distances considered in Coifman and Lafon (2006); Nadler et al. (2005), the implied metric on $\mathcal{V}$ in this case is given by $d_{\boldsymbol{M}}(v_i, v_j) = \sqrt{(\boldsymbol{e}_i - \boldsymbol{e}_j)^\top\boldsymbol{M}^+(\boldsymbol{e}_i - \boldsymbol{e}_j)}$.

$d^2_{\boldsymbol{L}}(v_i, c(v_i)) \leq \frac{1}{k_j}$. Thus, for the graph $\mathcal{K}$, (3.9) implies that

$$\mathcal{R}^{\text{trs}}_m(\mathcal{H}_\phi) \leq \sqrt{\frac{N}{m}} + \sqrt{\frac{N\phi}{mn}}. \tag{3.10}$$

Accounting for the cluster structure here offers significant improvement since the resistance distance between vertices in separate cliques is much larger (and on weighted graphs can be arbitrarily large).

Comparison to VC-dimension bounds

We now compare the result (3.9) to the bound of Kleinberg et al. (2004) on the VC-dimension of $\mathcal{H}_\phi$ for unweighted graphs:

$$\text{VC}(\mathcal{H}_\phi) = \mathcal{O}\left(\frac{\phi}{\phi^\star}\right), \tag{3.11}$$

where $\phi^\star$ is the minimum number of edges that must be removed in order to disconnect the graph. Since $\mathcal{R}_m(\mathcal{H}) = \mathcal{O}\left(\sqrt{\frac{\text{VC}(\mathcal{H})}{m}}\right)$, $\mathcal{R}_m(\mathcal{H})$ should be directly compared to $\sqrt{\frac{\text{VC}(\mathcal{H})}{m}}$.

We first consider the (n^2, n)-lollipop graph, see Figure 3.2, and compare the bounds for $\text{VC}(\mathcal{H}_\phi)$ and $\mathcal{R}_m(\mathcal{H}_\phi)$. For large n, since the VC dimension is independent of the distribution over vertices $\text{VC}(\mathcal{H}_\phi)$ measures the complexity of $\mathcal{H}_\phi$ on a path graph (the handle of the lollipop): for $n > \phi$, the VC dimension is equal to the VC dimension on the n-path graph, $\text{VC}(\mathcal{H}_\phi) = \phi + 1$. Whereas $\mathcal{R}_m(\mathcal{H}_\phi)$ will (approximately) measure the complexity of $\mathcal{H}_\phi$ on a n^2-clique, since the majority of vertices will be sampled from the lolly: from the argument above the bound (3.10) implies, for large n, $\mathcal{R}_m(\mathcal{H}_\phi) \lesssim \frac{1}{\sqrt{m}} + \sqrt{\frac{\phi}{mn^2}}$. Thus, even though the bound provided by (3.11) is tight (upto constants) in this case, a comparison of the bounds for $\mathcal{R}_m(\mathcal{H})$ and $\sqrt{\frac{\text{VC}(\mathcal{H})}{m}}$ show that the Rademacher bound is a significant improvement by a factor of approximately $\frac{1}{|\mathcal{V}|}$. This is a symptom of the VC dimension failing to capture the underlying distribution of instances.

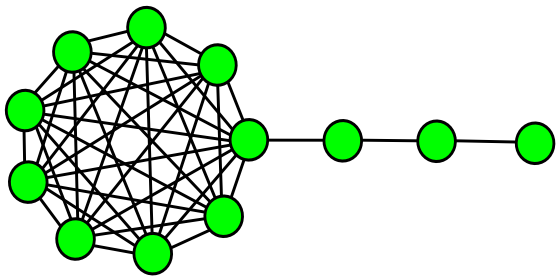


Figure 3.2: A (9,3)-lollipop

More generally, for an unweighted collection of cliques $\mathcal{K}$ which is fairly easily disconnected[7], e.g. $\phi^\star < \frac{n}{N}$, the bound (3.10) can be preferred to $\sqrt{\frac{\text{VC}(\mathcal{H}_\phi)}{m}} = \mathcal{O}\left(\sqrt{\frac{\phi}{m\phi^\star}}\right)$ for ϕ reasonably large, e.g. $\phi > N\phi^\star$. We note that because of the appearance of the $\sqrt{\frac{1}{n}}$ term in the bound (3.10) there is a lot of slack to relax the connectivity of the graph while still maintaining a good bound.

[7]Note that $\phi^\star$ doesn't reveal much about graph structure and could realistically be as small as 1 in practical applications

We note however that at the other end of the connectivity spectrum the bound (3.9) degrades: for example, for an unweighted path graph (3.9) becomes vacuous, at least for small m, and the VC bound is tight. This situation is improved by passing to p-resistances of Chapter 5: essentially the bound (3.8) holds simultaneously over a family of p-norms defined on the graph labellings and p-resistance, for $p \to 1$, is more suitable when the graph is sparse and to a large extent solves the problem encountered here. We note however that (3.9) degrades here because $d^2_{\boldsymbol{L}}(v_i, c(v_i)) = \mathcal{O}(n)$ for a path graph, and this situation is far from typical in practical machine learning applications, and, in anycase, we can always upper bound $\mathcal{R}_m(\mathcal{H})$ with an equivalent VC term in any bound whenever the latter is sharper (see e.g. Kääriäinen).

3.4.2 Transductive risk analysis

The following risk bound essentially due to Pelckmans and Suykens (2007) but slightly generalized here, is valid in the transductive setting[8]. For completeness a proof is supplied in Appendix H.

Theorem 3.4.3. *(Pelckmans and Suykens, 2007) For a given loss function $\ell : \mathcal{D} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$, K-Lipschitz in its first argument, bounded by C, for any $\delta > 0$, simultaneously for all $h \in \mathcal{H}$,*

$$\mathbb{P}_{\mathcal{S}} \Bigg(\mathrm{risk}^{\ell}_{\mathcal{T}}(h) \leq \widehat{\mathrm{risk}}^{\ell}_{\mathcal{S}}(h) + 2K \frac{m+u}{\max(m,u)} \mathcal{R}^{\mathrm{trs}}_{\min(m,u)}(\mathcal{H}) \\ + C \left(\frac{1}{m} + \frac{1}{u} \right) \sqrt{\frac{\min(m,u)}{2} \log \frac{1}{\delta}} \Bigg) \geq 1 - \delta,$$

where probability is w.r.t. the (uniform, without replacement) draw of the training sample $\mathcal{S} = \{(\boldsymbol{X}_{s_1}, Y_{s_1}), ...(\boldsymbol{X}_{s_m}, Y_{s_m})\}$ from $\mathcal{Z}$ and $\mathcal{T} \cup \mathcal{S} = \mathcal{Z}$.

We specialize this to the case of predicting the binary labeling of a graph $\mathcal{G}$ and apply the bound (3.9). Let $\mathcal{H} = \{-1, 1\}^n$ and $F_{\boldsymbol{L}}(\boldsymbol{h}) = \frac{1}{2}\boldsymbol{h}^\top \boldsymbol{L}\boldsymbol{h}$ where $\boldsymbol{L}$ is the Laplacian of $\mathcal{G}$. For simplicity we suppose $m < u$. We have $\mathcal{D} = \mathcal{Y} = \{-1, 1\}$ and by choosing the $0-1$ loss, which is $\frac{1}{2}$-Lipschitz for this function class, and bounded by 1, we have the following result bounding transductive binary classification risk:

Theorem 3.4.4. *Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ for any clustering $\mathcal{C}$ of $\mathcal{V}$, for any $\delta > 0$, with probability at least $1 - \delta$ over the draw of $\mathcal{S}$, simultaneously for all $\boldsymbol{h} \in \{-1, 1\}^n$,*

$$\mathrm{risk}_{\mathcal{T}}(\boldsymbol{h}) - \widehat{\mathrm{risk}}_{\mathcal{S}}(\boldsymbol{h}) \leq \frac{n}{u} \left(\mathbb{E}_{\mathcal{S}} \left[\sqrt{\frac{|\mathcal{C}_{\mathcal{S}}|}{m}} \right] + 2\sqrt{\frac{F'_{\boldsymbol{L}}(\boldsymbol{h})\rho}{m}} + \sqrt{\frac{\log \frac{1}{\delta}}{2m}} \right), \tag{3.12}$$

where $\rho := \frac{1}{n}\sum_{i=1}^n d^2_{\boldsymbol{L}}(v_i, c(v_i))$, $F'_{\boldsymbol{L}}(\boldsymbol{h}) := \min_{r \in \{1,2,...\}} \max\left(\phi_r, 2\frac{r+1}{r} F_{\boldsymbol{L}}(\boldsymbol{h})\right)$, $\phi_r := \frac{r \log 2}{2\rho}$ and $\mathcal{C}_{\mathcal{S}} := \{\mathcal{C}_k \in \mathcal{C} \,:\, \mathcal{S} \cap \mathcal{C}_k \neq \emptyset\}$ is the clustering restricted to the sample $\mathcal{S}$.

Proof. Define the stratification[9]: $\mathcal{H}^{(0)} = \{\}$ and, for $t \in \{1, 2, ...\}$, $\mathcal{H}^{(t)} = \mathcal{H}_{\phi_t}$. Theorem 3.4.3 implies

[8] As $u \to \infty$ we recover the inductive bound of Theorem 3.2.1.

[9] This technique is similar to that employed in (Balcan and Blum, 2010, Theorem 12).

that with probability at least $1 - \frac{\delta}{2^t}$ simultaneously for all $\boldsymbol{h} \in \mathcal{H}^{(t)} \backslash \mathcal{H}^{(t-1)}$ we have,

$$\begin{aligned}
\mathrm{risk}_{\mathcal{T}}(\boldsymbol{h}) - \widehat{\mathrm{risk}}_{\mathcal{S}}(\boldsymbol{h}) &\leq \frac{n}{u}\left(\mathcal{R}_m^{\mathrm{trs}}(\mathcal{H}_{\phi_t}) + \sqrt{\frac{\log\frac{2^t}{\delta}}{2m}}\right) \\
&\leq \frac{n}{u}\left(\mathbb{E}_{\mathcal{S}}\left[\sqrt{\frac{|\mathcal{C}_{\mathcal{S}}|}{m}}\right] + \sqrt{\frac{\phi_t \rho}{m}} + \sqrt{\frac{t\log 2}{2m}} + \sqrt{\frac{\log\frac{1}{\delta}}{2m}}\right) \\
&\leq \frac{n}{u}\left(\mathbb{E}_{\mathcal{S}}\left[\sqrt{\frac{|\mathcal{C}_{\mathcal{S}}|}{m}}\right] + 2\sqrt{\frac{\phi_t \rho}{m}} + \sqrt{\frac{\log\frac{1}{\delta}}{2m}}\right). \qquad (3.13)
\end{aligned}$$

Now noting that for $r \in \{1, 2, ...\}$, $\phi_t > \phi_r$ implies that $t \geq r + 1$ and $\phi_t \leq \frac{t}{t-1}\phi_{t-1} \leq \frac{r+1}{r}\phi_{t-1}$, so

$$\phi_t \leq \min_{r \in \{1,2,\ldots\}} \max\left(\phi_r, \frac{r+1}{r}\phi_{t-1}\right) \leq F'_{\boldsymbol{L}}(\boldsymbol{h}) \qquad (3.14)$$

The result follows by combining (3.14) with (3.13) and applying the union bound over all $t \in \{1, 2, ...\}$.

□

This bound resembles the bounds of Herbster (2008) for graph label prediction in the online framework which are related to a cover in the resistance metric.

This bound gives means of analyzing the transductive classification risk of any algorithm which produces a binary labeling of a graph, in terms of the structure of the underlying graph, including the harmonic energy minimization algorithm of Zhu et al. (2003a), the regularization of Belkin et al. (2004), the TSVM (Joachims, 1999), Mincut (Blum and Chawla, 2001) and the algorithm of Pelckmans et al. (2007). It also suggests an algorithm obtained by minimizing the bound simultaneously over classifiers and clusterings: essentially a Laplacian regularization whose regularization parameters are determined by the cluster structure of the graph.

Comparison

We compare Theorem 3.4.4 to similar bounds in the literature.

The following bound[10] is provided in Hanneke (2006).

Theorem 3.4.5. *(Hanneke, 2006, Corollary 2) With probability at least $1 - \delta$ simultaneously for all $\boldsymbol{h} \in \{-1, 1\}^n$,*

$$\mathrm{risk}_{\mathcal{T}}(\boldsymbol{h}) \leq \widehat{\mathrm{risk}}_{\mathcal{S}}(\boldsymbol{h}) + \sqrt{\frac{n(u+1)}{u^2}\frac{\frac{F_{\boldsymbol{L}}(\boldsymbol{h})}{\phi^\star}\ln n + \ln\frac{2(QW+1)}{\delta}}{2m}} \qquad (3.15)$$

where $\phi^\star$ is the minimum number of edges that must be removed to disconnect the graph, $W := \sum_{(i,j)\in\mathcal{E}} A_{ij}$, where $\boldsymbol{A}$ is the (weighted) adjacency of $\mathcal{G}$, and Q is the smallest positive rational number such that $QA_{ij} \in \mathbb{Z}$ for all $(i, j) \in \mathcal{E}$.

[10] We note that Hanneke (2006) provides a sharper implicit bound. Since we are interested in the essential dependence of these bounds on structural quantities of the graph we compare, for simplicity, to the explicit bound only.

Since this is essentially equivalent to a bound derived from the VC-dimension bound (3.11), we note that (ignoring multiplicative constants) (3.12) will be preferred to (3.15) whenever the Rademacher complexity bound (3.9) is preferred to the VC-dimension bound (3.11), and we refer the reader to the discussion of that subject in Section 3.4.1: for clustered graphs which are fairly easily disconnected, (3.12) seems preferable, nevertheless (3.15) remains tighter for sparser graphs, such as a path graph.

The following result relates the cardinality of $\mathcal{H}_\phi$ and transductive classification risk to the spectrum $\{\lambda_i\}_{i=1}^n$ of the graph Laplacian:

Theorem 3.4.6. *(Pelckmans et al., 2007, Theorem 1 and Theorem 2) With probability at least* $1-\delta$*,*

$$\sup_{\boldsymbol{h}\in\mathcal{H}_\phi} |\mathrm{risk}_{\mathcal{T}}(\boldsymbol{h}) - \widehat{\mathrm{risk}}_{\mathcal{S}}(\boldsymbol{h})| \leq \sqrt{\frac{2(n-m+1)}{nm}\log\frac{|\mathcal{H}_\phi|}{\delta}}$$

with $|\mathcal{H}_\phi| \leq \left(\frac{en}{n_\phi}\right)^{n_\phi}$ *where* $n_\phi := |\{\lambda_i : \lambda_i \leq \phi\}|$.

We compare these results with that given by (3.12). For the simple toy example given in Figure 3.1, $n_\phi = |\mathcal{V}|$ for $\phi \geq 3$ and so the bound on $|\mathcal{H}_\phi|$ is vacuous. For a practical comparison we consider the MNIST data set of hand-written digits (Lecun and Cortes) and form a 4-NN graph from 500 instances each of the digits "0" and "1". The two approaches to bounding the richness of $\mathcal{H}_\phi$ on this data set and graph are summarized in Table 3.1 (results are averaged over 5 randomly chosen sets of data). The (average) true labeling $\boldsymbol{y}$ has a cut of 8, and so $\boldsymbol{y}^\top \boldsymbol{L}\boldsymbol{y} = 32$.

Table 3.1: Practical evaluation of complexity bounds

ϕ	n_ϕ	$\lvert\mathcal{H}_\phi\rvert$ (Thm. 3.4.6)	$\mathcal{R}_m^{\mathrm{trs}}(\mathcal{H}_\phi)$ (Eq. (3.9))
10	902	$\left(\frac{1000e}{902}\right)^{902}$	$\frac{1}{\sqrt{m}}\left(\sqrt{2}+0.57\sqrt{10}\right)$
25	1000	e^{1000}	$\frac{1}{\sqrt{m}}\left(\sqrt{2}+0.57\sqrt{25}\right)$
50	1000	e^{1000}	$\frac{1}{\sqrt{m}}\left(\sqrt{2}+0.57\sqrt{50}\right)$

A comparison of the consequent bounds given by Theorem 3.4.6 and Theorem 3.4.3 apparently demonstrate that the bound (3.9) on the Rademacher complexity of $\mathcal{H}_\phi$ yields a sharper quantification of the richness of $\mathcal{H}_\phi$ on this data set. Note that n_ϕ tends to be very large on this particular dataset, rendering the bounds of Theorem 3.4.6 weak. One explanation for this is that graphs used in this context in machine learning tend to be very sparse (e.g. k-nearest neighbour graphs where $k \ll n$) the eigenvalues of sparse Laplacians are all small:

Claim 3.4.7. *Let* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ *be an unweighted* n*-vertex graph with maximal degree* $d = \max_{v\in\mathcal{V}} \mathrm{degree}(v)$*. Then let* $\boldsymbol{u} \in \mathbb{R}^n$ *be a normalized eigenvetor of the Laplacian* L *of* $\mathcal{G}$ *with eigenvalue* λ*. Then* $\lambda = \boldsymbol{u}^\top \boldsymbol{L}\boldsymbol{u} \leq 2d$*.*

Proof. We consider only the case where all vertices have degree d and prove that on such a graph any normalised vector $\boldsymbol{u}$ satisfies $\boldsymbol{u}^\top \boldsymbol{L}\boldsymbol{u} \leq 2d$ (the claim will then follow since removing edges stricly

decreases the value of $\boldsymbol{u}^\top \boldsymbol{L}\boldsymbol{u}$). We have

$$\begin{aligned}
\boldsymbol{u}^\top \boldsymbol{L}\boldsymbol{u} &= \sum_{(i,j)\in\mathcal{E}} (u_i - u_j)^2 \\
&= \sum_{(i,j)\in\mathcal{E}} u_i^2 + u_j^2 - 2u_i u_j \\
&= \frac{1}{2}\sum_i \sum_{j\sim i} u_i^2 + \frac{1}{2}\sum_j \sum_{i\sim j} u_j^2 - 2\sum_{(i,j)\in\mathcal{E}} u_i u_j \\
&= d + \sum_j u_j \sum_{i\sim j} u_i \\
&\leq d + \sqrt{\sum_j u_j^2}\sqrt{\sum_j \left(\sum_{i\sim j} u_i\right)^2} \\
&\leq d + d\sqrt{\sum_j \left(\frac{1}{d}\sum_{i\sim j} u_i\right)^2} \\
&\leq d + d\sqrt{\frac{1}{d}\sum_j \sum_{i\sim j} u_i^2} \\
&= d + d\sqrt{\frac{1}{d}\sum_i \sum_{j\sim i} u_i^2} \\
&\leq 2d.
\end{aligned}$$

□

When building k-nearest neighbour greaphs the maximal degree will often be small, and thus all eigenvalues will be small. Further the bound of Claim 3.4.7 is very crude and could clearly be generalized and improved; clearly in practice most eigenvalues on such graphs will be much smaller than even the bound suggests.

3.5 Application to semi-supervised learning

We indicate how the above ideas can be applied in a typical semi-supervised analysis to provide a semi-supervised bound in which the complexity is related to the cluster structure of the data sample.

The setting we consider is this: we are given a set $\mathcal{S} = \{(\boldsymbol{X}_{s_1}, Y_{s_1}), ...(\boldsymbol{X}_{s_m}, Y_{s_m})\}$ of m labeled instances drawn i.i.d. from P_{XY} over $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ and a set $\mathcal{X}_{\mathcal{T}} = \{\boldsymbol{X}_{t_1}, ...\boldsymbol{X}_{t_u}\}$ of u unlabeled instances drawn i.i.d. from the marginal P_X. Let $\mathcal{X}_{\mathcal{S}} := \{\boldsymbol{X}_{s_1}, ...\boldsymbol{X}_{s_m}\}$ and $\mathcal{I} := \mathcal{X}_{\mathcal{T}} \cup \mathcal{X}_{\mathcal{S}}$ denote the set of all $n = m + u$ instances. Consider a space $\mathcal{H}$ of bounded hypotheses mapping $\mathcal{X}$ to $\mathcal{D}$ and a complexity measure $F : \mathcal{H} \to \mathbb{R}_{\geq 0}$, κ-strongly convex w.r.t. a norm $||\cdot||_F$ on $\mathcal{H}$ and which is not informed by the sample of data instances, and let $\mathcal{H}_\alpha := \{h \in \mathcal{H} \ : \ F(h) \leq \alpha\}$. We then consider the space $\widetilde{\mathcal{H}}$ of functions from $\mathcal{H}$ with "modified" complexity $\widetilde{F} : \mathcal{H} \to \mathbb{R}_{\geq 0}$, $\widetilde{\kappa}$-strongly convex w.r.t. a norm $||\cdot||_{\widetilde{F}}$ on $\mathcal{H}$, which can take into account an empirical complexity measure derived from the entire sample of instances $\mathcal{I}$. The following semi-supervised bound on hypotheses from the empirically

defined $\widetilde{\mathcal{H}}_\beta := \{h \in \mathcal{H}_\alpha \ : \ \widetilde{F}(h) \leq \beta\}$ is essentially a version of the sample complexity result (Balcan and Blum, 2005, Theorem 5) for the statistical learning theory framework and specialized to our cluster structure method. Note that the statistical analysis relies on an initial hypothesis class $\mathcal{H}_\alpha$ chosen before the data is available as a tool to prove the convergence of transductive risk to inductive risk, and we perform two structurings of this hypothesis class, one which is data-dependent and one which is not.

Theorem 3.5.1. *Let $\ell : \mathcal{D} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ be a loss function, K-Lipschitz in its first argument and bounded by C. Then simultaneously for all $h \in \widetilde{\mathcal{H}}_\beta$ we have,*

$$\mathbb{P}\left(\mathrm{risk}^\ell(h) \leq \widehat{\mathrm{risk}}^\ell_{\mathcal{S}}(h) + 2K\mathcal{R}^{\mathrm{trs}}_m(\widetilde{\mathcal{H}}_\beta) + 2K\widehat{\mathcal{R}}^{\mathrm{ind}}_{\mathcal{I}}(\mathcal{H}_\alpha) + C\left(\sqrt{\frac{1}{2m}\log\frac{2}{\delta}} + 3\sqrt{\frac{1}{2n}\log\frac{4}{\delta}}\right)\right) \geq 1-\delta, \tag{3.16}$$

where probability is w.r.t. the draw of the labeled and unlabeled data from P_{XY}. Further, for all clusterings $\mathcal{C}, \mathcal{C}'$ of $\mathcal{I}$,

$$\mathcal{R}^{\mathrm{trs}}_m(\widetilde{\mathcal{H}}_\beta) \leq B\mathbb{E}\left[\sqrt{\frac{|\mathcal{C}_{\mathcal{X}_\mathcal{S}}|}{m}}\right] + \sqrt{\frac{2\beta}{mn\widetilde{\kappa}}\sum_{\boldsymbol{x}\in\mathcal{I}} d^2_{\widetilde{F}}(\boldsymbol{x}, c(\boldsymbol{x}))},$$

and,

$$\widehat{\mathcal{R}}^{\mathrm{ind}}_{\mathcal{I}}(\mathcal{H}_\alpha) \leq B\sqrt{\frac{|\mathcal{C}'|}{n}} + \frac{1}{n}\sqrt{\frac{2\alpha}{\kappa}\sum_{\boldsymbol{x}\in\mathcal{I}} d^2_F(\boldsymbol{x}, c'(\boldsymbol{x}))},$$

where $\mathcal{C}_{\mathcal{X}_\mathcal{S}} := \{\mathcal{C}_k \in \mathcal{C} \ : \ \mathcal{X}_\mathcal{S} \cap \mathcal{C}_k \neq \emptyset\}$ is the clustering restricted to the labeled instances, expectation is with respect to the (uniform without replacement) draw of $\mathcal{X}_\mathcal{S}$ from $\mathcal{I}$, $d_F(\cdot,\cdot)$ and $d_{\widetilde{F}}(\cdot,\cdot)$ are the metrics on $\mathcal{X}$ implied by $||\cdot||_F$ and $||\cdot||_{\widetilde{F}}$, and $B := \sup_{h\in\mathcal{H}_\alpha, \boldsymbol{x}\in\mathcal{X}} |\langle h, \boldsymbol{x}\rangle|$.

Proof. Let $\mathcal{T} := \{(\boldsymbol{X}_{t_1}, Y_{t_1}), ...(\boldsymbol{X}_{t_u}, Y_{t_u})\}$ where the Y_{t_i} are drawn from the conditional $P_{Y|X}$. The transductive bound Theorem 3.4.3 implies that,

$$\mathbb{P}\left(\sup_{h\in\widetilde{\mathcal{H}}_\beta}\left(\mathrm{risk}^\ell_{\mathcal{S}\cup\mathcal{T}}(h) - \widehat{\mathrm{risk}}^\ell_{\mathcal{S}}(h)\right) \leq 2K\mathcal{R}^{\mathrm{trs}}_m(\widetilde{\mathcal{H}}_\beta) + C\sqrt{\frac{1}{2m}\log\frac{2}{\delta}}\right) \geq 1 - \frac{\delta}{2},$$

where $\mathrm{risk}^\ell_{\mathcal{S}\cup\mathcal{T}}(h) := \frac{1}{|\mathcal{S}\cup\mathcal{T}|}\sum_{(X,Y)\in\mathcal{S}\cup\mathcal{T}} \ell(h(X), Y)$. The empirical counterpart of Theorem 3.2.1 (see e.g. Bousquet et al., 2003a) implies that

$$\mathbb{P}\left(\sup_{h\in\mathcal{H}_\alpha}\left(\mathrm{risk}^\ell(h) - \mathrm{risk}^\ell_{\mathcal{S}\cup\mathcal{T}}(h)\right) \leq 2K\widehat{\mathcal{R}}^{\mathrm{ind}}_{\mathcal{I}}(\mathcal{H}_\alpha) + 3C\sqrt{\frac{1}{2n}\log\frac{4}{\delta}}\right) \geq 1 - \frac{\delta}{2},$$

and (3.16) follows from the union bound. The final results follow from the bounds (3.8,3.2) on the transductive Rademacher complexity and empirical inductive Rademacher complexity. □

In particular, when $\widetilde{\mathcal{H}}$ is a "warped" RKHS of the form discussed in Section 3.3.1 then $\kappa = \widetilde{\kappa} = 1$ and we know the form of the implied metrics precisely.

The idea here is that the terms relating to the (non-empirical) hypothesis space $\mathcal{H}_\alpha$ decay as $\mathcal{O}(\frac{1}{\sqrt{n}})$ and so with plenty of unlabeled data[11] these terms are small, and we have a data-dependent hypothesis

[11] It is argued in Bennett and Demiriz (1998), for example, that unlabeled data should be almost free.

space $\widetilde{\mathcal{H}}_\beta$ within which we should be able to find a low risk hypothesis, even when β is small, if the true classifier respects the data structure; the advantage offered by unlabeled data is therefore seen to be that we can form an empirically defined hypothesis space (which should be more suitable for the learning process), and (with enough unlabeled data) obtain bounds similar to standard inductive bounds which are usually restricted to hypothesis spaces chosen before seeing the data and therefore without knowledge of the true data distribution. We have given a bound in which the complexity terms are entirely related to structure in the observed data using the results developed in previous sections.

3.6 Discussion

We have related the cluster structure in data to a common distribution-dependent notion of the capacity of a class of functions defined over the input space. This demonstrates the intuitive notion that when a data distribution exhibits a good k-means clustering in a metric which is related to our learning assumptions by a natural duality then our performance guarantees for the learning method will be sharper. We showed that accounting for cluster structure in this way can offer significant improvement in the quantification of the Rademacher capacity of a function class.

We specialized this observation to the case of transductive learning over a graph: when a graph exhibits a good k-means clustering in the resistance metric, relating the richness of the class of binary labellings defined over the vertices of the graph to this cluster structure appears to allow a sharp accounting of the richness of the function class. It is unknown whether the information contained in the clustering which is relevant to learning is contained in a simpler object like the Laplacian spectrum; after all finding the best k-means clustering is in general an NP-hard problem (though good clusterings can usually be found more easily). An open problem is therefore to improve the spectral approach or to identify other relevant structural features which can be tightly related to the performance of a learned classifier.

Cluster structure is very likely prevalent in "small world" networks which have been a huge focus of recent interest (Chung and Lu, 2006; Durrett, 2006) due to their apparent ubiquity in diverse domains of the information age – the web, social and biological networks etc.. Such networks are known to be generally very sparse but highly concentrated around "hubs" with low degree nodes belonging to dense (low resistance) subgraphs – precisely the type of graphs which exhibit clustering in the resistance metric.

Chapter 4

Efficient transduction by graph linearization

Abstract

We study the problem of efficiently learning the labeling of a large graph in the online setting, presenting algorithms with performance guarantees related to natural quantities associated with the graph. We show a fundamental limitation of a standard Laplacian-based interpolation algorithm: the number of mistakes made may be proportional to the square root of the number of vertices, even when tackling simple problems. We present an efficient algorithm which achieves a logarithmic mistake bound. It is based on the notion of a *spine*, a path graph which provides a linear embedding of the original graph. In practice, graphs may exhibit cluster structure; thus in the last part, we present a modified algorithm which achieves the "best of both worlds": it performs well *locally* in the presence of cluster structure, and *globally* on large diameter graphs.

4.1 Introduction

Huge data sets with a high degree of data-defined structure are increasingly common in practical machine learning applications meaning that efficient methods are vital. Particularly common is the situation in which data is represented as a graph. Many semi-supervised and transductive learning methods do not scale well in the amount of (unlabelled) data. Standard methods involving minimising a cost function derived from a graph Laplacian, for example, typically require the inversion of the Laplacian matrix. Online learning is inherently an efficient learning strategy, and in the graph prediction setting we must think of ways to exploit the structure defined by the graph in an efficient way without, for example, inverting the full Laplacian matrix. This research is about *efficiently* learning the labelling of a graph in the online framework, while maintaining good performance guarantees relative to natural quantities associated with the graph: we present an efficient algorithm which attains an upper bound on the number of mistakes superior to a lower bound we prove for a standard technique of transductive learning, that of harmonic energy minimization (Zhu et al., 2003a) discussed in Section 1.3.1i. The general idea is to define certain Markov random fields over the labellings of a graph. It is well-known that marginalizing

a discrete random field is intractable for general graphs. On trees marginalization can be achieved in time complexity linear in the number of vertices, by belief propagation. Here, by utilising a certain linear embedding, we define a random field that can be marginalized to produce sequential predictions in logarithmic time.

We study the problem of predicting the labelling of a graph in the online learning framework. The strength of the methods in Herbster and Pontil (2007); Herbster (2008) is in the case when the graph exhibits "cluster structure". The apparent deficiency of these methods is that they have poor bounds when the graph diameter is large relative to the number of vertices. We observe that this weakness is not due to insufficiently tight bounds, but is a problem in their performance. In particular, we discuss an example of a n-vertex labelled graph with a *single edge* between disagreeing label sets. On this graph, sequential prediction using the common method of harmonic energy minimization based upon minimising a cost derived from the graph Laplacian , subject to constraints, incurs $\Omega(\sqrt{n})$ mistakes (see Proposition 4.2.1). The expectation is that the number of mistakes incurred by an optimal online algorithm is bounded by $O(\ln n)$.

We solve this problem by observing that there exists an approximate structure-preserving embedding of any graph into a path graph. In particular the cut-size of any labelling is increased by no more than a factor of two. We call this embedding a *spine* of the graph. The spine is the foundation on which we build two algorithms. Firstly we study prediction on the spine with the majority vote classifier for a particular Markov random field. In the noiseless case we demonstrate that this equivalent to the 1-nearest-neighbor algorithm acting on the spine. A logarithmic mistake bound for learning on a path graph is proved by the Halving algorithm analysis. We further consider this algorithm in the presence of noise. Secondly, we use the spine of the graph as a foundation to add a binary *support tree* to the original graph. This enables us to prove a bound which is the "best of both worlds" – if the predicted set of vertices has cluster-structure we will obtain a bound appropriate for that case but if, instead, the predicted set exhibits a large diameter we will obtain a polylogarithmic bound.

Denoting by $\Phi_{\mathcal{G}}(\boldsymbol{u})$ the size of the *cut* induced by a binary labelling $\boldsymbol{u} \in \{-1,1\}^n$ of an n-vertex graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ (see (4.1)) our first algorithm predicts every vertex of $\mathcal{G}$ in $\mathcal{O}(|\mathcal{V}| \log |\mathcal{V}| + |\mathcal{E}|)$ time with a mistake bound of $\mathcal{O}\left(\Phi_{\mathcal{G}}(\boldsymbol{u}) \log\left(\frac{|\mathcal{V}|}{\Phi_{\mathcal{G}}(\boldsymbol{u})}\right)\right)$.

4.1.1 Previous Work

In Herbster and Pontil (2007); Herbster (2008) the online graph labelling problem was studied. An aim of those papers was to provide a natural interpretation of the bound on the cumulative mistakes of the kernel Perceptron when the kernel is the pseudoinverse of the graph Laplacian – bounds in this case being relative to the cut and (resistance) diameter of the graph. The online graph labelling problem is also studied in Pelckmans and Suykens (2008) and Cesa-Bianchi et al. (2009a), and here the graph structure is not given initially. A slightly weaker logarithmic bound for the online graph labelling problem has also been independently derived via a connection to an online routing problem in Fakcharoenphol and

Kijsirikul (2008).

4.2 Background

In this section, we describe the problem of online graph labelling, recall some common approaches to this problem and point out weaknesses of these methods, which motivate our subsequent analysis.

4.2.1 The online graph labelling problem

Recalling the notions outlined in Section 1.3.1 we let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected n-vertex graph and in this chapter, we consider only *connected* graphs, that is, graphs for which there exists a path between any two vertices.

We study the problem of predicting the labelling of a graph in the online learning framework. Consider the following game for predicting the labelling of a graph: *Nature* presents a graph; *nature* queries a vertex v_{i_1}; the *learner* predicts the label of the vertex $\hat{y}_1 \in \{-1, 1\}$; *nature* presents a label y_1; *nature* queries a vertex v_{i_2}; the *learner* predicts $\hat{y}_2$; and so forth. The learner's goal is to minimise the total number of mistakes $M = |\{t : \hat{y}_t \neq y_t\}|$. If nature is adversarial, the learner will always mispredict, but if nature is regular or simple, there is hope that a learner may make only a few mispredictions. Thus, a central goal of on-line learning is to design algorithms whose total mispredictions can be bounded relative to the complexity of nature's labelling.

We shall study the case of a consistent labelling, but also address the case of noisy labelling. The former case means that whenever a vertex appears more than once in the trial sequence, nature will always present the same label for it. Hence, in this case we may speak of a "true" underlying labeling of the graph. The latter permits an inconsistent trial sequence.

As in earlier chapters, the complexity measure which plays a central role in our analysis, is the *cut size* of a graph labelling $\boldsymbol{u} \in \{-1, 1\}^n$, which is defined as

$$\Phi_{\mathcal{G}}(\boldsymbol{u}) = \sum_{(i,j)\in\mathcal{E}} A_{ij}(u_i - u_j)^2 = \frac{1}{4}\boldsymbol{u}^\top \boldsymbol{L}\boldsymbol{u}. \tag{4.1}$$

We also say that a *cut* occurs on edge (i, j) if $u_i \neq u_j$, so that $\Phi_{\mathcal{G}}(\boldsymbol{u})$ measures the number of such cuts. Sometimes, we will evaluate equation (4.1) at continuous labellings $\boldsymbol{u} \in \mathbb{R}^n$, still referring to it as the cut size.

4.2.2 Markov random fields and Gibbs measures

Given a working assumption that the true labelling of a graph induces a small cut, a natural learning methodology is to encode this assumption as a probability measure over labellings, which would be updated according to knowledge acquired during the learning process and used to produce predictions. One practical way of achieving this is by defining a *Markov random field* (MRF) over the labellings of a graph (see, for example, Kinderman and Snell (1980)).

Definition Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph and, for every vertex $v_i \in \mathcal{V}$, define its neighborhood set as $\mathcal{N}_i := \{v_j \in \mathcal{V} : \ (i,j) \in \mathcal{E}\}$. A collection of random variables $\{U_1, ...U_n\}$ drawn from a probability distribution P is a Markov random field with respect to $\mathcal{G}$ if it satisfies the equation,

$$P(U_i = u_i | U_j = u_j, \ \forall j \neq i) = P(U_i = u_i | U_j = u_j, \ \forall v_j \in \mathcal{N}_i), \qquad \forall \ v_i \in \mathcal{V}.$$

It is difficult to analyze MRFs using this conditional dependency structure alone, but it turns out that they arise only from the following specific class of probability measures which factorize over *cliques*. Recall that a subgraph $\mathcal{C}$ of $\mathcal{G}$ is a *clique* if every pair of two vertices in $\mathcal{C}$ are connected by an edge. We denote by $C(\mathcal{G})$ the set of all cliques of $\mathcal{G}$.

Definition A probability distribution P defined over a space $\Omega \subseteq \mathbb{R}^n$ of labellings of an n-vertex graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a *Gibbs measure* if its probability (mass or density) function $p(\cdot)$ factorizes over cliques,

$$p(\boldsymbol{u}) = \frac{1}{Z} \exp\left(- \sum_{\mathcal{C} \in C(\mathcal{G})} E_{\mathcal{C}}(\boldsymbol{u}_{\mathcal{C}}) \right),$$

for some choice of "potential functions" $E_{\mathcal{C}}$ and where the sum is over the set $C(\mathcal{G})$ all cliques of $\mathcal{G}$, and $Z = \sum_{\boldsymbol{u} \in \Omega} \exp\left(- \sum_{\mathcal{C} \in C(\mathcal{G})} E_{\mathcal{C}}(\boldsymbol{u}_{\mathcal{C}}) \right)$, or $Z = \int_{\Omega} \exp\left(- \sum_{\mathcal{C} \in C(\mathcal{G})} E_{\mathcal{C}}(\boldsymbol{u}_{\mathcal{C}}) \right) \mathrm{d}\boldsymbol{u}$ if $p(\cdot)$ is a density, are normalizing "partition functions".

It is a fundamental result of Hammersley-Clifford that a collection of random variables with distribution P with a positive mass or density function is a MRF with respect to a graph $\mathcal{G}$ if and only if P is a Gibbs measure over the labellings of $\mathcal{G}$ (see, for example, Grimmett (1973) and Lauritzen (1996) for a rigorous presentation). It readily follows, for every $\gamma > 0$, that the Gibbs measure

$$p(\boldsymbol{u}) = \frac{1}{Z} \exp\left(-\gamma \Phi_{\mathcal{G}}(\boldsymbol{u}) \right), \quad \boldsymbol{u} \in \{-1, 1\}^n, \tag{4.2}$$

is a MRF with respect to $\mathcal{G}$ (as is the relaxation to the continuous distribution over real-valued labellings); a factorization of $\Phi_{\mathcal{G}}(\boldsymbol{u})$ into 2-cliques identified with the edges is given by equation (4.1).

4.2.3 Predicting the labelling of a graph with Markov random fields and Gibbs measures

Online learning is concerned with proving "worst-case" bounds without probabilistic assumptions on the data generation process. Surprisingly, however this goal is achieved by the adaptation of probabilistically motivated algorithms. We describe two algorithms of this kind, which play a central role in our development. We denote with a bold capital letter the vector valued random variable drawn from a MRF. Given a trial sequence $\mathcal{S} = \{(v_{i_1}, y_1), , ...(v_{i_m}, y_m)\}$, when P is a discrete distribution, we use the notation $p(\boldsymbol{u}|\mathcal{S})$ and $p(u_i|\mathcal{S})$ as shorthands for $P(\mathbf{U} = \boldsymbol{u} | U_{i_t} = y_t, \ t \leq m)$ and $P(U_i = u_i | U_{i_t} = y_t, \ t \leq m)$, respectively. When P is continuous $p(\boldsymbol{u}|\mathcal{S})$ denotes the conditional density at $\boldsymbol{u}$.

First, we consider the majority vote classifier associated with a Gibbs measure (4.2) on $\mathcal{G}$. This classifier is obtained by marginalising the posterior distribution at a given vertex i and taking the

weighted average prediction,

$$u_i^{\text{VOTED}} := \operatorname{sgn}\left(\sum_{u_i\in\{-1,1\}} u_i p(u_i|\mathcal{S})\right) \tag{4.3}$$
$$= \operatorname*{argmax}_{u_i\in\{-1,1\}} p(u_i|\mathcal{S})$$
$$= \operatorname{sgn}\left(\sum_{\boldsymbol{u}\in\{-1,1\}^n} p(\boldsymbol{u}|\mathcal{S})u_i\right).$$

We appeal to Haussler and Barron (1993) to note that this classifier is equivalent to the *Bayes classifier* for the 0-1 loss, under the assumption that the data are generated according to the same Gibbs measure. However, the majority vote classifier may be of limited utility for generic graphs as even the computation of the partition function (see Definition 4.2.2) is known to be $\sharp P$-complete (Jerrum and Sinclair, 1993, Theorem 15). However, by approximating the original graph with a tree the required marginalization may then be computed in linear time with belief propagation (see e.g. Yedidia et al., 2003) (see, for example, Mackay (2002) for a description of the method). Such an algorithm has been investigated on trees in the context of binary classification in Blum et al. (2004). We go further: in the next section we shall show that there exists a path graph (*spine*) which 2-approximates the original graph in cut-size (meaning that the cut size of no vertex labelling increases by more than a factor of 2 when the representation is transferred to the spine, see Section 4.3), such that the computation of the Bayes classifier is improved to $\mathcal{O}(\log n)$ time.

The second algorithm we consider is based on *minimum semi-norm interpolation*, whose value at the vertex i is defined as

$$u_i^{\text{INTERPOLANT}} = \left(\operatorname*{argmax}_{\boldsymbol{u}\in\mathbb{R}^n} p(\boldsymbol{u}|\mathcal{S})\right)_i$$
$$= \left(\operatorname*{argmin}_{\boldsymbol{u}\in\mathbb{R}^n}\{\boldsymbol{u}^\top \boldsymbol{L}\boldsymbol{u}|u_{i_t} = y_t,\ t\le m\}\right)_i. \tag{4.4}$$

A binary prediction is then obtained by taking the sign of the above quantity. Predicting with the minimum semi-norm interpolant (MNI) is called the method of *harmonic energy minimization* or *regularized interpolation* (Zhu et al., 2003a; Belkin and Niyogi, 2004) and is a foundational methodology in semi-supervised learning. It may be interpreted as a MAP prediction obtained from a relaxation of the discrete measure (4.2) to a continuous probability density, corresponding to a Gaussian random field[1]. Unlike the previous method, the minimum seminorm interpolation classifier can be computed efficiently as the optimization problem (4.4) consists in solving a linear system of equations.

Although MNI is computationally appealing, in Section 4.2.4 we shall show that this method does not enjoy a good mistake bound in the online framework. Specifically, we will provide an example of a simple graph, on which MNI may incur at least $\Omega(\sqrt{n})$ mistakes. On the other hand, if the trial sequence is well clustered relative to the resistance distance, MNI is known to perform significantly better than

[1]The semi-norm interpolant is also equal to both the mean and the majority vote classifier arising from this Gaussian random field, conditioned on observations. These facts follow from elementary properties of the Gaussian distribution.

the Bayes classifier (Herbster, 2008). Finally, we will present a new algorithm which leverages the two algorithms described above and establish a mistake bound for this algorithm, which is the "best of both worlds".

4.2.4 Limitations of online minimum semi-norm interpolation

Unfortunately, a deficiency of MSNI is that it may perform poorly on graphs with large diameters, whereas the majority vote classifier may still have a nontrivial $\log n$ upper bound. Specifically, we provide an example of a graph for which there exists a trial sequence on which MSNI will make at least $\Omega(\sqrt{n})$ mistakes.

Definition An *octopus graph* of size d is defined to be d path graphs (the *tentacles*) of length d (that is, with $d+1$ vertices) all adjoined at a common end vertex, to which a further single *head* vertex is attached, so that $n = |\mathcal{V}| = d^2 + 2$.

Proposition 4.2.1. *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an octopus graph of size d and $\boldsymbol{y} = (y_1, \ldots, y_{|\mathcal{V}|})$ the labelling such that $y_i = 1$ if v_i is the head vertex and $y_i = -1$ otherwise, see Figure 4.1. There exists a trial sequence for which online minimum semi-norm interpolation makes $\Omega(\sqrt{|\mathcal{V}|})$ mistakes.*

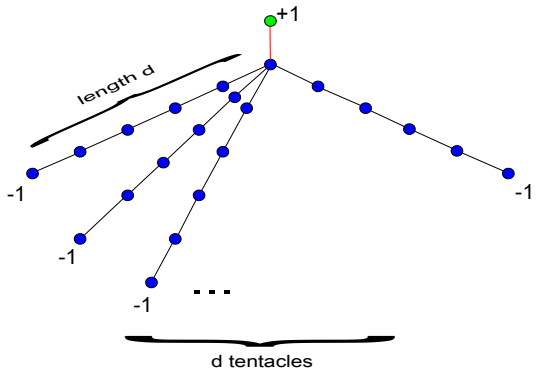


Figure 4.1: Partially labelled octopus graph

Proof. Let the first query vertex be the head vertex, and let the end vertex of a tentacle be queried at each subsequent trial. We show that this strategy forces at least d mistakes. The solution, $\boldsymbol{u}^{\text{INTERPOLANT}}$, to this minimum semi-norm interpolation with boundary values problem is precisely the *harmonic solution* (Doyle and Snell, 2000) – that is, for every unlabeled vertex v_j,

$$\sum_{i=1}^{n} A_{ij}(u_i^{\text{INTERPOLANT}} - u_j^{\text{INTERPOLANT}}) = 0.$$

If the graph is connected $\boldsymbol{u}^{\text{INTERPOLANT}}$ is unique and the graph labelling problem is identical to that of identifying the potential at each vertex of a resistive network defined on the graph where each edge

corresponds to a resistor of 1 unit; the harmonic principle corresponds to Kirchoff's current law in this case. Using this analogy, suppose that the end points of $k < d$ tentacles are labelled and that the end vertex v_q of an unlabelled tentacle is queried. Suppose a current of $k\lambda$ flows from the head to the body of the graph. By Kirchoff's law, a current of λ flows along each labelled tentacle (in order to obey the harmonic principle at every vertex it is clear that no current flows along the unlabelled tentacles). By Ohm's law $\lambda = \frac{2}{d+k}$. Minimum semi-norm interpolation therefore results in the solution,

$$u_q^{\text{INTERPOLANT}} = 1 - \frac{2k}{d+k} \geq 0 \text{ iff } k \leq d.$$

Hence the minimum semi-norm solution predicts incorrectly whenever $k < d$ and the algorithm makes at least d mistakes. □

The above demonstrates a limitation in the method of MSNI for predicting a graph labeling. We note that similar arguments can be formulated for other online algorithms based on the graph Laplacian, in particular the perceptron algorithm.

4.3 Graph linearization

We demonstrate a method of embedding data represented as a connected graph $\mathcal{G}$ into a path graph, we call it a *spine* of $\mathcal{G}$, which partially preserves the structure of $\mathcal{G}$. This construction will be used in the algorithms studied below. Let $\mathbb{P}_n$ be the set of path graphs with n vertices. We would like to find a path graph with the same vertex set as $\mathcal{G}$, which solves,

$$\min_{\mathcal{P} \in \mathbb{P}_n} \max_{\boldsymbol{u} \in \{-1,1\}^n} \frac{\Phi_{\mathcal{P}}(\boldsymbol{u})}{\Phi_{\mathcal{G}}(\boldsymbol{u})}.$$

If a Hamiltonian path $\mathcal{H}$ of $\mathcal{G}$ (a path on $\mathcal{G}$ which visits each vertex precisely once) exists, then the approximation ratio is $\frac{\Phi_{\mathcal{H}}(\boldsymbol{u})}{\Phi_{\mathcal{G}}(\boldsymbol{u})} \leq 1$. The problem of finding a Hamiltonian path is NP-complete however, and such a path is not guaranteed to exist. As we shall see, a spine $\mathcal{P}_{\text{spine}}$ of $\mathcal{G}$ may be found efficiently and satisfies $\frac{\Phi_{\mathcal{P}_{\text{spine}}}(\boldsymbol{u})}{\Phi_{\mathcal{G}}(\boldsymbol{u})} \leq 2$.

We now detail the construction of a spine of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}_{\mathcal{G}})$, with $|\mathcal{V}| = n$. Starting from any node, $\mathcal{G}$ is traversed in the manner of a *depth-first search* (that is, each vertex is fully explored before backtracking to the last unexplored vertex), and an ordered list $\mathcal{L} = \{v_{\ell_1}, v_{\ell_2}, \ldots, v_{\ell_{2k+1}}\}$ of the vertices ($k \leq |\mathcal{E}|$) in the order that they are visited is formed, allowing repetitions when a vertex is visited more than once. Note that each edge in $\mathcal{E}_{\mathcal{G}}$ is traversed no more than twice when forming $\mathcal{L}$. For a given list $\mathcal{L}$ define an edge multiset $\mathcal{E}_{\mathcal{L}} = \{(\ell_1, \ell_2), (\ell_2, \ell_3), \ldots, (\ell_{2k}, \ell_{2k+1})\}$ – the set of pairs of consecutive vertices in $\mathcal{L}$. Let $\boldsymbol{u}$ be an arbitrary labelling of $\mathcal{G}$ and denote, as usual, $\Phi_{\mathcal{G}}(\boldsymbol{u}) = \frac{1}{4}\sum_{(i,j)\in\mathcal{E}_{\mathcal{G}}}(u_i - u_j)^2$ and $\Phi_{\mathcal{L}}(\boldsymbol{u}) = \frac{1}{4}\sum_{(i,j)\in\mathcal{E}_{\mathcal{L}}}(u_i - u_j)^2$. Since the multiset $\mathcal{E}_{\mathcal{L}}$ contains every element of $\mathcal{E}_{\mathcal{G}}$ no more than twice, $\Phi_{\mathcal{L}}(\boldsymbol{u}) \leq 2\Phi_{\mathcal{G}}(\boldsymbol{u})$.

We then take any subsequence $\mathcal{L}'$ of $\mathcal{L}$ containing every vertex in $\mathcal{V}$ exactly once. A spine $\mathcal{P}_{\text{spine}} = (\mathcal{V}, \mathcal{E}_{\mathcal{L}'})$ is a graph formed by connecting each vertex in $\mathcal{V}$ to its immediate neighbours in the subsequence

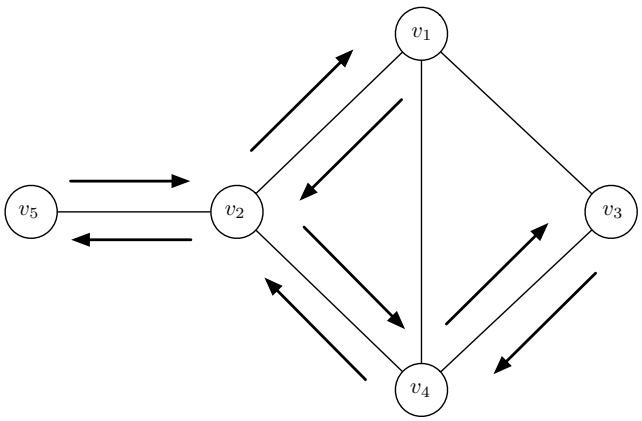


Figure 4.2: Example of spine construction.

$\mathcal{V}'_{\mathcal{L}}$ with an edge. Since a cut occurs between connected vertices v_i and v_j in $\mathcal{P}_{\text{spine}}$ only if a cut occurs on some edge in $\mathcal{E}_{\mathcal{L}}$ located between the corresponding vertices in the list $\mathcal{L}$ we have,

$$\Phi_{\mathcal{P}_{\text{spine}}}(\boldsymbol{u}) \leq \Phi_{\mathcal{L}}(\boldsymbol{u}) \leq 2\Phi_{\mathcal{G}}(\boldsymbol{u}). \tag{4.5}$$

Thus we have reduced the problem of learning the cut on a generic graph to that of learning the cut on a path graph. In the following we see that 1-nearest neighbour (1-NN) algorithm is an implementation of the majority vote classifier (4.3) from a natural MRF on the path. Note that the 1-NN algorithm does not perform well on general graphs; on the octopus graph discussed above, for example, it can make at least $\Omega(\sqrt{n})$ mistakes, and even $\Omega(n)$ mistakes on a related graph construction (Herbster and Pontil, 2007).

Finally, we note that the problem of embedding a graph in a path graph was considered in Hall (1970); Atkins et al. (1999). The essential idea is based on the computation of the eigenvector associated with the smallest nonzero eigenvalue of the graph Laplacian. The elements of this vector are then sorted and a path graph is obtained by reordering the vertex set accordingly. Although simple and elegant, it is apparent that this graph linearization does not enjoy the same approximation guarantee as in (4.5).

4.4 Predicting with a spine

4.4.1 The majority vote classifier defined on a spine

The general idea is to reduce an arbitrary graph to a simpler structure using the spine construction and define a Markov random field on the embedding obtained and predict with the majority vote classifier (4.3) from the posterior MRF obtained by a certain inference process. The key point of our construction is that the marginalisation required to produce the majority vote classifier from our predictive distribution can be performed in logarithmic time, rather than the linear time it generally takes to marginalize a discrete Markov random field on a tree.

We detail the construction of a particular Gibbs distribution over the labellings of a path graph. Let $\mathcal{P} = (\mathcal{V}, \mathcal{E})$ be a path graph, where $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ is the set of vertices and $\mathcal{E} =$

$\{(1,2),(2,3),\ldots,(n-1,n)\}$. Consider the probability distribution P over the space of all labellings $\boldsymbol{u} \in \{-1,1\}^n$ of $\mathcal{P}$ obtained by allowing a cut to occur on any given edge with probability α, independently of all other cuts; $P(U_{i+1} \neq U_i) = \alpha \; \forall i < n$. The position of all cuts fixes the labelling up to flipping every label, and each of these two resulting possible arrangements are equally likely. This recipe associates with each possible labelling $\boldsymbol{u} \in \{-1,1\}^n$ a probability $p(\boldsymbol{u})$ which is a function of the labelling's cut size

$$p(\boldsymbol{u}) = \frac{1}{2}\alpha^{\Phi_{\mathcal{P}}(\boldsymbol{u})}(1-\alpha)^{n-1-\Phi_{\mathcal{P}}(\boldsymbol{u})}. \tag{4.6}$$

This induces a full joint probability distribution on the space of vertex labels. In fact (4.6) is a Gibbs measure defined by (4.2) with $\gamma = -\ln\frac{\alpha}{1-\alpha}$, and as such defines a Markov random field over the space of binary labellings of $\mathcal{P}$.

Given a trial sequence $\mathcal{S} = \{(v_{i_1}, y_1), (v_{i_2}, y_2), ...(v_{i_m}, y_m)\}$, the Markov random field (4.6) can be used to define a posterior predictive measure $p(\cdot|\mathcal{S})$ over $\boldsymbol{u} \in \{-1,1\}^n$ by simply conditioning on the observed vertices,

$$p(\boldsymbol{u}|\mathcal{S}) := \frac{p(\mathcal{S}|\boldsymbol{u})p(\boldsymbol{u})}{p(\mathcal{S})} \tag{4.7}$$

where (with abuse of notation[2]) $p(\mathcal{S}|\boldsymbol{u})$ is the probability of observing the labels $\{y_1, ...y_m\}$ given the query sequence $\{v_{i_1}, ...v_{i_m}\}$ and the true labelling $\boldsymbol{u}$, which in general depends upon the noise model which is to be chosen, and $p(\mathcal{S}) = \sum_{\boldsymbol{u}\in\{-1,1\}^n} p(\mathcal{S}|\boldsymbol{u})p(\boldsymbol{u})$. Thus the posterior predictive measure is that obtained by Bayesian inference under a certain noise model. The noise model we assume is that each observed label in the trial sequence may be flipped with probability $\beta < \frac{1}{2}$, independently of all other observations. Thus, denoting for a given trial sequence $\mathcal{S}$, $M_{\mathcal{S}}(\boldsymbol{u}) := |\{(v_{i_t}, y_t) \in \mathcal{S} \,:\, u_{i_t} \neq y_t\}|$ the number of mistakes incurred by the hypothesis $\boldsymbol{u}$ on $\mathcal{S}$ we have,

$$\begin{aligned} p(\mathcal{S}|\boldsymbol{u}) &= \beta^{\frac{1}{2}\sum_{t\leq m}|y_t - u_{i_t}|}(1-\beta)^{\frac{1}{2}\sum_{t\leq m}|y_t + u_{i_t}|} \\ &= \beta^{M_{\mathcal{S}}(\boldsymbol{u})}(1-\beta)^{m-M_{\mathcal{S}}(\boldsymbol{u})}, \end{aligned}$$

and the predictive distribution (4.7) becomes,

$$p(\boldsymbol{u}|\mathcal{S}) = \frac{1}{2p(\mathcal{S})}\alpha^{\Phi_{\mathcal{P}}(\boldsymbol{u})}(1-\alpha)^{n-1-\Phi_{\mathcal{P}}(\boldsymbol{u})}\beta^{M_{\mathcal{S}}(\boldsymbol{u})}(1-\beta)^{m-M_{\mathcal{S}}(\boldsymbol{u})}. \tag{4.8}$$

In the case of no assumed noise we define $0^0 := 1$.

The majority vote classifier, defined by (4.3), from the predictive distribution (4.8) is then used to produce predictions on any new queried vertex v_i of $\mathcal{P}$,

$$\begin{aligned} u_i^{\text{VOTED}} &:= \operatorname{sgn}\left(\sum_{\boldsymbol{u}\in\{-1,1\}^n} p(\boldsymbol{u}|\mathcal{S})u_i\right) \\ &= \operatorname*{argmax}_{u_i\in\{-1,1\}} p(u_i|\mathcal{S}) \end{aligned} \tag{4.9}$$

[2] Since there is no distribution over the vertices queried, only over the labelling.

Using this method we can predict the labelling of an arbitrary graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, by first transferring the data representation to that of a spine $\mathcal{P}_{\text{spine}}$ of $\mathcal{G}$, and predicting with the majority vote classifier (4.9) acting on $\mathcal{P}_{\text{spine}}$. This methodology can be used to produce sequential predictions in the online setting, see Algorithm 1 in Figure 4.3.

Input: A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a trial sequence $\mathcal{S} := \{(v_{i_1}, y_1), ...(v_{i_m}, y_m)\}$

Parameters: $0 < \alpha, \beta < 0.5$

Create: A spine $\mathcal{P}_{\text{spine}}$ of $\mathcal{G}$

Initialization: $M := 0$

for $t = 1, ...m$ **do**

  Define: $\mathcal{S}_t := \{(v_{i_1}, y_1), ...(v_{i_{t-1}}, y_{t-1})\}$

  Define: the posterior predictive distribution $p(\boldsymbol{u}|\mathcal{S}_t)$ as in (4.8)

  Receive: $i_t \in \{1, ...n\}$

  Predict: $\hat{y}_t = u_{i_t}^{\text{VOTED}} = \text{argmax}_{u_{i_t} \in \{-1,1\}} p(u_{i_t}|\mathcal{S}_t)$

  Receive: y_t

  if $\hat{y}_t \neq y_t$ **then** $M = M + 1$

end

Figure 4.3: Algorithm 1: prediction with a spine

We observe that Algorithm 1 is in fact equivalent to (a version of) the Weighted Majority algorithm[3] (Littlestone and Warmuth, 1989) with prior weights $w(\boldsymbol{u}) = p(\boldsymbol{u})$ as defined by (4.6) and such that the weight of each labelling is multiplied by the update factor $\eta = \frac{\beta}{1-\beta}$ whenever it incurs a mistake, producing the posterior weighting

$$w(\boldsymbol{u}|\mathcal{S}) = \frac{1}{2}\alpha^{\Phi_{\mathcal{P}}(\boldsymbol{u})}(1-\alpha)^{n-1-\Phi_{\mathcal{P}}(\boldsymbol{u})}\left(\frac{\beta}{1-\beta}\right)^{M_{\mathcal{S}}(\boldsymbol{u})}, \tag{4.10}$$

which is equal to $p(\boldsymbol{u}|\mathcal{S})$ as defined by (4.8) up to a constant multiple. In the special case $\beta = 0$ Algorithm 1 is therefore equivalent to the halving algorithm (Barzdin and Frievald, 1972) (so-called because at each mistake hypotheses contributing at least half of the total probability mass are 'deleted').

4.4.2 Noiseless case

Assuming no noise, so that $\beta = 0$, we have that $p(\mathcal{S}|\boldsymbol{u}) = \mathbb{I}_{\{u_{i_t} = y_t \ \forall t \leq m\}}$, where $\mathbb{I}$ denotes the indicator function, and so the predictive distribution (4.7) is simply the Markov random field (4.6) conditioned on observed vertices,

$$p(\boldsymbol{u}|\mathcal{S}) := p(\boldsymbol{u}|u_{i_1} = y_{i_1}, ...u_{i_m} = y_{i_m}). \tag{4.11}$$

[3]We recall that the Weighted Majority algorithm, for binary classifiers proceeds as follows: a positive weight $w(f)$ is associated with each binary classifier f in a pool. At each trial Weighted Majority makes the prediction in agreement with the largest total weight in the pool. The weight of each function f which erred on the trial is then multiplied by a fixed *update factor* $0 \leq \eta < 1$, $w(f) \leftarrow \eta w(f)$.

Giving the majority vote classifier,

$$u_i^{\text{VOTED}} := \underset{u_i \in \{-1,1\}}{\operatorname{argmax}}\ p(u_i | u_{i_1} = y_{i_1}, ...u_{i_m} = y_{i_m}). \tag{4.12}$$

We now show that predicting with the majority vote classifier (4.12) on a path graph is equivalent to predicting with the nearest neighbour algorithm for all $0 < \alpha < \frac{1}{2}$. The nearest neighbour algorithm, in the standard online learning framework described above, attempts to predict a graph labelling by producing, for each query vertex v_{i_t}, the prediction $\hat{y}_t$ which is consistent with the label of the closest labelled vertex (and predicts randomly in the case of a tie). This equivalence gives an efficient implementation of Algorithm 1 in the noiseless case.

To demonstrate the equivalence we note a few properties of the distribution P defined by the measure (4.6). Since P defines a Markov random field it satisfies the Markov property

$$P(U_i = u | U_j = u_j \ \forall j \neq i) = P(U_i = u | U_j = u_j \ \forall v_j \in \mathcal{N}_i), \tag{4.13}$$

where here $\mathcal{N}_i$ is the set of vertices neighbouring v_i – those connected to v_i by an edge. We will give an equivalent Markov property which allows a more general conditioning to reduce to that over *boundary vertices*.

Definition Given a path graph $\mathcal{P} = (\mathcal{V}, \mathcal{E})$, a set of vertices $\mathcal{V}' \subset \mathcal{V}$ and a vertex $v_i \in \mathcal{V}$, we define the boundary vertices v_ℓ, v_r (either of which may be vacuous) to be the two vertices in $\mathcal{V}'$ that are closest to v_i in each direction along the path; its *nearest neighbours* in each direction.

The distribution P induced by (4.6) satisfies the following Markov property; given a partial labelling of $\mathcal{P}$ defined on a subset $\mathcal{V}' \subset \mathcal{V}$, the label of any vertex v_i is independent of all labels on $\mathcal{V}'$ except those on the vertices v_ℓ, v_r (either of which could be vacuous)

$$P(U_i = u | U_j = u_j, \ \forall j : v_j \in \mathcal{V}') \quad = \quad P(U_i = u | U_\ell = u_\ell, \ U_r = u_r). \tag{4.14}$$

Given the construction of the probability distribution formed by independent cuts on graph edges, we can evaluate conditional probabilities. For example, $p(U_j = u | U_k = u)$ is the probability of an even number of cuts between vertex v_j and vertex v_k. Since cuts occur with probability α and there are $\binom{|k-j|}{s}$ possible arrangements of s cuts we have

$$P(U_j = u | U_k = u) = \sum_{s \text{ even}} \binom{|k-j|}{s} \alpha^s (1-\alpha)^{|k-j|-s} = \frac{1}{2}(1 + (1-2\alpha)^{|k-j|}), \tag{4.15}$$

and likewise we have that

$$P(U_j \neq u | U_k = u) = \sum_{s \text{ odd}} \binom{|k-j|}{s} \alpha^s (1-\alpha)^{|k-j|-s} = \frac{1}{2}(1 - (1-2\alpha)^{|k-j|}), \tag{4.16}$$

which follow by forming the binomial expansion of $1 = (\alpha + (1-\alpha))^{|k-j|}$ and $(1-2\alpha)^{|k-j|} = (-\alpha + (1-\alpha))^{|k-j|}$ and adding and subtracting the resulting expressions. Note also that for any single vertex we have $P(U_i = u) = \frac{1}{2}$ for $u \in \{-1, 1\}$.

Theorem 4.4.1. *Given the task of predicting the labelling of an n-vertex path graph online, the nearest neighbour algorithm is an implementation of the majority vote classifier (4.9) for any $0 < \alpha < \frac{1}{2}$.*

Proof. Suppose that $t-1$ trials have been performed so that we have a partial labelling of a subset $\mathcal{V}' = \{(v_{i_1}, y_1), (v_{i_2}, y_2), \ldots, (v_{i_{t-1}}, y_{t-1})\} \subset \mathcal{V}$. Suppose the label of vertex v_{i_t} is queried so that Algorithm 1 makes the following prediction $\hat{y}_t$ for vertex v_{i_t}: $\hat{y}_t = y$ if $P(U_{i_t} = y | U_{i_j} = y_j \; \forall \, 1 \leq j < t) > \frac{1}{2}$, $\hat{y}_t = -y$ if $P(U_{i_t} = y | U_{i_j} = y_j \; \forall \, 1 \leq j < t) < \frac{1}{2}$ (and predicts randomly if this probability is equal to $\frac{1}{2}$). We first consider the case where the conditional labelling includes vertices on both sides of v_{i_t}. We have, by (4.14), that

$$\begin{aligned} P(U_{i_t} = y | U_{i_j} = y_j \; \forall \, 1 \leq j < t) &= P(U_{i_t} = y | U_\ell = y_{\tau(\ell)}, U_r = y_{\tau(r)}) \\ &= \frac{P(U_\ell = y_{\tau(\ell)} | U_r = y_{\tau(r)}, U_{i_t} = y) P(U_r = y_{\tau(r)}, U_{i_t} = y)}{P(U_\ell = y_{\tau(\ell)}, U_r = y_{\tau(r)})} \\ &= \frac{P(U_\ell = y_{\tau(\ell)} | U_{i_t} = y) P(U_r = y_{\tau(r)} | U_{i_t} = y)}{P(U_\ell = y_{\tau(\ell)} | U_r = y_{\tau(r)})} \end{aligned} \tag{4.17}$$

where v_ℓ and v_r are the boundary vertices and $\tau(\ell)$ and $\tau(r)$ are trials at which vertices v_ℓ and v_r are queried, respectively. We can evaluate the right hand side of this expression using (4.15, 4.16). To show equivalence with the nearest neighbour method whenever $\alpha < \frac{1}{2}$, we have from (4.15, 4.16, 4.17)

$$P(U_{i_t} = y | U_\ell = y, U_r \neq y) = \frac{(1 + (1-2\alpha)^{|\ell - i_t|})(1 - (1-2\alpha)^{|r - i_t|})}{2(1 - (1-2\alpha)^{|\ell - r|})}$$

which is greater than $\frac{1}{2}$ if $|\ell - i_t| < |r - i_t|$ and less than $\frac{1}{2}$ if $|\ell - i_t| > |r - i_t|$. Hence, this produces predictions exactly in accordance with the nearest neighbour scheme. We also have more simply that for all i_t, ℓ and r and $\alpha < \frac{1}{2}$

$$P(U_{i_t} = y | U_\ell = y, U_r = y) > \frac{1}{2}, \quad \text{and} \quad P(U_{i_t} = y | U_\ell = y) > \frac{1}{2}.$$

This proves the theorem for all cases. □

Performance analysis

We recall the following mistake bound for the Weighted Majority algorithm:

Theorem 4.4.2. *(Littlestone and Warmuth, 1994, Theorem 2.1) The number of mistakes, M, incurred by Weighted Majority on any sequence of instances and binary labels satisfies,*

$$M \leq \frac{\log_2 (w_{\text{init}} / w_{\text{fin}})}{\log_2 (2/(1+\eta))}.$$

where η is the update factor and w_{init}, w_{fin} are the totals of all initial and final weights respectively.

We now prove a mistake bound for Algorithm 1 in the noise free case.

Theorem 4.4.3. *Given the task of predicting the labelling of any unweighted, connected, n-vertex graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in the online framework, the number of mistakes, M, incurred by Algorithm 1 satisfies*

$$M \leq 2\Phi_{\mathcal{G}}(\boldsymbol{u}) \max \left[0, \log_2 \left(\frac{n-1}{2\Phi_{\mathcal{G}}(\boldsymbol{u})} \right) \right] + \frac{2\Phi_{\mathcal{G}}(\boldsymbol{u})}{\ln 2} + 1, \tag{4.18}$$

where $\boldsymbol{u} \in \{-1, 1\}^n$ is any labelling consistent with the trial sequence.

Proof. A direct application of the well-known bound for the halving algorithm (Theorem 4.4.2 with $\eta = 0$) gives,

$$M \leq \log_2\left(\frac{1}{p(\boldsymbol{u})}\right) = \log_2\left(\frac{2}{\alpha^{\Phi_{\mathcal{P}_{\text{spine}}}(\boldsymbol{u})}(1-\alpha)^{n-1-\Phi_{\mathcal{P}_{\text{spine}}}(\boldsymbol{u})}}\right),$$

where $\boldsymbol{u}$ is any labelling consistent with the trial sequence. By Theorem 4.4.1, for all $0 < \alpha < \frac{1}{2}$, the algorithm is independent of α, we choose $\alpha = \min(\frac{\Phi_{\mathcal{P}}(\boldsymbol{u})}{n-1}, \frac{1}{2})$ (note that the bound is vacuous when $\frac{\Phi_{\mathcal{P}}(\boldsymbol{u})}{n-1} > \frac{1}{2}$ since M is necessarily upper bounded by n) giving

$$\begin{aligned} M &\leq \Phi_{\mathcal{P}_{\text{spine}}}(\boldsymbol{u}) \log_2\left(\frac{n-1}{\Phi_{\mathcal{P}_{\text{spine}}}(\boldsymbol{u})}\right) + (n-1-\Phi_{\mathcal{P}_{\text{spine}}}(\boldsymbol{u})) \log_2\left(1 + \frac{\Phi_{\mathcal{P}_{\text{spine}}}(\boldsymbol{u})}{n-1-\Phi_{\mathcal{P}_{\text{spine}}}(\boldsymbol{u})}\right) + 1 \\ &\leq \Phi_{\mathcal{P}_{\text{spine}}}(\boldsymbol{u}) \log_2\left(\frac{n-1}{\Phi_{\mathcal{P}_{\text{spine}}}(\boldsymbol{u})}\right) + \frac{\Phi_{\mathcal{P}_{\text{spine}}}(\boldsymbol{u})}{\ln 2} + 1, \end{aligned}$$

since $\ln(1+x) \leq x$ for $x \geq 0$. Since this is an increasing function of $\Phi_{\mathcal{P}_{\text{spine}}}(\boldsymbol{u})$ for $\Phi_{\mathcal{P}_{\text{spine}}}(\boldsymbol{u}) \leq n-1$ and is vacuous at $\Phi_{\mathcal{P}_{\text{spine}}}(\boldsymbol{u}) \geq n-1$ (M is necessarily upper bounded by n) we upper bound by substituting $\Phi_{\mathcal{P}_{\text{spine}}}(\boldsymbol{u}) \leq 2\Phi_{\mathcal{G}}(\boldsymbol{u})$ (equation (4.5)). □

We observe that predicting with the spine is a minimax improvement over Laplacian minimal semi-norm interpolation: recall Proposition 4.2.1, there we showed that there exists a trial sequence such that Laplacian minimal semi-norm interpolation incurs $\Omega(\sqrt{n})$ mistakes. In fact this trivially generalizes to $\Omega(\sqrt{\Phi_{\mathcal{G}}(\boldsymbol{u})n})$ mistakes by creating a colony of $\Phi_{\mathcal{G}}(\boldsymbol{u})$ octopi then identifying each previously separate head vertex as a single central vertex. The upper bound (4.18) is smaller than the prior lower bound.

Efficient Implementation/Complexity analysis

The computational complexity for this algorithm is $O(|\mathcal{E}| + |\mathcal{V}| \log |\mathcal{V}|)$ time. We compute the spine in $O(|\mathcal{E}|)$ time by simply listing vertices in the order in which they are first visited during a depth-first search traversal of $\mathcal{G}$. Using online 1-NN requires $O(|\mathcal{V}| \log |\mathcal{V}|)$ time to predict an arbitrary vertex sequence using a self-balancing binary search tree (e.g., a red-black tree) as the insertion of each vertex into the tree and determination of the nearest left and right neighbour is $O(\log |\mathcal{V}|)$.

4.4.3 Noisy case

We now consider the more general case in which the vertex labels might be subject to noise. As well as giving bounds for the case of noisy observations, this allows us to prove regret bounds in which the number of mistakes incurred by our algorithm is related to the performance of any fixed classifier, and not necessarily a classifier which is correct on all trials, which might give tighter bounds even in the noiseless case.

Performance analysis

Theorem 4.4.4. *Given the task of predicting the labelling of any unweighted, connected, n-vertex graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in the online framework, the number of mistakes, M, incurred by Algorithm 1 on a trial*

sequence $\mathcal{S}$ satisfies,

$$M \leq \frac{1}{1+\log_2(1-\beta)}\left(\Phi_{\mathcal{P}_{\text{spine}}}(\boldsymbol{u})\log_2\left(\frac{1}{\alpha}\right)+(n-1-\Phi_{\mathcal{P}_{\text{spine}}}(\boldsymbol{u}))\log_2\left(\frac{1}{1-\alpha}\right)+1\right) + \frac{\log_2(1-\beta)-\log_2(\beta)}{1+\log_2(1-\beta)}M_{\mathcal{S}}(\boldsymbol{u}). \tag{4.19}$$

where $\boldsymbol{u} \in \{-1,1\}^n$ is any labelling of $\mathcal{G}$.

Proof. We observed in Section 4.4.1 that Algorithm 1 is identical to the Weighted Majority algorithm with posterior weights defined by (4.10) (an update factor $\eta = \frac{\beta}{1-\beta}$). The result then follows immediately from Theorem 4.4.2 by noting that $w_{\text{init}} = 1$ and $w_{\text{fin}} \geq w(\boldsymbol{u}|\mathcal{S}) = \frac{1}{2}\alpha^{\Phi_{\mathcal{P}}(\boldsymbol{u})}(1-\alpha)^{n-1-\Phi_{\mathcal{P}}(\boldsymbol{u})}\left(\frac{\beta}{1-\beta}\right)^{M_{\mathcal{S}}(\boldsymbol{u})}$. □

For the sake of illustrating what can be hoped for if the parameters are chosen well we now present a bound in which we suppose that the trial sequence $\mathcal{S}$ and number of mistakes $M_{\mathcal{S}}(\boldsymbol{u})$ incurred by each labelling is known a-priori so that the learner may tune the above bound by choosing α and β to be dependent on these quantities. Clearly this is an unrealistic learning setting (in fact with this information there is no learning to be done) we are simply illustrating the type of bound that can be achieved when the learning parameters to be chosen happen to perfectly align with the learning problem. The concept and proof are due to Cesa-Bianchi et al. (1997) – we repeat it here for convenience.

Corollary 4.4.5. *Given the task of predicting the labelling of any unweighted, connected, n-vertex graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in the online framework, the number of mistakes, M, incurred by Algorithm 1 with the tuning $\alpha := \min(\frac{\Phi_{\mathcal{P}_{\text{spine}}}(\boldsymbol{u})}{n-1}, \frac{1}{2})$ and $\frac{\beta}{1-\beta} = g\left(\sqrt{\frac{-\ln p(\boldsymbol{u})}{M_{\mathcal{S}}(\boldsymbol{u})}}\right)$, for any particular $\boldsymbol{u}$, where,*

$$g(z) := \frac{1}{1+2z+\frac{z^2}{\ln 2}},$$

satisfies,

$$M \leq 2M_{\mathcal{S}}(\boldsymbol{u}) + 2\sqrt{M_{\mathcal{S}}(\boldsymbol{u})\ln 2B(\boldsymbol{u})} + B(\boldsymbol{u})$$

where $\boldsymbol{u} \in \{-1,1\}^n$ is any labelling of $\mathcal{G}$ and

$$B(\boldsymbol{u}) := 2\Phi_{\mathcal{G}}(\boldsymbol{u})\max\left[0, \log_2\left(\frac{n-1}{2\Phi_{\mathcal{G}}(\boldsymbol{u})}\right)\right] + \frac{2\Phi_{\mathcal{G}}(\boldsymbol{u})}{\ln 2} + 1.$$

Proof. We again recall the equivalence with the Weighted Majority algorithm with the update factor $\eta = \frac{\beta}{1-\beta}$ as discussed in Section 4.4.1. We denote $x = \sqrt{\frac{-\ln p(\boldsymbol{u})}{M_{\mathcal{S}}(\boldsymbol{u})}}$ and we observe that $w_{\text{init}} = 1$ and

$w_{\text{fin}} \geq p(\boldsymbol{u})\eta^{M_{\mathcal{S}}(\boldsymbol{u})}$. Thus Theorem 4.4.2 implies that,

$$\begin{aligned}
M &\leq \frac{-\log_2(w_{\text{fin}})}{\log_2\left(\frac{2}{1+\eta}\right)} \\
&\leq \frac{-\ln(p(\boldsymbol{u})) - M_{\mathcal{S}}(\boldsymbol{u})\ln\eta}{\ln\left(\frac{2}{1+\eta}\right)} \\
&= 2M_{\mathcal{S}}(\boldsymbol{u}) + \frac{-\ln(p(\boldsymbol{u}))}{\ln\left(\frac{2}{1+\eta}\right)} + 2M_{\mathcal{S}}(\boldsymbol{u})\left(\frac{-\ln\eta}{2\ln\left(\frac{2}{1+\eta}\right)} - 1\right) \\
&= 2M_{\mathcal{S}}(\boldsymbol{u}) + 2M_{\mathcal{S}}(\boldsymbol{u})\left(\frac{x^2 - \ln g(x)}{2\ln\left(\frac{2}{1+g(x)}\right)} - 1\right) \\
&\leq 2M_{\mathcal{S}}(\boldsymbol{u}) + 2M_{\mathcal{S}}(\boldsymbol{u})\left(x + \frac{x^2}{2\ln 2}\right),
\end{aligned}$$

where in the final line we applied the following inequality, valid for any $z > 0$, (see (Cesa-Bianchi et al., 1997, Lemma 4.4.1)),

$$\frac{z^2 - \ln g(z)}{2\ln(\frac{2}{1+g(z)})} \leq 1 + z + \frac{z^2}{2\ln 2}.$$

We therefore have,

$$M \leq 2M_{\mathcal{S}}(\boldsymbol{u}) + 2\sqrt{M_{\mathcal{S}}(\boldsymbol{u})\ln 2\log_2\left(\frac{1}{p(\boldsymbol{u})}\right)} + \log_2\left(\frac{1}{p(\boldsymbol{u})}\right),$$

and $\log_2\left(\frac{1}{p(\boldsymbol{u})}\right) = \log_2\left(\frac{2}{\alpha^{\Phi_{\mathcal{P}_{\text{spine}}}(\boldsymbol{u})}(1-\alpha)^{n-1-\Phi_{\mathcal{P}_{\text{spine}}}(\boldsymbol{u})}}\right)$ is bounded exactly as in the proof of Theorem 4.4.3 (for the same choice of α). □

Efficient implementation

To demonstrate an efficient implementation of Algorithm 1 in the noisy case we show equivalence with predicting using the majority vote classifier from a certain conditional Markov random field defined on a graph construction called a *comb*.

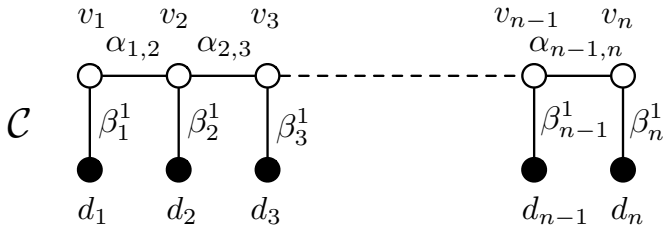


Figure 4.4: Comb

Given a spine $\mathcal{P}_{\text{spine}} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V} = \{v_1, ...v_n\}$ we define a new set of vertices $\mathcal{D} := \{d_1, ...d_n\}$ called *dongles* and define the comb $\mathcal{C} := (\mathcal{V} \cup \mathcal{D}, \mathcal{E} \cup \{(v_1, d_1), ...(v_n, d_n)\})$. With each edge (v_j, d_j) is associated a weight $\beta_j \in [0, 1]$, and each edge (v_j, v_{j+1}) a weight $\alpha_{j,j+1}$. For any

labelling $\hat{\boldsymbol{u}} \in \{-1,1\}^{2n}$ of $\mathcal{C}$ denote by $\hat{u}(v_j)$ the label of vertex v_j, and likewise $\hat{u}(d_j)$ the label of dongle d_j. Denote by $\boldsymbol{u} = (\hat{u}(v_j)) \in \{-1,1\}^n$ the labelling of the spine $\mathcal{P}_{\text{spine}}$. We define the following Markov random field over the space $\{-1,1\}^{2n}$ of labellings of $\mathcal{C}$,

$$q(\hat{\boldsymbol{u}}) = \frac{1}{2}\prod_{k=1}^{n-1} \alpha_{k,k+1}^{\frac{1}{2}|\hat{u}(v_k)-\hat{u}(v_{k+1})|}(1-\alpha_{k,k+1})^{\frac{1}{2}|\hat{u}(v_k)+\hat{u}(v_{k+1})|}\prod_{j=1}^{n}\beta_j^{\frac{1}{2}|\hat{u}(v_j)-\hat{u}(d_j)|}(1-\beta_j)^{\frac{1}{2}|\hat{u}(v_j)+\hat{u}(d_j)|}, \tag{4.20}$$

i.e. precisely in the manner of the measure (4.6) with a cut occurring on each edge (v_j, d_j) with probability β_j and on (v_j, v_{j+1}) with probability $\alpha_{j,j+1}$.

Given a trial sequence $\mathcal{S} := \{(v_{i_1}, y_1), ...(v_{i_m}, y_m)\}$, we set

$$\beta_j := \frac{\beta^{n_{j,1}}(1-\beta)^{n_{j,-1}}}{\beta^{n_{j,1}}(1-\beta)^{n_{j,-1}} + \beta^{n_{j,-1}}(1-\beta)^{n_{j,1}}},$$

where $n_{j,1} := |\{(v_{i_t}, 1) \in \mathcal{S} \,:\, i_t = j\}|$ and $n_{j,-1} := |\{(v_{i_t}, -1) \in \mathcal{S} \,:\, i_t = j\}|$ denote the number of times vertex v_j receives a positive and negative label. The measure (4.20) can then be used to define the following posterior predictive measure on the restriction $\boldsymbol{u}$ of the labelling $\hat{\boldsymbol{u}}$ to $\mathcal{P}_{\text{spine}}$, which is obtained by conditioning (4.20) on each dongle having a positive label,

$$q(\boldsymbol{u}|\mathcal{S}) := q(\hat{\boldsymbol{u}}|\hat{u}(d_i) = 1\ \forall i). \tag{4.21}$$

When $\alpha_{k,k+1} = \alpha$ for all k, we have,

$$\begin{aligned} q(\boldsymbol{u}|\mathcal{S}) &:= \frac{q(\boldsymbol{u}, \hat{u}(d_i) = 1\ \forall i)}{q(\hat{u}(d_i) = 1\ \forall i)} \\ &= \frac{\alpha^{\Phi_{\mathcal{P}_{\text{spine}}}(\boldsymbol{u})}(1-\alpha)^{n-1-\Phi_{\mathcal{P}_{\text{spine}}}(\boldsymbol{u})}\prod_{j=1}^{n}\beta^{\frac{1}{2}n_{j,1}|u_j-1|}(1-\beta)^{\frac{1}{2}n_{j,-1}|u_j-1|}\beta^{\frac{1}{2}n_{j,-1}|u_j+1|}(1-\beta)^{\frac{1}{2}n_{j,1}|u_j+1|}}{2q(\hat{u}(d_i) = 1\ \forall i)\prod_{j=1}^{n}\beta^{n_{j,1}}(1-\beta)^{n_{j,-1}} + \beta^{n_{j,-1}}(1-\beta)^{n_{j,1}}} \\ &= \frac{\alpha^{\Phi_{\mathcal{P}_{\text{spine}}}(\boldsymbol{u})}(1-\alpha)^{n-1-\Phi_{\mathcal{P}_{\text{spine}}}(\boldsymbol{u})}\beta^{M_{\mathcal{S}}(\boldsymbol{u})}(1-\beta)^{m-M_{\mathcal{S}}(\boldsymbol{u})}}{2q(\hat{u}(d_i) = 1\ \forall i)\prod_{j=1}^{n}\beta^{n_{j,1}}(1-\beta)^{n_{j,-1}} + \beta^{n_{j,-1}}(1-\beta)^{n_{j,1}}}. \end{aligned} \tag{4.22}$$

Since the denominators in both (4.22) and (4.8) are independent of $\boldsymbol{u}$ (and so are just normalization constants) when $\alpha_{k,k+1} = \alpha$ the predictive measure (4.21) is clearly seen to be identical to the predictive measure of Algorithm 1 defined by (4.8), i.e. $q(\boldsymbol{u}|\mathcal{S}) = p(\boldsymbol{u}|\mathcal{S})$. This leads to the alternative implementation of Algorithm 1 defined in Figure 4.5.

The remainder of this section will be a demonstration that the sequential updates required to marginalise the conditional Markov random field (4.21) and perform predictions at each trial in Figure 4.5 can be calculated in logarithmic time.

This is achieved by constructing a stack of combs each derived from that below using a *4-comb to 2-comb transform*. We explain this structure below. We first explain the basic 4-comb $\rightarrow$ 2-comb transform and refer to Figure 4.6. Given a 4-comb $\mathcal{C} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V} = \{v_1, v_2, v_3, v_4, d_1, d_1, d_3, d_4\}$ and weights $\alpha_{1,2}, \alpha_{2,3}, \alpha_{3,4}, \beta_1, \beta_2, \beta_3, \beta_4$ we wish to find a 2-comb $\mathcal{C}' = (\{v_1, v_4, d_1, d_4\}, \{(v_1, v_4), (v_1, d_1), (v_4, d_4)\})$ with weights $\alpha'_{1,4}, \beta'_1, \beta'_2$ such that for all $a, b \in \{-1, 1\}$ we have,

$$q'(\hat{U}'(v_1) = a, \hat{U}'(v_4) = b|\hat{U}'(d_j) = 1\ \forall j) = q(\hat{U}(v_1) = a, \hat{U}(v_4) = b|\hat{U}(d_j) = 1\ \forall j), \tag{4.23}$$

Input: A graph $\mathcal{G}$, a trial sequence $\mathcal{S} := \{(v_{i_1}, y_1), ...(v_{i_m}, y_m)\}$

Parameters: $0 < \alpha, \beta < 0.5$

Create a spine $\mathcal{P}_{\text{spine}} = (\mathcal{V}, \mathcal{E})$ of $\mathcal{G}$

Create a comb $\mathcal{C}$ on $\mathcal{P}_{\text{spine}}$

Initialization: $M = 0$, $\beta_j = \frac{1}{2}\ \forall j \leq n$, $\alpha_{j,j+1} = \alpha\ \forall j < n$

for $t = 1, ...m$ **do**

Define: $\mathcal{S}_t := \{(v_{i_1}, y_1), ...(v_{i_{t-1}}, y_{t-1})\}$

Define: the posterior predictive distribution $q(\boldsymbol{u}|\mathcal{S}_t)$ on $\mathcal{P}_{\text{spine}}$ as in (4.21)

Receive: $i_t \in \{1, ...n\}$

Predict: $\hat{y}_t = u_{i_t}^{\text{VOTED}} = \text{argmax}_{u_{i_t} \in \{-1,1\}}\, q(u_{i_t}|\mathcal{S}_t)$

Receive: y_t

Update: $n_{i_t,1}$, $n_{i_t,-1}$

Modify: $\beta_{i_t} \leftarrow \frac{\beta^{n_{i_t,1}}(1-\beta)^{n_{i_t,-1}}}{\beta^{n_{i_t,1}}(1-\beta)^{n_{i_t,-1}} + \beta^{n_{i_t,-1}}(1-\beta)^{n_{i_t,1}}}$

if $\hat{y}_t \neq y_t$ **then** $M = M + 1$

end

Figure 4.5: Algorithm 1 – comb implementation

where q and q' are the Gibbs distributions defined by (4.20) over labellings $\hat{U}$, $\hat{U}'$ on $\mathcal{C}$ and $\mathcal{C}'$ respectively.

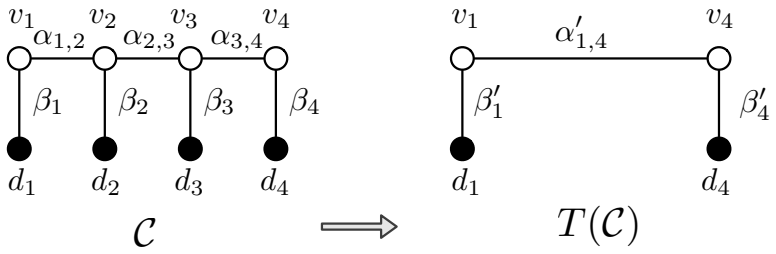


Figure 4.6: 4-comb to 2-comb transform

We denote $\mathcal{C}' = T(\mathcal{C})$ and we have the following result:

Lemma 4.4.6. *Given an arbitrary 4-comb $\mathcal{C}$ the transform $\mathcal{C}' = T(\mathcal{C})$ exists and the required weights are given by,*

$$\beta'_1 = f^{-1}\left(\sqrt{\frac{q_{1,1}}{q_{-1,-1}}\frac{q_{1,-1}}{q_{-1,1}}}\right)$$
$$\beta'_4 = f^{-1}\left(\sqrt{\frac{q_{1,1}}{q_{-1,-1}}\frac{q_{-1,1}}{q_{1,-1}}}\right)$$
$$\alpha'_{1,4} = f^{-1}\left(\sqrt{\frac{q_{1,1}q_{-1,-1}}{q_{-1,1}q_{1,-1}}}\right),$$

where $f^{-1}(y) = \frac{1}{1+y}$ *and* $q_{a,b}$ *denotes the quantity*[4]*,*

$$q_{a,b} := q(\hat{U}(v_1) = a, \hat{U}(v_4) = b | \hat{U}(d_j) = 1 \; \forall j) \qquad \forall a, b \in \{-1, 1\}. \tag{4.24}$$

Proof. We define $f(x) := (1-x)/x$. Using the identity,

$$q(\hat{U}'(v_1) = a, \hat{U}'(v_4) = b | \hat{U}'(d_1) = 1, \hat{U}'(d_4) = 1) = \frac{q(\hat{U}'(v_1) = a, \hat{U}'(v_4) = b, \hat{U}'(d_1) = 1, \hat{U}'(d_4) = 1)}{q(\hat{U}'(d_1) = 1, \hat{U}'(d_4) = 1)}, \tag{4.25}$$

the system of equations (4.23) is equivalent to,

$$\begin{aligned} q_{1,1} &= (1-\beta_1')(1-\alpha_{1,4}')(1-\beta_4')/z \\ q_{1,-1} &= (1-\beta_1')\alpha_{1,4}'\beta_4'/z \\ q_{-1,1} &= \beta_1'\alpha_{1,4}'(1-\beta_4')/z \\ q_{-1,-1} &= \beta_1'(1-\alpha_{1,4}')\beta_4'/z, \end{aligned}$$

with,

$$\begin{aligned} z :&= q(\hat{U}'(d_1) = 1, \hat{U}'(d_4) = 1) \\ &= (1-\beta_1')(1-\alpha_{1,4}')(1-\beta_4') + (1-\beta_1')\alpha_{1,4}'\beta_4' + \beta_1'\alpha_{1,4}'(1-\beta_4') + \beta_1'(1-\alpha_{1,4}')\beta_4'. \end{aligned}$$

We have,

$$\frac{q_{1,1}}{q_{-1,-1}} = f(\beta_1')f(\beta_4'); \quad \frac{q_{1,-1}}{q_{-1,1}} = \frac{f(\beta_1')}{f(\beta_4')}; \quad \frac{q_{1,1}}{q_{1,-1}} = f(\alpha_{1,4}')f(\beta_4'), \tag{4.26}$$

from which the solution follows from straightforward manipulations. □

In the following theorem we show that by transforming any sub-4-comb of an arbitrary comb $\mathcal{C}$, marginalizations (at any remaining vertices) of the Markov random field defined by (4.20) are unaffected by the transform. We first need a simple lemma:

Lemma 4.4.7. *For any random variables* A, B*, conditionally independent given* X *we have,*

$$P(X = x | A = a, B = b) = \frac{P(A = a | X = x) P(X = x | B = b)}{\sum_{x'} P(A = a | X = x') P(X = x' | B = b)}.$$

The following theorem now shows that the transform T preserves the marginal distributions of the conditional MRFs defined by (4.21).

Theorem 4.4.8. *Given any n-comb* $\mathcal{C}$*, with* $n \geq 4$*, let* $\mathcal{C}'$ *be obtained by applying the basic transform* T *to any sub-4-comb of* $\mathcal{C}$*. Let* q, q' *be the Gibbs measures defined by (4.20) on* $\mathcal{C}$ *and* $\mathcal{C}'$ *respectively. For any spine vertex* v_i *of* $\mathcal{C}$ *(not deleted by the transform) we have that* $q'(\hat{U}'(v_i) = a | \hat{U}'(d_j) = 1 \; \forall j) = q(\hat{U}(v_i) = a | \hat{U}(d_j) = 1 \; \forall j)$.

[4]Note that these quantities are easily calculated: $q_{a_1,a_4} = \frac{\sum_{a_2,a_3 \in \{-1,1\}} q(\hat{U}(v_j) = a_j, \hat{U}(d_j) = 1 \; \forall j \in \{1,\ldots 4\})}{\sum_{a_1,a_2,a_3,a_4 \in \{-1,1\}} q(\hat{U}(v_j) = a_j, \hat{U}(d_j) = 1 \; \forall j \in \{1,\ldots 4\})}$.

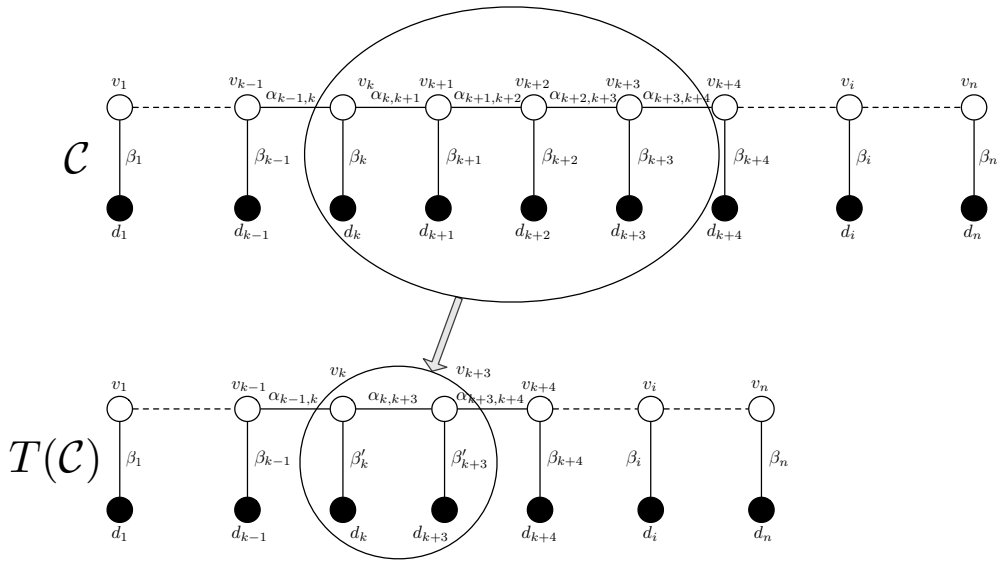


Figure 4.7: Embedded 4-comb to 2-comb transform

Proof. Let the transformed sub-4-comb have vertex set $\{v_k, v_{k+1}, v_{k+2}, v_{k+3}, d_k, d_{k+1}, d_{k+2}, d_{k+3}\}$, and the 2-comb thus have vertex set $\{v_k, v_{k+3}, d_k, d_{k+3}\}$. We can suppose w.l.o.g. that $i \geq k+3$ (the case $i \leq k$ follows by symmetry), so that the construction is as shown in Figure 4.7. Denote for convenience,

$$\mathcal{D}_A := \{\hat{U}(d_j) = 1 \,\forall j < k\} \qquad \mathcal{D}'_A := \{\hat{U}'(d_j) = 1 \,\forall j < k\}$$
$$\mathcal{D}_B := \{\hat{U}(d_k) = 1, ...\hat{U}(d_{k+3}) = 1\} \qquad \mathcal{D}'_B := \{\hat{U}'(d_k) = 1, \hat{U}'(d_{k+3}) = 1\}$$
$$\mathcal{D}_C := \{\hat{U}(d_j) = 1 \,\forall j > k+3\} \qquad \mathcal{D}'_C := \{\hat{U}'(d_j) = 1 \,\forall j > k+3\}.$$

First note the following,

$$\begin{aligned} q(\hat{U}(v_{k+3}) = a|\mathcal{D}_A, \mathcal{D}_B) &= \sum_{b\in\{-1,1\}} q(\hat{U}(v_{k+3}) = a, \hat{U}(v_k) = b|\mathcal{D}_A, \mathcal{D}_B) \\ &= \sum_{b\in\{-1,1\}} q(\hat{U}(v_{k+3}) = a|\mathcal{D}_A, \mathcal{D}_B, \hat{U}(v_k) = b) q(\hat{U}(v_k) = b|\mathcal{D}_A, \mathcal{D}_B) \\ &= \frac{\sum_{b\in\{-1,1\}} q(\hat{U}(v_{k+3}) = a|\mathcal{D}_B, \hat{U}(v_k) = b) q(\mathcal{D}_A|\hat{U}(v_k) = b) q(\hat{U}(v_k) = b|\mathcal{D}_B)}{\sum_{x\in\{-1,1\}} q(\mathcal{D}_A|\hat{U}(v_k) = x) q(\hat{U}(v_k) = x|\mathcal{D}_B)}, \end{aligned} \tag{4.27}$$

where (4.27) follows from the Markov property and Lemma 4.4.7. By an identical argument we have,

$$\begin{aligned} &q'(\hat{U}'(v_{k+3}) = a|\mathcal{D}'_A, \mathcal{D}'_B) \\ &= \frac{\sum_{b\in\{-1,1\}} q'(\hat{U}'(v_{k+3}) = a|\mathcal{D}'_B, \hat{U}'(v_k) = b) q'(\mathcal{D}'_A|\hat{U}'(v_k) = b) q'(\hat{U}'(v_k) = b|\mathcal{D}'_B)}{\sum_{x\in\{-1,1\}} q'(\mathcal{D}'_A|\hat{U}'(v_k) = x) q'(\hat{U}'(v_k) = x|\mathcal{D}'_B)}. \end{aligned} \tag{4.28}$$

That the r.h.s. of equations (4.27) and (4.28) are equal follows from the defining properties of the 4-comb $\to$ 2-comb transform and the definition of the Gibbs measures q and q', thus,

$$q'(\hat{U}'(v_{k+3}) = a|\mathcal{D}'_A, \mathcal{D}'_B) = q(\hat{U}(v_{k+3}) = a|\mathcal{D}_A, \mathcal{D}_B). \tag{4.29}$$

Now we have,

$$
\begin{aligned}
&q(\hat{U}(v_i) = a|\mathcal{D}_A, \mathcal{D}_B, \mathcal{D}_C) \\
&= \sum_{b\in\{-1,1\}} q(\hat{U}(v_i) = a|\hat{U}(v_{k+3}) = b, \mathcal{D}_A, \mathcal{D}_B, \mathcal{D}_C) q(\hat{U}(v_{k+3}) = b|\mathcal{D}_A, \mathcal{D}_B, \mathcal{D}_C) \\
&= \sum_{b\in\{-1,1\}} q(\hat{U}(v_i) = a|\hat{U}(v_{k+3}) = b, \mathcal{D}_C) \frac{q(\hat{U}(v_{k+3}) = b, \mathcal{D}_C|\mathcal{D}_A, \mathcal{D}_B)}{q(\mathcal{D}_C|\mathcal{D}_A, \mathcal{D}_B)} \\
&= \frac{\sum_{b\in\{-1,1\}} q(\hat{U}(v_i) = a|\hat{U}(v_{k+3}) = b, \mathcal{D}_C) q(\hat{U}(v_{k+3}) = b, \mathcal{D}_C|\mathcal{D}_A, \mathcal{D}_B)}{\sum_{x\in\{-1,1\}} q(\hat{U}(v_{k+3}) = x, \mathcal{D}_C|\mathcal{D}_A, \mathcal{D}_B)} \\
&= \frac{\sum_{b\in\{-1,1\}} q(\hat{U}(v_i) = a|\hat{U}(v_{k+3}) = b, \mathcal{D}_C) q(\mathcal{D}_C|\hat{U}(v_{k+3}) = b, \mathcal{D}_A, \mathcal{D}_B) q(\hat{U}(v_{k+3}) = b|\mathcal{D}_A, \mathcal{D}_B)}{\sum_{x\in\{-1,1\}} q(\mathcal{D}_C|\hat{U}(v_{k+3}) = x, \mathcal{D}_A, \mathcal{D}_B) q(\hat{U}(v_{k+3}) = x|\mathcal{D}_A, \mathcal{D}_B)} \\
&= \frac{\sum_{b\in\{-1,1\}} q(\hat{U}(v_i) = a|\hat{U}(v_{k+3}) = b, \mathcal{D}_C) q(\mathcal{D}_C|\hat{U}(v_{k+3}) = b) q(\hat{U}(v_{k+3}) = b|\mathcal{D}_A, \mathcal{D}_B)}{\sum_{x\in\{-1,1\}} q(\mathcal{D}_C|\hat{U}(v_{k+3}) = x) q(\hat{U}(v_{k+3}) = x|\mathcal{D}_A, \mathcal{D}_B)}. \qquad (4.30)
\end{aligned}
$$

By an identical derivation we have,

$$
\begin{aligned}
&q'(\hat{U}'(v_i) = a|\mathcal{D}'_A, \mathcal{D}'_B, \mathcal{D}'_C) \\
&= \frac{\sum_{b\in\{-1,1\}} q'(\hat{U}'(v_i) = a|\hat{U}'(v_{k+3}) = b, \mathcal{D}'_C) q'(\mathcal{D}'_C|\hat{U}'(v_{k+3}) = b) q'(\hat{U}'(v_{k+3}) = b|\mathcal{D}'_A, \mathcal{D}'_B)}{\sum_{x\in\{-1,1\}} q'(\mathcal{D}'_C|\hat{U}'(v_{k+3}) = x) q'(\hat{U}'(v_{k+3}) = x|\mathcal{D}'_A, \mathcal{D}'_B)},
\end{aligned} \qquad (4.31)
$$

and (4.31), and (4.30) are seen to be identical from the defining properties of the 4-comb $\rightarrow$ 2-comb transform and from the identity (4.29). □

Now we describe the efficient implementation of our algorithm. We restrict our description to the case $n = 2^k$ for simplicity. Over the course of the learning process we maintain a stack of combs in which each comb in a higher tier is derived by applying a 4-comb to 2-comb transform T to every 4-comb in the collection of sub-4-combs which comprise the lower tier, as shown in Figure 4.8 (for the case $n = 16$). At each trial t we build a comb $\widehat{\mathcal{C}}$ of smallest possible size containing v_{i_t} (and v_1 and v_n) and comprising only of transformed 2-combs and their necessary connecting edges. In Figure 4.8 such a minimal comb is highlighted for the query vertex v_6. The size of this minimal comb is $\mathcal{O}(\log n)$. Since this comb can be obtained from the initial comb by applying a succession of basic transforms, by Theorem 4.4.8 marginalization of the conditional MRF defined by (4.21) on $\widehat{\mathcal{C}}$ produces predictions equivalent to marginalizing the conditional MRF defined by (4.21) on the original comb $\mathcal{C}$. Thus Algorithm 1 can be implemented by marginalizing $\hat{q}$ defined by (4.21) on $\widehat{\mathcal{C}}$. Thus prediction is achieved by marginalizing a MRF defined on a tree of size $\mathcal{O}(\log n)$ and therefore has complexity $\mathcal{O}(\log n)$ using e.g. Belief Propogation. After each trial t the 'modify' step of Figure 4.5 of updating β_{i_t} is performed which is a constant time operation. The stack of combs must then be "repaired" by recalculating every 4-comb to 2-comb transform which involved β_{i_t} or any quantity derived from it, i.e. every transform appearing above v_{i_t} in the stack must be calculated. There are $\log n$ such transforms, one for each tier, and each calculation is a constant time operation. The overall time complexity of this implementation is therefore $\mathcal{O}(\log n)$ per query plus a one-time $\mathcal{O}(n)$ operation to calculate all n initial transforms.

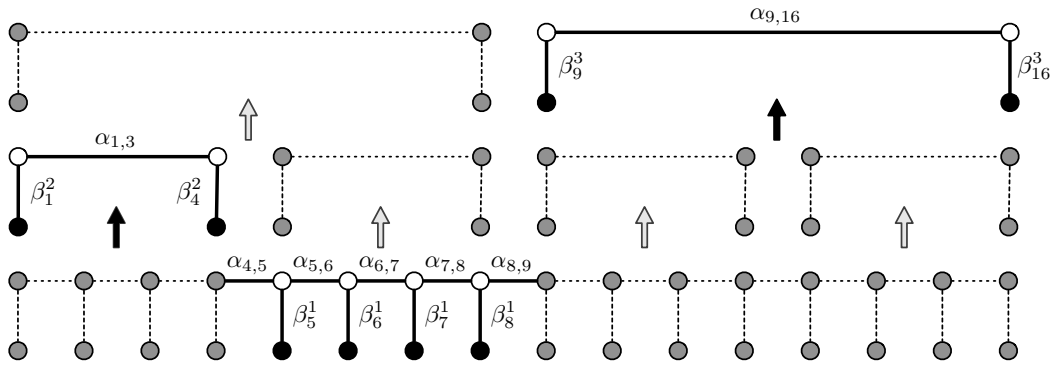


Figure 4.8: The stack of combs

4.5 Prediction with a binary support tree

In this section, we turn our attention to the MSNI algorithm in equation (4.4). We begin by stating the mistake bound for MSNI, which will be proved in the form given here in Corollary 5.4.1 (with $p = 2$), but which was first proved (with slightly worse constants) in Herbster (2008). We must recall the notion of resistance distance, as discussed in Section 3.3.1, Example 2 (and which will be further discussed in Chapter 5), which we denote by $r_{\mathcal{G}} : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ and is given by the formula $r_{\mathcal{G}}(v_i, v_j) = (\boldsymbol{e}_i - \boldsymbol{e}_j)^\top \boldsymbol{L}^+ (\boldsymbol{e}_i - \boldsymbol{e}_j)$.

Theorem 4.5.1. *Given the task of predicting the labelling of an unweighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in the online learning framework, the number of mistakes, M, incurred by MSNI defined by (4.4) on a trial sequence $\mathcal{S}$ satisfies, for any $\rho > 0$, the bound*

$$M \leq N(\mathcal{V}', \rho, r_{\mathcal{G}}) + 4\Phi_{\mathcal{G}}(\boldsymbol{u})\rho, \tag{4.32}$$

where $\boldsymbol{u} \in \mathbb{R}^n$ is any labelling consistent with the trial sequence, $\mathcal{V}' = \{i_1, i_2, \dots\} \subseteq \mathcal{V}$ is the set of trial vertices and $N(\mathcal{V}', \rho, r_{\mathcal{G}})$ is the minimum number of balls of radius at most ρ, required to cover $\mathcal{V}'$ according to the effective resistance distance.

This bound indicates that the predictive MSNI performs well on graphs with pronounced cluster structure. The mistake bound (4.32) can be preferable to (4.18) whenever the inputs are sufficiently clustered and so has a cover of small diameter sets. For example, consider two $(m + 1)$-cliques, one labeled "+1", one "−1" with cm arbitrary interconnecting edges $(c \geq 1)$ here the bound (4.18) is vacuous while (4.32) is $M \leq 8c + 3$ (with $\rho = \frac{2}{m}$, $N(X, \rho, r_{\mathcal{G}}) = 2$, and $\Phi_{\mathcal{G}}(\boldsymbol{u}) = cm$).

An graph $\mathcal{G}$ may have both local cluster structure yet have a large diameter. Imagine a "universe" such that vertices are distributed into many dense clusters such that some sets of clusters are tightly packed but overall the distribution is quite diffuse. A given set of trial vertices $\mathcal{V}' \subseteq \mathcal{V}$ may then be centered on a few clusters or alternatively encompass the entire space. Thus, for practical purposes, we would like a prediction algorithm which achieves the "best of both worlds", that is a mistake bound which is no greater, in order of magnitude, than the minimum of (4.18) and (4.32). This section is

directed towards this goal.

We introduce the notion of binary support tree, detail the use of minimum semi-norm interpolation method in the support tree setting and then prove the desired result.

Definition Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with $|\mathcal{V}| = n$, and spine $\mathcal{P}_{\text{spine}}$, we define a *binary support tree* of $\mathcal{G}$ to be any binary tree $\mathcal{T} = (\mathcal{V}_\mathcal{T}, \mathcal{E}_\mathcal{T})$ of least possible depth, D, whose leaves are the vertices of $\mathcal{P}_{\text{spine}}$, in order. Note that $D < \log_2(n) + 1$.

We show that there is a weighting of the support tree which ensures that the resistance diameter of the support tree is small, but also such that any labelling of the leaf vertices can be extended to the support tree such that its cut size remains small. This enables effective learning via the support tree. A related construction has been used to build preconditioners for solving linear systems (Gremban et al., 1995).

Lemma 4.5.2. *Given any spine graph $\mathcal{P}_{\text{spine}} = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = n$, and labelling $\boldsymbol{u} \in \{-1, 1\}^n$, with support tree $\mathcal{T} = (\mathcal{V}_\mathcal{T}, \mathcal{E}_\mathcal{T})$, there exists a weighting $\boldsymbol{A}$ of $\mathcal{T}$, and a labelling $\bar{\boldsymbol{u}} \in [-1, 1]^{|\mathcal{V}_\mathcal{T}|}$ of $\mathcal{T}$ such that $\bar{\boldsymbol{u}}$ and $\boldsymbol{u}$ are identical on $\mathcal{V}$, $\Phi_\mathcal{T}(\bar{\boldsymbol{u}}) < \Phi_{\mathcal{P}_{\text{spine}}}(\boldsymbol{u})$ and $R_\mathcal{T} \leq (\log_2 n+1)(\log_2 n+4)(\log_2(\log_2 n+2))^2$.*

Proof. Let v_r be the root vertex of $\mathcal{T}$. Suppose each edge $(i, j) \in \mathcal{E}_\mathcal{T}$ has a weight A_{ij}, which is a function of the edge's depth $d = \max\{d_\mathcal{T}(v_i, v_r), d_\mathcal{T}(v_j, v_r)\}$, $A_{ij} = W(d)$ where $d_\mathcal{T}(v, v')$ is the number of edges in the shortest path from v to v'. Consider the unique labelling $\bar{\boldsymbol{u}}$ such that, for $1 \leq i \leq n$ we have $\bar{u}_i = u_i$ and such that for every other vertex $v_p \in \mathcal{V}_\mathcal{T}$, with child vertices v_{c_1}, v_{c_2}, we have $\bar{u}_p = \frac{\bar{u}_{c_1} + \bar{u}_{c_2}}{2}$, or $\bar{u}_p = \bar{u}_c$ in the case where v_p has only one child, v_c. Suppose the edges $(p, c_1), (p, c_2) \in \mathcal{E}_\mathcal{T}$ are at some depth d in $\mathcal{T}$, and let $\mathcal{V}' \subset \mathcal{V}$ correspond to the leaf vertices of $\mathcal{T}$ descended from v_p. Define $\Phi_{\mathcal{P}_{\text{spine}}}(\boldsymbol{u}_{\mathcal{V}'})$ to be the cut of $\boldsymbol{u}$ restricted to vertices in $\mathcal{V}'$. If $\bar{u}_{c_1} = \bar{u}_{c_2}$ then $(\bar{u}_p - \bar{u}_{c_1})^2 + (\bar{u}_p - \bar{u}_{c_2})^2 = 0 \leq 2\Phi_{\mathcal{P}_{\text{spine}}}(\boldsymbol{u}_{\mathcal{V}'})$, and if $\bar{u}_{c_1} \neq \bar{u}_{c_2}$ then $(\bar{u}_p - \bar{u}_{c_1})^2 + (\bar{u}_p - \bar{u}_{c_2})^2 \leq 2 \leq 2\Phi_{\mathcal{P}_{\text{spine}}}(\boldsymbol{u}_{\mathcal{V}'})$. Hence

$$W(d)\left((\bar{u}_p - \bar{u}_{c_1})^2 + (\bar{u}_p - \bar{u}_{c_2})^2\right) \leq 2W(d)\Phi_{\mathcal{P}_{\text{spine}}}(\boldsymbol{u}_{\mathcal{V}'}) \tag{4.33}$$

(a similar inequality is trivial in the case that v_p has only one child). Since the sets of leaf descendants of all vertices at depth d form a partition of $\mathcal{V}$, summing (4.33) first over all parent nodes at a given depth and then over all integers $d \in [1, D]$ gives

$$4\Phi_\mathcal{T}(\bar{\boldsymbol{u}}) \leq 2\sum_{d=1}^{D} W(d)\Phi_{\mathcal{P}_{\text{spine}}}(\boldsymbol{u}). \tag{4.34}$$

We then choose

$$W(d) = \frac{1}{(d+1)(\log_2(d+1))^2} \tag{4.35}$$

and note that $\sum_{d=1}^{\infty} \frac{1}{(d+1)(\log_2(d+1))^2} \leq \frac{1}{2} + \ln^2 2 \int_2^\infty \frac{1}{x \ln^2 x} \mathrm{d}x = \frac{1}{2} + \ln 2 < 2$.

Further, $R_{\mathcal{T}} = 2\sum_{d=1}^{D}(d+1)(\log_2(d+1))^2 \leq D(D+3)(\log_2(D+1))^2$ and so $D \leq \log_2 n + 1$ gives the resistance bound. □

Definition Given the task of predicting the labelling of an unweighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ the *augmented minimum semi-norm interpolation algorithm* proceeds as follows: An augmented graph $\bar{\mathcal{G}} = (\bar{\mathcal{V}}, \bar{\mathcal{E}})$ is formed by attaching a binary support tree of $\mathcal{G}$, with weights defined as in (4.35), to $\mathcal{G}$; formally let $\mathcal{T} = (\mathcal{V}_{\mathcal{T}}, \mathcal{E}_{\mathcal{T}})$ be such a binary support tree of $\mathcal{G}$, then $\bar{\mathcal{G}} = (\mathcal{V}_{\mathcal{T}}, \mathcal{E} \cup \mathcal{E}_{\mathcal{T}})$. The minimum semi-norm interpolation algorithm is then used to predict the (partial) labelling defined on $\bar{\mathcal{G}}$.

Theorem 4.5.3. *Given the task of predicting the labelling of any unweighted, connected, n-vertex graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in the online framework, the number of mistakes, M, incurred by the augmented minimum semi-norm interpolation algorithm satisfies*

$$M \leq \min_{\rho>0}\{N(X, \rho, r_{\mathcal{G}}) + 12\Phi_{\mathcal{G}}(\boldsymbol{u})\rho\} + 1, \tag{4.36}$$

where $N(X, \rho, r_{\mathcal{G}})$ is the covering number of the input set $X = \{v_{i_1}, v_{i_2}, \dots\} \subseteq \mathcal{V}$ relative to the resistance distance $r_{\mathcal{G}}$ of $\mathcal{G}$ and $\boldsymbol{u} \in \mathbb{R}^n$ is any labelling consistent with the trial sequence. Furthermore,

$$M \leq 12\Phi_{\mathcal{G}}(\boldsymbol{u})(\log_2 n + 1)(\log_2 n + 4)(\log_2(\log_2 n + 2))^2 + 2. \tag{4.37}$$

Proof. Let $\boldsymbol{u}$ be some labelling consistent with the trial sequence. By (4.5) we have that $\Phi_{\mathcal{P}_{\text{spine}}}(\boldsymbol{u}) \leq 2\Phi_{\mathcal{G}}(\boldsymbol{u})$ for any spine $\mathcal{P}_{\text{spine}}$ of $\mathcal{G}$. Moreover, by the arguments in Lemma 4.5.2 there exists some labelling $\bar{\boldsymbol{u}}$ of the weighted support tree $\mathcal{T}$ of $\mathcal{G}$, consistent with $\boldsymbol{u}$ on $\mathcal{V}$, such that $\Phi_{\mathcal{T}}(\bar{\boldsymbol{u}}) < \Phi_{\mathcal{P}_{\text{spine}}}(\boldsymbol{u})$. We then have

$$\Phi_{\bar{\mathcal{G}}}(\bar{\boldsymbol{u}}) = \Phi_{\mathcal{T}}(\bar{\boldsymbol{u}}) + \Phi_{\mathcal{G}}(\boldsymbol{u}) < 3\Phi_{\mathcal{G}}(\boldsymbol{u}). \tag{4.38}$$

By Rayleigh's monotonicity law the addition of the support tree does not increase the resistance between any vertices on $\mathcal{G}$, hence

$$N(X, \rho, r_{\bar{\mathcal{G}}}) \leq N(X, \rho, r_{\mathcal{G}}). \tag{4.39}$$

Combining inequalities (4.38) and (4.39) with the minimum semi-norm interpolation bound (4.32) for predicting $\bar{\boldsymbol{u}}$ on $\bar{\mathcal{G}}$, yields

$$M \leq N(X, \rho, r_{\bar{\mathcal{G}}}) + 4\Phi_{\bar{\mathcal{G}}}(\bar{\boldsymbol{u}})\rho + 1 \leq N(X, \rho, r_{\mathcal{G}}) + 12\Phi_{\mathcal{G}}(\boldsymbol{u})\rho + 1.$$

which proves (4.36). We prove (4.37) by covering $\bar{\mathcal{G}}$ with single ball so that $M \leq 4\Phi_{\bar{\mathcal{G}}}(\bar{\boldsymbol{u}})R_{\bar{\mathcal{G}}} + 2 \leq 12\Phi_{\mathcal{G}}(\boldsymbol{u})R_{\mathcal{T}} + 2$ and the result follows from the bound on $R_{\mathcal{T}}$ in Lemma 4.5.2. □

4.6 Conclusion

Existing techniques for predicting the labelling of a graph do not scale well in the size of the graph. We have explored a further theoretical deficiency with existing techniques for predicting the labelling of a

graph online. As a solution, we have presented an approximate cut-preserving embedding of any graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ into a simple path graph, which we call a spine, such that efficient realization of the weighted majority algorithm can be performed. This achieves a mistake bound which is logarithmic in the size of the vertex set for any graph, and the complexity of this algorithm is of $O(|\mathcal{E}| + |\mathcal{V}| \ln |\mathcal{V}|)$. We further applied the insights gained to a second algorithm – an augmentation of the Pounce algorithm, which achieves a polylogarithmic performance guarantee, but can further take advantage of clustered data, in which case its bound is relative to any cover of the graph.

Chapter 5

p-norm algorithms for learning the labelling of a graph

Abstract

We study the problem of predicting the labelling of a graph. The graph is given and a trial sequence of (vertex,label) pairs is then incrementally revealed to the learner. On each trial a vertex is queried and the learner predicts a boolean label. The true label is then returned. The learner's goal is to minimise mistaken predictions. We propose *minimum p-seminorm interpolation* to solve this problem. To this end we give a p-seminorm on the space of graph labellings. Thus on every trial we predict using the labelling which *minimises* the p-seminorm and is also *consistent* with the revealed (vertex, label) pairs. When $p = 2$ this is the *harmonic energy minimisation* procedure of Zhu et al. (2003a), also called (Laplacian) *interpolated regularisation* in Belkin et al. (2004). In the limit as $p \to 1$ this is equivalent to predicting with a label-consistent mincut. We give mistake bounds relative to a label-consistent mincut and a resistive cover of the graph. We say an edge is *cut* with respect to a labelling if the connected vertices have disagreeing labels. We find that minimising the p-seminorm with $p = 1 + \epsilon$ where $\epsilon \to 0$ as the graph diameter $D \to \infty$ gives a bound of $\mathcal{O}(\Phi^2 \log D)$ versus a bound of $\mathcal{O}(\Phi D)$ when $p = 2$ where Φ is the number of cut edges.

5.1 Introduction

As in Chapter 4 we study the online graph labelling problem and recall the definitions of Section 4.2.1.

5.1.1 The p-norm algorithms

In previous work (Herbster and Pontil, 2007; Herbster, 2008) a norm induced by the graph Laplacian was used to predict the labelling of a graph in the online setting with algorithms such as the Perceptron. In Kivinen et al. (1997) it was shown that the perceptron, "online SVM"s and similar algorithms applied to the problem of learning *sparse* linear classifiers in Euclidean space, suffer from the limitation that

there exist example sequences such that these algorithms incur mistakes linearly in the dimension of the examples. These lower bounds should be contrasted to upper bounds for multiplicative algorithms such as Winnow (Littlestone, 1988) and the "quasi additive" p-norm Perceptron (Grove et al., 1997; Gentile, 2003) which are *logarithmic* in the dimension of the examples. An analogous observation for the graph labelling problem (which will be presented in Chapter 4) demonstrated that there exists an n-vertex graph with a single cut edge for which the foundational semi-supervised method of Harmonic Energy Minimization (or "Regularized interpolation") (Zhu et al., 2003a; Belkin et al., 2004) incurs $\Omega(\sqrt{n})$ mistakes.

Inspired by the results for the p-norm perceptron's ability to learn sparse concepts in $\mathbb{R}^n$, with a mistake bound logarithmic in n, we consider a similar idea for building classifiers on graphs. We thus introduce a family of seminorms defined on the labellings of a graph – we term them Laplacian p-seminorms which include the smoothness functional of Belkin et al. (2004); Zhu et al. (2003a) and the label-consistent graph cut (Blum and Chawla, 2001) as limiting cases. We present an online algorithm for learning concepts defined on graphs based upon minimum p-seminorm interpolation. We derive a mistake bound for this algorithm in which the graph cut of a labelling is the measure of the complexity of the learning task. In the graph setting the dual seminorm gives rise to a generalisation of the notion of resistance between graph vertices (Klein and Randić, 1993; Doyle and Snell, 2000), which we term p-resistance and show that it is a natural measure of similarity between graph vertices. We give a brief survey of its fundamental properties by extending a well-known analogy with resistive networks. Cluster structure in the graph w.r.t. the p-resistance distance (captured via covering number of the vertex set) features as the "structural" term in our mistake bound. Expressing the bound in this way helps to demonstrate that our algorithm exploits connectivity and cluster structure in data.

We demonstrate that, in natural cases, the optimal choice for the parameter p (inasmuch as optimizing our bounds) results in an algorithm which lies between the mincut ($p = 1$) of Blum and Chawla (2001) and the method of Harmonic Energy Minimization ($p = 2$) of Belkin et al. (2004); Zhu et al. (2003a). In a further parallel with the behaviour of the p-norm Perceptron we demonstrate that we can choose the parameter p (using only information available a-priori to the learner) to ensure a performance guarantee which is logarithmic with regard to graph diameter. The bound also decreases with the edge connectivity of the graph or clusters thereof as a consequence of the p-resistance term.

5.2 Background and preliminaries

If $\boldsymbol{z} \in \mathbb{R}^n$ then let $\|\boldsymbol{z}\|_p := \sqrt[p]{\sum_{i=1}^n |z_i|^p}$ denote the p-norm when $p \in [1, \infty)$. More generally, if $\boldsymbol{\Psi} : \mathbb{R}^n \to \mathbb{R}^m$ is any linear map we define the associated $(\boldsymbol{\Psi}, p)$-seminorm as,

$$\|\boldsymbol{u}\|_{\boldsymbol{\Psi},p} := \|\boldsymbol{\Psi}\boldsymbol{u}\|_p. \tag{5.1}$$

If $\{\mathbf{0}\} = \{\boldsymbol{u} \in \mathbb{R}^n : \boldsymbol{\Psi}\boldsymbol{u} = \mathbf{0}\}$ then $\|\cdot\|_{\boldsymbol{\Psi},p}$ defines a norm since we have a unique minimal vector. Given a seminorm $\|\cdot\| : \mathbb{R}^n \to \mathbb{R}$ the (unique) *dual* seminorm $\|\cdot\|^* : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is defined on the

vector space of continuous linear functionals $Z : \mathbb{R}^n \to \mathbb{R}$ as,

$$\|Z\|^* := \sup_{\boldsymbol{w}\in\mathbb{R}^n, ||\boldsymbol{w}||\neq 0} \left\{ \frac{|Z(\boldsymbol{w})|}{\|\boldsymbol{w}\|} \right\} = [\inf_{\boldsymbol{w}\in\mathbb{R}^n} \{\|\boldsymbol{w}\| : Z(\boldsymbol{w}) = 1\}]^{-1}. \tag{5.2}$$

We immediately recover the useful "generalized Hölder" inequality,

$$|Z(\boldsymbol{w})| \leq \|Z\|^* \|\boldsymbol{w}\|.$$

Since the dual space of $\mathbb{R}^n$ is isometrically isomorphic to $\mathbb{R}^n$ we can identify each linear functional with an element of $\mathbb{R}^n$ with evaluation corresponding to the dot product between vectors. The canonical basis vectors of $\mathbb{R}^n$ we denote as $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n$ and we are particularly interested in the corresponding linear functionals $E_i(\boldsymbol{w}) := \boldsymbol{e}_i^\top \boldsymbol{w}$.

Given a set $\mathcal{U} \subseteq \mathcal{X}$, a *cover* of $\mathcal{U}$ is a collection $\mathcal{C} = \{\mathcal{C}_i\}_{i=1}^k$ of subsets $\mathcal{C}_i \subseteq \mathcal{X}$ such that $\mathcal{U} \subseteq \cup_{i=1}^k \mathcal{C}_i$. For a given symmetric *discrepancy* function $d : \mathcal{U} \times \mathcal{U} \to \mathbb{R}$ ($d(x, y) = d(y, x)$) and any $\rho > 0$, the *covering number* $N(\mathcal{U}, \rho, d(\cdot, \cdot))$ of $\mathcal{U}$ is the cardinality of the smallest cover $\mathcal{C}$ such that for each $\mathcal{C}_i \in \mathcal{C}$ we have $d(x, x') \leq \rho$ if $x, x' \in \mathcal{C}_i$.

We consider *undirected* graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ so that $\mathcal{E} := \{(i, j)|i \sim j\}$ is the set of unordered pairs of adjacent vertex indexes. Associated with each edge $(i, j) \in \mathcal{E}$ is a weight $A_{ij} > 0$ and $A_{ij} = 0$ if $(i, j) \notin \mathcal{E}$, so that $\boldsymbol{A}$ is the (weighted) symmetric *adjacency matrix*. Typically we consider n-vertex graphs with $\hat{n}$ edges. We say that $\mathcal{G}$ is *unweighted* if $A_{ij} \in \{0, 1\}$.

We say $\mathcal{G}'$ is a *subgraph* of $\mathcal{G}$ whenever $\mathcal{V}'_\mathcal{G} \subseteq \mathcal{V}_\mathcal{G}$ and $\mathcal{E}'_\mathcal{G} \subseteq \mathcal{E}_\mathcal{G}$ and we write $\mathcal{G}' \subseteq \mathcal{G}$. If $\mathcal{V}'_\mathcal{G} \subseteq \mathcal{V}_\mathcal{G}$ then the *induced subgraph* is $(\mathcal{V}'_\mathcal{G}, \mathcal{E}'_\mathcal{G})$ with $\mathcal{E}'_\mathcal{G} := \{(i, j) \in \mathcal{E}_\mathcal{G} : v_i, v_j \in \mathcal{V}'_\mathcal{G}\}$.

A *path graph* $\mathcal{P}$ is a graph of the form $\mathcal{V}_\mathcal{P} = \{v_0, v_1 ... v_n\}$, $\mathcal{E}_\mathcal{P} = \{(0, 1), (1, 2) ... (n - 1, n)\}$ and we define the length, $\ell(\mathcal{P})$, of any path $\mathcal{P}$ by $\ell(\mathcal{P}) := \sum_{(i,j)\in\mathcal{E}_\mathcal{P}} \frac{1}{A_{ij}}$. The *distance* between any two vertices $v_i, v_j \in \mathcal{V}_\mathcal{G}$ is the length of the shortest path containing v_i and v_j,

$$\delta(i, j) := \min_{\{\mathcal{P}\subseteq\mathcal{G}: v_i, v_j \in \mathcal{V}_\mathcal{P}\}} \ell(\mathcal{P}),$$

and is equal to ∞ if no path exists. We define the *diameter* of $\mathcal{G}$, $D(\mathcal{G}) := \max_{i,j} \delta(i, j)$. In this chapter, we generally consider *connected* graphs (that is, graphs in which a path connects any two vertices).

We denote $\mathbb{N}_n := \{1, 2, , ..., n\}$.

5.2.1 Laplacian (Ψ, p)-seminorms on functions over a graph

A *labelling* $\boldsymbol{u} \in \mathbb{R}^n$ of an n-vertex graph $\mathcal{G}$ is viewed as a function $\boldsymbol{u} : \mathcal{V}_\mathcal{G} \to \mathbb{R}$ defined on the vertices of $\mathcal{G}$ whereby u_i corresponds to the label of v_i. If $\mathcal{G} = (\mathcal{V}, \mathcal{E} = \{(i_1, j_1), \ldots, (i_{\hat{n}}, j_{\hat{n}})\})$ is a graph then an associated *edge map* [1] $\boldsymbol{\Psi}_\mathcal{G} : \mathbb{R}^n \to \mathbb{R}^{\hat{n}}$ (with p implicit) is a linear map such that,

$$\boldsymbol{\Psi}_\mathcal{G}\boldsymbol{u} = (A_{i_1 j_1}^{\frac{1}{p}}(u_{i_1} - u_{j_1}), \ldots, A_{i_{\hat{n}} j_{\hat{n}}}^{\frac{1}{p}}(u_{i_{\hat{n}}} - u_{j_{\hat{n}}}))^\top. \tag{5.3}$$

[1] Corresponding to a weighted, oriented *incidence matrix*.

When $p = 2$, the $n \times n$ matrix $\boldsymbol{L} = \boldsymbol{\Psi}_{\mathcal{G}}^{\top}\boldsymbol{\Psi}_{\mathcal{G}}$ is the graph *Laplacian*. We introduce a class of *Laplacian* $(\boldsymbol{\Psi}, p)$*-seminorms* defined on the space of graph labellings: if $\boldsymbol{u} \in \mathbb{R}^n$ then,

$$\|\boldsymbol{u}\|_{\mathcal{G},p} := \|\boldsymbol{u}\|_{\boldsymbol{\Psi}_{\mathcal{G}},p} = \left(\sum_{(i,j)\in\mathcal{E}_{\mathcal{G}}} A_{ij}|u_i - u_j|^p\right)^{\frac{1}{p}}. \tag{5.4}$$

These seminorms generalise the "smoothness functional" $\boldsymbol{u}^T\boldsymbol{L}\boldsymbol{u}$ (Belkin et al., 2004; Zhu et al., 2003a) which corresponds to the case $p = 2$, and as such measure the complexity of graph labellings. In fact one feature map associated to the natural kernel $K : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ defined by the pseudoinverse of the graph Laplacian $K(v_i, v_j) := \boldsymbol{L}^{+}_{ij}$ (see, e.g. Smola and Kondor, 2003) is the edge map

$$\begin{aligned}(\Psi_{\mathcal{G}}^{\top})^{+} :\ & \mathcal{V} \to \mathbb{R}^{\widehat{n}} \\ :\ & v_i \to (\Psi_{\mathcal{G}}^{\top})^{+}\boldsymbol{e}_i,\end{aligned}$$

where the inner product in the feature space $\mathbb{R}^{\widehat{n}}$ is then the Euclidean inner product. A binary labelling $\boldsymbol{u} \in \mathbb{R}^n$ has a small graph cut precisely when the image $(\Psi_{\mathcal{G}}^{\top})^{+}\boldsymbol{u}$ has a small 1-norm (and so is "sparse") in this particular feature space. These p-smoothness functionals have also been considered by Bühler and Hein (2009) in the context of spectral clustering and Singaraju et al. (2009) in the context of computer vision. Closely related are other notions of discrete p-Dirichlet forms (e.g. Zhou and Schölkopf, 2005) which are a discrete counterpart to the continuous p-Laplacian which has been studied to a much greater extent (Heinonen et al., 1993).

When the labelling is restricted to $\boldsymbol{u} \in \{-1, 1\}^n$ we say that edge (i, j) is *cut* if $u_i \neq u_j$ and we define the *weighted cut size* of $\boldsymbol{u}$ as,

$$\Phi_{\mathcal{G}}(\boldsymbol{u}) := \frac{1}{2^p}\|\boldsymbol{u}\|^p_{\mathcal{G},p} = \frac{1}{2^p}\sum_{(i,j)\in\mathcal{E}} A_{ij}|u_i - u_j|^p. \tag{5.5}$$

The cut-size is independent of p and if the graph is unweighted it is just the number of cut edges.

We will use the dual norm $\|\cdot\|^*_{\mathcal{G},p}$ to give a discrepancy $r_{\mathcal{G},p}(\cdot,\cdot)$ called *effective p-resistance* between vertices by identifying vertices v_i and v_j with the functionals E_i and E_j so that,

$$r_{\mathcal{G},p}(i, j) = \left(\|E_i - E_j\|^*_{\mathcal{G},p}\right)^p. \tag{5.6}$$

When $p = 2$ there is an established natural connection (Doyle and Snell, 2000) between graphs and resistive networks where each edge $(i, j) \in \mathcal{E}_{\mathcal{G}}$ is viewed as a resistor with resistance $\frac{1}{A_{ij}}$. The *effective resistance* $r_{\mathcal{G}}(i, j) = r_{\mathcal{G},2}(i, j)$ is the potential difference needed to induce a unit current flow between v_i and v_j.The effective resistance may be computed with the formula (Klein and Randić, 1993) (and see the derivation in Section 3.3 Example 2),

$$r_{\mathcal{G}}(i, j) = (\boldsymbol{e}_i - \boldsymbol{e}_j)^T\boldsymbol{L}^{+}(\boldsymbol{e}_i - \boldsymbol{e}_j),$$

where $\boldsymbol{L}^{+}$ denotes the pseudoinverse of $\boldsymbol{L}$.

The *p-resistance* (*diameter*) of a graph $\mathcal{G}$ is defined $R_p(\mathcal{G}) := \max_{\{v_i,v_j\in\mathcal{V}_{\mathcal{G}}\}} r_{\mathcal{G},p}(i, j)$ ($R(\mathcal{G}) = R_2(\mathcal{G})$). In this chapter the notion of (*effective) p-resistance* will be a key to our bounds and is further developed in Section 5.4.1.

5.2.2 Previous work

The problem of learning a labeling of a graph is a natural problem in the online learning setting, as well as a foundational technique for a variety of semi-supervised learning methods (Blum and Chawla, 2001; Kondor and Lafferty, 2002; Zhu et al., 2003a; Belkin et al., 2004).

The problem of predicting the labelling of a graph in the online framework was first considered in Herbster et al. (2005) and a mistake bound for the kernel perceptron was given in (Herbster and Pontil, 2007, Theorem 4.2 (with $b = R(\mathcal{G}); c = 0$)) of ,

$$|\mathcal{M}| \leq 8\Phi_{\mathcal{G}}(\boldsymbol{u})R(\mathcal{G}) + 2,$$

where $\boldsymbol{u}$ is any labelling consistent with the trial sequence.

In Herbster (2008) the Pounce on-line prediction technique was developed to exploit any cluster structure in a graph. The algorithm achieves the mistake bound,

$$|\mathcal{M}| \leq N(\mathcal{V}', \rho, r_{\mathcal{G}}) + 4\Phi_{\mathcal{G}}(\boldsymbol{u})\rho + 1,$$

for any $\rho > 0$. Here, $\boldsymbol{u} \in \mathbb{R}^n$ is any labelling consistent with the trial sequence, $\mathcal{V}' = \{v_{i_1}, v_{i_2}, \ldots v_{i_m}\} \subseteq \mathcal{V}$ is the set of inputs and the covering number $N(\mathcal{V}', \rho, \sqrt{r_{\mathcal{G}}})$ is the minimum number of vertex sets of resistance diameter no greater than ρ^2 required to cover $\mathcal{V}'$ (see Section 5.2). The Pounce algorithm therefore captures the notion of cluster structure through a graph cover of low resistance vertex sets. For a definition of Pounce see the projection algorithms defined in Section 5.5.1, where Pounce corresponds to the choice $p = 2$.

In Chapter 4 a limitation of existing methods for predicting the labelling of a graph online was identified. In particular for an online version of the foundational Harmonic Energy Minimization method of minimising the smoothness functional given by (5.4), when $p = 2$, an n-vertex graph construction for which the algorithms incur $\Omega(\sqrt{\Phi_{\mathcal{G}}(\boldsymbol{u})n})$ mistakes was constructed. It is further demonstrated that any unweighted graph can be embedded into a path graph in such a way that an efficient Bayes optimal classifier used to predict the labelling of the embedding (and, therefore, of the underlying graph) obtains a mistake bound which grows only logarithmically in the size of the graph,

$$|\mathcal{M}| \leq 2\Phi_{\mathcal{G}}(\boldsymbol{u}) \max\left[0, \log_2\left(\frac{n-1}{2\Phi_{\mathcal{G}}(\boldsymbol{u})}\right)\right] + \frac{2\Phi_{\mathcal{G}}(\boldsymbol{u})}{\ln 2} + 1. \tag{5.7}$$

This algorithm, however, involves the corruption of the graph structure resulting in a drawback: the method does not exploit graph connectivity – in fact the mistake bound (5.7) improves if the graph is replaced by any spanning tree – and is therefore not demonstrably suitable for the case of dense or clustered data. A further algorithm to utilise an embedding of $\mathcal{G}$ into a simpler structure was presented in Cesa-Bianchi et al. (2009b) and here the reduction is to a tree $\mathcal{T}$. A mistake bound of,

$$|\mathcal{M}| \leq \mathcal{O}(\Phi_{\mathcal{T}}(\boldsymbol{u}) \log D(\mathcal{C})),$$

is derived, where here $\Phi_{\mathcal{T}}(\boldsymbol{u})$ is the cut size of the true labelling $\boldsymbol{u}$ on $\mathcal{T}$ and $D(\mathcal{C})$ is the maximum diameter of any cluster (unitarily labelled) of vertices which $\mathcal{T}$ is partitioned into by $\boldsymbol{u}$.

A goal of research in this area is to present an algorithm which fully exploits cluster structure and connectivity in graphs and obtains a logarithmic performance guarantee. In this chapter we present an algorithm with a mistake bound in terms of a revealing resistance feature and demonstrate that this is upper bounded by a logarithmic function of the graph diameter. The algorithm therefore exploits cluster structure and connectivity but is also suitable in the case in which a graph exhibits a sparse structure or large diameter.

5.3 Minimum $(\mathbf{\Psi}, p)$-seminorm interpolation

Given the problem of predicting a labelling of a set of objects, a natural approach is to specify a norm on the labelling of those objects and to choose a labelling which is then both consistent and minimal in norm; this approach is known as *minimum norm interpolation*. Recalling Section 5.2, in this chapter we investigate interpolation with $(\mathbf{\Psi}, p)$-*seminorms*, $\|\cdot\|_{\mathbf{\Psi},p}$, which are specified by choosing a linear map $\mathbf{\Psi} : \mathbb{R}^n \to \mathbb{R}^{\hat{n}}$ and a $p \in (1, 2]$. In the case when $p = 2$ and when $\mathbf{\Psi}$ has a rank of n this is equivalent to using the Euclidean norm induced by the kernel matrix $K = (\mathbf{\Psi}^\top \mathbf{\Psi})^{-1}$. The intention is that $\mathbf{\Psi}$ is chosen so that the $(\mathbf{\Psi}, p)$-seminorm captures our assumptions about the complexity of the true labelling or acts as a regularizer suitable for the problem in question, and in our application it will capture the geometry of a graph; our assumption will be that the labelling is smooth over a graph. First we present the algorithm in a general abstract form and specialize later to the case of predicting the labelling of a graph.

Given a $(\mathbf{\Psi}, p)$-seminorm and a sequence of online trials $t \in \{1, 2, 3, ...\}$ in which (index,label) pairs (i_t, y_t) are revealed, our algorithm (see Figure 5.1) maintains a weight vector $\boldsymbol{w}_t \in \mathbb{R}^n$ such that $\operatorname{sgn}(\boldsymbol{e}_{i_t}^\top \boldsymbol{w}_t)$ is the hypothesised label for indexed object i_t at trial t. On trial t, the weight vector is updated by choosing that vector consistent with all previous examples[2] which attains the least $(\mathbf{\Psi}, p)$-seminorm, if there are multiple minimisers an arbitrary vector is chosen[3].

We bound the mistakes of our interpolation algorithm in the following theorem.

Theorem 5.3.1. *The number of mistakes, $|\mathcal{M}|$, incurred by minimum $(\mathbf{\Psi}, p)$-seminorm interpolation, for any $\rho > 0$, is bounded by,*

$$|\mathcal{M}| \leq N(\mathcal{X}', \rho, d_{\mathbf{\Psi},p}) + \frac{\rho^2 \|\boldsymbol{u}\|_{\mathbf{\Psi},p}^2}{p-1}, \tag{5.8}$$

where $\boldsymbol{u} \in \mathbb{R}^n$ is any labelling such that $u_{i_t} = y_t \; \forall t \leq m$, and $N(\mathcal{X}', \rho, d_{\mathbf{\Psi},p})$ is the covering number of the input set $\mathcal{X}' = \{i_1, i_2, ..., i_m\}$ relative to the distance,

$$d_{\mathbf{\Psi},p}(i, j) := \|E_i - E_j\|_{\mathbf{\Psi},p}^*. \tag{5.9}$$

The bound above is for the abstract general case of $(\mathbf{\Psi}, p)$-seminorm interpolation for an arbitrary linear map $\mathbf{\Psi}$. In the following we will study the case corresponding to prediction of the labelling of

[2]The conservative version of the algorithm where a vector is chosen consistent with only the "mistaken" examples obtains the same bound as Theorem 5.3.1.

[3]If $\mathbf{\Psi}$ is the edge map of a connected graph this will never occur.

Parameters: A linear map $\mathbf{\Psi} : \mathbb{R}^n \to \mathbb{R}^{\hat{n}}$ and $p \in (1, 2]$

Initialization: $\boldsymbol{w}_1 = \mathbf{0}$; $\mathcal{M} = \{\}$

Input: $\{(i_t, y_t)\}_{t=1}^m \in (\mathbb{N}_n \times \{-1, 1\})^m$

for $t = 1, \ldots, m$ **do**

Receive: $i_t \in \{1, \ldots, n\}$

Predict: $\hat{y}_t = \text{sign}(\boldsymbol{e}_{i_t}^\top \boldsymbol{w}_t)$

Receive: y_t

if $\hat{y}_t \neq y_t$ **then** $\mathcal{M} = \mathcal{M} \cup \{t\}$

$\boldsymbol{w}_{t+1} \in \text{argmin}_{\boldsymbol{u} \in \mathbb{R}^n} \{\|\boldsymbol{u}\|_{\mathbf{\Psi},p} : u_{i_1} = y_1, \ldots, u_{i_t} = y_t\}$

end

Figure 5.1: Minimum $(\mathbf{\Psi}, p)$-seminorm interpolation

a graph where $\|\boldsymbol{u}\|_{\mathbf{\Psi},p}^2$ will correspond to a function of the cut size (see (5.5)) of the labelling $\boldsymbol{u}$ and $d_{\mathbf{\Psi},p}(i, j)$ will be identified with a measure closely related to resistance in an electrical network. We first provide a proof of Theorem 5.3.1.

5.3.1 Mistake bound analysis (proof of Theorem 5.3.1)

In this section we recall Bregman divergence and develop some properties relevant to our application. Then we show that the minimum $(\mathbf{\Psi}, p)$-seminorm interpolation algorithm is equivalent to successive projections with regard to a Bregman divergence and we complete our proof.

Properties of Bregman projections

Bregman (1967) introduced the *Bregman divergence* for convex programming.

Definition Let $F : \mathbb{R}^n \to \mathbb{R}$ be a $\mathscr{C}^2$ convex function. Denote by $D_F(\boldsymbol{u}, \boldsymbol{w})$ the Bregman divergence w.r.t. F;

$$D_F(\boldsymbol{u}, \boldsymbol{w}) = F(\boldsymbol{u}) - F(\boldsymbol{w}) - (\boldsymbol{u} - \boldsymbol{w})^\top \boldsymbol{\nabla} F(\boldsymbol{w}). \tag{5.10}$$

The Bregman divergence is generally defined in terms of a *strictly* convex potential function F where "strictness" ensures the uniqueness of a projection. In our application we will use the nonstrictly convex potential $F(\boldsymbol{v}) = \|\boldsymbol{v}\|_{\mathbf{\Psi},p}^2$ and thus projection (see (5.11)) will not necessarily be unique. The Bregman divergence is nonnegative as the convexity of F guarantees that the first order approximation $F(\boldsymbol{u}) \approx F(\boldsymbol{w}) + (\boldsymbol{u} - \boldsymbol{w})^\top \boldsymbol{\nabla} F(\boldsymbol{w})$ is not an overestimate. We will use the following notation $D_p := D_{\|\cdot\|_p^2}$ and $D_{\mathbf{\Psi},p} := D_{\|\cdot\|_{\mathbf{\Psi},p}^2}$.

We define the projection of $\boldsymbol{w}$ onto a non-empty set $\mathcal{U} \subseteq \mathbb{R}^n$ with respect to D_F as,

$$\text{proj}_F(\mathcal{U}; \boldsymbol{w}) := \operatorname*{argmin}_{\boldsymbol{u} \in \mathcal{U}} D_F(\boldsymbol{u}, \boldsymbol{w}). \tag{5.11}$$

We note that the argmin is not necessarily unique.

Lemma 5.3.2. *If $\mathcal{U} \subseteq \mathbb{R}^n$ is a nonempty affine set and $\boldsymbol{w} \in \mathbb{R}^n$, then $\mathrm{proj}_{\mathbf{\Psi},p}(\mathcal{U}; \boldsymbol{w})$ is non-empty.*

Proof. We recall that a *direction of recession* of a convex function is any direction in which the function is non-increasing (Rockafellar, 1972, p. 69). We observe that any direction of recession $\boldsymbol{x}$ of $D_{\mathbf{\Psi},p}(\cdot, \boldsymbol{w})$ is exactly one such that $\mathbf{\Psi}\boldsymbol{x} = 0$ and in these directions $D_{\mathbf{\Psi},p}(\cdot, \boldsymbol{w})$ is constant. It then follows that $\mathrm{proj}_{\mathbf{\Psi},p}(\mathcal{U}; \boldsymbol{w})$ is non-empty by (Rockafellar, 1972, Theorem 27.3) which in particular guarantees that a continuous convex function on $\mathbb{R}^n$ attains its minima on a given affine constraint set if the function is constant in every common direction of recession between the function and the constraint set. □

The following is the well-known Pythagorean equality for Bregman divergences.

Lemma 5.3.3. *If $\boldsymbol{w}' \in \mathbb{R}^n$ is a projection of $\boldsymbol{w} \in \boldsymbol{R}^n$ to the affine set $\mathcal{U} \subseteq \mathbb{R}^n$ with regard to the Bregman divergence D_F, then $\forall \boldsymbol{u} \in \mathcal{U}$ we have,*

$$D_F(\boldsymbol{u}, \boldsymbol{w}) = D_F(\boldsymbol{w}', \boldsymbol{w}) + D_F(\boldsymbol{u}, \boldsymbol{w}'). \tag{5.12}$$

Proof. Let $\mathcal{U} = \cap_{i=1}^k \{\boldsymbol{u} : \boldsymbol{u}^\top \boldsymbol{x}_i = y_i\}$. By expanding D_F in (5.12) we obtain the equivalent form,

$$\left(\boldsymbol{\nabla} F(\boldsymbol{w}) - \boldsymbol{\nabla} F(\boldsymbol{w}')\right)^\top (\boldsymbol{u} - \boldsymbol{w}') = 0. \tag{5.13}$$

Recalling the method of Lagrange multipliers to compute $\boldsymbol{w}'$, we note that the unconstrained minimum of the Lagrangian,

$$L(\boldsymbol{\lambda}, \boldsymbol{v}) = D_F(\boldsymbol{v}, \boldsymbol{w}) + \sum_{i=1}^k \lambda_i (\boldsymbol{x}_i^\top \boldsymbol{v} - y_i),$$

occurs at $\boldsymbol{v} = \boldsymbol{w}'$. Thus,

$$\begin{aligned} \mathbf{0} &= \boldsymbol{\nabla}_{\boldsymbol{v}} L(\boldsymbol{\lambda}, \boldsymbol{v}) \mid_{\boldsymbol{v}=\boldsymbol{w}'} \\ &= \boldsymbol{\nabla} F(\boldsymbol{w}') - \boldsymbol{\nabla} F(\boldsymbol{w}) + \sum_{i=1}^k \lambda_i \boldsymbol{x}_i. \end{aligned}$$

Thus, since $\boldsymbol{u}, \boldsymbol{w}' \in \mathcal{U}$,

$$\begin{aligned} \left(\boldsymbol{\nabla} F(\boldsymbol{w}) - \boldsymbol{\nabla} F(\boldsymbol{w}')\right)^\top (\boldsymbol{u} - \boldsymbol{w}') &= \left(\sum_{i=1}^k \lambda_i \boldsymbol{x}_i\right)^\top (\boldsymbol{u} - \boldsymbol{w}') \\ &= 0, \end{aligned}$$

as required. □

We build on the following lemma, which requires the linearity of $\mathbf{\Psi}$, to prove the important Lemma 5.3.5.

Lemma 5.3.4. *Given a linear map $\mathbf{\Psi}$ then,*

$$D_{\mathbf{\Psi},p}(\boldsymbol{u}, \boldsymbol{w}) = D_p(\mathbf{\Psi}\boldsymbol{u}, \mathbf{\Psi}\boldsymbol{w}).$$

Proof. As $\|\boldsymbol{z}\|_{\boldsymbol{\Psi},p} = \|\boldsymbol{\Psi}\boldsymbol{z}\|_p$ we have, by applying the chain rule,

$$\begin{aligned} D_{\boldsymbol{\Psi},p}(\boldsymbol{u},\boldsymbol{w}) &= \|\boldsymbol{u}\|^2_{\boldsymbol{\Psi},p} - \|\boldsymbol{w}\|^2_{\boldsymbol{\Psi},p} - (\boldsymbol{u}-\boldsymbol{w})^\top \boldsymbol{\nabla}_{\boldsymbol{z}} \, \|\boldsymbol{z}\|^2_{\boldsymbol{\Psi},p}\Big|_{\boldsymbol{z}=\boldsymbol{w}} \\ &= \|\boldsymbol{\Psi}\boldsymbol{u}\|^2_p - \|\boldsymbol{\Psi}\boldsymbol{w}\|^2_p - (\boldsymbol{u}-\boldsymbol{w})^\top \boldsymbol{\nabla}_{\boldsymbol{z}} \, \|\boldsymbol{\Psi}\boldsymbol{z}\|^2_p\Big|_{\boldsymbol{z}=\boldsymbol{w}} \\ &= \|\boldsymbol{\Psi}\boldsymbol{u}\|^2_p - \|\boldsymbol{\Psi}\boldsymbol{w}\|^2_p - \boldsymbol{\Psi}(\boldsymbol{u}-\boldsymbol{w})^\top \boldsymbol{\nabla}_{\boldsymbol{z}'} \, \|\boldsymbol{z}'\|^2_p\Big|_{\boldsymbol{z}'=\boldsymbol{\Psi}\boldsymbol{w}} \qquad (5.14) \\ &= D_p(\boldsymbol{\Psi}\boldsymbol{u}, \boldsymbol{\Psi}\boldsymbol{w}), \end{aligned}$$

where (5.14) follows from the chain rule. □

The following lemma is inspired directly by arguments upper bounding the quadratic remainder term in the Taylor's series expansion of the squared p-norm in Grove et al. (1997). We will need only the first inequality.

Lemma 5.3.5.

$$(p-1)\|\boldsymbol{w}' - \boldsymbol{w}\|^2_{\boldsymbol{\Psi},p} \leq D_{\boldsymbol{\Psi},p}(\boldsymbol{w}', \boldsymbol{w}) \quad p \in (1,2] \qquad (5.15)$$

$$D_{\boldsymbol{\Psi},p}(\boldsymbol{w}', \boldsymbol{w}) \leq (p-1)\|\boldsymbol{w}' - \boldsymbol{w}\|^2_{\boldsymbol{\Psi},p} \quad p \in [2,\infty) \qquad (5.16)$$

Proof. We first recall the Hölder inequality (for functions on discrete spaces). If $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n$ and $\frac{1}{r} + \frac{1}{s} = 1$, then,

$$\sum_{i=1}^n |a_i b_i| \leq \|\boldsymbol{a}\|_r \|\boldsymbol{b}\|_s \qquad r \in (1,\infty). \qquad (5.17)$$

Now, if $\boldsymbol{\xi} = \boldsymbol{w}' - \boldsymbol{w}$ then, for $p \geq 2$ by Taylor's theorem there is some point $\boldsymbol{\zeta} \in \mathbb{R}^n$ such that,

$$||\boldsymbol{w}'||^2_p - ||\boldsymbol{w}||^2_p - \nabla \|\boldsymbol{z}\|^2_p \,|_{\boldsymbol{z}=\boldsymbol{w}} \cdot \boldsymbol{\xi} = \frac{1}{2}\sum_{ij} \frac{\partial^2 \|\boldsymbol{z}\|^2_p}{\partial z_i \partial z_j}\Big|_{\boldsymbol{z}=\boldsymbol{\zeta}} \xi_i \xi_j$$

$$D_p(\boldsymbol{w}', \boldsymbol{w}) = \frac{1}{2}\sum_{ij} \frac{\partial^2 (\|\boldsymbol{z}\|^2_p)}{\partial z_i \partial z_j}\Big|_{\boldsymbol{z}=\boldsymbol{\zeta}} \xi_i \xi_j$$

We have,

$$\frac{\partial (||\boldsymbol{z}||^2_p)}{\partial z_i} = 2||\boldsymbol{z}||^{2-p}_p z_i^{p-1} \mathrm{sgn}(z_i)\,,$$

and for $i \neq j$,

$$\begin{aligned} \frac{\partial^2 (||\boldsymbol{z}||^2_p)}{\partial z_i \partial z_j} &= \frac{\partial}{\partial z_j}\left(2||\boldsymbol{z}||^{2-p}_p z_i^{p-1}\mathrm{sgn}(z_i)\right) \\ &= 2(2-p)||\boldsymbol{z}||^{2-2p}_p (z_i z_j)^{p-1}\mathrm{sgn}(z_i z_j), \end{aligned}$$

and,

$$\frac{\partial^2 (||\boldsymbol{z}||^2_p)}{\partial z_i^2} = 2(2-p)\|\boldsymbol{z}\|^{2-2p}_p |z_i|^{2p-2} + 2(p-1)\|\boldsymbol{z}\|^{2-p}_p |z_i|^{p-2},$$

which exist for all $\boldsymbol{z} \neq \mathbf{0}$ for $p \geq 2$. Thus,

$$\begin{aligned} D_p(\boldsymbol{w}', \boldsymbol{w}) &= (2-p)||\boldsymbol{\zeta}||_p^{2-2p} \sum_{i,j=1}^{n} \xi_i \xi_j (\zeta_i \zeta_j)^{p-1} \operatorname{sgn}(z_i z_j) \\ &\quad + (p-1)\|\boldsymbol{\zeta}\|_p^{2-p} \sum_{i=1}^{n} \xi_i^2 |\zeta_i|^{p-2} \\ &= (2-p)||\boldsymbol{\zeta}||_p^{2-2p} \left[\sum_{i=1}^{n} \xi_i \zeta_i^{p-1} \right]^2 \\ &\quad + (p-1)\|\boldsymbol{\zeta}\|_p^{2-p} \sum_{i=1}^{n} \xi_i^2 |\zeta_i|^{p-2}. \end{aligned}$$

For $p \geq 2$ the first term here is not positive while the second term is bounded above with equation (5.17) with $r = \frac{p}{2}, s = \frac{p}{p-2}$ giving,

$$D_p(\boldsymbol{w}', \boldsymbol{w}) \leq (p-1)\|\boldsymbol{\xi}\|_p^2 \qquad p \geq 2. \tag{5.18}$$

With reference to Appendix E, this is equivalent to the $(p-1)$-strong smoothness of the function $\frac{1}{2}||\cdot||_p^2$ with respect to the norm $||\cdot||_p$. This function has Fenchel conjugate $\frac{1}{2}||\cdot||_q^2$, where $\frac{1}{p} + \frac{1}{q} = 1$, and by the duality of strong convexity and strong smoothness, Theorem E.0.12, we therefore have that $\frac{1}{2}||\cdot||_q^2$ is $(q-1)$-strongly convex w.r.t. $||\cdot||_q$, and so,

$$D_p(\boldsymbol{w}', \boldsymbol{w}) \geq (p-1)||\boldsymbol{\xi}||_p^2 \qquad 1 < p \leq 2. \tag{5.19}$$

Finally, since $\|\boldsymbol{z}\|_{\boldsymbol{\Psi},p} = \|\boldsymbol{\Psi}\boldsymbol{z}\|_p$ an application of Lemma 5.3.4 to (5.18) and (5.19) gives the result. □

Successive Bregman projection and interpolation

We prove that minimum $(\boldsymbol{\Psi}, p)$-seminorm interpolation is equivalent to the sequential composition of Bregman projections in Corollary 5.3.7. First we show that Bregman projections to affine sets compose using the following well-known lemma.

Lemma 5.3.6. *If $\mathcal{U}_1$ and $\mathcal{U}_2$ are affine sets and $\mathcal{U}_2 \subseteq \mathcal{U}_1$ then*

$$\operatorname{proj}_{\boldsymbol{\Psi},p}(\mathcal{U}_2; \boldsymbol{w}_0) = \operatorname{proj}_{\boldsymbol{\Psi},p}(\mathcal{U}_2; \operatorname{proj}_{\boldsymbol{\Psi},p}(\mathcal{U}_1; \boldsymbol{w}_0)) \tag{5.20}$$

Proof. Let $\boldsymbol{w}_1 = \operatorname{proj}_{\boldsymbol{\Psi},p}(\mathcal{U}_1; \boldsymbol{w}_0)$ and $\boldsymbol{w}_2 = \operatorname{proj}_{\boldsymbol{\Psi},p}(\mathcal{U}_2; \boldsymbol{w}_1)$. We have the following string of inequalities which hold for every $\boldsymbol{u} \in \mathcal{U}_2$,

$$D(\boldsymbol{w}_1, \boldsymbol{w}_0) = D(\boldsymbol{u}, \boldsymbol{w}_0) - D(\boldsymbol{u}, \boldsymbol{w}_1), \tag{5.21}$$

$$D(\boldsymbol{w}_2, \boldsymbol{w}_1) = D(\boldsymbol{u}, \boldsymbol{w}_1) - D(\boldsymbol{u}, \boldsymbol{w}_2), \tag{5.22}$$

$$D(\boldsymbol{w}_1, \boldsymbol{w}_0) + D(\boldsymbol{w}_2, \boldsymbol{w}_1) = D(\boldsymbol{u}, \boldsymbol{w}_0) - D(\boldsymbol{u}, \boldsymbol{w}_2), \tag{5.23}$$

$$D(\boldsymbol{w}_2, \boldsymbol{w}_0) = D(\boldsymbol{u}, \boldsymbol{w}_0) - D(\boldsymbol{u}, \boldsymbol{w}_2), \tag{5.24}$$

where equations (5.21) and (5.22) follow from the Pythagorean theorem (Lemma 5.3.3) equation (5.24) then follows from setting $\boldsymbol{u} = \boldsymbol{w}_2$ in (5.21) then substituting into (5.23). Equation (5.24) implies $\boldsymbol{w}_2$ is the projection of $\boldsymbol{w}_0$ onto $\mathcal{U}_2$. □

We now observe that the minimum p-seminorm interpolation (Figure 5.1) is equivalent to a proxy method of successive projections which minimise the Bregman divergence $D_{\boldsymbol{\Psi},p}(\boldsymbol{w}_{t+1}, \boldsymbol{w}_t)$. Since $\boldsymbol{\nabla}||\boldsymbol{u}||^2_{\boldsymbol{\Psi},p}|_{\boldsymbol{u}=\mathbf{0}} = \mathbf{0}$, we have the following corollary.

Corollary 5.3.7. *If $\boldsymbol{w}_1 := \mathbf{0}$ and we recursively define,*

$$\boldsymbol{w}_{t+1} := \operatorname*{argmin}_{\boldsymbol{u}\in\mathbb{R}^n}\{D_{\boldsymbol{\Psi},p}(\boldsymbol{u}, \boldsymbol{w}_t) : u_{i_s} = y_s \; \forall s \leq t\}.$$

then,

$$\boldsymbol{w}_{m+1} = \operatorname*{argmin}_{\boldsymbol{u}\in\mathbb{R}^n}\{\|\boldsymbol{u}\|_{\boldsymbol{\Psi},p} : u_{i_s} = y_{i_s} \; \forall s \leq m\}$$

Proof of Theorem 5.3.1

Proof. In Corollary 5.3.7 we noted that the minimum $(\boldsymbol{\Psi}, p)$-seminorm interpolation algorithm is identical to a successive Bregman projection algorithm. We prove a bound for the latter. Let $\boldsymbol{u} \in \mathbb{R}^n$ be such that $u_{i_t} = y_t$ for all trials $t \leq m$. From (5.12) we have,

$$\sum_{t=1}^{m} D_{\boldsymbol{\Psi},p}(\boldsymbol{w}_{t+1}, \boldsymbol{w}_t) = D_{\boldsymbol{\Psi},p}(\boldsymbol{u}, \boldsymbol{w}_1) - D_{\boldsymbol{\Psi},p}(\boldsymbol{u}, \boldsymbol{w}_{m+1}). \tag{5.25}$$

Using Lemma 5.3.5 we lower bound $D_p(\boldsymbol{w}_{t+1}, \boldsymbol{w}_t)$,

$$(p-1)\|\boldsymbol{w}_{t+1} - \boldsymbol{w}_t\|^2_{\boldsymbol{\Psi},p} \leq D_{\boldsymbol{\Psi},p}(\boldsymbol{w}_{t+1}, \boldsymbol{w}_t). \tag{5.26}$$

Note that there is a mistake, by convention, on the first trial since $\boldsymbol{w}_1 = \mathbf{0}$. Now, for each mistaken trial $t \in \mathcal{M}$ with $t \geq 2$, recalling Section 5.2 we define the linear functional $Z_t = E_{i_t} - E_{\eta_{i_t}}$, where,

$$\eta_{i_t} = \operatorname*{argmin}_{i_s}\{\|E_{i_t} - E_{i_s}\|^*_{\boldsymbol{\Psi},p} : s \in \mathcal{M}, s < t\},$$

so that,

$$\begin{aligned}
1 &\leq |Z_t(\boldsymbol{w}_{t+1}) - Z_t(\boldsymbol{w}_t)| && t \geq 2 \\
&= |Z_t(\boldsymbol{w}_{t+1} - \boldsymbol{w}_t)| && t \geq 2 \\
&\leq \|Z_t\|^*_{\boldsymbol{\Psi},p}\|\boldsymbol{w}_{t+1} - \boldsymbol{w}_t\|_{\boldsymbol{\Psi},p} && t \geq 2 \\
&\leq \|Z_t\|^{*^2}_{\boldsymbol{\Psi},p}\|\boldsymbol{w}_{t+1} - \boldsymbol{w}_t\|^2_{\boldsymbol{\Psi},p} && t \geq 2,
\end{aligned} \tag{5.27}$$

thus on a mistaken trial $t \geq 2$ combining (5.26) and (5.27) gives,

$$\frac{p-1}{\|Z_t\|^{*^2}_{\boldsymbol{\Psi},p}} \leq D_p(\boldsymbol{w}_{t+1}, \boldsymbol{w}_t) \qquad t \geq 2. \tag{5.28}$$

We follow a technique introduced in Herbster (2008). Recalling Section 5.2, consider any cover $\mathcal{C} = \cup_k \mathcal{C}_k$ which covers $\mathcal{X}' = \{i_1, i_2, \ldots, i_m\}$ with regard to the distance,

$$d_{\boldsymbol{\Psi},p}(i, j) := \|E_i - E_j\|^*_{\boldsymbol{\Psi},p},$$

with $N(\mathcal{X}', \rho, d_{\boldsymbol{\Psi},p})$ covering sets of diameter no greater than ρ. Let $\mathcal{F}$ be the set of trials in which a mistake first occurred on a cover set, $\mathcal{F} = \cup_k\{\min\{t : i_t \in \mathcal{C}_k\}\}$. Setting $\boldsymbol{w}_1 = \mathbf{0}$ we deduce from

(5.25) and (5.28),

$$\begin{aligned}\sum_{t\in\mathcal{M}\setminus\mathcal{F}}\frac{1}{\|Z_t\|^{*2}_{\boldsymbol{\Psi},p}} &\leq \sum_{t\in\mathcal{M}\setminus\{1\}}\frac{1}{\|Z_t\|^{*2}_{\boldsymbol{\Psi},p}}\\ &\leq \frac{1}{p-1}\sum_{t\in\mathcal{M}\setminus\{1\}} D_{\boldsymbol{\Psi},p}(\boldsymbol{w}_{t+1},\boldsymbol{w}_t)\\ &\leq \frac{1}{p-1}\sum_{t=1}^{m} D_{\boldsymbol{\Psi},p}(\boldsymbol{w}_{t+1},\boldsymbol{w}_t)\\ &\leq \frac{\|\boldsymbol{u}\|^2_{\boldsymbol{\Psi},p}}{p-1}.\end{aligned}$$

Recall that,

$$\|Z_t\|^*_{\boldsymbol{\Psi},p} = d_{\boldsymbol{\Psi},p}(i_t,\eta_{i_t}).$$

Hence for any $t\in\mathcal{M}\setminus\mathcal{F}$ we have $\|Z_t\|^*_{\boldsymbol{\Psi},p}\leq\rho$. Hence as $|\mathcal{F}|\leq N(\mathcal{X}',\rho,d_{\boldsymbol{\Psi},p})$

$$\begin{aligned}\sum_{t\in\mathcal{M}-\mathcal{F}} 1 &\leq \frac{\rho^2\|\boldsymbol{u}\|^2_{\boldsymbol{\Psi},p}}{p-1}\\ |\mathcal{M}| &\leq N(\mathcal{X}',\rho,d_{\boldsymbol{\Psi},p}) + \frac{\rho^2\|\boldsymbol{u}\|^2_{\boldsymbol{\Psi},p}}{p-1}.\end{aligned}$$

□

5.4 Interpolation on a graph

We proceed to our intended application of predicting the labelling of a given graph $\mathcal{G}$ by choosing $\boldsymbol{\Psi}$ to be an edge map $\boldsymbol{\Psi}_{\mathcal{G}}$ of $\mathcal{G}$ (recall (5.3)), so that $\|\boldsymbol{u}\|_{\boldsymbol{\Psi}_{\mathcal{G}},p}$ measures smoothness of functions over the vertices $\mathcal{V}$. If we denote the adjacency of $\mathcal{G}$ by $\boldsymbol{A}$, $(\boldsymbol{\Psi}_{\mathcal{G}},p)$-seminorm interpolation on $\mathcal{G}$ is therefore the process of choosing the labelling $\boldsymbol{u}$ of $\mathcal{G}$ which minimises the seminorm (recalling (5.4)),

$$\|\boldsymbol{u}\|_{\boldsymbol{\Psi}_{\mathcal{G}},p} = \left(\sum_{(i,j)\in\mathcal{E}_{\mathcal{G}}} A_{ij}|u_i-u_j|^p\right)^{\frac{1}{p}},$$

subject to the constraints imposed by the revealed vertex labels. The dual norm term (5.9) of our mistake bound for $(\boldsymbol{\Psi},p)$-seminorm interpolation now corresponds to the following generalization of effective resistance.

Definition Given a graph $\mathcal{G}$, we define the (effective) p-resistance between any two vertices $v_a,v_b\in\mathcal{V}_{\mathcal{G}}$ as,

$$r_{\mathcal{G},p}(a,b) := (||E_a-E_b||^*_{\mathcal{G},p})^p. \tag{5.29}$$

Thus when $p=2$ this is the usual effective resistance and as $p\to 1$ then $r_{\mathcal{G},p}(s,t)\to\frac{1}{\text{“st-mincut”}}$. We will see that for $1<p\leq 2$ effective p-resistance provides a natural measure of similarity between vertices on a graph.

Rewriting Theorem 5.3.1 with the substitution (5.29) we now have the following corollary, which is the main result of this chapter.

Corollary 5.4.1. *After m trials we have, for any $\rho > 0$,*

$$|\mathcal{M}| \leq N(\mathcal{V}', \rho, r_{\mathcal{G},p}) + \frac{\rho^{\frac{2}{p}} \|\boldsymbol{u}\|^2_{\mathcal{G},p}}{p-1}, \tag{5.30}$$

where $\boldsymbol{u} \in \mathbb{R}^n$ is any labelling of $\mathcal{G}$ such that $u_{i_t} = y_t \; \forall t \leq m$, $p \in (1, 2]$, and $N(\mathcal{V}', \rho, r_{\mathcal{G},p})$ is the covering number of the input set $\mathcal{V}' = \{v_{i_1}, v_{i_2}, ...v_{i_m}\}$ relative to the p-resistance $r_{\mathcal{G},p}$.

Proof. This is a special case of Theorem 5.3.1 with $\boldsymbol{\Psi} = \boldsymbol{\Psi}_{\mathcal{G}}$. We must simply recall the identity relating dual norm and graph p-resistance, Definition 5.4. □

We have, then, a bound which relates the number of mistakes to cluster structure in data via the notion of resistive cover of a graph. Note that the bound is valid for all choices of ρ, and so always valid for the optimal cover, which the learner is never required to calculate.

We proceed to develop an interpretation of the bound (5.30) to culminate in Corollary 5.4.10. The norm of the classifier $\|\boldsymbol{u}\|^2_{\mathcal{G},p}$ is relatively simple to interpret while the properties of the p-resistance measure $r_{\mathcal{G},p}$ are less immediate. We therefore next establish an instructive theory of the p-resistance which will both clarify the bound above and provide guidance on the tuning of the parameter p. We will see that the resistance defined by (5.29) is a natural measure of (dis)similarity between vertices on a graph and that this construction admits a surprisingly rich theory which extends a common analogy between graphs and electrical networks.

5.4.1 Theory of p-resistive networks

We now build on a popular connection between the graph labelling problem and the problem of identifying the potential at the nodes of an electric network derived from the graph (Zhu et al., 2003a; Doyle and Snell, 2000). We describe the notion of a network as parallel to a partially labelled graph, in which each edge is a resistive conduit along which electric charge flows between vertices. The label u_i of a vertex v_i is equivalent to it's electric *potential* (or voltage). A partial labelling constrains the potential on the corresponding subset of vertices in the network, through which current then flows along edges according to the laws of the electric network theory. The foundation of our theory here differs from standard theory in a single respect – energy is produced in resistors according to a purely hypothetical formulation of power. This results in changes to other familiar key concepts, such as Ohm's law.

A *p-resistive network* $\mathcal{N} = (\mathcal{G}, \mathcal{S}, p)$ consists of an n-vertex weighted connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with adjacency $\boldsymbol{A}$, a set $\mathcal{S} = \{(v_{i_1}, y_1), \ldots, (v_{i_m}, y_m)\} \in (\mathcal{V}_{\mathcal{G}} \times \mathbb{R})^m$ of $0 \leq m \leq n$ feasible potential constraints and a constant $p \in (1, 2]$. The potential constraints can be viewed as (the effect of) voltage sources applied to the relevant vertices. Denote by $\mathcal{V}_{\mathcal{S}}$ the set of constrained vertices. The resistance of an edge, $\pi_{ij} := \frac{1}{A_{ij}} \in (0, \infty)$, measures the resistance of (i, j) to current flow and is constant. Given a network $\mathcal{N}$ a *state* is an assignment of potentials $\boldsymbol{u} \in \mathbb{R}^n$ to $\mathcal{V}_{\mathcal{G}}$. In the following we will additionally define for any network, a *power* $P(\mathcal{N}, \cdot) : \mathbb{R}^n \to [0, \infty)$, a *current* $\boldsymbol{I}(\mathcal{N}) : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ satisfying

$I_{ij} = -I_{ji}$, and $I_{ij} = 0$ whenever $A_{ij} = 0$, and when $\mathcal{G}$ is clear from the context we will abbreviate the effective p-resistance $r_{\mathcal{G},p}$ to r_p.

Central to electric network theory is the notion of a *flow*.

Definition A flow is any map $\boldsymbol{J} : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ satisfying $J_{ij} = -J_{ji}$, $J_{ij} = 0$ whenever $A_{ij} = 0$ and *Kirchoff's junction law* $\sum_{j:j\sim i} J_{ij} = 0$ for v_i not constrained.

Denote the total flow leaving any vertex v_i by $J_i = \sum_{j:j\sim i} J_{ij}$. If $J_i > 0$ we say that v_i is a source and write $v_i \in \mathcal{V}_{\text{source}}$. If $J_i < 0$ we say that v_i is a sink and write $v_i \in \mathcal{V}_{\text{sink}}$. A k-flow is such that $\sum_{i\,:\,v_i\in\mathcal{V}_{\text{source}}} J_i = k$. A 1-flow is a unit flow.

Fundamental properties

To draw a parallel with our graph labelling problem we define the *power* of potential state $\boldsymbol{u}$ as,

$$P(\boldsymbol{u}) := \sum_{(i,j)\in\mathcal{E}} \frac{|u_i - u_j|^p}{\pi_{ij}}. \tag{5.31}$$

and the corresponding power of any edge (i, j) as,

$$P_{ij}(\boldsymbol{u}) := \frac{|u_i - u_j|^p}{\pi_{ij}}.$$

The standard electric network theory corresponds to the choice $p = 2$, and all other choices result in hypothetical theories. Determining the labelling with minimal p-seminorm (5.4) subject to certain label constraints is equivalent to determining the potential state which minimises (5.31) under the same corresponding potential constraints. Given a network $\mathcal{N} = (\mathcal{G}, \mathcal{S}, p)$, if the potential constraints $\mathcal{S} = \{(v_{i_1}, y_1), \ldots, (v_{i_m}, y_m)\} \neq \emptyset$ then let $\boldsymbol{w}(\mathcal{N})$ denote the unique minimiser

$$\boldsymbol{w}(\mathcal{N}) = \operatorname*{argmin}_{\boldsymbol{u}\in\mathbb{R}^n} \{P(\boldsymbol{u}) : u_{i_1} = y_1, \ldots, u_{i_m} = y_m\}.$$

A p-resistive network operates according to the principle of minimising (5.31) and so a set of potential constraints $\mathcal{S}$ induces the minimal potential state $\boldsymbol{w}(\mathcal{N})$ on the network. The *power of a network* $\mathcal{N}$ is therefore defined as the power of the minimal feasible state,

$$P(\mathcal{N}) := \min_{\boldsymbol{u}\in\mathbb{R}^n} \{P(\boldsymbol{u}) : u_{i_1} = y_1, \ldots, u_{i_m} = y_m\}.$$

At the minimum we have,

$$\begin{aligned} \frac{\partial P(\boldsymbol{u})}{\partial u_i}\Big|_{\boldsymbol{u}=\boldsymbol{w}} &= 0 && v_i \notin \mathcal{V}_{\mathcal{S}} \\ \sum_{j:j\sim i} \frac{|w_i - w_j|^{p-1}\operatorname{sgn}(w_i - w_j)}{\pi_{ij}} &= 0 && v_i \notin \mathcal{V}_{\mathcal{S}}. \end{aligned} \tag{5.32}$$

We define the *current* from vertex v_i to v_j of a network,

$$I_{ij}(\mathcal{N}) := \frac{|w_i - w_j|^{p-1}\operatorname{sgn}(w_i - w_j)}{\pi_{ij}}, \tag{5.33}$$

(if $p = 2$ this is *Ohm's law*) and the *net current* from vertex v_i as,

$$I_i := \sum_{j:j\sim i} I_{ij}\,.$$

Since $\pi_{ij} \geq 0$ we see that current flows from vertices with high potential to those with low potential. We see that (5.32) is *Kirchoff's current law* for $\boldsymbol{I}$,

$$0 = I_i \quad v_i \notin \mathcal{V}_{\mathcal{S}}, \tag{5.34}$$

so that current is a flow and we can alternatively express power of a potential state $\boldsymbol{u}$ via *Joule's law*,

$$P_{ij}(\boldsymbol{u}) = (u_i - u_j)I_{ij}. \tag{5.35}$$

Lemma 5.4.2. *Given a network $\mathcal{N} = (\mathcal{G}, \mathcal{S}, p)$ with potential constraints $\mathcal{S} = \{(v_a, y_a), (v_b, y_b)\}$, then,*

$$P(\mathcal{N}) = (w_a - w_b)I_a, \tag{5.36}$$

where $\boldsymbol{w}$ and $\boldsymbol{I}$ are the minimal potential state and the current induced by $\mathcal{S}$.

Proof. The power of a network is sum of the power along the edges $P(\mathcal{N}) = \sum_{(i,j)\in\mathcal{E}} P_{ij}(\boldsymbol{w})$ and thus by Joule's law (5.35) we have,

$$\begin{aligned}
P(\mathcal{N}) &= \sum_{(i,j)\in\mathcal{E}} (w_i - w_j)I_{ij} \\
&= \sum_i \sum_{j:j<i} w_i I_{ij} - \sum_j \sum_{i:i>j} w_j I_{ij} \\
&= \sum_i \sum_{j:j<i} w_i I_{ij} + \sum_i \sum_{j:j>i} w_i I_{ij} \\
&= \sum_j w_a I_{aj} + \sum_j w_b I_{bj} + \sum_{i:i\neq a,b} \sum_j w_i I_{ij},
\end{aligned}$$

and the result follows since $I_a = \sum_j I_{aj} = -\sum_j I_{bj}$ and $\sum_j I_{ij} = 0\ \forall i \neq a, b$. □

We now demonstrate that the construction (5.29) can indeed naturally be interpreted as a resistance feature in our electric network analogy, via an identity similar to Ohm's Law relating potential, current and effective p-resistance.

Lemma 5.4.3. *Given a network $\mathcal{N} = (\mathcal{G}, \mathcal{S}, p)$ with potential constraints $\mathcal{S} = \{(v_a, y_a), (v_b, y_b)\}$ and $y_a \neq y_b$, then,*

$$P(\mathcal{N}) = \frac{|w_a - w_b|^p}{r_p(a, b)}, \tag{5.37}$$

and,

$$r_p(a, b) = \frac{|w_a - w_b|^{p-1}\mathrm{sgn}(w_a - w_b)}{I_a}, \tag{5.38}$$

where $\boldsymbol{w}$ and $\boldsymbol{I}$ are the minimal potential state and the current induced by $\mathcal{S}$.

Proof. We have, by the definition of power:

$$\begin{aligned} P(\mathcal{N}) &= \min_{\boldsymbol{u}\in\mathbb{R}^n} \left\{ \|\boldsymbol{u}\|_{\mathcal{G},p}^p : u_a = y_a, u_b = y_b \right\} \\ &= |y_a - y_b|^p \min_{\boldsymbol{u}\in\mathbb{R}^n} \left\{ \|\boldsymbol{u}\|_{\mathcal{G},p}^p : u_a - u_b = 1 \right\}. \end{aligned}$$

Substituting (5.2) into (5.29) gives,

$$r_p(a,b) = \left(\min_{\boldsymbol{u}\in\mathbb{R}^n} \left\{ \|\boldsymbol{u}\|_{\mathcal{G},p}^p : u_a - u_b = 1 \right\} \right)^{-1},$$

now equation (5.37) follows since $w_a = y_a$ and $w_b = y_b$. Finally if we apply Lemma 5.4.2 by substituting $P(\mathcal{N}) = (w_a - w_b)I_a$ into (5.37) we obtain (5.38). □

The following final observations will not be required in the sequel but are included to complete the theory.

Definition The power associated with any flow $\boldsymbol{J}$ is defined,

$$P(\boldsymbol{J}) = \sum_{(i,j)\in\mathcal{E}} |J_{ij}|^{\frac{p}{p-1}} \pi_{ij}^{\frac{1}{p-1}}, \tag{5.39}$$

and note that this equals the power of a potential assignment when the flow has a corresponding assignment - that is when the flow is the current induced by some set of potential constraints. Note that not every flow has a consistent assignment of potentials.

We prove that the unit electrical current $\boldsymbol{I}$ between any pair of vertices v_a to v_b is the unit flow of minimal power.

Lemma 5.4.4. *(Thompson's Principle) The unit flow of minimal power between any pair of vertices v_a to v_b is the unit electrical current.*

Proof. We wish to minimise

$$P(\boldsymbol{J}) = \sum_{(i,j)\in\mathcal{E}} |J_{ij}|^{\frac{p}{p-1}} \pi_{ij}^{\frac{1}{p-1}}$$

such that

$$\begin{aligned} \sum_{j\sim i} J_{ij} &= 0 && \forall i \neq a,b \\ \sum_{j\sim a} J_{aj} &= 1 = -\sum_{j\sim b} J_{bj} \\ J_{ij} &= -J_{ji} && \forall i,j \qquad (5.40) \\ J_{ii} &= 0 && \forall i \qquad (5.41) \end{aligned}$$

We form the Lagrangian,

$$L(\boldsymbol{\lambda}, \boldsymbol{J}) = \sum_{(i,j)\in\mathcal{E}} |J_{ij}|^{\frac{p}{p-1}} \pi_{ij}^{\frac{1}{p-1}} + \sum_i \lambda_i \left(\sum_{j\sim i} J_{ij} - \mathbb{I}_{\{i=a\}} + \mathbb{I}_{\{i=b\}} \right),$$

where $\mathbb{I}$ denotes the indicator function. Using (5.40) and (5.41) we can consider only the upper triangular part of $\boldsymbol{J}$,

$$L(\boldsymbol{\lambda}, \boldsymbol{J}) = 2 \sum_{(i,j)\in\mathcal{E}:i>j} |J_{ij}|^{\frac{p}{p-1}} \pi_{ij}^{\frac{1}{p-1}} + \sum_i \lambda_i \left(\sum_{j\sim i: j>i} J_{ij} - \mathbb{I}_{i=a} \right) - \sum_j \lambda_j \left(\sum_{i\sim j: i<j} J_{ij} - \mathbb{I}_{j=b} \right).$$

Differentiating w.r.t. J_{st} for $s < t$ we see that $\nabla L = 0$ occurs when $J_{ij} = \frac{|u_i - u_j|^{p-1}\mathrm{sgn}(u_i - u_j)}{\pi_{ij}}$, $\lambda_i = -\frac{p}{p-1} u_i$ if such a consistent set of $\{u_i\}_{i=1}^n$ exists. We know that such an assignment does exist - any potential which induces unit current from v_a to v_b. Since $P(\boldsymbol{J})$ is strictly convex and the constraints are linear this is the unique minimizer. □

Bounding the p-resistance

The holy grail of this section would be a closed form for the dual norm $\|\boldsymbol{v}\|_{\boldsymbol{\Psi},p}^* = \sup_{\boldsymbol{u}\in\mathbb{R}^n, \|\boldsymbol{u}\|_{\boldsymbol{\Psi},p}\neq 0} \frac{|\boldsymbol{v}^\top \boldsymbol{u}|}{\|\boldsymbol{u}\|_{\boldsymbol{\Psi},p}}$ of a $(\boldsymbol{\Psi}, p)$-seminorm $\|\cdot\|_{\boldsymbol{\Psi},p}$ as defined by (5.1) for an arbitrary linear map $\boldsymbol{\Psi}$ on $\mathbb{R}^n$. This would provide a closed form for the p-resistance as defined by (5.6). This problem remains open (as does establishing the existence of a closed form) but we offer the following upper bound.

Claim 5.4.5. *Given a connected graph $\mathcal{G}$ with edge map $\boldsymbol{\Psi}$, the dual norm $\|\boldsymbol{v}\|_{\boldsymbol{\Psi},p}^* = \sup_{\boldsymbol{u}\in\mathbb{R}^n, \|\boldsymbol{u}\|_{\boldsymbol{\Psi},p}\neq 0} \frac{|\boldsymbol{v}^\top \boldsymbol{u}|}{\|\boldsymbol{u}\|_{\boldsymbol{\Psi},p}}$ of (5.4) has the following form over $\mathbb{R}^n$*

$$\|\boldsymbol{v}\|_{\boldsymbol{\Psi},p}^* \leq ||\boldsymbol{v}||_{(\boldsymbol{\Psi}^+)^\top,q} \qquad\qquad \boldsymbol{v}\perp\mathbf{1},$$

where $\frac{1}{p} + \frac{1}{q} = 1$, and is equal to $+\infty$ when $\boldsymbol{v} \not\perp \mathbf{1}$.

Proof. For $\boldsymbol{v}\perp\mathbf{1}$ we have,

$$\begin{aligned}
\|\boldsymbol{v}\|_{\boldsymbol{\Psi},p}^* &= \sup_{\boldsymbol{u}\in\mathbb{R}^n, \|\boldsymbol{u}\|_{\boldsymbol{\Psi},p}\neq 0} \frac{\boldsymbol{v}^\top \boldsymbol{u}}{\|\boldsymbol{u}\|_{\boldsymbol{\Psi},p}} \\
&= \sup_{\boldsymbol{u}\in\mathbb{R}^n, \|\boldsymbol{\Psi}\boldsymbol{u}\|_p\neq 0} \frac{\boldsymbol{v}^\top \boldsymbol{\Psi}^+ \boldsymbol{\Psi}\boldsymbol{u}}{\|\boldsymbol{\Psi}\boldsymbol{u}\|_p} \\
&= \sup_{\boldsymbol{\Psi}\boldsymbol{u}\in\boldsymbol{\Psi}(\mathbb{R}^n), \|\boldsymbol{\Psi}\boldsymbol{u}\|_p\neq 0} \frac{((\boldsymbol{\Psi}^+)^\top \boldsymbol{v})^\top (\boldsymbol{\Psi}\boldsymbol{u})}{\|\boldsymbol{\Psi}\boldsymbol{u}\|_p} \\
&= \sup_{\boldsymbol{w}\in\mathrm{col}(\boldsymbol{\Psi}), ||\boldsymbol{w}||_p\neq 0} \frac{((\boldsymbol{\Psi}^+)^\top \boldsymbol{v})^\top \boldsymbol{w}}{\|\boldsymbol{w}\|_p} \\
&\leq ||(\boldsymbol{\Psi}^+)^\top \boldsymbol{v}||_q \qquad (5.42) \\
&= ||\boldsymbol{v}||_{(\boldsymbol{\Psi}^+)^\top,q},
\end{aligned}$$

where we applied Hölder's inequality in (5.42) and $\mathrm{col}(\boldsymbol{\Psi})$ denotes the column space of $\boldsymbol{\Psi}$. □

Inequality in (5.42) is generally strict because the vector $\boldsymbol{w}$ required to attain equality in Hölder's inequality will in general not be in the column space $\mathrm{col}(\boldsymbol{\Psi})$ of $\boldsymbol{\Psi}$: a simple counter example which demonstrates that this is typically not the case is provided by the (3,1)-lollipop graph. For $p = 2$ and for trees equality occurs in (5.42). For graphs with $\hat{n}$ edges and n vertices this upper bound will therefore presumably be large when $\hat{n} \gg n$ (since $\mathrm{col}(\boldsymbol{\Psi})$ is in that case a much smaller (rank $n-1$) subspace of

$\mathbb{R}^{\hat{n}}$), and note that finding a precise form for the dual norm amounts to finding the dual of the p-norm when confined to a linear subspace of $\mathbb{R}^{\hat{n}}$.

Black box principles in electric circuit theory are useful tools that allow the simplification of complex networks. In the p-resistive framework we give analogues of the classic "series" (Lemma 5.4.6) and "parallel" laws (Lemma 5.4.7). The fact that we can compose sequential applications of these laws is guaranteed by the seemingly intuitive Thevenin-type theorem (Theorem 5.4.8).

Lemma 5.4.6. *(Resistors in series) Consider a path graph $\mathcal{P}$, with $\mathcal{V}_\mathcal{P} = \{v_1, v_2 ... v_n\}, \mathcal{E}_\mathcal{P} = \{(1,2),(2,3)...(n-1,n)\}$ and edge resistance π_{ij} for each $i \sim j$. Then,*

$$r_p(1,n) = \left(\sum_{i=1}^{n-1} \pi_{i,i+1}^{\frac{1}{p-1}} \right)^{p-1}.$$

Proof. Given a network $\mathcal{N} = (\mathcal{P}, \mathcal{S}, p)$ with potential constraints $\mathcal{S} = \{(v_1, y_1), (v_n, y_n)\}$ let $\boldsymbol{w}$ and $\boldsymbol{I}$ denote the minimal potential state and current induced on $\mathcal{N}$. We have, from Lemma 5.4.3 and (5.33),

$$w_1 - w_n = \sum_{i=1}^{n-1} w_i - w_{i+1}$$
$$|I_1|^{\frac{1}{p-1}} r_p(1,n)^{\frac{1}{p-1}} = \sum_{i=1}^{n-1} |I_{i,i+1}|^{\frac{1}{p-1}} \pi_{i,i+1}^{\frac{1}{p-1}},$$

and the result follows since, by (5.34), we have that $I_1 = I_{i,i+1}$ for $i < n$. □

Lemma 5.4.7. *(Resistors in parallel) Consider a multigraph $\mathcal{G}$ with two vertices $\mathcal{V}_\mathcal{G} = \{v_a, v_b\}$ joined by m resistive edges with resistances $\{\pi_k\}_{k=1}^m$. Then,*

$$r_p(a,b) = \left(\sum_{k=1}^{m} \frac{1}{\pi_k} \right)^{-1}.$$

Proof. Given a network $\mathcal{N} = (\mathcal{G}, \mathcal{S}, p)$ with potential constraints $\mathcal{S} = \{(v_a, y_a), (v_b, y_b)\}$ let $\boldsymbol{w}$ denote the minimal potential state on $\mathcal{N}$. Then by (5.37) we have the following identity for the power $P(\mathcal{N})$,

$$\frac{|w_a - w_b|^p}{r_p(a,b)} = \sum_{k=1}^{m} \frac{|w_a - w_b|^p}{\pi_k},$$

and the result follows immediately. □

We define the notion of a resistive *unit* $\mathcal{U} = (\mathcal{V}_\mathcal{U}, \mathcal{E}_\mathcal{U})$ as any combination of resistors and vertices with two *terminal* vertices $\mathcal{V}_\mathcal{U}^T = \{v_a, v_b\} \subseteq \mathcal{V}_\mathcal{U}$. We refer to the non-terminal vertices $\mathcal{V}_\mathcal{U}^I = \mathcal{V}_\mathcal{U} \backslash \mathcal{V}_\mathcal{U}^T$ as the *interior* vertices. Any unit $\mathcal{U}$ can be treated as a component in a larger graph $\mathcal{G} = (\mathcal{V}_\mathcal{G}, \mathcal{E}_\mathcal{G})$, such that $\mathcal{U} \subseteq \mathcal{G}$ and whenever $v \in \mathcal{V}_\mathcal{U}, v' \in \mathcal{V}_\mathcal{G} \backslash \mathcal{V}_\mathcal{U}$ and $v \sim v'$ then $v \in \mathcal{V}_\mathcal{U}^T$.

Theorem 5.4.8. *(Thevenin) Any resistive unit $\mathcal{U}$ with two terminals v_a and v_b and with effective p-resistance $r_{\mathcal{U},p}(a,b)$ is electrically identical to a single edge with p-resistance $\pi_{ab} = r_{\mathcal{U},p}(a,b)$. In particular, in any given network in which $\mathcal{U}$ is a component and $\mathcal{V}_\mathcal{U}^I$ is unconstrained we can "black box" $\mathcal{U}$, and replace it with a single edge of p-resistance $r_{\mathcal{U},p}(a,b)$ without affecting current or potential in the external network.*

Proof. Consider a network $\mathcal{N} = (\mathcal{G}, \mathcal{S}, p)$ in which $\mathcal{U}$ is a component of an n-vertex graph $\mathcal{G} = (\mathcal{V}_\mathcal{G}, \mathcal{E}_\mathcal{G})$ with adjacency $\boldsymbol{A}$. Suppose that the non-empty potential constraints $\mathcal{S}$ are defined on a subset of vertices $\mathcal{V}_\mathcal{S} \subseteq \mathcal{V}_\mathcal{G} \backslash \mathcal{V}_\mathcal{U}^I$ not in the interior of $\mathcal{U}$. Denote by $\boldsymbol{w}$ and $\boldsymbol{I}$ the minimal feasible potential state and current, and by $P(\mathcal{N})$ the induced power. Define the power produced across $\mathcal{U}$ by potential state $\boldsymbol{u} \in \mathbb{R}^n$ as $P_\mathcal{U}(\boldsymbol{u}) = \sum_{(i,j)\in\mathcal{E}_\mathcal{U}} A_{ij}|u_i - u_j|^p$.

Consider a second network $\mathcal{N}' = (\mathcal{G}', \mathcal{S}, p)$ formed by replacing $\mathcal{U}$ with a single edge (a, b); $\mathcal{V}_{\mathcal{G}'} = \mathcal{V}_\mathcal{G} \backslash \mathcal{V}_\mathcal{U}^I$, $\mathcal{E}_{\mathcal{G}'} = (\mathcal{E}_\mathcal{G} \backslash \mathcal{E}_\mathcal{U}) \cup \{(a, b)\}$. Let $\pi_{ab} = r_{\mathcal{U},p}(a, b)$, $|\mathcal{V}_{\mathcal{G}'}| = n'$ and denote the adjacency of $\mathcal{G}'$ by $\boldsymbol{A}'$. Let $\boldsymbol{w}'$ denote the minimal feasible potential state induced by $\mathcal{S}$ on $\mathcal{N}'$.

The potential at no vertex $v \in \mathcal{V}_\mathcal{U}^I$ is constrained by $\mathcal{S}$ and so $P_\mathcal{U}(\boldsymbol{w})$ is equal to the power produced across $\mathcal{U}$ when it is considered as an isolated circuit with the terminal vertices constrained to $\{(v_a, w_a), (v_b, w_b)\}$. Since such a circuit satisfies the conditions for Lemma 5.4.3 we have,

$$\begin{aligned} P_\mathcal{U}(\boldsymbol{w}) &= \frac{|w_a - w_b|^p}{r_{\mathcal{U},p}(a, b)} \\ &= \frac{|w_a - w_b|^p}{\pi_{ab}}. \end{aligned}$$

Thus $P_\mathcal{U}(\boldsymbol{w})$ is always identical to the power produced across a single edge with resistance $\pi_{ab} = r_{\mathcal{U},p}(a, b)$ and,

$$\begin{aligned} P(\mathcal{N}') &= \min_{\boldsymbol{u}\in\mathbb{R}^{n'}} \left\{ \sum_{(i,j)\in\mathcal{E}_{\mathcal{G}'}} |u_i - u_j|^p A'_{ij} : \mathcal{S} \right\} \\ &= \min_{\boldsymbol{u}\in\mathbb{R}^{n'}} \left\{ \sum_{(i,j)\in\mathcal{E}_\mathcal{G}\backslash\mathcal{E}_\mathcal{U}} |u_i - u_j|^p A_{ij} + \frac{|u_a - u_b|^p}{\pi_{ab}} : \mathcal{S} \right\} \\ &= \min_{\boldsymbol{u}\in\mathbb{R}^{n'}} \left\{ \sum_{(i,j)\in\mathcal{E}_\mathcal{G}\backslash\mathcal{E}_\mathcal{U}} |u_i - u_j|^p A_{ij} + \frac{|u_a - u_b|^p}{r_{\mathcal{U},p}(a, b)} : \mathcal{S} \right\} \\ &= \min_{\boldsymbol{u}\in\mathbb{R}^{n}} \left\{ \sum_{(i,j)\in\mathcal{E}_\mathcal{G}} |u_i - u_j|^p A_{ij} : \mathcal{S} \right\} \\ &= P(\mathcal{N}). \end{aligned} \tag{5.43}$$

It is then sufficient to notice that $\boldsymbol{w}'$ must be identical to $\boldsymbol{w}$ on $\mathcal{V}_\mathcal{G} \backslash \mathcal{V}_\mathcal{U}^I$ since by (5.43) they then produce the same (minimal) power: $P_\mathcal{N}(\boldsymbol{w}) = P_{\mathcal{N}'}(\boldsymbol{w}')$. That current on the external circuits is identical follows from (5.33). □

We demonstrate that the effective p-resistance satisfies an equivalent of Rayleigh's monotonicity law – suppose that the weighting of some edge of $\mathcal{G}$ is increased (equivalently, its resistance is decreased) or a new edge created, then the effective p-resistance between any two vertices of $\mathcal{G}$ does not increase.

Lemma 5.4.9. *(Rayleigh's Monotonicity Principle) Given $\mathcal{G}$ with adjacency matrix $\boldsymbol{A}$. Let $\mathcal{G}'$, with adjacency $\boldsymbol{A}'$, be identical to $\mathcal{G}$ except for the increase in the weight of one arbitrary edge (a, b), so that $A'_{ab} = A'_{ba} = A_{ab} + \delta$ for $\delta > 0$. Then for arbitrary vertices i and j,*

$$r_{\mathcal{G},p}(i, j) \geq r_{\mathcal{G}',p}(i, j).$$

Proof. Given any $v_i, v_j \in \mathcal{V}_\mathcal{G}$, let,

$$\boldsymbol{w} = \underset{\boldsymbol{u}\in\mathbb{R}^n}{\operatorname{argmin}}\{\|\boldsymbol{u}\|^p_{\mathcal{G},p} : u_i - u_j = 1\}.$$

Suppose that we can find a labelling $\boldsymbol{w}'$ of $\mathcal{G}'$ such that $w'_i - w'_j = 1$ and $\|\boldsymbol{w}'\|^p_{\mathcal{G}',p} < \|\boldsymbol{w}\|^p_{\mathcal{G},p}$, then note that,

$$\begin{aligned}
&\sum_{(k,\ell)\in\mathcal{E}_\mathcal{G}} |w'_k - w'_\ell|^p A_{k\ell} \\
&\qquad = \sum_{(k,\ell)\in\mathcal{E}_{\mathcal{G}'}} |w'_k - w'_\ell|^p A'_{k\ell} - |w'_a - w'_b|^p \delta \\
&\qquad \leq \sum_{(k,\ell)\in\mathcal{E}_{\mathcal{G}'}} |w'_k - w'_\ell|^p A'_{k\ell} \\
&\qquad < \sum_{(k,\ell)\in\mathcal{E}_\mathcal{G}} |w_k - w_\ell|^p A_{k\ell},
\end{aligned}$$

which contradicts the minimality of $\boldsymbol{w}$. Hence,

$$\min_{\boldsymbol{u}\in\mathbb{R}^n}\{\|\boldsymbol{u}\|^p_{\mathcal{G}',p} : u_i - u_j = 1\} \geq \|\boldsymbol{w}\|^p_{\mathcal{G},p},$$

from which (5.29) implies,

$$r_{\mathcal{G}',p}(i,j) \leq r_{\mathcal{G},p}(i,j).$$

□

Further we also have monotonicity in p so that for a graph $\mathcal{G}$ and vertices i and j if $p \leq q$ then,

$$r_{\mathcal{G},p}(i,j) \leq r_{\mathcal{G},q}(i,j).$$

5.4.2 Analysing the mistake bound for unweighted graphs

We are now better equipped with an understanding of effective p-resistance to analyse the mistake bound, Corollary 5.4.1. We see, through Lemmas 5.4.6 and 5.4.7, that p-resistance is a distance measure which captures both connectivity and the length of paths connecting points. Since it is difficult to evaluate the behaviour of (5.30) through p-resistance directly, we choose a more tractable approximation: we generalize the notion of graph diameter to that of (unweighted) wide diameter (Hsu, 1994). This approximation captures connectivity in the graph structure.

The k-wide distance $\delta_k(i,j)$ is the minimum value ℓ such that there exists k edge disjoint paths each containing v_i and v_j of length no more than ℓ (and $\delta_k(i,j) = \infty$ if no such k paths exist). We then define the k-wide diameter $\Delta_k(\mathcal{G}) := \max_{i,j}(\delta_k(i,j))$. Thus $\Delta_1(\mathcal{G})$ is just the usual diameter and if,

$$\Phi^0_\mathcal{G} := \min_{\boldsymbol{u}\in\{-1,1\}^n}\{\Phi_\mathcal{G}(\boldsymbol{u}) : \Phi_\mathcal{G}(\boldsymbol{u}) \geq 1\},$$

then by Menger's theorem (Diestel, 2005) then there exists $\Phi^0_\mathcal{G}$ edge-disjoint paths between all pairs of vertices. Thus if $k \leq \Phi^0_\mathcal{G}$ then $\Delta_k(\mathcal{G}) \leq n$. We can now bound the p-resistance diameter of an

unweighted graph $\mathcal{G}$ by,

$$R_p(\mathcal{G}) \leq \frac{\Delta_k(\mathcal{G})^{p-1}}{k}. \tag{5.44}$$

This follows immediately from application of resistors in parallel and series laws (Lemmas 5.4.6 and 5.4.7) to the set of k edge disjoints paths determined by the wide diameter $\Delta_k(\mathcal{G})$ and an application of Rayleigh's monotonicity principle (Lemma 5.4.9). We observe that (5.44) becomes tight as $p \to 1$ hence,

$$\lim_{p\to 1} R_p(\mathcal{G}) = \frac{1}{\Phi^0_{\mathcal{G}}}\,.$$

In the following we use the upper bound (5.44) to investigate the mistake bound (5.30). In Chapter 4 it is demonstrated that the case $p = 2$ (which is an online version of the harmonic energy minimisation of Zhu et al. (2003a); Belkin et al. (2004)) suffers a limitation – there exist graphs for which the algorithm makes $\Omega(\sqrt{|\mathcal{V}_{\mathcal{G}}|})$ mistakes. It has been demonstrated that simple online algorithms with a logarithmic mistake bound exist (Herbster et al., 2008; Cesa-Bianchi et al., 2009b). We will demonstrate that it is possible to choose p to ensure that $(\mathbf{\Psi}_{\mathcal{G}}, p)$-seminorm interpolation achieves a logarithmic guarantee.

The choice of p

A natural question arises: how does the behaviour of the $(\mathbf{\Psi}_{\mathcal{G}}, p)$-seminorm interpolation algorithm differ for various choices of p? To begin an investigation into this question we first deduce a mistake bound for the unweighted graph case in terms of a graph's wide diameter, and consider a simple tuning of p for unweighted graphs. For any vertex set partition $\mathcal{V}_1 \cup \ldots \cup \mathcal{V}_N = \mathcal{V}_{\mathcal{G}}$ with induced subgraphs $\mathcal{G}_1, \ldots, \mathcal{G}_N$ of maximum wide diameter $\Delta_k := \max\{\Delta_k(\mathcal{G}_i) : i = 1, \ldots, N\}$ we have as an immediate consequence of Corollary 5.4.1,

$$|\mathcal{M}| \leq N + \frac{4\Delta_k^2}{p-1}\left(\frac{\Phi(\boldsymbol{u})}{k\Delta_k}\right)^{\frac{2}{p}}, \tag{5.45}$$

for any $\boldsymbol{u} \in \{-1, 1\}^n$ correct on all trials. For the purpose of investigating the dependence of the bound (5.45) on the parameter p, we consider the hypothetical situation in which the graph cut $\Phi(\boldsymbol{u})$ is known to the learner a-priori and consider tuning (5.45) with regard to p. Note that, for $k\Delta_k > e^2\Phi(\boldsymbol{u})$ the quantity $\frac{1}{p-1}\left(\frac{\Phi(\boldsymbol{u})}{k\Delta_k}\right)^{\frac{2}{p}}$ is minimised when,

$$p = p^* = \log\left(\frac{k\Delta_k}{\Phi(\boldsymbol{u})}\right) - \sqrt{\left(\log\left(\frac{k\Delta_k}{\Phi(\boldsymbol{u})}\right)\right)^2 - 2\log\left(\frac{k\Delta_k}{\Phi(\boldsymbol{u})}\right)},$$

and we have that $1 < p^* < 2$. Of course, the value of $k\Delta_k$ is dependent upon the (optimal) choice of graph partition. Typically, when the diameter of a graph is large relative to the cut, lower values of p optimise (5.45). Note that the optimal value for p depends upon the unknown value of the cut of the true labelling. One situation in which the cut is typically likely to be small relative to the diameter is when the graph is sparse and has a large diameter. The situation is not simple, however, due to the connectivity element; below we demonstrate a dense, clustered graph for which a small choice of p is equally reasonable.

A simple tuning

We now give a simpler tuning (near-optimal) which will be used to evaluate the behaviour of p-seminorm interpolation in instructive cases. In a parallel with the logarithmic behaviour of the p-norm Perceptron, we show that it is possible to choose p (using information known to the learner a-priori) to ensure a performance guarantee which is logarithmic in the graph diameter.

Corollary 5.4.10. *Given the task of predicting the labelling of any unweighted, connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in the online framework, the number of mistakes, $|\mathcal{M}|$, incurred by minimum $(\mathbf{\Psi}_{\mathcal{G}}, p)$-seminorm interpolation with $p := \frac{c}{c-1}$ is bounded by,*

$$|\mathcal{M}| \leq \begin{cases} N + \frac{4e^2\Phi^2(\boldsymbol{u})[\log(k\Delta_k) - \log(\widehat{\Phi}) - 1]}{k^2} & \frac{k\Delta_k}{\widehat{\Phi}} > e^2 \\ N + \frac{4\Phi(\boldsymbol{u})\Delta_k}{k} & \frac{k\Delta_k}{\widehat{\Phi}} \leq e^2, \end{cases}$$

where $c = \max(\log[\frac{k\Delta_k}{\widehat{\Phi}}], 2)$ and $\mathcal{V}_1 \cup \ldots \cup \mathcal{V}_N = \mathcal{V}_{\mathcal{G}}$ is any vertex set partition with induced subgraphs $G_1, \ldots, G_N$ of maximum wide diameter $\Delta_k := \max\{\Delta_k(G_i) : i = 1, \ldots, N\}$, $\widehat{\Phi}$ is any constant $1 \leq \widehat{\Phi} \leq \Phi(\boldsymbol{u})$ and $\boldsymbol{u} \in \{-1, 1\}^n$ is any labelling consistent with the trial sequence.

Note immediately that by choosing $k = 1, \widehat{\Phi} = 1$, for $\Delta_1 = \max_i D(\mathcal{G}_i) > e^2$, we recover a mistake bound which is a logarithmic function of the graph diameter. In the following we consider three examples with varying degrees of connectivity. The *tree*, a prototypically sparse graph, is minimally connected with $k = 1$. The $2m$-vertex dense *barbell*, an idealized model of two clusters, has connectivity $k = m - 1$. Finally the mD-vertex *cylinder* has an intermediate connectivity $k = m$. This intermediate case more generally includes graphs with spatially extended clusters whose internal connectivity equals or exceeds the cut between clusters. The bounds for these intermediately connected graphs uniformly improve on the results in Herbster (2008); Herbster et al. (2008); Cesa-Bianchi et al. (2009b).

Tree graph

Consider a tree. We take $N = 1, k = 1, \Delta_k = D = \max_i D(\mathcal{G}_i)$ in Corollary 5.4.10. For $\frac{D}{\widehat{\Phi}} > e^2$ the first tuning ($p < 2$) in Corollary 5.4.10 is preferred and we derive,

$$|\mathcal{M}| \leq 1 + 4e^2\Phi^2(\boldsymbol{u})[\log(D) - \log(\widehat{\Phi}) - 1].$$

For $\frac{D}{\widehat{\Phi}} \leq e^2$ we derive, from the second tuning ($p = 2$)

$$|\mathcal{M}| \leq 1 + 4\Phi(\boldsymbol{u})D.$$

Barbell graph

Consider the barbell graph: two m-cliques joined by Φ connecting cut edges. We take $N = 2, \Delta_k = 2, k = m - 1$ in Corollary 5.4.10. For $\frac{2(m-1)}{\widehat{\Phi}} > e^2$ the first tuning ($p < 2$) in Corollary 5.4.10 is preferred and we derive,

$$|\mathcal{M}| \leq 2 + \frac{4e^2\Phi^2(\boldsymbol{u})[\log(2(m-1)) - \log(\widehat{\Phi}) - 1]}{(m-1)^2}.$$

For $\frac{2(m-1)}{\hat{\Phi}} \leq e^2$ we derive, from the second tuning ($p = 2$),

$$|\mathcal{M}| \leq 2 + \frac{8\Phi(\boldsymbol{u})}{m-1}.$$

Note that a bound of 2 is optimal for this barbell graph labelling problem.

Cylinder graph

Consider the "cylindrical" graph that is the Cartesian product of an m-clique with a path graph of D vertices. This cylinder may be visualized as D "aligned" cliques. We assume the cylinder is labeled with two classes by an m-edge cut that partitions into two cylinders. Assuming $D > e^2 - 1$ then choosing $p = 1 + \frac{1}{\log(D+1)-1}$ (with $N = 1$, $k = m$, and $\Delta_k = D + 1$) and substituting into Corollary 5.4.10 we derive,

$$|\mathcal{M}| \leq 4e^2 \log(D+1)\,.$$

If instead we tune with $p = 2$ we have $|\mathcal{M}| \leq 5 + 4D$. Further this bound improves on the "spine" method in Chapter 4 which has a bound of $O(k \log D)$ for this problem.

5.5 Towards efficient p-norm projections

(*This section is essentially useless without a closed form for the dual norm in the graph case. It can be skipped without loss. It is included as a point of interest and to show that the problem of obtaining a closed form has an immediate useful application.*)

In Corrolary 5.3.7 we observed that the online minimum $(\boldsymbol{\Psi}, p)$-seminorm interpolation algorithm is equivalent to performing successive Bregman projections to the intersection of constraint sets. Each single such projection could itself be performed by successive projections to each hyperplane constraint – such a succession of projections will converge to the projection to the intersection of constraints (Csiszar, 1975; Bauschke and Borwein, 1997). However this would not be an efficient implementation.

In this section we present a related algorithm which, at each (mistaken) trial performs a single projection to a single hyperplane and obtains exactly the same mistake bound provided by Theorem 5.3.1 and Corollary 5.4.1 for the $(\boldsymbol{\Psi}, p)$-seminorm interpolation algorithm.

5.5.1 Projection algorithm

Let $\mathcal{H}^\perp := \{\boldsymbol{u} \in \mathbb{R}^n \ : \ \boldsymbol{u}^\top \mathbf{1} = 0\}$. Let $F(\boldsymbol{u}) = \frac{1}{2}\|\boldsymbol{u}\|^2_{\boldsymbol{\Psi},p}$ be defined over all $\boldsymbol{u} \in \mathcal{H}^\perp$. Denote by $f^\star$ the Legendre-Fenchel conjugate of a convex function f and, with reference to Appendix E, note that $F^\star(\boldsymbol{v}) = \frac{1}{2}\|\boldsymbol{v}\|^{*^2}_{\boldsymbol{\Psi},p}$. Define the Bregman divergence,

$$D_F(\boldsymbol{u}, \boldsymbol{w}) = F(\boldsymbol{u}) - F(\boldsymbol{w}) - (\boldsymbol{u} - \boldsymbol{w})^\top \boldsymbol{\nabla} F(\boldsymbol{w}).$$

Define the *Bregman projection* of $\boldsymbol{w} \in \mathcal{H}^\perp$ to a convex set $\mathcal{C} \subseteq \mathcal{H}^\perp$,

$$\mathrm{proj}_{\mathcal{C}}(\boldsymbol{w}) = \operatorname*{argmin}_{\boldsymbol{u} \in \mathcal{C}} D_F(\boldsymbol{u}, \boldsymbol{w}).$$

We consider the following Pounce-like (Herbster, 2008) algorithm of Figure 5.2. Note that Pounce Herbster (2008) corresponds to the case $p = 2$.

Parameters: A linear map $\boldsymbol{\Psi} : \mathbb{R}^n \to \mathbb{R}^{\hat{n}}$ and $p \in (1, 2]$

Initialization: $\boldsymbol{w}_1 = \mathbf{0}$; $\mathcal{M} = \{\}$

Input: $\{(i_t, y_t)\}_{t=1}^m \in (\mathbb{N}_n \times \{-1, 1\})^m$

for $t = 1, \ldots, m$ **do**

Receive: $i_t \in \{1, \ldots, n\}$

Predict: $\hat{y}_t = \operatorname{sgn}((\boldsymbol{e}_{i_t} - \boldsymbol{e}_{i_{\eta_t}})^\top \boldsymbol{w}_t + y_{\eta_t})$

Receive: y_t

if $\hat{y}_t \neq y_t$ **then**

$\mathcal{M} = \mathcal{M} \cup \{t\}$

$\boldsymbol{w}_{t+1} = \operatorname{proj}_{\mathcal{U}_t}(\boldsymbol{w}_t)$, where $\mathcal{U}_t$ is the hyperplane $\mathcal{U}_t := \{\boldsymbol{u} \in \mathcal{H}^\perp \ : \ (\boldsymbol{e}_{i_t} - \boldsymbol{e}_{i_{\eta_t}})^\top \boldsymbol{u} = y_t - y_{\eta_t}\}$

end

Figure 5.2: Minimum $(\boldsymbol{\Psi}, p)$-Bregman projection

Form of the projection

Note that F is Legendre (closed, proper and essentially smooth and strictly convex on the relative interior of its domain (e.g. Rockafellar, 1972)) on $\mathcal{H}^\perp$. To find the projection $\boldsymbol{w}_{t+1} = \operatorname{proj}_{\mathcal{U}_t}(\boldsymbol{w}_t)$ it is then just a case of forming the Lagrangian of the relevent constrained convex minimisation problem, and following standard techniques (see, for example, (Dhillon and Tropp, 2008, Section 3.1)). We have the following convex problem,

$$\min_{\boldsymbol{u} \in \mathbb{R}^n} D_F(\boldsymbol{u}, \boldsymbol{w}_t) \qquad \text{subject to :} \qquad (\boldsymbol{e}_{i_t} - \boldsymbol{e}_{i_{\eta_t}})^\top \boldsymbol{u} = y_t - y_{\eta_t}$$
$$\mathbf{1}^\top \boldsymbol{u} = 0.$$

Forming the Lagrangian for this problem and taking the derivative in $\boldsymbol{u}$, which must be zero at $\boldsymbol{w}_{t+1}$, implies that,

$$\begin{aligned} \boldsymbol{\nabla}_{\boldsymbol{u}} D_F(\boldsymbol{u}, \boldsymbol{w}_t)|_{\boldsymbol{u}=\boldsymbol{w}_{t+1}} - \xi(\boldsymbol{e}_{i_t} - \boldsymbol{e}_{i_{\eta_t}}) - \zeta \mathbf{1} &= 0 \\ \boldsymbol{\nabla} F(\boldsymbol{w}_{t+1}) = \boldsymbol{\nabla} F(\boldsymbol{w}_t) + \xi(\boldsymbol{e}_{i_t} - \boldsymbol{e}_{i_{\eta_t}}) + \zeta \mathbf{1} & \end{aligned} \tag{5.46}$$

where ξ and ζ are Lagrange multipliers. By taking the inner product with $\mathbf{1}$, (5.46) implies $\zeta = 0$ (as expected) and since the inverse gradient of a strictly convex function is the gradient of its conjugate we have,

$$\boldsymbol{w}_{t+1} = \boldsymbol{\nabla} F^\star(\xi(\boldsymbol{e}_{i_t} - \boldsymbol{e}_{i_{\eta_t}}) + \boldsymbol{\nabla} F(\boldsymbol{w}_t)), \tag{5.47}$$

where ξ is such that,

$$(\boldsymbol{e}_{i_t} - \boldsymbol{e}_{i_{\eta_t}})^\top \boldsymbol{\nabla} F^\star(\xi(\boldsymbol{e}_{i_t} - \boldsymbol{e}_{i_{\eta_t}}) + \boldsymbol{\nabla} F(\boldsymbol{w}_t)) = y_t - y_{\eta_t}. \tag{5.48}$$

Note that, equivalently, ξ is the minimum of the strictly convex univariate function,

$$J(x) = F^\star(x(\boldsymbol{e}_{i_t} - \boldsymbol{e}_{i_{\eta_t}}) + \boldsymbol{\nabla} F(\boldsymbol{w}_t)) - (y_t - y_{\eta_t})x. \tag{5.49}$$

The projection (5.47) can then be found by either solving for ξ analytically in (5.48) or by, for example, numerically finding the minimiser of (5.49).

Note then that calculating this projection amounts to finding a closed form for the dual norm $\|\boldsymbol{v}\|^*_{\mathbf{\Psi},p}$, since without this closed form none of (5.47), (5.48) or (5.49) have closed form expressions.

Mistake bound

This attains the same mistake bound as the algorithm in Herbster and Lever (2009): using the notation of Theorem 5.3.1 we have:

Theorem 5.5.1. *The number of mistakes, $|\mathcal{M}|$, incurred by the $(\mathbf{\Psi}, p)$-Bregman projection algorithm, for any $\rho > 0$, is bounded by,*

$$|\mathcal{M}| \leq N(\mathcal{X}', \rho, d_{\mathbf{\Psi},p}) + \frac{\rho^2 \|\boldsymbol{u}\|^2_{\mathbf{\Psi},p}}{p-1},$$

where $\boldsymbol{u} \in \mathbb{R}^n$ is any labelling such that $u_{i_t} = y_t \; \forall t \leq m$, and $N(\mathcal{X}', \rho, d_{\mathbf{\Psi},p})$ is the covering number of the input set $\mathcal{X}' = \{i_1, i_2, ..., i_m\}$ relative to the distance,

$$d_{\mathbf{\Psi},p}(i,j) := \|E_i - E_j\|^*_{\mathbf{\Psi},p}.$$

Proof. (sketch) This is proved as in Theorem 5.3.1. The key points are as follows: we have a Pythagorean theorem for each projection on a (mistaken) trial t,

$$D_F(\boldsymbol{u}, \boldsymbol{w}_t) = D_F(\boldsymbol{w}_{t+1}, \boldsymbol{w}_t) + D_F(\boldsymbol{u}, \boldsymbol{w}_{t+1}) \qquad \forall \boldsymbol{u} \in \mathcal{U}_t,$$

and so over all trials $t \in [1, m]$ we have,

$$D_F(\boldsymbol{u}^*, \boldsymbol{w}_t) = D_F(\boldsymbol{w}_{t+1}, \boldsymbol{w}_t) + D_F(\boldsymbol{u}^*, \boldsymbol{w}_{t+1}) \qquad \forall \boldsymbol{u}^* \in \mathcal{U}^*, \tag{5.50}$$

where $\mathcal{U}^* = \cap_{t=1}^{\top} \mathcal{U}_t$. Note that $\mathcal{U}^*$ is nonempty in the realisable case since, for example, the projection of the true labelling of $\mathcal{G}$ onto $\mathcal{H}^\perp$ is always in $\mathcal{U}^*$. Summing (5.50) gives,

$$\begin{aligned} \sum_{t=1}^{\top} D_F(\boldsymbol{w}_{t+1}, \boldsymbol{w}_t) &= D_F(\boldsymbol{u}^*, \boldsymbol{w}_1) - D_F(\boldsymbol{u}^*, \boldsymbol{w}_{T+1}) \\ &\leq D_F(\boldsymbol{u}^*, \boldsymbol{0}) \\ &= F(\boldsymbol{u}^*). \end{aligned}$$

The rest of the proof follows as in Theorem 5.3.1. □

5.6 Transductive risk bound for the minimum $(\mathbf{\Psi}, p)$-seminorm algorithm

We recall the setting of transductive learning discussed in Section 1.3.1. In this section we prove a bound on the transductive classification risk as defined in (1.14) for the minimum $(\mathbf{\Psi}, p)$-seminorm algorithm.

Several methods to derive bounds on the generalization ability of a classifier which is learned through an online process have been proposed (Littlestone, 1989; Cesa-Bianchi et al., 2001; Graepel et al., 2005; Cesa-Bianchi and Gentile, 2008). Such methodologies can yield tight risk tail bounds for common algorithms such as the perceptron (Graepel et al., 2005). These techniques for deducing risk bounds from online analyses require the assumption that draws of instances from some underlying input space are i.i.d. and as discussed in Section 1.3.1 this assumption is atypical in the transductive setting. Here we adapt the result of Cesa-Bianchi et al. (2001) to the case in which instances are sampled uniformly without replacement from a finite set. Thus we extend this methodology for deriving a risk bound from an online analysis to make it more naturally applicable to transduction.

Recalling Section 1.3.1, in the transductive learning framework it is common to assume that instances are uniformly sampled without replacement from the finite set $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ of labeled inputs. Let $\mathcal{X}$ be a finite input set and $\mathcal{Y}$ the corresponding label space so that $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ is the joint space of labeled inputs. Consider an online algorithm A acting on an (ordered) trial sequence $\mathcal{S} = \{(x_1, y_1), (x_2, y_2)...(x_m, y_m)\} \subseteq \mathcal{Z}$ and let $h_t : \mathcal{X} \to \mathcal{D}$ denote the hypothesis formed after trial t, where $\mathcal{D}$ is a decision space. Let $\mathcal{H}_{\mathcal{S}} = \{h_0, h_1...h_{m-1}\}$ be the ensemble of hypotheses produced when A is run on $\mathcal{S}$ (note the exclusion of the final hypothesis h_m). Denote $|\mathcal{Z}| = n$ so that $u = n - m$ is the size of the "test set" (i.e. that part of $\mathcal{Z}$ remaining unlabelled and on which a labelling must be inferred).

Let (X_t, Y_t) denote the pair of random variables, taking values in $\mathcal{Z}$, drawn at trial t according to some distribution. We suppose that labeled instances are sampled uniformly without replacement from $\mathcal{Z}$. Denote by $P_t(\cdot) = P_{(X_t,Y_t)}(\cdot \mid (X_1, Y_1), ...(X_{t-1}, Y_{t-1}))$ the probability measure for the tth draw from $\mathcal{Z}$, that is, the uniform probability measure over the draw of instances from the finite set $\mathcal{Z}_t = \mathcal{Z} \backslash \{(X_1, Y_1), ...(X_{t-1}, Y_{t-1})\}$ of labeled inputs remaining at trial t, and $\mathbb{E}_t[\cdot]$ the corresponding expectation. We note that the results of Cesa-Bianchi et al. (2001), regarding the risk of learned hypotheses, are no longer valid under the above assumptions on the distribution of the draw of instances. (However if we are willing to adopt the less typical assumption of an i.i.d. draw of instances then the standard bounds are immediately applicable.)

Let $\ell_{0-1} : \mathcal{D} \times \mathcal{Y} \to \{0, 1\}$ denote the zero-one loss function. We recall the notion of transductive (classification) risk as defined in (1.14),

$$\text{risk}_{\mathcal{T}}(h_t) = \frac{1}{u} \sum_{i=m+1}^{n} \ell_{0-1}(h_t(X_i), Y_i). \tag{5.51}$$

For each $h_t \in \mathcal{H}_{\mathcal{S}}$ we also define the following measure of risk,

$$\overline{\text{risk}}(h_{t-1}) := \mathbb{E}_t(\ell_{0-1}(h_{t-1}(X_t), Y_t)) \tag{5.52}$$

$$= \frac{1}{|n - t + 1|} \sum_{i=t}^{n} \ell_{0-1}(h_{t-1}(X_i), Y_i) \tag{5.53}$$

which follows since the draw from $\mathcal{Z}_t = \{(X_t, Y_t), ...(X_n, Y_n)\}$ is uniform. We ultimately derive a bound on the transductive risk for a specific classifier by first proving a bound for the quantity defined by (5.53).

Let algorithm A act on a trial sequence $\mathcal{S} = \{(x_1, y_1), (x_2, y_2)...(x_m, y_m)\} \subseteq \mathcal{Z}$. In complete analogue to Cesa-Bianchi et al. (2001) we demonstrate that a particular hypothesis from the ensemble $\mathcal{H}_\mathcal{S}$ which has low risk with high probability is that which minimizes the notion of *penalized empirical risk*.

Definition The *empirical risk* of h_t is defined $\widehat{\text{risk}}(h_t, t+1) = \frac{1}{m-t}\sum_{i=t+1}^{m} \ell_{0-1}(h_t(X_i), Y_i)$.

Definition The δ-*penalized empirical risk* of h_t is defined

$$\widehat{\text{risk}}^{(\delta)}(h_t, t+1) = \widehat{\text{risk}}(h_t, t+1) + c_\delta(m-t) \tag{5.54}$$

where $c_\delta(x) = \sqrt{\frac{1}{2x}\ln\frac{m(m+1)}{\delta}}$ for $x = 1, 2, ...m$.

We denote $\hat{h} = \text{argmin}_{h_t \in \mathcal{H}_\mathcal{S}}\{\widehat{\text{risk}}^{(\frac{\delta}{2})}(h_t, t+1)\}$ and we will prove the following bound for $\overline{\text{risk}}(\hat{h})$ which is the counterpart of (Cesa-Bianchi et al., 2001, Theorem 4).

Theorem 5.6.1. *Suppose an online algorithm A is run on a trial sequence $\mathcal{S} = \{(x_1, y_1), ...(x_m, y_m)\}$ of labeled instances, drawn uniformly without replacement from a discrete labeled input set $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. For any $\delta \in (0, 1]$ let $\hat{h} \in \{h_0, h_1, ...h_{m-1}\}$ be that hypothesis which minimizes the $\frac{\delta}{2}$-penalized empirical risk. Then,*

$$\mathbb{P}\left(\overline{\text{risk}}(\hat{h}) \geq \frac{M(\mathcal{S})}{m} + 6\sqrt{\frac{1}{m}\ln\frac{2(m+1)}{\delta}}\right) \leq \delta \tag{5.55}$$

where the probability is w.r.t. the draw of the training sample and $M(\mathcal{S})$ is (any upper bound on) the number of mistakes incurred by A on $\mathcal{S}$.

The result and technique of the proof are due almost entirely to Cesa-Bianchi et al. (2001), but the argument is repeated in Appendix F with the changes required for our application highlighted. In general it is fairly straightforward to derive risk tail bounds valid for the transductive setting which are analogous to those from the inductive setting: it is simply a case of analysing not the tails of the binomial distribution but the (shallower) tails of the hypergeometric distribution which can be done using, for example, Serfling's inequality (Serfling, 1974).

The following corollary provides a bound on the transductive classification risk as defined in (5.51) for the minimum $(\mathbf{\Psi}, p)$-seminorm interpolation algorithm.

Corollary 5.6.2. *Suppose the minimum $(\mathbf{\Psi}, p)$-seminorm interpolation algorithm is run on a trial sequence $\mathcal{S} = \{(v_{i_1}, y_1), ...(v_{i_m}, y_m)\}$ of labeled vertices, drawn uniformly without replacement from $\mathcal{G}$. For any $\delta \in (0, 1]$ let $\hat{h}$ be that hypothesis which minimizes the $\frac{\delta}{2}$-penalized empirical risk, and let $\hat{t}$ be such that $\hat{h} = h_{\hat{t}}$. We have, for any $\rho > 0$,*

$$\mathbb{P}\Bigg(\text{risk}_\mathcal{T}(\hat{h}) \leq \frac{n-\hat{t}}{n-m}\left(\frac{N(\mathcal{V}_\mathcal{S}, \rho, r_p) + \frac{[\rho\,\Phi_p(\boldsymbol{u})]^{\frac{2}{p}}}{p-1}}{m} + 6\sqrt{\frac{1}{m}\ln\frac{2(m+1)}{\delta}}\right) - \frac{1}{n-m}\sum_{i=\hat{t}+1}^{m}\ell_{0-1}(\hat{h}(x_i), y_i)\Bigg) \geq 1-\delta \tag{5.56}$$

where the probability is w.r.t. the draw of the training sample and $\boldsymbol{u} \in \mathbb{R}^n$ *is any labeling of* $\mathcal{G}$ *such that* $u_{i_t} = y_t \; \forall t \leq m$, $p \in (1, 2]$, *and* $N(\mathcal{V}_\mathcal{S}, \rho, r_p)$ *is the covering number of the input set* $\mathcal{V}_\mathcal{S} = \{v_{i_1}, v_{i_2}, ...v_{i_m}\}$ *of labeled vertices relative to the* p*-resistance* r_p.

Proof. We first note the equality, valid for all $t < m$,

$$\begin{aligned} \mathrm{risk}_\mathcal{T}(h_t) &= \frac{n-t}{n-m}\left(\frac{1}{n-t}\sum_{s=m+1}^{n} \ell_{0-1}(h_t(v_{i_s}), y_s)\right) \\ &= \frac{n-t}{n-m}\left(\overline{\mathrm{risk}}(h_t) - \frac{1}{n-t}\sum_{s=t+1}^{m} \ell_{0-1}(h_t(v_{i_s}), y_s)\right). \end{aligned} \tag{5.57}$$

The term $\overline{\mathrm{risk}}(\hat{h})$ can then be bounded using (5.55) and then we finally plug the mistake bound of Corollary 5.4.1 into (5.57). □

Let's make some observations: typically we expect $n \gg m$ so that the multiplicative term in the bound (5.56) satisfies $\frac{n-\hat{t}}{n-m} \approx 1$. However when $n \approx m$ the bound is crude (and there is a convergence issue) – we note this could probably be corrected with a more thorough argument. Nonetheless, this bound should be compared favourably to typical risk bounds for the transductive setting. For example, the rate of convergence of the complexity term is $\mathcal{O}(\frac{1}{m})$ compared to the typical $\mathcal{O}(\frac{1}{\sqrt{m}})$ bounds of, for example, Theorem 1.3.1 and the bounds of Theorem 3.4.5 and Theorem 3.4.6 discussed in Chapter 3.

5.7 Discussion

We have presented an algorithm for predicting the labelling of a graph which achieves bounds of a similar form to those of the p-norm Perceptron of Grove et al. (1997). A main argument of this chapter is that intermediate values of p in our $(\mathbf{\Psi}, p)$-seminorm interpolation algorithm may have important advantages over the extreme cases of $p = 1$ and $p = 2$. This is in agreement with the practical observations in Bühler and Hein (2009); Singaraju et al. (2009), which considers an algorithm similar to $(\mathbf{\Psi}, p)$-seminorm interpolation for interactive image segmentation. As with the p-norm perceptron there is a direct argument that gives bounds which scale logarithmically with the dimension n of the input space. We refined these "$\mathcal{O}(\log n)$" to "$\mathcal{O}(\log D)$" bounds in section 5.4.2 using the geometrical results on p-resistive networks from 5.4.1: it is possible for the learner to tune p using known geometrical quantities of the graph to obtain a $\mathcal{O}(\log D)$ bound. The bounds may be further improved by recognizing that the diameter D of the input space is replaceable by the diameter of the balls that constitute a cover of the inputs. This was accomplished by adapting the methods of Herbster (2008) to a p-norm framework. We remark that as $p \to 1$ the bound 5.8 diverges since the strong convexity arguments are not tight. A sharper analysis of the case $p \to 1$ may be possible by other methods.

We note the following open problem. As discussed for trees we obtain a bound of $\mathcal{O}(\Phi^2 \log D)$. In Herbster et al. (2008) and in Cesa-Bianchi et al. (2009b) efficient online algorithms were proposed with mistake bounds of $\mathcal{O}(\Phi \log \frac{n}{\Phi} + \Phi)$ and $\mathcal{O}(\Phi \log D)$ respectively. The drawbacks of these algorithms are that they are not able to fully exploit additional connectivity in non-tree graphs as typified by

barbell or cylinder graphs. This leaves as an open problem the discovery of an algorithm that can obtain $\mathcal{O}(\Phi \log D)$ on trees but also exploit edge-connectivity as typified by Corollary 5.4.10.

We also remark that Theorem 5.3.1 holds for a much wider class of interpolation algorithms in which the complexity penalty is κ-strongly convex – the denominator $p-1$ in the bound would be replaced by κ. Lemma 5.3.4 and Lemma 5.3.5 are then just seen as a proof of the $p-1$-strong convexity of the complexity $\frac{1}{2}\|\boldsymbol{u}\|^2_{\boldsymbol{\Psi},p}$. However we do not know how to recover the clustering aspect of the bound under a relaxation of the constraint that the comparison function is exactly correct on the entire trial sequence (which seems strict in the general case).

Chapter 6

Summary of online graph label prediction algorithms

Since the original conference publication of the results of Chapters 4 and 5 (Herbster et al., 2008; Herbster and Lever, 2009) more research in this field has emerged, and we here compare these existing methods and tabulate a overview of the key results. We recall some notation: $D(\mathcal{G})$ is the diameter of a graph $\mathcal{G}$; $N(\mathcal{X}, \rho, d)$ is the minimum number of sets of diameter no greater than ρ in the metric $d : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}_{\geq 0}$ required to cover a set $\mathcal{X} \subseteq \mathcal{V}$; $r_{\mathcal{G}}$ is the resistance distance metric; $R_{\mathcal{G}}$ is the resistance diameter of $\mathcal{G}$; $\Phi(\boldsymbol{u}) = \sum_{(i,j)\in\mathcal{E}} A_{ij}|u_i - u_j|^2$ is the cut of $\boldsymbol{u}$. Recalling Chapter 5 $r_{\mathcal{G},p}$ is the *p-resistance* on $\mathcal{G}$ induced by a p-norm $||\boldsymbol{u}||^2_{\mathcal{G},p} = \left(\sum_{(i,j)\in\mathcal{E}} A_{ij}|u_i - u_j|^p\right)^{\frac{2}{p}}$ which generalizes the cut which is obtained when $p = 2$.

The online graph labelling problem was first studied in Herbster et al. (2005) where a bound for the kernel Perceptron was derived which related learning to the resistance diameter of the graph, and the cut size induced by the underlying binary labelling. This bound was improved upon by a second algorithm, Pounce (Herbster, 2008), which furhter exploited cluster structure in the resistance metric (along with demonstrating that this is not the case for the Perceptron). Both of these algorithms have quadratic runtime in the number of vertices. The algorithm of Chapter 4 (originally in Herbster et al. (2008)) was the fisrt demonstration of an algorithm with a mistake bound (Theorem 4.4.3) which is always logarithmic in the number of vertices of the graph. This algorithm has loglinear runtime in the number of vertices and this was accompanied by a second algorithm which exploited cluster structure in the sense of Pounce (Theorem 4.5.3). Slightly weaker logarithmic bounds were later presented in Fakcharoenphol and Kijsirikul (2008). More-or-less simultaneously two algorithms were presented which attained a mistake bound which is logarithmic in the diameter of the graph – the algorithm of Chapter 5 and the algorithm of (Cesa-Bianchi et al., 2009b). The former exploits cluster structure in the sense of Pounce and the primary mistake bound is in terms of the cluster structure in the p-resistance metric, when compared purely in terms of the simultaneous dependence upon the graph cut and the diamter terms the latter bound has a favourable linear dependence on the cut (compared to a quadratic dependence for the

algorithm of Chapter 5 when p is chosen to ensure logarithmic dependence on the diameter) and has runtime which is quadratic in the number of vertices.

The following table presents the results, time complexity is the (amortized per trial) time required to predict all $|\mathcal{V}|$ vertices of a graph. When two bounds exist for a given algorithm, both hold simultaneously. Note that Algorithms A, B, D and E require an expensive initialization of computing the pseudoinverse of a graph Laplacian, which would often be of complexity greater than $\mathcal{O}(|\mathcal{V}|^2)$ and is not included in the complexity below. The improved amortized time per trial complexity of Algorithm C was proved recently in Cesa-Bianchi et al. (2010), even though the worst case time per trial is $\mathcal{O}(\log|\mathcal{V}|)$. The 'initialization' of algorithms C and D of finding a spine by performing a depth first search of the graph is included and accounts for the $\mathcal{O}(|\mathcal{E}|)$ term.

Table 6.1: Comparison of algorithms predicting all vertices of an unweighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

	Algorithm	Mistake Bound	Time Complexity
A	Perceptron (Herbster and Pontil, 2007)	$\mathcal{O}(\Phi(\boldsymbol{u})R_{\mathcal{G}})$	$\mathcal{O}(\|\mathcal{V}\|^2)$
B	Pounce (Herbster, 2008)	$\forall\rho>0:\ \mathcal{O}(N(\mathcal{V},\rho,r_{\mathcal{G}})+\Phi(\boldsymbol{u})\rho)$	$\mathcal{O}(\|\mathcal{V}\|^2)$
C	Prediction with a spine Theorem 4.4.3	$\mathcal{O}\left(\Phi(\boldsymbol{u})\log\left(\frac{\|\mathcal{V}\|}{\Phi(\boldsymbol{u})}\right)\right)$	$\mathcal{O}(\|\mathcal{V}\|+\|\mathcal{E}\|)$
D	Prediction with a support tree Theorem 4.5.3	$\forall\rho>0:\ \mathcal{O}\left(N(\mathcal{V},\rho,r_{\mathcal{G}})+\Phi(\boldsymbol{u})\rho\right)$ $\mathcal{O}(\Phi(\boldsymbol{u})(\log\|\mathcal{V}\|)^2\log(\log\|\mathcal{V}\|))$	$\mathcal{O}(\|\mathcal{V}\|^2+\|\mathcal{E}\|)$
E	p-seminorm interpolation Theorem 5.4.1	$\forall\rho>0:\ \mathcal{O}\left(N(\mathcal{V},\rho,r_{\mathcal{G},p})+\frac{\|\|\boldsymbol{u}\|\|^2_{\mathcal{G},p}\rho^{2/p}}{p-1}\right)$	N/A
F	(Cesa-Bianchi et al., 2009b)	$\mathcal{O}(\Phi(\boldsymbol{u})\log D(\mathcal{G}))$	$\mathcal{O}(\|\mathcal{V}\|^2)$

Notation

Recurrent notation is listed below. Notation that is introduced and used only locally is ommitted. Section numbers are given in brackets.

Sets, spaces and related objects

$\mathcal{X}, \mathcal{Y}, \mathcal{Z} = \mathcal{X} \times \mathcal{Y}$	Input space, label space joint space of labelled inputs (1.1.1)
$\mathcal{D}$	Decision space (1.1.1)
$\mathcal{H}, \mathcal{H}_K$	Hypothesis class (1.1.1), reproducing kernel Hilbert space (1.2.2)
$\mathcal{H}_\alpha$	Hypotheses with α-bounded complexity (3.2)
$\mathcal{S}, \mathcal{S}_{\text{labelled}}, \mathcal{S}_{\text{unlabelled}}$	Training sample (1.1.1), labelled sample, unlabelled sample (1.3.1)
$\mathcal{T}$	Test set (1.3.1) (also a tree in Chapter 4)
$(\mathcal{X}, \Sigma, \nu)$	Measure space (1.2.2)
$\mathcal{L}^2(\mathcal{X}, \Sigma, \nu)$	Square integrable functions on $(\mathcal{X}, \Sigma, \nu)$ (1.2.2)
$\ell^2, \mathbb{R}^\infty$	Square-summable real-valued sequences, Real vlaued sequences
$\mathcal{C} = \{\mathcal{C}_i\}_i$	Clustering or cover (comb graph in Chapter 4)
$N(\mathcal{U}, \rho, d)$	Covering number of $\mathcal{U}$ w.r.t. distance d

Loss, risk and mistakes

ℓ, ℓ_{0-1}	Loss function, classification loss (1.1.1)
$\text{risk}^\ell, \widehat{\text{risk}}^\ell_{\mathcal{S}}$	Risk associated to ℓ, empirical risk on $\mathcal{S}$ (1.1.1)
$\text{risk}, \widehat{\text{risk}}_{\mathcal{S}}$	classification risk, empirical classification risk on $\mathcal{S}$ (1.1.1)
$\text{risk}^\ell_{\mathcal{T}}$	Transductive risk on $\mathcal{T}$ (1.3.1)
$M_{\mathcal{S}}(\boldsymbol{u})$	Number of mistakes on $\mathcal{S}$ by $\boldsymbol{u}$ (4.4.1)

Operators, matrices, special functions, norms

$K(\cdot,\cdot)$	Kernel function (1.2.2)
κ	$\sup_{x\in\mathcal{X}} K(x,x)$ (2.4.2)
$d_K(\cdot,\cdot)$	Feature space metric associated to K (2.4.2)
A_K	Integral operator associated to a kernel K (1.2.2)
$F_P(\cdot), F_Q(\cdot)$	Energy functions or regularizers (2.2.1)
$\widehat{U}_{\mathcal{S}}(\cdot), U(\cdot)$	Smoothness functional and its expectation (2.3.3)
$\widehat{U}_{\mathcal{S}}(\cdot), U(\cdot)$	Generic U-statistic and its expectation (2.3.3)
$D_\Phi(\cdot,\cdot)$	Bregman divergence associated to Φ (2.4.2)
$\|\|\cdot\|\|, \|\|\cdot\|\|_*(\|\|\cdot\|\|^*)$	Norm and its dual (occasional usage)
$\|\|\cdot\|\|_p$	p-norm (5.2)
E_i	Linear functional $E_i(\boldsymbol{v}) = \boldsymbol{e}_i^\top \boldsymbol{v}$ (5.2)
$F_{\boldsymbol{L}}(\cdot)$	Laplacian complexity (3.3.1)
$d_F(\cdot,\cdot)$	Distance *implied* by F (3.3.1,3.3.2)
D_p	$D_{\|\cdot\|_p^2}$ (5.3.1)
$\mathrm{proj}_F(\mathcal{U};\cdot)$	Projection onto $\mathcal{U}$ operator (5.3.1)

Probability

$\mathbb{P}(A)$	The probability of event A
$\mathbb{P}_{\mathcal{S}}(\cdot)$	Probability over the draw of $\mathcal{S}$
$\mathbb{E}[\cdot], \mathbb{E}_X[\cdot]$	Expectation, expectation w.r.t. r.v. X
P, Q	Prior and posterior distributions over hypotheses (2.2)
$\mathrm{KL}(\cdot,\cdot), \mathrm{kl}(\cdot,\cdot)$	KL divergence between distributions and between Bernouilli distributions (2.2)

Complexity

$\mathrm{VC}(\cdot)$	The VC dimension (1.4.1)
$\widehat{\mathcal{R}}_{\mathcal{S}}(\cdot), \mathcal{R}_m(\cdot)$	Empirical Rademacher complexity, Rademacher complexity (3.2)
$\mathcal{R}_m^{\mathrm{ind}}(\cdot), \mathcal{R}_m^{\mathrm{trs}}(\cdot)$	Inductive and transductive Rademacher complexity (3.4.1)

Graphs and related objects

$\mathcal{V}, \mathcal{E}, \mathcal{G} = (\mathcal{V}, \mathcal{E})$	Vertex set, edge set, graph on $\mathcal{V}, \mathcal{E}$ (1.3.1)
$\boldsymbol{L}, \boldsymbol{D}, \boldsymbol{A}$	Graph Laplacian, degree matrix and Adjacency (1.3.1)
$\nabla_{\mathcal{G}}, \mathrm{div}_{\mathcal{G}}$	Gradient and divergence operator on $\mathcal{G}$ (1.3.1)
$\mathcal{P}, \mathcal{P}_{\mathrm{spine}}$	Path graph, spine (4.3)
$\ell(\mathcal{P})$	length of $\mathcal{P}$ (5.2)
$\Phi_{\mathcal{G}}(\cdot)$	Cut functional on $\mathcal{G}$ (4.2.1)
$\mathcal{N}_i$	Neighbourhood of vertex v_i (4.2.2)
$\mathcal{C},\ \mathcal{D} = \{d_i\}_i$	Comb graph and dongles (4.4.3)
$r_{\mathcal{G}},\ r_{\mathcal{G},p}$	Resistance metric, p-resistance (3.3.1,5.2.1)
$R,\ R_p$	Resistance diameter, p-resistance diameter (5.2.1)
$\boldsymbol{\Psi}_{\mathcal{G}}$	Edge map (5.2.1)
$\|\cdot\|_{\boldsymbol{\Psi},p}, \|\cdot\|_{\mathcal{G},p}$	$(\boldsymbol{\Psi}, p)$-seminorms (5.2.1)
$D_{\boldsymbol{\Psi},p}$	$D_{\|\cdot\|^2_{\boldsymbol{\Psi},p}}$ (5.3.1)
$\mathcal{N} = (\mathcal{G}, \mathcal{S}, p)$	p-resistive network (5.4.1)
$P,\ \boldsymbol{I},\ \boldsymbol{J}$	Power, current, flow (5.4.1)

Bibliography

A. Aizerman, E. M. Braverman, and L. I. Rozoner. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837, 1964.

A. Ambroladze, E. Parrado-Hernández, and J. Shawe-Taylor. Tighter pac-bayes bounds. In *NIPS*, pages 9–16, 2006.

M. Anthony and P. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.

N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68 (3):337–404, 1950.

J. Atkins, E. Boman, B. Hendrickson, et al. A spectral algorithm for seriation and the consecutive ones problem. *SIAM Journal on Computing*, 28(1):297–310, 1999.

K. Azuma. Weighted sums of certain dependent random variables. *Tohoku Mathematical Journal*, 68: 357–367, 1967.

M. Balcan and A. Blum. A discriminative model for semi-supervised learning. *JACM*, 57(3), 2010.

M.-F. Balcan and A. Blum. A pac-style model for learning from labeled and unlabeled data. In *COLT*, pages 111–126, 2005.

P. L. Bartlett and S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

P. L. Bartlett, O. Bousquet, and S. Mendelson. Localized rademacher complexities. In *COLT*, pages 44–58, 2002.

J. M. Barzdin and R. V. Frievald. On the prediction of general recursive functions. *Soviet Math. Doklady*, 13:1224–1228, 1972.

H. H. Bauschke and J. M. Borwein. Legendre functions and the method of random bregman projections. *Journal of convex analysis*, 4, 1997.

M. Belkin and P. Niyogi. Semi-supervised learning on riemannian manifolds. *Machine Learning*, 56: 209–239, 2004.

M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

M. Belkin, I. Matveeva, and P. Niyogi. Regularization and semi-supervised learning on large graphs. In *COLT*, pages 624–638, 2004.

M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

S. Ben-David, T. Lu, and D. Pál. Does unlabeled data provably help? worst-case analysis of the sample complexity of semi-supervised learning. In *COLT*, pages 33–44, 2008.

G. M. Benedek and A. Itai. Learnability by fixed distributions. In *COLT*, pages 80–90, 1988.

K. P. Bennett and A. Demiriz. Semi-supervised support vector machines. In *NIPS*, pages 368–374, 1998.

T. D. Bie and N. Cristianini. Convex methods for transduction. In *NIPS*, pages 73–80, 2003.

T. D. Bie, J. A. K. Suykens, and B. D. Moor. Learning from general label constraints. In *SSPR/SPR*, pages 671–679, 2004.

P. Billingsley. *Probability and measure*. Wiley, New York, NY, USA, 1995.

G. Blanchard and F. Fleuret. Occam's hammer. In *COLT*, pages 112–126, 2007.

A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *ICML*, pages 19–26, 2001.

A. Blum and J. Langford. Pac-mdl bounds. In *COLT*, pages 344–357, 2003.

A. Blum and T. M. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT*, pages 92–100, 1998.

A. Blum, J. D. Lafferty, M. R. Rwebangira, and R. Reddy. Semi-supervised learning using randomized mincuts. In *ICML*, 2004.

B. Bollobas. *Modern Graph Theory*. Springer, 1998.

D. Bonchev and D. H. Rouvray. *Chemical Graph Theory. Introduction and Fundamentals*. Gordon and Breach, 1991.

S. Boucheron, O. Bousquet, and G. Lugosi. Theory of classification: A survey of some recent advances. *ESAIM: Probability and Statistics*, 9:323 – 375, 2005.

C. Bouman and K. Sauer. A generalized gaussian image model for edge-preserving map estimation. *IEEE Trans. on Image Processing*, 2:296–310, 1993.

O. Bousquet and A. Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2: 499–526, 2002.

O. Bousquet, S. Boucheron, and G. Lugosi. Introduction to statistical learning theory. In *Advanced Lectures on Machine Learning*, pages 169–207, 2003a.

O. Bousquet, O. Chapelle, and M. Hein. Measure based regularization. In *NIPS*, 2003b.

L. M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7:200–217, 1967.

T. Bühler and M. Hein. Spectral clustering based on the graph p-laplacian. In *ICML*, pages 11–18, 2009.

V. Castelli and T. M. Cover. On the exponential value of labeled samples. *Pattern Recognition Letters*, 16(1):105–111, 1995.

V. Castelli and T. M. Cover. The relative value of labeled and unlabeled samples in pattern recognition with an unknown mixing parameter. *IEEE Transactions on Information Theory*, 42(6):2102–2117, 1996.

O. Catoni. *PAC-Bayesian surpevised classification: the thermodynamics of statistical learning*. Monograph series of the Institute of Mathematical Statistics, December 2007.

N. Cesa-Bianchi and C. Gentile. Improved risk tail bounds for on-line algorithms. *IEEE Transactions on Information Theory*, 54(1):386–390, 2008.

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, New York, NY, USA, 2006.

N. Cesa-Bianchi, Y. Freund, D. Haussler, D. P. Helmbold, R. E. Schapire, and M. K. Warmuth. How to use expert advice. *J. ACM*, 44(3):427–485, 1997.

N. Cesa-Bianchi, A. Conconi, and C. Gentile. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, 50:2050–2057, 2001.

N. Cesa-Bianchi, C. Gentile, and F. Vitale. Learning unknown graphs. In *ALT*, pages 110–125, 2009a.

N. Cesa-Bianchi, C. Gentile, and F. Vitale. Fast and optimal prediction on a labeled tree. *COLT*, 2009b.

N. Cesa-Bianchi, C. Gentile, F. Vitale, and G. Zappella. Random spanning trees and the prediction of weighted graphs. In *ICML*, pages 175–182, 2010.

O. Chapelle and A. Zien. Semi-supervised classification by low density separation. In *AISTATS*, 2005.

O. Chapelle, J. Weston, and B. Schölkopf. Cluster kernels for semi-supervised learning. In *NIPS*, pages 585–592, 2002.

O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2006.

F. Chung and L. Lu. *Complex Graphs and Networks*. American Mathematical Society, 2006.

F. R. K. Chung. *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92)*. American Mathematical Society, February 1997.

R. R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1): 5–30, July 2006.

M. Collins and N. Duffy. Convolution kernels for natural language. In *NIPS*, 2001.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley, 2 edition, 1991.

N. Cristianini and J. Shawe-Taylor. *An introduction to support vector machines : and other kernel-based learning methods*. Cambridge University Press, 2000.

I. Csiszar. i-divergence geometry of probability distributions and minimization problems. *The Annals of Probability*, 3(1):146–158, 1975.

F. Cucker and S. Smale. On the mathematical foundations of learning. *Bulletin of the American Mathematical Society*, 39:1–49, 2002.

G. Da Prato. *An Introduction to Infinite-Dimensional Analysis*. Springer, 2006.

P. Derbeko, R. El-Yaniv, and R. Meir. Explicit learning curves for transduction and application to clustering and compression algorithms. *Journal of Artificial Intelligence Research*, 22:2004, 2004.

L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition (Stochastic Modelling and Applied Probability)*. Springer, 1996.

I. S. Dhillon and J. A. Tropp. Matrix nearness problems with bregman divergences. *SIAM J. Matrix Anal. Appl.*, 29(4):1120–1146, 2008.

R. Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer-Verlag, Heidelberg, third edition, 2005.

P. G. Doyle and J. L. Snell. *Random Walks and Electric Networks*. Mathematical Association of America, 2000.

R. Durrett. *Random Graph Dynamics (Cambridge Series in Statistical and Probabilistic Mathematics)*. Cambridge University Press, New York, NY, USA, 2006.

R. El-Yaniv and D. Pechyony. Transductive rademacher complexity and its applications. In *COLT*, pages 157–171, 2007.

A. P. Eriksson, C. Olsson, and F. Kahl. Normalized cuts revisited: A reformulation for segmentation with linear grouping constraints. In *ICCV*, pages 1–8, 2007.

J. Fakcharoenphol and B. Kijsirikul. Low congestion online routing and an improved mistake bound for online prediction of graph labeling. *CoRR*, abs/0809.2075, 2008.

M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(98):298–305, 1973.

B. A. Frigyik, S. Srivastava, and M. R. Gupta. Functional bregman divergence and bayesian estimation of distributions. *IEEE Transactions on Information Theory*, 54:51305139, 2008.

C. Gentile. The robustness of the p-norm algorithms. *Machine Learning*, 53(3):265–299, 2003.

P. Germain, A. Lacasse, F. Laviolette, and M. Marchand. Pac-bayesian learning of linear classifiers. In *ICML*, page 45, 2009.

T. Graepel, R. Herbrich, and J. Shawe-Taylor. Pac-bayesian compression bounds on the prediction error of learning algorithms for classification. *Machine Learning*, 59(1-2):55–76, 2005.

K. Gremban, G. Miller, and M. Zagha. Performance evaluation of a new parallel preconditioner. *Parallel Processing Symposium, International*, 0:65, 1995.

G. R. Grimmett. A theorem about random fields. *Bulletin of the London Mathematical Society*, (5): 81–84, 1973.

A. J. Grove, N. Littlestone, and D. Schuurmans. General convergence results for linear discriminant updates. In *COLT*, pages 171–183, 1997.

K. Hall. An r-dimensional quadratic placement algorithm. *Management Science*, 17(3):219–229, 1970.

S. Hanneke. An analysis of graph cut size for transductive learning. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, 2006.

Z. Harchaoui. Image classification with segmentation graph kernels. In *In Proc. CVPR*, 2007.

D. Haussler and A. Barron. How well does the bayes method work for online prediction of $\{-1, +1\}$ values? In *Proceedings of 3rd NEC Symposium*, pages 74–100. Siam, 1993.

M. Hein. Uniform convergence of adaptive graph-based regularization. In *COLT*, pages 50–64, 2006.

M. Hein, J.-Y. Audibert, and U. von Luxburg. Graph laplacians and their convergence on random neighborhood graphs. *Journal of Machine Learning Research*, 8:1325–1370, 2007.

J. Heinonen, T. Kilpeläinen, and O. Martio. *Nonlinear Potential Theory of Degenerate Elliptic Equations*. Oxford University Press, 1993.

M. Herbster. Exploiting cluster-structure to predict the labeling of a graph. In *ALT*, pages 54–69, 2008.

M. Herbster and G. Lever. Predicting the labelling of a graph via p-norm interpolation. In *COLT*, 2009.

M. Herbster and M. Pontil. Prediction on a graph with a perceptron. In *NIPS*, pages 577–584. MIT Press, 2007.

M. Herbster, M. Pontil, and L. Wainer. Online learning over graphs. In *ICML*, pages 305–312, 2005.

M. Herbster, G. Lever, and M. Pontil. Online prediction on large diameter graphs. In *NIPS*, pages 649–656, 2008.

D. Hsu. On container width and length in graphs, groups, and networks. *IEICE Trans. Fundamental of Electronics, Comm., and Computer Sciences*, A(4):668–680, 1994.

M. Jerrum and A. Sinclair. Polynomial-time approximation algorithms for the ising model. *SIAM J. Comput.*, 22(5):1087–1116, 1993.

T. Joachims. Transductive inference for text classification using support vector machines. In *ICML*, pages 200–209, 1999.

T. Joachims. Transductive learning via spectral graph partitioning. In *ICML*, pages 290–297, 2003.

M. Kääriäinen. Relating the rademacher and vc bounds. In *Technical Report, University of Helsinki*.

S. M. Kakade, K. Sridharan, and A. Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In *NIPS*, pages 793–800, 2008.

S. M. Kakade, S. Shalev-Shwartz, and A. Tewari. Applications of strong convexity–strong smoothness duality to learning with matrices. *CoRR*, abs/0910.0610, 2009.

O. Kallenberg and R. Sztencel. Some dimension-free features of vector-valued martingales. *Probability Theory and Related Fields*, 88:215–247, 1991.

R. Kinderman and J. L. Snell. *Markov Random Fields and Their Applications*. Amer. Math. Soc., Providence, RI, 1980.

J. Kivinen, M. K. Warmuth, and P. Auer. The perceptron algorithm versus winnow: Linear versus logarithmic mistake bounds when few input variables are relevant (technical note). *Artif. Intell.*, 97 (1-2):325–343, 1997.

D. Klein and M. Randić. Resistance distance. *Journal of Mathematical Chemistry*, 12(1):81–95, 1993.

J. M. Kleinberg, M. Sandler, and A. Slivkins. Network failure detection and graph connectivity. In *SODA*, pages 76–85, 2004.

V. Koltchinskii and D. Y. Panchenko. Empirical margin distributions and bounding the generalization error of combined classifiers. *Annals of Statistics*, 30:2002, 2000.

R. I. Kondor and J. D. Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *ICML*, pages 315–322, 2002.

M. Krein. Hermitian-positive kernels on homogeneous spaces. *American Mathematical Society Translations series 2*, 34:69–164, 1963.

R. Kumar, J. Novak, and A. Tomkins. Structure and evolution of online social networks. In *KDD*, pages 611–617, 2006.

J. Langford. Tutorial on practical prediction theory for classification. *Journal of Machine Learning Research*, 6:273–306, 2005.

J. Langford and J. Shawe-taylor. Pac-bayes and margins. In *NIPS*, pages 439–446, 2002.

S. L. Lauritzen. *Graphical Models (Oxford Statistical Science Series)*. Oxford University Press, 1996.

Y. Lecun and C. Cortes. The mnist database of handwritten digits. URL `http://yann.lecun.com/exdb/mnist/`.

M. Ledoux and M. Talagrand. *Probability in Banach Spaces*. Springer, 1991.

G. Lever. Relating function class complexity and cluster structure in the function domain with applications to transduction. *AISTATS*, 2010.

G. Lever, F. Laviolette, and J. Shawe-Taylor. Distribution dependent pac-bayes priors. *ALT*, 2010.

N. Littlestone. From on-line to batch learning. In *COLT*, pages 269–284, 1989.

N. Littlestone. Learning when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1988.

N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Inf. Comput.*, 108(2):212–261, 1994.

N. Littlestone and M. K. Warmuth. The weighted majority algorithm. In *IEEE Symposium on Foundations of Computer Science*, pages 256–261, 1989.

D. J. C. Mackay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, 2002.

A. Maurer. A note on the pac bayesian theorem. *CoRR*, cs.LG/0411099, 2004.

D. A. McAllester. Pac-bayesian model averaging. In *COLT*, pages 164–170, 1999.

R. Meir and T. Zhang. Generalization error bounds for bayesian mixture algorithms. *Journal of Machine Learning Research*, 4:839–860, 2003.

J. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Philos. Trans. Roy. Soc. London*, 1909.

M. Minsky and S. Papert. *Perceptrons; an introduction to computational geometry*. Cambridge, MA: MIT Press, 1969.

B. Nadler, S. Lafon, R. R. Coifman, and I. G. Kevrekidis. Diffusion maps, spectral clustering and eigenfunctions of fokker-planck operators. In *NIPS*, 2005.

K. Nigam, A. McCallum, S. Thrun, and T. M. Mitchell. Learning to classify text from labeled and unlabeled documents. In *AAAI/IAAI*, pages 792–799, 1998.

A. B. Novikoff. On convergence proofs for perceptrons. In *Proceedings of the Symposium on the Mathematical Theory of Automata*, volume 12, pages 615–622, 1963.

K. Pelckmans and J. A. Suykens. An online algorithm for learning a labeling of a graph. In *MLG*, 2008.

K. Pelckmans and J. A. K. Suykens. Transductive rademacher complexities for learning over a graph. In *MLG*, 2007.

K. Pelckmans, J. Shawe-Taylor, J. A. K. Suykens, and B. D. Moor. Margin based transductive graph cuts using linear programming. In *AISTATS*, 2007.

L. Ralaivola, M. Szafranski, and G. Stempfel. Chromatic pac-bayes bounds for non-iid data: Applications to ranking and stationary β-mixing processes. *Journal of Machine Learning Research*, 11: 1927–1956, 2010.

J. Ratsaby and S. S. Venkatesh. Learning from a mixture of labeled and unlabeled examples with parametric side information. In *COLT*, pages 412–417, 1995.

R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1972.

F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psych. Rev.*, 65:386–407, 1958. (Reprinted in *Neurocomputing* (MIT Press, 1988).).

S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

B. Schölkopf and A. J. Smola. *Learning with kernels : support vector machines, regularization, optimization, and beyond*. Adaptive computation and machine learning. MIT Press, 2002.

M. Seeger. Pac-bayesian generalisation error bounds for gaussian process classification. *Journal of Machine Learning Research*, 3:233–269, 2002.

R. Serfling. *Approximation Theorems of Mathematical Statistics*. Wiley, 1980.

R. J. Serfling. Probability inequalities for the sum in sampling without replacement. *The Annals of Statistics*, 2(1):39–48, 1974.

S. Shalev-Shwartz. *Online Learning: Theory, Algorithms and Applications*. PhD thesis, The Hebrew University, 2007.

R. Sharan and T. Ideker. Modeling cellular machinery through biological network comparison. *Nature biotechnology*, 24(4):427–433, 2006.

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

J. Shawe-Taylor, P. L. Bartlett, R. C. Williamson, and M. Anthony. Structural risk minimization over data-dependent hierarchies. *IEEE Transactions on Information Theory*, 44(5):1926–1940, 1998.

J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

V. Sindhwani, P. Niyogi, and M. Belkin. Beyond the point cloud: from transductive to semi-supervised learning. In *ICML*, pages 824–831, 2005.

D. Singaraju, L. Grady, and R. Vidal. P-brush: Continuous valued mrfs with normed pairwise distributions for image segmentation. In *CVPR*, pages 1303–1310, 2009.

A. J. Smola and R. I. Kondor. Kernels and regularization on graphs. In *COLT*, pages 144–158, 2003.

A. J. Smola, B. Schölkopf, and K.-R. Müller. The connection between regularization operators and support vector kernels. *Neural Netw.*, 11(4):637–649, 1998.

I. Steinwart and A. Christmann. *Support Vector Machines*. Springer, 2008.

J. B. Tenenbaum, V. Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-Posed Problems*. V. H. Winston & Sons, Washington, D.C.: John Wiley & Sons, New York,, 1977.

L. G. Valiant. A theory of the learnable. In *STOC*, pages 436–445, 1984.

V. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer, 1982.

V. N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, September 1998.

V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probab. and its Applications*, 16(2):264–280, 1971.

V. G. Vovk. Aggregating strategies. In *COLT*, pages 371–386, 1990.

G. Wahba. *Spline models for observational data*. SIAM, 1990.

T. Washio and H. Motoda. State of the art of graph-based data mining. *SIGKDD Explor. Newsl.*, 5(1): 59–68, 2003.

K. Q. Weinberger and L. K. Saul. Unsupervised learning of image manifolds by semidefinite programming. In *CVPR (2)*, pages 988–995, 2004.

D. H. Wolpert. The lack of a priori distinctions between learning algorithms. *Neural Comput.*, 8(7): 1341–1390, 1996.

Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(11), 1993.

J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. Technical report, Mitsubishi Electric Research Laboratories, 2003.

C. Zălinescu. *Convex analysis in general vector spaces*. World Scientific, 2002.

T. Zhang and F. J. Oles. A probability analysis on the value of unlabeled data for classification problems. In *ICML*, pages 1191–1198, 2000.

D. Zhou and B. Schölkopf. Regularization on discrete spaces. In *Pattern Recognition*, pages 361–368. Springer, 2005.

D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *NIPS*, 2003.

X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, pages 912–919, 2003a.

X. Zhu, J. Lafferty, and Z. Ghahramani. Semi-supervised learning: From gaussian fields to gaussian processes. Technical report, School of CS, CMU, 2003b.

X. Zhu, J. S. Kandola, Z. Ghahramani, and J. D. Lafferty. Nonparametric transforms of graph kernels for semi-supervised learning. In *NIPS*, 2004.

Appendix A

Kernels and Green's functions

Given a linear operator equation,

$$Af(x) = g(x), \tag{A.1}$$

we seek a function $G(\cdot, \cdot)$ such that, for all $x \in \mathcal{X}$

$$\int_{\mathcal{X}} g(x')G(x', x)\nu(\mathrm{d}x') = f(x).$$

That is, G provides an integral operator representation of the inverse function $A^{-1} : \mathcal{F} \to \mathcal{F}$. If it exists, such a function G is called the Green's function for the operator A (or the equation (A.1)). Denote $G_x(\cdot) := G(\cdot, x)$ then

$$\begin{aligned} f(x) &= \langle g, G_x \rangle_{\mathcal{L}^2} \\ &= \langle Af, G_x \rangle_{\mathcal{L}^2} \\ &= \langle f, A^* G_x \rangle_{\mathcal{L}^2}, \end{aligned}$$

where A^* represents the adjoint of A. In other words,

$$(A^* G_x)(z) = \delta(x, z). \tag{A.2}$$

Equation (A.2) is usually the definition of the Green's function for A^*.

We specialize to the case where A is a self-adjoint linear operator $A_R := R^*R : \mathcal{L}^2(\mathcal{X}, \Sigma, \nu) \to \mathcal{L}^2(\mathcal{X}, \Sigma, \nu)$, where R is some linear operator[1] such that $\mathrm{reg} : h \to \langle h, A_R h \rangle_{\mathcal{L}^2} = \langle Rh, Rh \rangle$ is a regularization operator in the sense of Tikhonov and Arsenin (1977). The Green's function $G(\cdot, \cdot)$ is the function which satisfies $\delta(x, z) = (A_R G_x)(z)$ so it is seen that G plays the role analogue to that of a matrix right inverse for operators on a Hilbert space. It is shown by Smola et al. (1998) that G is a Hilbert-Schmidt kernel and we have the following result:

Claim A.0.1. *Let $\mathcal{H}_K := \overline{\mathrm{span}\{K(x, \cdot)\}_{x \in \mathcal{X}}}$ be an RKHS whose Mercer kernel K is the Green's function for the regularization operator A_R. Then the inner product $\langle \cdot, \cdot \rangle_K$ in $\mathcal{H}_K$ (on finitely generated[2]*

[1] For example R might be the gradient operator.

[2] i.e. elements $h \in \mathcal{H}_K$ such that $h(\cdot) := \sum_{i=1}^{m} \alpha_i K(x_i, \cdot)$ for finite m.

elements of $\mathcal{H}_K$) can be represented as the bilinear form

$$\begin{aligned}\langle h, g\rangle_K &= \langle h, A_R g\rangle_{\mathcal{L}^2} \\ &= \int_{\mathcal{X}} h(x)(A_R g)(x)\nu(\mathrm{d}x).\end{aligned}$$

Proof. Let,

$$\begin{aligned}h(\cdot) &:= \sum_{i=1}^{m} \alpha_i K(x_i, \cdot) \\ g(\cdot) &:= \sum_{i=1}^{n} \beta_i K(x_i, \cdot).\end{aligned}$$

Then,

$$\begin{aligned}\int_{\mathcal{X}} h(x)(A_R g)(x)\nu(\mathrm{d}x) &= \int_{\mathcal{X}} \sum_{i,j} \alpha_i\beta_j K(x_i, x)(A_R K(x_j, \cdot))(x)\nu(\mathrm{d}x) \\ &= \int_{\mathcal{X}} \sum_{i,j} \alpha_i\beta_j K(x_i, x)(A_R G_{x_j})(x)\nu(\mathrm{d}x) \\ &= \int_{\mathcal{X}} \sum_{i,j} \alpha_i\beta_j K(x_i, x)\delta(x_j, x)\nu(\mathrm{d}x) \\ &= \sum_{i,j} \alpha_i\beta_j K(x_i, x_j) \\ &= \langle h, g\rangle_K.\end{aligned}$$

□

Corollary A.0.2. *The RKHS norm $||\cdot||_K$ (on finitely generated elements of $\mathcal{H}_K$) can be represented in the following 'regularizer' form,*

$$||h||_K^2 = \langle h, A_R h\rangle_{\mathcal{L}^2}. \tag{A.3}$$

Often, the above argument can be reversed: the natural $\mathcal{L}^2$ regularizer corresponding to an RKHS norm is the Green's function of the integral operator (1.8) corresponding to the kernel.

Appendix B

Technical lemmas

Theorem B.0.3. *(Bouman and Sauer, 1993, Theorem 1) Let $\mathcal{U}$ and $\mathcal{V}$ be Euclidean metric spaces. Let $f(\cdot,\cdot)$ be a continuous functional $f : \mathcal{U} \times \mathcal{V} \to \mathbb{R}$ such that for all $v \in \mathcal{V}$, $f(\cdot, v)$ is strictly convex with a local minimum. Then,*

$$\operatorname*{argmin}_{u\in\mathcal{U}} f(u, v),$$

is a unique and continuous function of v.

Lemma B.0.4. *(Hoeffding's lemma) Let X be a random variable with $\mathbb{E}[X] = 0$ and $a < X < b$ then for $t > 0$,*

$$\mathbb{E}[e^{tX}] \leq e^{\frac{t^2(b-a)^2}{8}}.$$

The following theorem demonstrates that many key properties of martingales are independent of their dimension. The authors note that it is true for any Hilbert space-valued martingale but the proof is just for martingales in ℓ^2.

Theorem B.0.5. *(Kallenberg and Sztencel, 1991, Theorem 3.1) Let $\{V_t\}$ be a martingale in $\mathbb{R}^d$ or ℓ^2. Then there exists a martingale $\{U_t\}$ in $\mathbb{R}^2$ such that $||V_t|| = ||U_t||$ $a.s.$ and $||V_t - V_{t-1}|| = ||U_t - U_{t-1}||$ $a.s.$.*

Given the above result all that we must do to obtain a large deviation inequality for ℓ^2-valued martingales is to demonstrate a variation of Azuma-Hoeffding inequality for a martingale in $\mathbb{R}^2$, which is elementary if we are not concerned with obtaining the best constants.

Corollary B.0.6. *For a martingale $\{V_i\}_{i=1}^m$ in $\mathbb{R}^d$ or ℓ^2, such that, for all i,*

$$||V_i - V_{i-1}|| \leq c_i,$$

we have for all $\delta > 0$,

$$\mathbb{P}\left(||V_m - V_0|| \leq 2\sqrt{\sum_{i=1}^m c_i^2 \ln\frac{4}{\delta}}\right) \geq 1 - \delta.$$

Proof. Consider a martingale $\{U_i\}_{i=1}^m$ in $\mathbb{R}^2$ such that,

$$||U_i - U_{i-1}|| \leq c_i. \tag{B.1}$$

Let $U_i = (U_i^{(1)}, U_i^{(2)})$, so that we have that $\{U_i^{(1)}\}_{i=1}^n$ and $\{U_i^{(2)}\}_{i=1}^n$ are clearly martingales and that,

$$|U_i^{(1)} - U_{i-1}^{(1)}| \leq c_i$$
$$|U_i^{(2)} - U_{i-1}^{(2)}| \leq c_i.$$

Now,

$$\begin{aligned}
\mathbb{P}\left(||U_m - U_0|| \geq \epsilon\right) &= \mathbb{P}\left((U_m^{(1)} - U_0^{(1)})^2 + (U_m^{(2)} - U_0^{(2)})^2 \geq \epsilon^2\right) \\
&\leq \mathbb{P}\left(|U_m^{(1)} - U_0^{(1)}| \geq \frac{\epsilon}{\sqrt{2}}\right) + \mathbb{P}\left(|U_m^{(2)} - U_0^{(2)}| \geq \frac{\epsilon}{\sqrt{2}}\right) \\
&\leq 4\exp\left(-\frac{\epsilon^2}{4\sum_{i=1}^m c_i^2}\right),
\end{aligned}$$

where the last line follows by the Hoeffding-Azuma inequality (e.g. Azuma, 1967). The result then follows by theorem B.0.5. □

Appendix C

Gaussian measures on infinite-dimensional Hilbert space

We recall some fundamental facts about Gaussian measures on infinite-dimensional Hilbert space. These results are the focus of (Da Prato, 2006, chapter 1) and here we just sketch the main ideas.

Let $\mathcal{H}$ be a Hilbert space with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ and M a self-adjoint positive-definite compact operator on $\mathcal{H}$. Thus M provides a countable orthonormal basis $\{\phi_i\}$ for $\mathcal{H}$, comprising its eigenfunctions, with corresponding eigenvalues λ_i. Suppose further that M is of trace class, $\sum_{i=1}^{\infty} \lambda_i < \infty$. Define $h_i := \langle h, \phi_i \rangle_{\mathcal{H}}$ and the isomorphism $I : \mathcal{H} \to \ell^2$ by $I(h) = (h_i)$.

We define by N_{a_i,λ_i} the Gaussian measure on the Borel σ-algebra $\mathcal{B}(\mathbb{R})$ of $\mathbb{R}$ with mean a_i and variance λ_i. Define,

$$N_{a,M} := \prod_{i=1}^{\infty} N_{a_i,\lambda_i}. \tag{C.1}$$

Theorem C.0.7. *(Da Prato, 2006, Theorem 1.9) $N_{a,M}$ is a probability measure on $\mathbb{R}^{\infty}$.*

Proof. (Sketch.) The measure is first defined on the ring[1] $\mathcal{C}$ of *cylinder sets*, defined for each $\mathcal{A} \in \mathcal{B}(\mathbb{R}^n)$ by $I_{n,\mathcal{A}} = \{(x_k) \in \mathbb{R}^{\infty} \ : \ (x_1, ...x_n) \in \mathcal{A}\}$ which contain the whole of $\mathbb{R}^{\infty}$ in all but finitely many dimensions (and correspond to a Borel measurable set in those dimensions). The product $N_{a,M}$ can be shown to be σ-additive on $\mathcal{C}$ and so, by Carathéodory's extension theorem (e.g. Billingsley (1995)) $N_{a,M}$ extends uniquely to a probability measure on the σ-algebra generated by $\mathcal{C}$ which is shown to be $\mathcal{B}(\mathbb{R}^{\infty})$ (the Borel σ-algebra generated by the product topology). □

[1] Meaning, in this case, a set $\mathcal{R}$ such that,

$$\emptyset \in \mathcal{R}$$
$$A, B \in \mathcal{R} \Rightarrow B \backslash A \in \mathcal{R}$$
$$A, B \in \mathcal{R} \Rightarrow A \cup B \in \mathcal{R}.$$

The following two results show that although defined on $\mathbb{R}^\infty$ the support of $N_{a,M}$ is precisely ℓ^2, i.e. $\mathcal{H}$ under the above isomorphism. Thus we refer to $N_{a,M}$ as a measure on ℓ^2 and also on $\mathcal{H}$ via isomorphism.

Theorem C.0.8. *(Da Prato, 2006, Proposition 1.11)* $N_{a,M}(\ell^2) = 1$.

Theorem C.0.9. *(Da Prato, 2006, Proposition 1.25) Let $\mathcal{A}$ be any non-empty open subset of $\mathcal{H}$, then* $N_{a,M}(\mathcal{A}) > 0$.

We denote by $M^{\frac{1}{2}}$, $M^{-\frac{1}{2}}$, M^{-1}, the operators on $\mathcal{H}$ such that $M^{\frac{1}{2}}\phi_i = \lambda^{\frac{1}{2}}\phi_i$, $M^{-\frac{1}{2}}\phi_i = \lambda^{-\frac{1}{2}}\phi_i$, $M^{-1}\phi_i = \lambda^{-1}\phi_i$ (which are not necessarily continuous). The following is attributed as a particular version of the Cameron-Martin formula:

Theorem C.0.10. *(Da Prato, 2006, Theorem 2.8)*

(a) If $a \notin M^{\frac{1}{2}}(\mathcal{H})$ then $N_{a,M}$ and $N_{0,M}$ are singular.

(b) If $a \in M^{\frac{1}{2}}(\mathcal{H})$ then $N_{a,M}$ and $N_{0,M}$ are equivalent.

(c) If $N_{a,M}$ and $N_{0,M}$ are equivalent[4] then the Radon-Nikodym derivative is given by,

$$\frac{\mathrm{d}N_{a,M}}{\mathrm{d}N_{0,M}}(h) = \exp\left(\langle h, M^{-1}a\rangle_{\mathcal{H}} - \frac{1}{2}||M^{-\frac{1}{2}}a||^2_{\mathcal{H}}\right),$$

where equality is as a function in $\mathcal{L}^1(\mathcal{H}, N_{0,M})$.

Appendix D

The Karhunen-Loève theorem applied to a Gaussian process

We recall the Karhunen-Loève expansion of a Gaussian process (e.g. Wahba, 1990, page 5). Consider any zero-mean Gaussian process $\{G_{\boldsymbol{x}}\}_{\boldsymbol{x}\in\mathcal{X}}$ on a measure space $(\mathcal{X}, \Sigma, \nu)$ with covariance $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Suppose that K is a Mercer kernel and therefore has expansion,

$$K(\boldsymbol{x}, \boldsymbol{x}') = \sum_{i=1}^{\infty} \lambda_i \phi_i(\boldsymbol{x}) \phi_i(\boldsymbol{x}'),$$

where $\{\phi_i\}_{i=1}^{\infty} \subset \mathcal{L}^2(\mathcal{X}, \Sigma, \mathcal{X})$ are the eigenfunctions and $\{\lambda_i\}$ eigenvalues of the corresponding integral operator A_K on $\mathcal{L}^2(\mathcal{X}, \Sigma, \nu)$ i.e. such that,

$$A_K \phi_i(\boldsymbol{x}) = \int_{\mathcal{X}} K(\boldsymbol{x}, \boldsymbol{x}') \phi_i(\boldsymbol{x}') \nu(\mathrm{d}\boldsymbol{x}') = \lambda_i \phi_i(\boldsymbol{x}) \qquad \forall \boldsymbol{x} \in \mathcal{X}.$$

Under certain conditions on $(\mathcal{X}, \Sigma, \nu)$ – for example, when $\mathcal{X}$ is compact and ν a finite Borel measure – $G_{\boldsymbol{x}}$ has (quadratic mean) representation,

$$G_{\boldsymbol{x}} = \sum_{i=1}^{\infty} V_i \phi_i(\boldsymbol{x}), \tag{D.1}$$

where V_i are independent zero-mean Gaussian random variables with $\mathbb{E}[V_i^2] = \lambda_i$. Convergence in (D.1) is in the quadratic mean (and so also in probability and distribution), and uniformly over $\mathcal{X}$, i.e.,

$$\mathbb{E}\left[\sup_{\boldsymbol{x}\in\mathcal{X}} \left(G_{\boldsymbol{x}} - \sum_{i=1}^{n} V_i \phi_i(\boldsymbol{x}) \right)^2 \right] \to 0, \tag{D.2}$$

as $n \to \infty$. Note that the Gaussian process $\{G_{\boldsymbol{x}}\}_{\boldsymbol{x}\in\mathcal{X}}$ corresponds to a distribution on the function space $\mathcal{L}^2(\mathcal{X}, \Sigma, \nu)$ which is in general much bigger than the RKHS $\mathcal{H}_K$ whenever more than a finite number of the λ_i are non-zero.

Appendix E

Convex analysis in general vector spaces

We recall some basic definitions and results from convex analysis. Some of the following concepts hold in more general settings but throughout let $\mathcal{V}$ be a normed vector space over the field of real numbers and denote by $\mathcal{V}^*$ the continuous dual space of continuous linear functionals on $\mathcal{V}$ and denote by $\langle \cdot, \cdot \rangle : \mathcal{V}^* \times \mathcal{V} \to \mathbb{R}$ the dual pairing.

Definition For a function $f : \mathcal{V} \to \mathbb{R} \cup \{+\infty\}$ we define the *convex (or Legendre-Fenchel) conjugate* $f^\star : \mathcal{V}^* \to \mathbb{R} \cup \{+\infty\}$,

$$f^\star(v^*) := \sup_{u \in \mathcal{V}} \langle v^*, u \rangle - f(u).$$

The Fenchel-Young inequality $\langle v^*, u \rangle \leq f^\star(v^*) + f(u)$ is an immediate consequence of the definition. Note that when $\mathcal{V}$ is a real Hilbert space (as is generally the case throughout this thesis) the continuous dual space is isometrically isomorphic to $\mathcal{V}$ and the dual pairing $\langle \cdot, \cdot \rangle$ can be identified with the Hilbert space inner product. An important case is the fact that the Legendre-Fenchel conjugate of a half norm squared is a half of the dual norm squared, i.e. if $f(\cdot) = \frac{1}{2}|| \cdot ||^2$ then $f^\star(\cdot) = \frac{1}{2} \left(|| \cdot ||_* \right)^2$.

Definition Given a convex function $f : \mathcal{V} \to \mathbb{R} \cup \{+\infty\}$ any element $v^* \in \mathcal{V}^*$ which satisfies,

$$\forall u \in \mathcal{V} : \qquad \langle u - v, v^* \rangle \leq f(u) - f(v),$$

is called a *subgradient* of f at v. The *subdifferential* $\partial f(v)$ of f at v is the set of all subgradients of f at v.

In particular if f is differentiable at v with derivative $\nabla f(v)$ then $\partial f(v) = \{\nabla f(v)\}$. In the following it is understood that $\mathcal{V}$ is a Banach space.

Definition A function $f : \mathcal{V} \to \mathbb{R}$ is κ-*strongly convex* w.r.t. a norm $|| \cdot ||$ on $\mathcal{V}$ if for all $u, v \in \mathcal{V}$ and $v^* \in \partial f(v)$ we have,

$$f(u) - f(v) - \langle v^*, u - v \rangle \geq \frac{\kappa}{2} ||u - v||^2.$$

It is not necessary for a function to be differentiable to be strongly convex, and strong convexity can be equivalently defined as follows (see e.g. (Shalev-Shwartz, 2007, Lemma 13) for the simpler case of finite dimensional spaces and (Zălinescu, 2002, Corollary 3.5.11) for the general case):

Lemma E.0.11. *A function $f : \mathcal{V} \to \mathbb{R}$ is κ-strongly convex w.r.t. a norm $||\cdot||$ on $\mathcal{V}$ if for all $u, v \in \mathcal{V}$ in the relative interior of the domain of f we have,*

$$f(\alpha u + (1-\alpha)v) \leq \alpha f(u) + (1-\alpha)f(v) - \frac{\kappa}{2}\alpha(1-\alpha)||u-v||^2.$$

Definition A function $f : \mathcal{V} \to \mathbb{R}$ is *κ-strongly smooth* w.r.t. a norm $||\cdot||$ on $\mathcal{V}$ if for all $u, v \in \mathcal{V}$ and $v^* \in \partial f(v)$ we have,

$$f(u) - f(v) - \langle v^*, u - v\rangle \leq \frac{\kappa}{2}||u-v||^2.$$

The following result has been of recent interest in the learning theory community. The proof can be found in (Zălinescu, 2002, Corollary 3.5.11) (see the equivalence of statements (i), (iii) and (viii) therein and in fact a more general case is studied), an accessible proof of a less general case is presented in Kakade et al. (2009).

Theorem E.0.12. *A lower semicontinuous convex function $f : \mathcal{V} \to \mathbb{R}$ is κ-strongly convex w.r.t. a norm $||\cdot||$ on $\mathcal{V}$ if and only if its Legendre-Fenchel conjugate $f^\star : \mathcal{V}^* \to \mathbb{R}$ is $\frac{1}{\kappa}$-strongly smooth w.r.t. the dual norm $||\cdot||_*$ on $\mathcal{V}^*$.*

Appendix F

Proof of Theorem 5.6.1

We try to maintain the structure and notation of the proof in Cesa-Bianchi et al. (2001) as closely as possible.

Proposition F.0.13. *(Cesa-Bianchi et al., 2001, Proposition 1) Let $\mathcal{H}_{\mathcal{S}} = \{h_0, h_1 ... h_{T-1}\}$ be the ensemble of hypotheses produced by an online algorithm A on a trial sequence $\mathcal{S} = \{(x_1, y_1), ...(x_T, y_T)\}$. For any $\delta \in (0, 1]$ we have,*

$$\mathbb{P}\left(\frac{1}{T}\sum_{t=1}^{T}\overline{\text{risk}}(h_{t-1}) \geq \frac{M(\mathcal{S})}{T} + \sqrt{\frac{2}{T}\ln\frac{1}{\delta}}\right) \leq \delta, \tag{F.1}$$

where $M(\mathcal{S})$ is (any upper bound on) the number of mistakes incurred by A on $\mathcal{S}$.

Proof. Set $S_0 = 0$ and for $1 \leq t \leq T$ set,

$$V_t = \overline{\text{risk}}(h_{t-1}) - \ell_{0-1}(h_{t-1}(X_t), (Y_t))$$
$$S_t = \sum_{i=1}^{t} V_i.$$

Note that $S_T = \sum_{t=1}^{T}\overline{\text{risk}}(h_{t-1}) - M(\mathcal{S})$. Note further that the sequence $\{S_t\}$ is a martingale with respect to the sequence $\{X_t\}$ since for all $1 \leq t \leq T$ we have $|V_t| \leq 1$ and,

$$\begin{aligned}\mathbb{E}[S_t \mid X_1, ...X_{t-1}] &= S_{t-1} + \mathbb{E}[V_t \mid X_1, ...X_{t-1}] \\ &= S_{t-1}.\end{aligned}$$

Thus we can use the Azuma-Hoeffding inequality (Azuma, 1967) to derive,

$$\mathbb{P}\left(\frac{1}{T}\sum_{t=1}^{T}\overline{\text{risk}}(h_{t-1}) \geq \frac{M(\mathcal{S})}{T} + \sqrt{\frac{2}{T}\ln\frac{1}{\delta}}\right) = \mathbb{P}\left(S_T \geq \sqrt{2T\ln\frac{1}{\delta}}\right) \leq \delta.$$

□

Lemma F.0.14. *(Cesa-Bianchi et al., 2001, Lemma 3) Let $\mathcal{H}_{\mathcal{S}} = \{h_0, h_1 ... h_{T-1}\}$ be the ensemble of hypotheses produced by an online algorithm A on a trial sequence $\mathcal{S} = \{(x_1, y_1), ...(x_T, y_T)\}$. For any $\delta \in (0, 1]$ let $\hat{h} = \text{argmin}_{h_t \in \mathcal{H}_{\mathcal{S}}}\{\widehat{\text{risk}}^{(\delta)}(h_t, t+1)\}$. We have,*

$$\mathbb{P}\left(\overline{\text{risk}}(\hat{h}) > \min_{0 \leq t \leq T}\{\overline{\text{risk}}(h_t) + 2c_\delta(T-t)\}\right) \leq \delta. \tag{F.2}$$

Proof. But for the final lines, this is proved as in (Cesa-Bianchi et al., 2001, Lemma 3). Setting $T^* = \mathrm{argmin}_{0\leq t\leq T}\{\overline{\mathrm{risk}}(h_t) + 2c_\delta(T-t)\}$, $h^* = h_{T^*}$ and $R_t = \widehat{\overline{\mathrm{risk}}}(h_t, t+1)$, the inequality,

$$\begin{aligned}\mathbb{P}\left(\overline{\mathrm{risk}}(\hat{h}) > \overline{\mathrm{risk}}(h^*) + 2c_\delta(T-T^*)\}\right) \quad &\leq \quad \sum_{t=0}^{T-1}\mathbb{P}(R_t \leq \overline{\mathrm{risk}}(h_t) - c_\delta(T-t)) \\ &+ \quad T\sum_{t=0}^{T-1}\mathbb{P}(R_t \geq \overline{\mathrm{risk}}(h_t) + c_\delta(T-t)),\end{aligned}$$

can be derived exactly as in Cesa-Bianchi et al. (2001). We then note that $R_t = \frac{1}{T-t}\sum_{i=t+1}^{T}\ell_{0-1}(h_t(X_i), Y_i)$ is not a sum of independent random variables (so we cannot use Chernoff-Hoeffding bounds as in Cesa-Bianchi et al. (2001)). Rather, $\sum_{i=t+1}^{T}\ell_{0-1}(h_t(X_i), Y_i)$ has a hypergeometric distribution. The result then follows by using Serfling's inequality for sums of random variables obtained by sampling uniformly without replacement (Serfling, 1974) [1]. □

Proof of Theorem 5.6.1. Theorem 5.6.1 can now be proved by following (Cesa-Bianchi et al., 2001, Theorem 5), invoking the results Proposition F.0.13 and Lemma F.0.14 in place of their counterparts therein.

[1] If Z has a hypergeometric distribution with $k \geq 1$ draws from a set of size N then $\mathbb{P}(Z \leq \mathbb{E}[Z] - k\epsilon) \leq e^{-2k\epsilon^2\frac{N}{N-k-1}} \leq e^{-2k\epsilon^2}$ and similarly $\mathbb{P}(Z \geq \mathbb{E}[Z] + k\epsilon) \leq e^{-2k\epsilon^2\frac{N}{N-k-1}} \leq e^{-2k\epsilon^2}$.

Appendix G

Structure dependent risk bound and regularization

Theorem 3.3.2 supplies a risk bound in terms of the observed cluster structure in the training sample.

Theorem G.0.15. *Using the notation of Theorem 3.3.2, and when $\ell(\cdot,\cdot)$ is positive and bounded by C, for all $h \in \mathcal{H}$,*

$$\mathbb{P}_{\mathcal{S}}\left(\mathrm{risk}^{\ell}(h) \leq \widehat{\mathrm{risk}}^{\ell}_{\mathcal{S}}(h) + 2K\left(B\sqrt{\frac{|\mathcal{C}|}{m}} + 2\sqrt{\frac{2F'(h)\rho_{\mathcal{S}}}{m\kappa}}\right) + 3C\sqrt{\frac{\log\frac{2}{\delta}}{2m}}\right) \geq 1-\delta.$$

where $F'(h) := \min_{r\in\{1,2,\ldots\}} \max\left(\alpha_r, \frac{r+1}{r}F(h)\right)$ *and* $\alpha_r := \frac{9C^2\kappa r \log 2}{16K^2\rho_{\mathcal{S}}}$.

Proof. Define the stratification: $\mathcal{H}^{(0)} = \{\}$ and, for $t \in \{1,2,\ldots\}$, $\mathcal{H}^{(t)} = \mathcal{H}_{\alpha_t}$. The empirical version of Theorem 3.2.1 (e.g. Boucheron et al., 2005) implies that with probability at least $1-\frac{\delta}{2^t}$ simultaneously for all $h \in \mathcal{H}^{(t)}\backslash\mathcal{H}^{(t-1)}$ we have,

$$\begin{aligned}
\mathrm{risk}^{\ell}(h) - \widehat{\mathrm{risk}}^{\ell}_{\mathcal{S}}(h) &\leq 2K\widehat{\mathcal{R}}_{\mathcal{S}}(\mathcal{H}_{\alpha_t}) + 3C\sqrt{\frac{\log\frac{2^{t+1}}{\delta}}{2m}} \\
&\leq 2K\left(B\sqrt{\frac{|\mathcal{C}|}{m}} + \sqrt{\frac{2\alpha_t\rho_{\mathcal{S}}}{m\kappa}}\right) + 3C\sqrt{\frac{t\log 2}{2m}} + 3C\sqrt{\frac{\log\frac{2}{\delta}}{2m}} \\
&\leq 2K\left(B\sqrt{\frac{|\mathcal{C}|}{m}} + 2\sqrt{\frac{2\alpha_t\rho_{\mathcal{S}}}{m\kappa}}\right) + 3C\sqrt{\frac{\log\frac{2}{\delta}}{2m}}. \qquad \text{(G.1)}
\end{aligned}$$

Now noting that for $r \in \{1,2,\ldots\}$, $\alpha_t > \alpha_r$ implies that $t \geq r+1$ and $\alpha_t \leq \frac{r+1}{r}\alpha_{t-1}$ so

$$\alpha_t \leq \min_{r\in\{1,2,\ldots\}} \max\left(\alpha_r, \frac{r+1}{r}\alpha_{t-1}\right) \leq F'(h) \qquad \text{(G.2)}$$

The result then follows by combining (G.2) with (G.1) and applying the union bound over all $t \in \{1,2,\ldots\}$.

□

Theorem G.0.15 suggests an algorithm: pick the classifier which minimizes the bound. this is simply regularization w.r.t. the complexity $F(\cdot)$ but the regularization parameters are determined by the

observed cluster structure in the data. In principle the information needed to choose the regularization parameter should be encoded in the data, so it would be of interest to understand this relationship and reduce the need for cross validation.

A special case of the above is RKHS regularization, obtained by picking the 1-strongly convex Hilbert space norm as a complexity, $F(h) = \frac{1}{2}||h||_K^2$. The cluster structure in this case is that in feature space.

Appendix H

Proof of Theorem 3.4.3

The theorem is due to Pelckmans and Suykens (2007), but no full proof could be found in the literature so we supply one here. The proof follows the familiar strategy of using a McDiarmid-type inequality followed by the introduction of a ghost sample, requiring a little more manipulation due to the transductive setting.

We require some preliminaries: let $\mathcal{P}$ be the set of all $n!$ permutations of $n = m + u$ objects $\mathcal{Z}$: for each $\boldsymbol{\pi} \in \mathcal{P}$, each π_i is a distinct element of $\mathcal{Z}$. Let $\boldsymbol{\pi}^{ij}$ be the permutation vector obtained by exchanging element i with j in $\boldsymbol{\pi}$. We use the following lemma.

Lemma H.0.16. *(El-Yaniv and Pechyony, 2007, Lemma 3) Suppose that, for each $\boldsymbol{\pi}$, $f : \mathcal{P} \to \mathbb{R}$ is symmetric on $(\pi_1, ...\pi_m)$ and on $(\pi_{m+1}, ...\pi_n)$ and $|f(\boldsymbol{\pi}) - f(\boldsymbol{\pi}^{ij})| \leq \beta$ for all i and j. Let $\boldsymbol{\pi}$ be drawn uniformly at random from $\mathcal{P}$, then*

$$\mathbb{P}_{\boldsymbol{\pi}}\left(f(\boldsymbol{\pi}) - \mathbb{E}_{\boldsymbol{\pi}}(f(\boldsymbol{\pi})) \geq \epsilon\right) \leq \exp\left(\frac{-2\epsilon^2}{\beta^2 \min(m,u)}\right).$$

We now prove the theorem.

Proof. Define $D(\mathcal{S}) := \sup_{h\in\mathcal{H}}\left(\mathrm{risk}^{\ell}_{\mathcal{T}}(h) - \widehat{\mathrm{risk}}^{\ell}_{\mathcal{S}}(h)\right)$ and notice that D satisfies the conditions of Lemma H.0.16 with $\beta = C(\frac{1}{m} + \frac{1}{u})$, thus with probability at least $1 - \delta$ over the draw of $\mathcal{S}$

$$D(\mathcal{S}) \leq \mathbb{E}_{\mathcal{S}}(D(\mathcal{S})) + C\left(\frac{1}{m} + \frac{1}{u}\right)\sqrt{\frac{\min(m,u)}{2}\log\frac{1}{\delta}}. \tag{H.1}$$

Denote $Z_i := (X_i, Y_i)$ for each (X_i, Y_i) drawn from $\mathcal{Z}$. For each $h \in \mathcal{H}$ denote $\ell_h(Z_i) := \ell(h(X_i), Y_i)$

so that $\mathcal{L}_{\mathcal{H}} := \{\ell_h \ : \ h \in \mathcal{H}\}$ is the class of loss functions indexed by $\mathcal{H}$ over $\mathcal{Z}$. We have

$$\begin{aligned}
\mathbb{E}_{\mathcal{S}}(D(\mathcal{S})) &= \mathbb{E}_{\mathcal{S}}\left[\sup_{h\in\mathcal{H}}\left(\mathrm{risk}^{\ell}_{\mathcal{T}}(h) - \widehat{\mathrm{risk}}^{\ell}_{\mathcal{S}}(h)\right)\right] \\
&= \mathbb{E}_{\mathcal{S}}\left[\sup_{h\in\mathcal{H}}\left(\frac{1}{u}\sum_{i=1}^{u}\ell_h(Z_{t_i}) - \frac{1}{m}\sum_{i=1}^{m}\ell_h(Z_{s_i})\right)\right] \qquad \text{(H.2)} \\
&= \mathbb{E}_{\mathcal{S}}\left[\sup_{h\in\mathcal{H}}\left(\frac{1}{u}\sum_{i=1}^{n}\ell_h(\boldsymbol{z}_i) - \left(\frac{1}{m}+\frac{1}{u}\right)\sum_{i=1}^{m}\ell_h(Z_{s_i})\right)\right] \\
&= \frac{n}{u}\mathbb{E}_{\mathcal{S}}\left[\sup_{h\in\mathcal{H}}\left(\frac{1}{n}\sum_{i=1}^{n}\ell_h(\boldsymbol{z}_i) - \frac{1}{m}\sum_{i=1}^{m}\ell_h(Z_{s_i})\right)\right] \\
&= \frac{n}{u}\mathbb{E}_{\mathcal{S}}\left[\sup_{h\in\mathcal{H}}\left(\mathbb{E}_{\mathcal{S}'}\left[\frac{1}{m}\sum_{i=1}^{m}\ell_h(Z'_{s_i})\right] - \frac{1}{m}\sum_{i=1}^{m}\ell_h(Z_{s_i})\right)\right] \qquad \text{(H.3)}
\end{aligned}$$

where $\mathcal{S}' = \{Z'_{s_1}, ...Z'_{s_m}\} = \{(X'_{s_1}, Y'_{s_1}), ...(X'_{s_m}, Y'_{s_m})\}$ is a familiar "ghost sample" drawn according to the same distribution as $\mathcal{S}$, that is, uniformly without replacement from $\mathcal{Z}$. Continuing, the r.h.s. of (H.3) is no greater than,

$$\frac{n}{u}\mathbb{E}_{\mathcal{S},\mathcal{S}'}\left[\sup_{h\in\mathcal{H}}\left(\frac{1}{m}\sum_{i=1}^{m}\ell_h(Z'_{s_i}) - \ell_h(Z_{s_i})\right)\right] \leq \frac{n}{u}\mathbb{E}_{\mathcal{S},\mathcal{S}',\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}}\left(\frac{1}{m}\sum_{i=1}^{m}\ell_h(\widetilde{Z}'_{s_i}) - \ell_h(\widetilde{Z}_{s_i})\right)\right] \qquad \text{(H.4)}$$

$$= \frac{n}{u}\mathbb{E}_{\mathcal{S},\mathcal{S}',\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}}\left(\frac{1}{m}\sum_{i=1}^{m}\sigma_i\ell_h(Z'_{s_i}) - \sigma_i\ell_h(Z_{s_i})\right)\right], \qquad \text{(H.5)}$$

where the $\{\sigma_i\}_{i=1}^m$ are independent Rademacher variables and where $\widetilde{Z}_{s_i} := \frac{1}{2}(1+\sigma_i)Z_{s_i} + \frac{1}{2}(1-\sigma_i)Z'_{s_i}$ and $\widetilde{Z}'_{s_i} := \frac{1}{2}(1-\sigma_i)Z_{s_i} + \frac{1}{2}(1+\sigma_i)Z'_{s_i}$. Inequality in (H.4) occurs because $\widetilde{\mathcal{S}} := \{\widetilde{Z}_{s_1}, ...\widetilde{Z}_{s_m}\}$ and $\widetilde{\mathcal{S}}' := \{\widetilde{Z}'_{s_1}, ...\widetilde{Z}'_{s_m}\}$ can each contain repeated instances and are less likely than $\mathcal{S}$ and $\mathcal{S}'$ to have in common a copy of the same labeled point, thus the expected supremum is larger[1]: we prove this formally, for a particular $\boldsymbol{\sigma}$, $\mathcal{S}$ and $\mathcal{S}'$ denote,

$$\mathcal{K} := \{(i,j) \ : \ Z_{s_i} = Z'_{s_j}\}$$
$$\widetilde{\mathcal{K}} := \{(i,j) \ : \ \widetilde{Z}_{s_i} = \widetilde{Z}'_{s_j}\},$$

and call such occurances "clashes". Put $N := |\mathcal{K}| - |\widetilde{\mathcal{K}}| \geq 0$ so that the action of $\boldsymbol{\sigma}$ on $\mathcal{S}$, $\mathcal{S}'$ swaps N clashes; there are N instances which $\mathcal{S}$ and $\mathcal{S}'$ had in common, which occur in one of $\widetilde{\mathcal{S}}$, $\widetilde{\mathcal{S}}'$ with multiplicity 2. Now let $M_0 = m - |\mathcal{K}|$ and define,

$$\Psi := \{\psi_1, ...\psi_{M_0}\} := \mathcal{S}\backslash\{Z_i : (i,j) \in \mathcal{K} \text{ for some } j\}$$
$$\Psi' := \{\psi'_1, ...\psi'_{M_0}\} := \mathcal{S}'\backslash\{Z'_j : (i,j) \in \mathcal{K} \text{ for some } i\}$$
$$\widetilde{\Psi} := \{\widetilde{\psi}_1, ...\widetilde{\psi}_{M_0+N}\} := \widetilde{\mathcal{S}}\backslash\{\widetilde{Z}_i : (i,j) \in \widetilde{\mathcal{K}} \text{ for some } j\}$$
$$\widetilde{\Psi}' := \{\widetilde{\psi}'_1, ...\widetilde{\psi}'_{M_0+N}\} := \widetilde{\mathcal{S}}'\backslash\{\widetilde{Z}'_j : (i,j) \in \widetilde{\mathcal{K}} \text{ for some } i\}$$

[1] In the inductive setting these steps are more straightforward since there the random variables Z'_{s_i} and Z_{s_i} have the same distribution for each i and an equality follows by symmetry.

so that, for example, Ψ is $\mathcal{S}$ with any elements common to $\mathcal{S}$ and $\mathcal{S}'$ removed. Note that $\{\psi_1, ...\psi_{M_0}, \psi'_1, ...\psi'_{M_0}\}$ are all distinct. Further, w.l.o.g. we order $\widetilde{\Psi}$ and $\widetilde{\Psi}'$ such that at least one copy of any elements which occur in either $\widetilde{\Psi}$ or $\widetilde{\Psi}'$ with multiplicity 2 (there are N such elements in total, shared between $\widetilde{\Psi}$ and $\widetilde{\Psi}'$) is placed in a position j where $M_0 < j \leq M_0 + N$. This ordering ensures $\{\widetilde{\psi}_1, ...\widetilde{\psi}_{M_0}, \widetilde{\psi}'_1, ...\widetilde{\psi}'_{M_0}\}$ are all distinct. Because of this, the sets $\{\psi_1, ...\psi_{M_0}, \psi'_1, ...\psi'_{M_0}\}$ and $\{\widetilde{\psi}_1, ...\widetilde{\psi}_{M_0}, \widetilde{\psi}'_1, ...\widetilde{\psi}'_{M_0}\}$ have the same distribution: they are both drawn uniformly without replacement from $\mathcal{Z}$. Now we set

$$h^* := \operatorname*{argmax}_{h\in\mathcal{H}} \sum_{i=1}^{M_0} \ell_h(\widetilde{\psi}'_i) - \ell_h(\widetilde{\psi}_i) \tag{H.6}$$

and note,

$$\begin{aligned}
&\mathbb{E}_{\mathcal{S},\mathcal{S}',\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}}\left(\frac{1}{m}\sum_{i=1}^{m}\ell_h(\widetilde{Z}'_{s_i}) - \ell_h(\widetilde{Z}_{s_i})\right) - \sup_{h\in\mathcal{H}}\left(\frac{1}{m}\sum_{i=1}^{m}\ell_h(Z'_{s_i}) - \ell_h(Z_{s_i})\right)\right] \\
&= \mathbb{E}_{\mathcal{S},\mathcal{S}',\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}}\left(\frac{1}{m}\sum_{i=1}^{M_0+N}\ell_h(\widetilde{\psi}'_i) - \ell_h(\widetilde{\psi}_i)\right) - \sup_{h\in\mathcal{H}}\left(\frac{1}{m}\sum_{i=1}^{M_0}\ell_h(\psi'_i) - \ell_h(\psi_i)\right)\right] \\
&\geq \mathbb{E}_{\mathcal{S},\mathcal{S}',\boldsymbol{\sigma}}\left[\frac{1}{m}\sum_{i=1}^{M_0+N}\ell_{h^*}(\widetilde{\psi}'_i) - \ell_{h^*}(\widetilde{\psi}_i) - \sup_{h\in\mathcal{H}}\left(\frac{1}{m}\sum_{i=1}^{M_0}\ell_h(\psi'_i) - \ell_h(\psi_i)\right)\right] \\
&= \mathbb{E}_{\mathcal{S},\mathcal{S}',\boldsymbol{\sigma}}\left[\frac{1}{m}\sum_{i=M_0+1}^{M_0+N}\ell_{h^*}(\widetilde{\psi}'_i) - \ell_{h^*}(\widetilde{\psi}_i)\right],
\end{aligned} \tag{H.7}$$

and we now show that the final line $(H.5) \geq 0$. Denote $\Psi_1 := \{\widetilde{\psi}'_i\}_{i=1}^{M_0}$, $\Psi_2 := \{\widetilde{\psi}_i\}_{i=1}^{M_0}$. The result will follow essentially because conditional on $\Psi_1 \cup \Psi_2$, elements of $\{\widetilde{\psi}'_{M_0+1}, ...\widetilde{\psi}'_{M_0+N}\}$ are drawn from $\mathcal{Z}\backslash\Psi_2$ and elements of $\{\widetilde{\psi}_{M_0+1}, ...\widetilde{\psi}_{M_0+N}\}$ are drawn from $\mathcal{Z}\backslash\Psi_1$: consider $\{\widetilde{\psi}'_i\}_{i=M_0+1}^{M_0+N}$, N_1 of these are drawn uniformly without replacement from Ψ_1, where $N_1 = \mathrm{Bin}(N, 1/2)$. Likewise $N_2 = N - N_1$ of the $\{\widetilde{\psi}_i\}_{i=M_0+1}^{M_0+N}$ are drawn uniformly without replacement from Ψ_2. Denote these by $\Xi_1 := \{\xi'_i\}_{i=1}^{N_1}$ and $\Xi_2 := \{\xi_i\}_{i=1}^{N_2}$ respectively and $\Xi := \Xi_1 \cup \Xi_2$. The remaining N elements of $\Omega := \left(\{\widetilde{\psi}'_i\}_{i=M_0+1}^{M_0+N} \cup \{\widetilde{\psi}_i\}_{i=M_0+1}^{M_0+N}\right)\backslash\Xi$ are drawn uniformly without replacement from $Z\backslash(\Psi_1\cup\Psi_2)$. Denote these by $\{\widetilde{\psi}'_i\}_{i=M_0+1}^{M_0+N}\backslash\Xi_1 =: \{\omega'_i\}_{i=1}^{N_2}$ and $\{\widetilde{\psi}_i\}_{i=M_0+1}^{M_0+N}\backslash\Xi_2 =: \{\omega_i\}_{i=1}^{N_1}$. Then we have,

$$\begin{aligned}
(H.5) &= \frac{1}{m}\mathbb{E}\left[\sum_{i=1}^{N_1}\ell_{h^*}(\xi'_i) - \sum_{i=1}^{N_2}\ell_{h^*}(\xi_i)\right] + \frac{1}{m}\mathbb{E}\left[\sum_{i=1}^{N_2}\ell_{h^*}(\omega'_i) - \sum_{i=1}^{N_1}\ell_{h^*}(\omega_i)\right] \\
&= \frac{1}{m}\mathbb{E}\left[\sum_{\xi'\in\Xi_1}\ell_{h^*}(\xi') - \sum_{\xi\in\Xi_2}\ell_{h^*}(\xi)\right] + \frac{1}{m}\mathbb{E}\left[\sum_{i=1}^{N_2}\ell_{h^*}(\omega'_i) - \sum_{i=1}^{N_1}\ell_{h^*}(\omega_i)\right].
\end{aligned}$$

The second summand is zero by symmetry (ω'_i and ω_i have the same distribution). The first summand is not less than zero since, conditioned on $\Psi_1 \cup \Psi_2$, Ξ_1 is drawn uniformly without replacement from Ψ_1, and Ξ_2 is drawn uniformly without replacement from Ψ_2 and otherwise we would have a contradiction on the definition (H.6) of h^*. Thus $(H.5) \geq 0$ and (H.4) holds.

To continue, we finally just note,

$$\begin{aligned}
(H.5) &\leq 2\frac{n}{u}\mathcal{R}_m^{\mathrm{trs}}(\mathcal{L}_\mathcal{H}) \\
&\leq 2K\frac{n}{u}\mathcal{R}_m^{\mathrm{trs}}(\mathcal{H}),
\end{aligned}$$

The final line is a consequence of the contraction inequality for Rademacher complexities, (Meir and Zhang, 2003, Theorem 7).

Finally, notice the symmetry in (H.2) for $m \leftrightarrow u$ and that by producing precisely the symmetrically opposite argument we would derive $\mathbb{E}_{\mathcal{S}}(D(\mathcal{S})) \leq 2K\frac{n}{m}\mathcal{R}_u^{\mathrm{trs}}(\mathcal{H})$, hence $\mathbb{E}_{\mathcal{S}}(D(\mathcal{S})) \leq \frac{2Kn}{\max(m,u)}\mathcal{R}_{\min(m,u)}^{\mathrm{trs}}(\mathcal{H})$. □