Exploiting Cluster-Structure to Predict the Labeling of a Graph

Mark Herbster

Department of Computer Science
University College London
Gower Street, London WC1E 6BT, England, UK
m.herbster@cs.ucl.ac.uk

Abstract. The *nearest neighbor* and the *perceptron* algorithms are intuitively motivated by the aims to exploit the "cluster" and "linear separation" structure of the data to be classified, respectively. We develop a new online perceptron-like algorithm, POUNCE, to exploit both types of structure. We refine the usual margin-based analysis of a perceptron-like algorithm to now additionally reflect the cluster-structure of the input space. We apply our methods to study the problem of predicting the labeling of a graph. We find that when both the quantity and extent of the clusters are small we may improve arbitrarily over a purely margin-based analysis.

1 Introduction

We study the problem of online learning over a graph. Consider the following game for predicting the labeling of a graph. *Nature* presents a vertex $\mathbf{v}_{i_1}$; the *learner* predicts the label of the vertex $\hat{y}_1 \in \{-1, 1\}$; *nature* presents a label y_1; *nature* presents a vertex $\mathbf{v}_{i_2}$; the *learner* predicts $\hat{y}_2$; and so forth. The learner's goal is minimize the total number of mistakes ($|\{t : \hat{y}_t \neq y_t\}|$). 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 methodological goal is to give learners whose total mispredictions can be bounded relative to the "complexity" of nature's labeling. In [16, 15], the *cut size* (the number of edges between disagreeing labels) and diameter were used as a measure of the complexity of a graph's labeling. We will show that such bounds may be improved arbitrarily by also considering the *cluster-structure* of the graph.

The problem of labeling a graph online is not only of theoretical interest but may also be practically motivated. For example, consider a system which serves advertisements on web pages. The web pages may be identified with the vertices of a graph and the edges as links between pages. The online prediction problem is then that, at a given time t the system may receive a request to serve an advertisement on a particular web page. For simplicity, we assume that there are two alternatives to be served: either advertisement "A" or advertisement "B". The system then interprets the feedback as the label and then may use this

information in responding to the next request to predict an advertisement for a requested web page.

Recently there has been extensive research into both transduction and semi-supervised learning in the batch setting. A motivating hypothesis is that the structure of the input space may be exploited to improve learning when either the *cluster* or *manifold* condition [5] is satisfied; informally that is

> **Cluster Condition:** If points are in the same cluster, they are likely to be of the same class.
> **Manifold Condition:** The (high-dimensional) data lie (roughly) on a low-dimensional manifold.

In this paper we will give bounds for an online perceptron-like algorithm which supports this hypothesis. In particular we will take advantage of "clumpiness" in the input space; thus when the inputs are distributed uniformly over a sphere we will find no advantage. However, if it is the case that the input space X is such that it is concentrated into a few dense clusters or if it lies on a smooth low-dimensional manifold, we can then *cover* the input space with a relatively moderate number of balls of modest diameter with respect to the extent of X.

In Section 4 we give a new online algorithm POUNCE with a mistake bound based on the size of the cover of an input space X. The minimum number of balls of (squared) diameter ρ that cover X is denoted as $\mathcal{N}(X, \rho)$. With the assumption that the input space is a subset of an inner-product space which is induced by a graph Laplacian, we refine the classical result of Novikoff [20] in Theorem 2 by incorporating the cover size to give

$$|\mathcal{M}| \leq \mathcal{N}(X, \rho) + \|\mathbf{u}\|^2 \rho + 1 \,.$$

Here $|\mathcal{M}|$ is the cumulative mistakes of our algorithm, and $\|\mathbf{u}\|^2$ is the squared semi-norm of a *separating* classifier. This result significantly improves on the Novikoff bound when the input space consists of a few dense clusters but not uniformly as the squared diameter *may* be four times larger than the origin-based squared radius of Novikoff bound and the additive constant "1".

A key issue in graph-based transductive learning is how we should use the basic inputs to build a graph for a given algorithm. In Section 5 we apply the previous result in order to understand implications of using the exponential embedding ("heat kernel" [2]) to build a graph. There we will additionally assume as with the "cluster condition" that points in the same ball will have the same label. In Figure 1 we illustrate our results with the classic two moons dataset [2]. Here each "moon" represents and contains points of only a single class. We can characterize this dataset via two types of now label-dependent covers of the input space. Their corresponding cover numbers $\mathcal{N}^\circ$ and $\mathcal{C}^\circ$ differ from $\mathcal{N}$ in that each set in these covers consist only of labeled points of the same class. The first kind of cover is illustrated by the $\mathcal{N}^\circ = 42$ balls which completely cover the input space. The diameter ρ of these balls must be less than the minimal separation between any two opposing labels δ. The second kind of cover $\mathcal{C}^\circ$ assumes now that we have particular knowledge of the unlabeled data. The second cover is

in terms of ρ-*path-connected* sets, i.e., every two points is connected by a path of points for which every point in the path is no more than ρ distant from its predecessor and successor in the path. Thus in Figure 1 the 30 black points cover the input space and define a ρ-path-connected cover of size $\mathcal{C}^\circ = 2$. Given such covers of the input space Theorem 8 then implies that there is an embedding of the input space with a graph Laplacian such that the mistakes of our algorithm may be then bounded by

$$|\mathcal{M}| \leq \mathcal{C}^\circ + 1 \leq \mathcal{N}^\circ + 1 \,.$$

This result gives an insight into the value of unlabeled data in transductive (semi-supervised) learning within the context of our setting. From the bound above we see that with no assumptions on the unlabeled data beyond the geometry of the input space, a priori, our mistake bound is $\mathcal{N}^\circ + 1$ whereas increasing the quantity of unlabeled data may induce a cover via a few path-connected subsets such that $\mathcal{C}^\circ \ll \mathcal{N}^\circ$, significantly decreasing the upper bound.

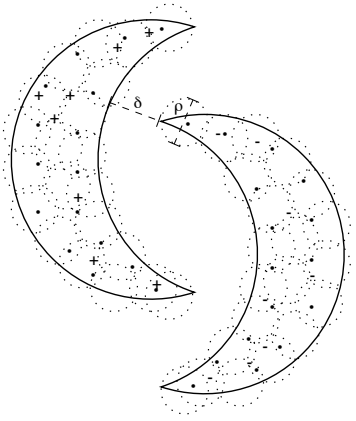

Fig. 1. Covering the two moons

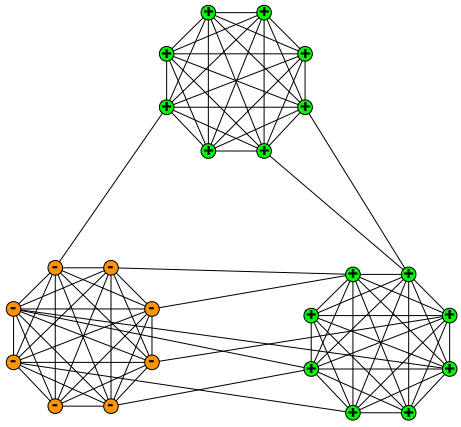

Fig. 2. Three clusters (one in isolation)

2 Background

Our research builds on the extensive literature concerned with algorithmic variants of the perceptron and their mistake bound analysis; a few recent examples include [18, 8, 10, 13, 4, 6]. Our particular concern is to factor into the usual margin-based analyses a model which can capture simplifying elements of the cluster-structure of the input space. A mistake bound analysis which also incorporates the structure of the input space is that of the second-order perceptron [4]; those results are based on the spectrum of the correlation matrix of the inputs. In the algorithmic luckiness framework [12] an analysis of the max-margin classifier is also given in terms of the cover number of the input space.

Semi-supervised and transductive learning methods suppose that unlabeled data can aid the learner. Thus if the input space is benign as characterized by the cluster and/or manifold conditions, it is expected that the unlabeled data may be exploited. A common approach is to use the labeled and unlabeled data to build a "graph" which is then used by the learning method. The seminal approach in [3] is to predict with a labeling which is consistent with a minimum label-separating cut. The approach which directly motivates our work is that based on the semi-norm which is induced by the graph Laplacian [22, 2, 17] which is either directly minimized subject to constraints, or used as a regularizer.

2.1 Prediction on a Graph with a Perceptron

In [16, 15] the online graph labeling 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. Results were given when the basic kernel was the pseudoinverse of the graph Laplacian. As the current work builds directly on the results in [16, 15] we will selectively summarize and elaborate on those prior results.

The graph Laplacian is a positive semidefinite matrix which is defined from the adjacency (weight) matrix of the graph. Let $\mathbf{A}$ be the $n \times n$ symmetric weight matrix of the graph such that $A_{ij} \geq 0$, and define the edge set $E(\mathbf{G}) := \{(i,j) : 0 < A_{ij}\}$; note edges are unordered pairs thus $(i,j) \equiv (j,i)$. The graph Laplacian $\mathbf{G}$ is then the $n \times n$ matrix defined as

$$\mathbf{G} := \mathbf{D} - \mathbf{A}, \tag{1}$$

where $\mathbf{D} = \mathrm{diag}(d_1, \ldots, d_n)$ and d_i is the *weighted* degree of vertex i, $d_i = \sum_{j=1}^{n} A_{ij}$. The induced semi-norm is then

$$\|\mathbf{u}\|^2 := \mathbf{u}^\top \mathbf{G} \mathbf{u} = \sum_{(i,j) \in E(\mathbf{G})} A_{ij}(u_i - u_j)^2. \tag{2}$$

The graph is naturally interpreted as an n-vertex resistive network where each edge $(i,j) \in E(\mathbf{G})$ is viewed as a resistor with resistance $\frac{1}{A_{ij}}$. Thus the *effective resistance* $r_{\mathbf{G}}(p,q)$ between vertex p and q is then the potential difference needed to induce a unit current flow between p and q. The effective resistance may be computed via [19],

$$r_{\mathbf{G}}(p,q) = (\mathbf{e}_p - \mathbf{e}_q)^\top \mathbf{G}^+ (\mathbf{e}_p - \mathbf{e}_q), \tag{3}$$

where "+" denotes the pseudoinverse and $\mathbf{e}_p$ is the p-th coordinate vector of $\mathbb{R}^n$. The resistance diameter of a graph $R_{\mathbf{G}} := \max_{1 \leq p < q \leq n} r_{\mathbf{G}}(p,q)$ is the maximum of the effective resistances between each of the pairs of vertices on the graph. Surprisingly, both $r_{\mathbf{G}}(\cdot,\cdot)$ and $\sqrt{r_{\mathbf{G}}(\cdot,\cdot)}$ are metrics on the graph [19].

In [15, Theorem 4.2 (with $b = R_{\mathbf{G}}; c = 0$)] using the pseudoinverse of the graph Laplacian mixed with the "constant" function as a kernel the cumulative mistakes of the kernel perceptron was upper bounded by

$$|\mathcal{M}| \leq 2\|\mathbf{u}^*\|^2 R_{\mathbf{G}} + 2. \tag{4}$$

In the above $\mathbf{u}^*$ is the optimal classifier which is correct on the examples; thus

$$\mathbf{u}^* := \arg\min_{\mathbf{u}\in\mathbb{R}^n} \{\|\mathbf{u}\|^2 : u_1 = y_1, \ldots, u_\ell = y_\ell\} \tag{5}$$

where $y_i \in \{-1, 1\}$ is the true label of vertex i. Thus if we view the graph as a resistive network and we fix the voltages at vertices $i = 1, \ldots, \ell$ to $y_1, \ldots, y_\ell$, respectively, then by Thomson's principle [7] $\mathbf{u}^*$ is then the vector of voltages that minimizes the energy dissipation (power). On an unweighted graph ($A_{ij} \in \{0, 1\}$) the energy dissipation and the resistance diameter may themselves be bounded by

$$\|\mathbf{u}^*\|^2 \leq 4\Phi(\mathbf{u}^*) \text{ and } R_{\mathbf{G}} \leq D_G,$$

four times the separating cut size and the geodesic diameter of the graph, respectively. The separating cut is the number of edges required to separate the positive and negative labels, and the geodesic diameter of a graph is the maximum of the geodesic distances between each of the pairs of vertices on the graph. In the central result of this paper, Theorem 2, we improve the leading term of the bound (equation (4)) by a factor of two. More significantly, we will see in the following subsection that we may improve over (4) without limit when the graph consists of three or more "dense" clusters.

2.2 Three Clusters are Hard for the Perceptron

First we will recall the analysis of a graph with two clusters as given in [15, p. 7]; there it was found that the mistakes of the perceptron could be upper bounded by a constant independent of the size of the clusters. Then we will observe that the two-cluster result does not generalize to 3+ clusters. In this discussion, for simplicity, we represent a "cluster" in an unweighted graph by an m-vertex clique.

Following [15] consider two m-cliques (one labeled "+1", one "-1") with ℓ arbitrary edges ($\ell < m$) connecting the cliques. Note that between any two vertices there are at least ℓ edge-disjoint paths of length no more than five, and therefore the resistance diameter $R_{\mathbf{G}}$ is at most $5/\ell$ and the cut size is $\Phi(\mathbf{u}^*) = \ell$. Hence by (4) the bound on the cumulative mistakes is the constant 42.

Now consider the addition of a third cluster (m-clique) such that the first two cliques are connected by ℓ edges in proportion to m ($cm < \ell < m$), but the third cluster is connected to the initial two by a constant number of edges independent of m and thus $R_{\mathbf{G}} = \Theta(1)$ (see Figure 2). Thus as m increases, in relative terms the third cluster becomes increasingly remote, but counter to geometric intuition the perceptron upper bound (4) increases as $\Theta(\ell)$. Whereas with POUNCE applied to the three cluster problem an upper bound on mistakes is the constant 20 (Equation (10) with $\rho = \frac{2}{m-1}$, $\mathcal{N}(X, \rho) = 3$, and $\Phi(\mathbf{u}^*) < 2\ell$). The difficulty of the three cluster problem for the perceptron is not only a problem in upper bound but also in performance, as there exists a parameterized three-example separable set such that the perceptron must incur mistakes linear in the parameter. In contrast, the upper bound of POUNCE is a constant. We omit this example for reasons of space, however, see [14].

3 Preliminaries

We denote matrices (conventionally $n \times n$) by capital bold letters and vectors (conventionally $n \times 1$) by small bold case letters. So $\mathbf{M}$ denotes the $n \times n$ matrix $(M_{ij})_{i,j=1}^n$ and $\mathbf{w}$ the n-dimensional column vector $(w_1, \ldots, w_n)^\top$ also denoted by $(\mathbf{w}(1), \ldots, \mathbf{w}(n))^\top$ where "$^\top$" denotes transposition. The identity matrix is denoted by $\mathbf{I}$. We also let $\mathbf{0}$ and $\mathbf{1}$ be the n-dimensional vectors all of whose components equal to zero and one respectively, and $\mathbf{e}_i$ the i-th coordinate vector of $\mathbb{R}^n$. Let $\mathbb{N}$ be the set of natural numbers and $\mathbb{N}_\ell := \{1, \ldots, \ell\}$. Let $\mathcal{H}$ denote a Hilbert space. If A and B are sets then the set difference is denoted $A \backslash B$ and the shorthand $A \backslash x := A \backslash \{x\}$.

A symmetric positive semidefinite matrix $\mathbf{M}$ induces a semi-inner product on $\mathbb{R}^n$ which is defined as

$$\langle \mathbf{u}, \mathbf{w} \rangle_{\mathbf{M}} := \mathbf{u}^\top \mathbf{M} \mathbf{w},$$

where $\|\mathbf{w}\|_{\mathbf{M}} := \langle \mathbf{w}, \mathbf{w} \rangle_{\mathbf{M}}$ denotes the associated semi-norm (note that the subscript "$\mathbf{M}$" in both $\langle \cdot, \cdot \rangle_{\mathbf{M}}$ and $\|\cdot\|_{\mathbf{M}}$ may be omitted when clear from the context). The reproducing kernel [1] associated with the above semi-inner product is $\mathbf{K} = \mathbf{M}^+$, where "$^+$" denotes the pseudoinverse. We also define the *coordinate spanning set*

$$\mathcal{V}_{\mathbf{M}} := \{\mathbf{v}_i := \mathbf{M}^+ \mathbf{e}_i : i = 1, \ldots, n\} \tag{6}$$

and let $\mathcal{H}(\mathbf{M}) := \text{span}(\mathcal{V}_{\mathbf{M}})$. The restriction of the semi-inner product $\langle \cdot, \cdot \rangle_{\mathbf{M}}$ to $\mathcal{H}(\mathbf{M})$ is an inner product on $\mathcal{H}(\mathbf{M})$. The set $\mathcal{V}_{\mathbf{M}}$ acts as "coordinates" for $\mathcal{H}(\mathbf{M})$, that is, if $\mathbf{w} \in \mathcal{H}(\mathbf{M})$ we have

$$\mathbf{w}(i) = \mathbf{e}_i^\top \mathbf{M}^+ \mathbf{M} \mathbf{w} = \mathbf{v}_i^\top \mathbf{M} \mathbf{w} = \langle \mathbf{v}_i, \mathbf{w} \rangle_{\mathbf{M}}, \tag{7}$$

although the vectors $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ are not necessarily normalized and are linearly independent only if $\mathbf{M}$ is positive definite. We note that equation (7) is simply the reproducing kernel property [1] for kernel $\mathbf{M}^+$.

A *discrepancy* function $d : \mathcal{X} \times \mathcal{X} \to [0, \infty]$ is symmetric $d(x, y) = d(y, x)$ and $(d(x, y) = 0) \iff (x = y)$. The diameter $D(X, d)$ of a set $X \subseteq \mathcal{X}$ is the maximum discrepancy between any two points in X, and thus $D(X, d) := \sup_{x, x' \in X} d(x, x')$. The covering number $\mathcal{N}(X, \rho, d)$ is the minimal number of sets of diameter ρ that contain set X. Thus

$$\mathcal{N}(X, \rho, d) := \min_{\{X_i'\}_{i=1}^k} \{k \in \mathbb{N}^+ : D(X_i', d) \leq \rho, \forall i \in \mathbb{N}_k, \cup_{i=1}^k X_i' \supseteq X\}; \tag{8}$$

if the minimum does not exist then $\mathcal{N}(X, \rho, d) := \infty$. When $X \subseteq \mathcal{H}(\mathbf{M})$ we assume a discrepancy which is a squared norm $d_{\mathbf{M}}(x, y) := \|x - y\|_{\mathbf{M}}^2$ for which we define the abbreviated notation $\mathcal{N}(X, \rho) := \mathcal{N}(X, \rho, d_{\mathbf{M}})$.

3.1 The Signed Laplacian

The graph Laplacian (see (1)) is naturally generalized by the class of symmetric diagonally dominant matrices with nonnegative diagonals which we will refer to

as *signed Laplacians*. Recall that a matrix $\mathbf{M}$ is a *symmetric diagonally dominant* iff $|M_{ii}| \geq \sum_{j \neq i} |M_{ij}|$ for every $i \in \mathbb{N}_n$.

Goldberg et al. [11] introduced the use of the signed Laplacian into semi-supervised learning to explicitly encode dissimilarity relations. The following lemma decomposes the quadratic form $\mathbf{u}^\top \mathbf{M}\mathbf{u}$ into edge and vertex contributions enabling the interpretation of $\mathbf{u}^\top \mathbf{M}\mathbf{u}$ as a smoothness measure of a labeling $\mathbf{u} \in \{-1, 1\}^n$ of an associated graph.

Lemma 1. *If* $\mathbf{M}$ *is symmetric and* $\mathbf{u} \in \mathbb{R}^n$ *then*

$$\mathbf{u}^\top \mathbf{M}\mathbf{u} = \sum_{(i,j)\in E^+} |M_{ij}|(u_i - u_j)^2 + \sum_{(i,j)\in E^-} |M_{ij}|(u_i + u_j)^2 + \sum_{i \in \mathbb{N}_n} [M_{ii} - (D_i^+ + D_i^-)]u_i^2, \quad (9)$$

where $E^+ := \{(i,j) : M_{ij} < 0,\, i < j\}$ *is the positive edge set and* $E^- := \{(i,j) : M_{ij} > 0,\, i < j\}$ *is the negative edge set and*

$$D_i^+ := \sum_{j \in \{k : M_{ik} < 0, k \neq i\}} |M_{ij}|\,, \qquad D_i^- := \sum_{j \in \{k : M_{ik} > 0, k \neq i\}} |M_{ij}|$$

are the positive and negative edge degrees of the i*th vertex respectively. If for all* $i \in \mathbb{N}_n$ *the vertex weight* $M_{ii} - (D_i^+ + D_i^-)$ *is nonnegative then* $\mathbf{M}$ *is positive semidefinite.*

Proof. Since $\mathbf{M}$ is symmetric then $\mathbf{u}^\top \mathbf{M}\mathbf{u} = 2\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} M_{ij} u_i u_j + \sum_{i=1}^{n} M_{ii} u_i^2$. Therefore as

$$2M_{ij}u_iu_j = \begin{cases} |M_{ij}|(u_i + u_j)^2 - |M_{ij}|(u_i^2 + u_j^2) & M_{ij} \geq 0 \\ |M_{ij}|(u_i - u_j)^2 - |M_{ij}|(u_i^2 + u_j^2) & M_{ij} \leq 0 \end{cases}$$

equation (9) follows immediately. Since the first two terms in (9) are nonnegative, and the third is nonnegative if the vertex weight $M_{ii} - (D_i^+ + D_i^-)$ is nonnegative $\forall i \in \mathbb{N}_n$, then $\mathbf{u}^\top \mathbf{M}\mathbf{u} \geq 0$ for $\mathbf{u} \in \mathbb{R}^n$, and thus $\mathbf{M}$ is positive semidefinite. □

Thus if all the vertex weights are nonnegative $\mathbf{M}$ is a signed Laplacian and $\|\cdot\|_{\mathbf{M}}$ is a semi-norm. The "sign" of the edges reflect dissimilarity (similarity) penalties as when two vertices are connected by a positive (negative) edge its contribution to the semi-norm is zero if the labels are the same (different) and positive otherwise. A matrix with a zero vertex weighting and an empty negative edge set corresponds to the Laplacian matrix defined in (1). A matrix $\mathbf{M}$ is *irreducible* iff there does not exist partitioning sets $P, Q \subset \mathbb{N}_n$, $P \cup Q = \mathbb{N}_n$ such that $M_{pq} = 0$ for each $p \in P$ and $q \in Q$; this is equivalent to stating that the associated graph is *connected*.

4 The POUNCE Algorithm

In a mistake-driven algorithm the mistaken examples implicitly generate a cover of the input space via the hypothesis vector. The motivating idea of the POUNCE *(Projection-Orientated-Using-Nearby-Cover-Elements)* algorithm (see Figure 3) is to explicitly use that cover to design an algorithm whose analysis directly

reflects both the margin of the separating hypothesis and the structure of the input space. Although it may be possible to achieve similar bounds for a "second order" algorithm our aim is to design an algorithm whose bound can improve on the kernel perceptron in natural scenarios with no increase in the order of computational complexity. Thus the time complexity of POUNCE is identical to the kernel perceptron that is $O(m_t)$ time required on trial t where m_t is the current total of the cumulative mistakes. Thus the total time required for ℓ trials is $O(m_\ell \ell)$. This is assuming the kernel is provided in advance. However, if we are interested in predicting the labeling of a graph with a (signed) Laplacian we must calculate the pseudoinverse of the (signed) Laplacian. The time required to initially compute this kernel in the general case of an n-vertex graph is pragmatically $O(n^3)$; however, see [9] for a significant improvement when the graph is a tree. As computing the kernel is a one-time cost, the existence of multiple problems or multiple validation experiments on the same graph may effectively offset this initial cost.

The input to POUNCE is an online sequence of vertices and labels from a graph with associated (signed) Laplacian $\mathbf{M}$. The first trial is automatically a mistake. On the tth trial vertex $i_t \in \mathbb{N}_n$ is input to the algorithm as its "cooordinate" $\mathbf{v}_{i_t} \in \mathcal{V}_\mathbf{M}$ (see (6)). In order to predict, POUNCE finds the nearest neighboring vertex $\mathbf{v}_{\eta_t}$ to $\mathbf{v}_{i_t}$, among those in the mistake set $\mathcal{M}$, in norm $\|\cdot\|_\mathbf{M}$. The prediction and update rule are based on both the vertex to be predicted and its nearest neighbor. Intuitively, POUNCE is thus a combination of a nearest neighbors algorithm and a perceptron-like algorithm.

Theorem 2. *Let* $\mathbf{M}$ *be either an irreducible Laplacian or a positive definite signed Laplacian. If* $X := \{\mathbf{v}_{i_t}\}_{t=1}^{\ell} \in \mathcal{V}_\mathbf{M}^{\ell}$ *and* $Y := \{y_t\}_{t=1}^{\ell} \in \{-1,1\}^{\ell}$ *are the sequences of inputs and associated labels and* $\mathcal{M}$ *is the set of trials in which the* POUNCE *algorithm predicted incorrectly, then the cumulative mistakes* $|\mathcal{M}|$ *are upper bounded by*

$$|\mathcal{M}| \leq \mathcal{N}(X,\rho) + \|\mathbf{u}\|^2 \rho + 1 \ , \tag{10}$$

for all $\rho > 0$*, and for all* $\mathbf{u} \in \mathbb{R}^n$ *such that* $\mathbf{u}(i_t) y_t \geq 1$ *for all* $t \in \mathcal{M}$*.*

Input: $\{(\mathbf{v}_{i_t}, y_t)\}_{t=1}^{\ell} \subseteq \mathcal{V}_\mathbf{M} \times \{-1,1\}$.
Initialization: $\mathbf{w}_2 = \mathbf{0}$; $\mathcal{M} = \{1\}$.
for $t = 2, \ldots, \ell$ **do**
  Receive: $i_t \in \{1, \ldots, n\}$
    $\eta_t = \arg\min_{j \in \mathcal{M}} \|\mathbf{v}_{i_t} - \mathbf{v}_{i_j}\|$
  Predict: $\hat{y}_t = \text{sign}(y_{\eta_t} + \mathbf{w}_t(i_t) - \mathbf{w}_t(i_{\eta_t}))$
  Receive: y_t
    if $\hat{y}_t = y_t$ **then**
      $\mathbf{w}_{t+1} = \mathbf{w}_t$
    else
      $\mathbf{w}_{t+1} = \mathbf{w}_t + \frac{y_t - y_{\eta_t} - (\mathbf{w}_t(i_t) - \mathbf{w}_t(i_{\eta_t}))}{\|\mathbf{v}_{i_t} - \mathbf{v}_{i_{\eta_t}}\|^2} (\mathbf{v}_{i_t} - \mathbf{v}_{i_{\eta_t}})$
      $\mathcal{M} = \mathcal{M} \cup \{t\}$
end

Fig. 3. The POUNCE Algorithm

4.1 Proof of Theorem 2

The update step of POUNCE is a projection. We recall the definition of projection.

Definition 3. *The projection of a point* $\mathbf{w} \in \mathcal{H}$ *onto a closed convex nonempty set* $\mathcal{U} \subseteq \mathcal{H}$ *is defined by*

$$P(\mathcal{U}; \mathbf{w}) := \arg\min_{\mathbf{u} \in \mathcal{U}} \|\mathbf{u} - \mathbf{w}\|. \tag{11}$$

We recall the following two facts about projection: the first is the pythagorean theorem, and the second the explicit equation for the projection to a hyperplane.

Lemma 4. *If* $\mathcal{U} \subseteq \mathcal{H}$ *is an affine set and* $\mathbf{w} \in \mathcal{H}$, $\mathbf{u} \in \mathcal{U}$ *then*

$$\|\mathbf{u} - \mathbf{w}\|^2 = \|\mathbf{u} - P(\mathcal{U}; \mathbf{w})\|^2 + \|P(\mathcal{U}; \mathbf{w}) - \mathbf{w})\|^2 \ , \tag{12}$$

and if $(\mathbf{x}, y) \in \mathcal{H} \backslash \mathbf{0} \times \mathbb{R}$ *and* $\mathbf{w} \in \mathcal{H}$ *then*

$$P(\{\mathbf{u} : \langle \mathbf{u}, \mathbf{x} \rangle = y\}; \mathbf{w}) = \mathbf{w} + \frac{y - \langle \mathbf{w}, \mathbf{x} \rangle}{\|\mathbf{x}\|^2} \mathbf{x} \, . \tag{13}$$

Inspired by the maximum principle [7, p. 7] for the graph Laplacian, we prove the following theorem which holds for the more general signed Laplacian.

Theorem 5. *If* $\mathbf{M}$ *is a signed Laplacian and* $\mathbf{y} \in \{-1, 1\}^\ell$ *then*

$$\min_{\mathbf{u} \in U_{\mathbf{y}}} \mathbf{u}^\top \mathbf{M} \mathbf{u} = \min_{\mathbf{u} \in \widetilde{U}_{\mathbf{y}}} \mathbf{u}^\top \mathbf{M} \mathbf{u} \tag{14}$$

with

$U_{\mathbf{y}} := \{\mathbf{u} \in \mathbb{R}^n : u_1 = y_1, \ldots, u_\ell = y_\ell\}$ *and* $\widetilde{U}_{\mathbf{y}} := \{\mathbf{u} \in \mathbb{R}^n : u_1 y_1 \geq 1, \ldots, u_\ell y_\ell \geq 1\}$.

Proof. Equation (14) is trivially true if $\mathbf{M} = 0$. Consider the case that $\mathbf{M}$ is irreducible. Suppose (14) is false then there exists (see [21, Corollary 27.3.1]) a possibly nonunique vector $\mathbf{v} \in \widetilde{U}_{\mathbf{y}}$, such that

$$\mathbf{v}^\top \mathbf{M} \mathbf{v} = \min_{\mathbf{u} \in \widetilde{U}_{\mathbf{y}}} \mathbf{u}^\top \mathbf{M} \mathbf{u} < \min_{\mathbf{u} \in U_{\mathbf{y}}} \mathbf{u}^\top \mathbf{M} \mathbf{u} \, .$$

Let $\iota = \arg\max_{j \in \mathbb{N}_n} |v_j|$ be the index of a component of $\mathbf{v}$ of maximal magnitude. By our supposition $1 < |v_\iota|$. Since the constraints are orthogonal we have that

$$\frac{\partial}{\partial u_\iota} (\mathbf{u}^\top \mathbf{M} \mathbf{u})|_{\mathbf{v}} = 0$$

otherwise $\mathbf{v}$ is not a minimum. Thus computing the partial and upper bounding gives

$$|v_\iota| = |\sum_{j \neq \iota} \frac{M_{\iota j} v_j}{M_{\iota\iota}}| \leq \sum_{j \neq \iota} \frac{|M_{\iota j}||v_j|}{|M_{\iota\iota}|} \, . \tag{15}$$

Since $\mathbf{M}$ is a signed Laplacian it is diagonally dominant, and thus

$$\sum_{j\neq\iota}\frac{|M_{\iota j}|}{|M_{\iota\iota}|}\leq 1\,. \tag{16}$$

Therefore as $|v_\iota|\geq|v_j|$ for each $j\in\mathbb{N}_n$ inequalities (15) and (16) imply that $1<|v_\iota|=|v_j|$ for each $j\in J:=\{j\in\mathbb{N}_n : M_{\iota j}\neq 0\}$.

For each $j\in J$ the above argument may be iterated and then as we have assumed $\mathbf{M}$ is irreducible we may continue iterating to conclude that

$$1<|v_1|=\ldots=|v_n|\,.$$

Thus $\frac{\mathbf{v}}{|v_1|}\in\widetilde{U}_{\mathbf{y}}$ and as $\frac{\mathbf{v}^\top}{|v_1|}\mathbf{M}\frac{\mathbf{v}}{|v_1|}<\mathbf{v}^\top\mathbf{M}\mathbf{v}$ this contradicts the assumption that $\mathbf{v}$ is minimal hence (14) follows for $\mathbf{M}$ irreducible.

Now, alternatively, suppose $\mathbf{M}$ is reducible then there exists k irreducible matrices $\mathbf{M}^{(1)},\ldots,\mathbf{M}^{(k)}$ such that

$$\mathbf{u}^\top\mathbf{M}\mathbf{u}=\mathbf{u}^{(1)\,\top}\mathbf{M}^{(1)}\mathbf{u}^{(1)}+\ldots+\mathbf{u}^{(k)\,\top}\mathbf{M}^{(k)}\mathbf{u}^{(k)}$$

for all $\mathbf{u}\in\mathbb{R}^n$ and $\mathbf{u}=(u_1^{(1)},u_2^{(1)},\ldots,u_1^{(k)},\ldots,u_{i_k}^{(k)})$ with $i_1+\cdots+i_k=n$. We conclude by applying (14) to the k independent problems. □

Proof of Theorem 2: From the algorithm we have that for each $t\in\mathcal{M}\backslash 1$,

$$\mathbf{w}_{t+1}:=P(\mathcal{U}_t;\mathbf{w}_t)=\mathbf{w}_t+\frac{y_t-y_{\eta_t}-(\mathbf{w}_t(i_t)-\mathbf{w}_t(i_{\eta_t}))}{\|\mathbf{v}_{i_t}-\mathbf{v}_{i_{\eta_t}}\|^2}(\mathbf{v}_{i_t}-\mathbf{v}_{i_{\eta_t}})\,,$$

which is the projection of $\mathbf{w}_t$ to the hyperplane

$$\mathcal{U}_t:=\{\mathbf{u}\in\mathcal{H}(\mathbf{M}):\langle\mathbf{u},\mathbf{v}_{i_t}-\mathbf{v}_{i_{\eta_t}}\rangle=y_t-y_{\eta_t}\}=\{\mathbf{u}\in\mathcal{H}(\mathbf{M}):\mathbf{u}(i_t)-\mathbf{u}(i_{\eta_t})=y_t-y_{\eta_t}\},$$

as follows from (13). Thus for every $t\in\{2,\ldots,\ell\}$

$$\|\mathbf{w}_t-\mathbf{w}_{t+1}\|^2=\|\mathbf{u}-\mathbf{w}_t\|^2-\|\mathbf{u}-\mathbf{w}_{t+1}\|^2,\quad(\mathbf{u}\in\mathcal{U}_t)$$

by Lemma 4 for $t\in\mathcal{M}\backslash 1$ and trivially otherwise. Summing these telescoping equalities for $t=2,\ldots,\ell$ then removing from the sum on the left hand side the terms when $t\notin\mathcal{M}$ as they are zero and on the right bounding the term $-\|\mathbf{u}-\mathbf{w}_{\ell+1}\|^2$ by zero gives

$$\sum_{t\in\mathcal{M}\backslash 1}\|\mathbf{w}_t-\boldsymbol{w}_{t+1}\|^2\leq\|\mathbf{u}\|^2\ \text{ for all }\ \mathbf{u}\in\mathcal{U}^*:=\bigcap_{t\in\mathcal{M}\backslash 1}\mathcal{U}_t\,. \tag{17}$$

Now if a mistake occurs on trial $t\neq 1$ we have that $1\leq|y_t-y_{\eta_t}-\boldsymbol{w}_t(i_t)+\boldsymbol{w}_t(\eta_t)|$. Since $\boldsymbol{w}_{t+1}\in\mathcal{U}_t$, then $1\leq|\boldsymbol{w}_{t+1}(i_t)-\boldsymbol{w}_{t+1}(\eta_t)-\boldsymbol{w}_t(i_t)+\boldsymbol{w}_t(\eta_t)|=|\langle\boldsymbol{w}_{t+1}-\boldsymbol{w}_t,\mathbf{v}_{i_t}-\mathbf{v}_{i_{\eta_t}}\rangle|$. We then apply the Cauchy-Schwarz inequality to obtain that

$$1\leq\|\boldsymbol{w}_{t+1}-\boldsymbol{w}_t\|\|\mathbf{v}_{i_t}-\mathbf{v}_{i_{\eta_t}}\|. \tag{18}$$

Now substituting (18) into (17) gives

$$\sum_{t\in\mathcal{M}\backslash 1}\frac{1}{\left\|\mathbf{v}_{i_t}-\mathbf{v}_{i_{\eta_t}}\right\|^2}\leq\|\mathbf{u}\|^2\qquad(\mathbf{u}\in\mathcal{U}^*)\,. \tag{19}$$

Given $X\subset\mathcal{H}$ and $\rho>0$ such that there are $\mathcal{N}(X,\rho)<\infty$ balls denoted $Z_1,\ldots,Z_{\mathcal{N}(X,\rho)}$ of diameter ρ which cover X then define $\mathcal{F}(X,\rho)$ to be the set of the initial trial indices in which a mistake first occurred in a ball (excepting $t=1$) thus $t\in\mathcal{F}(X,\rho)$ if $t\in\mathcal{M}\backslash 1$ and there exists a j such that $\mathbf{v}_{i_t}\in Z_j$ and there does not exist an $s<t$ such that $\mathbf{v}_{i_s}\in Z_j$ and $s\in\mathcal{M}\backslash 1$. We lower bound the left hand side of (19) by removing from sum over trials those trials which are in $\mathcal{F}(X,\rho)$, hence

$$\sum_{t\in(\mathcal{M}\backslash\mathcal{F}(X,\rho))\backslash 1}\frac{1}{\left\|\mathbf{v}_{i_t}-\mathbf{v}_{i_{\eta_t}}\right\|^2}\leq\|\mathbf{u}\|^2\qquad(\mathbf{u}\in\mathcal{U}^*)\,. \tag{20}$$

Since for every two points $\mathbf{v},\mathbf{v}'$ in the same ball we have that $\|\mathbf{v}-\mathbf{v}'\|^2\leq\rho$ then

$$\left\|\mathbf{v}_{i_t}-\mathbf{v}_{i_{\eta_t}}\right\|^2\leq\rho$$

for $t\in(\mathcal{M}\backslash\mathcal{F}(X,\rho))\backslash 1$. Thus substituting the equation above into (20) we have

$$|\mathcal{M}|-|\mathcal{F}(X,\rho)|-1\leq\|\mathbf{u}\|^2\rho\qquad(\mathbf{u}\in\mathcal{U}^*)\,. \tag{21}$$

Substituting the upper bound $|\mathcal{F}(X,\rho)|\leq\mathcal{N}(X,\rho)$ into (21) it follows that the mistake bound (10) holds for $\mathbf{u}\in\mathcal{U}^*$. We proceed to show that if $\mathbf{u}\notin\mathcal{U}^*$ and $\mathbf{u}(i_t)y_t\geq 1$ for all $t\in\mathcal{M}$ then there exists a proxy $\mathbf{u}'\in\mathcal{U}^*$ with $\|\mathbf{u}'\|^2\leq\|\mathbf{u}\|^2$ which hence proves the theorem.

If $\mathbf{u}\in\mathbb{R}^n$ and $\mathbf{u}(i_t)y_t\geq 1$ for $t\in\mathcal{M}$ then by Theorem 5 there exists a $\mathbf{u}'\in\mathbb{R}^n$ such that

$$(\mathbf{u}')^\top\mathbf{M}(\mathbf{u}')\leq\mathbf{u}^\top\mathbf{M}\mathbf{u}\text{ and }\mathbf{u}'(i_t)=y_t\text{ for }t\in\mathcal{M}.$$

If $\mathbf{M}$ is positive definite then $\mathbf{u}'\in\mathcal{U}^*\subset\mathbb{R}^n=\mathcal{H}(\mathbf{M})$ and we are done; otherwise $\mathbf{M}$ is an irreducible Laplacian. If $\mathbf{M}$ is an irreducible Laplacian then from (2) the vector $\mathbf{1}$ spans the null space of $\mathcal{H}(\mathbf{M})$ and thus

$$\mathbf{z}\in\mathbb{R}^n\text{ and }\mathbf{1}^\top\mathbf{z}=0\Longrightarrow\mathbf{z}\in\mathcal{H}(\mathbf{M})\,. \tag{22}$$

Set $\mathbf{u}''\coloneqq\mathbf{u}'-\mathbf{1}(\frac{\mathbf{1}^\top\mathbf{u}'}{n})$ then (22) implies $\mathbf{u}''\in\mathcal{H}(\mathbf{M})$ and

$$(\mathbf{u}'')^\top\mathbf{M}(\mathbf{u}'')=(\mathbf{u}')^\top\mathbf{M}(\mathbf{u}')\leq\mathbf{u}^\top\mathbf{M}\mathbf{u}\,.$$

Finally $\mathbf{u}''\in\mathcal{U}^*$ since $\mathbf{u}''(i_t)-\mathbf{u}''(\eta_t)=\mathbf{u}'(i_t)-(\frac{\mathbf{1}^\top\mathbf{u}'}{n})-(\mathbf{u}'(\eta_t)-(\frac{\mathbf{1}^\top\mathbf{u}'}{n}))=\mathbf{u}'(i_t)-\mathbf{u}'(\eta_t)=y_t-y_{\eta_t}$, holds for $t\in\mathcal{M}\backslash 1$. □

5 The Exponential Embedding

In transductive and semi-supervised learning if a graph is not inherent in the problem it is necessary to build the graph from the data. The usual procedure is to use a discrepancy function over the data and build a graph using edge weights derived by the k-NN, ϵ-ball, or the exponential (also known as the heat kernel [22, 2]) embedding.

Our model here is that a learning problem is determined by a possibly infinite set $\mathcal{X}$ and a *label function* $\mathcal{Y} : \mathcal{X} \to \{-1, 1\}$ which are both unknown to the learner. Structure is imposed on $\mathcal{X}$ through a discrepancy $d : \mathcal{X} \times \mathcal{X} \to [0, \infty]$. The learner is initially given a finite input set $X \subseteq \mathcal{X}$ and $d(X, X)$. Subsequently, the learner will predict a subset of the labels of X in an online fashion. In this section we study the performance of POUNCE if we predict by building a graph Laplacian using the exponential embedding of X.

Definition 6. *If $X = \{x_1, \ldots, x_n\} \subseteq \mathcal{X}$ is an indexed finite set, $d : \mathcal{X} \times \mathcal{X} \to [0, \infty]$ is a discrepancy function, and $a > 0$ is a scale parameter then the* exponential embedding *of (X, d, a) is the map $F_{(X,d,a)} : X \to \mathcal{H}(\mathbf{G}^a)$ constructed as follows. First define the Laplacian matrix*

$$G^a_{ij} := \begin{cases} -e^{-ad(x_i,x_j)} & i \neq j \\ \sum_{k \neq i}^{n} e^{-ad(x_i,x_k)} & i = j \end{cases} \tag{23}$$

then define the map from X to the coordinate spanning set $\mathcal{V}_{\mathbf{G}^a} \subset \mathcal{H}(\mathbf{G}^a)$ (recall (6)),

$$F_{(X,d,a)}(x_i) := \mathbf{v}_i = (\mathbf{G}^a)^+ \mathbf{e}_i \,. \tag{24}$$

The bound in the following theorem is based on label-dependent cover numbers. Thus we will require the following preliminary definitions. The points $x, x' \in X$ are *ρ-path-connected* if $d(x, x') \leq \rho$ or if there exists a point $x'' \in X$ such that $d(x, x'') \leq \rho$ and x'', x' are ρ-path-connected. The set X is *ρ-path-connected* if all pairs of points in X are thus connected. A set X is *connected* if there exists a $\rho > 0$ such that it is ρ-path-connected. The component covering number, $\mathcal{C}(X, \rho, d)$ is the minimal number of ρ-path-connected subsets (components) of X that cover X, thus

$$\mathcal{C}(X, \rho, d) := \min_{\{X'_i\}_{i=1}^k \subseteq X} \left\{ k \in \mathbb{N}^+ : X'_i \text{ is } \rho\text{-path-connected for } i \in \mathbb{N}_k, \cup_{i=1}^k X'_i = X \right\} . \tag{25}$$

Observe that $\mathcal{C}(X, \rho, d) \leq \mathcal{N}(X, \rho, d)$ and if $X \subseteq X'$ then $\mathcal{N}(X, \rho, d) \leq \mathcal{N}(X', \rho, d)$ but a superset of X may decrease or increase the component covering number. The label function $\mathcal{Y} : X \to \{-1, 1\}$ maps the input set to label set. The *separating-cover (component separating-cover)* number denoted $\mathcal{N}^\circ(X, \mathcal{Y}, d)$ ($\mathcal{C}^\circ(X, \mathcal{Y}, d)$) is the minimal number of sets of maximum diameter ρ (ρ-path-connected) that contain X such that every set contains points only with a single label and every two points with differing labels are more than ρ distant.

Definition 7. *If $X \subseteq \mathcal{X}$ is a set, $\mathcal{Y} : \mathcal{X} \to \{-1, 1\}$ is a label function and $d : \mathcal{X} \times \mathcal{X} \to [0, \infty]$ is a discrepancy function then the* separating-cover *number is*

$$\mathcal{N}^\circ(X, \mathcal{Y}, d) := \min_{0<\rho<\delta} \mathcal{N}(X, \rho, d) \tag{26}$$

and the component separating-cover *number is*

$$\mathcal{C}^\circ(X, \mathcal{Y}, d) := \min_{0<\rho<\delta} \mathcal{C}(X, \rho, d) \tag{27}$$

with $\delta = \inf\{d(x^+, x^-) : x^+ \in \mathcal{Y}^{-1}(1) \cap X,\, x^- \in \mathcal{Y}^{-1}(-1) \cap X\}$ and if the infimum is zero then the cover numbers are ∞.

Given an exponential embedding $F_{(X,d,a)}$ we consider the performance of the POUNCE algorithm in the following theorem as the parameter $a \to \infty$. This "tuning" of a minimizes the margin term in (10) thus increasing the cover term. Thus this tuning is optimized for the case in which the data is "aligned" with a small component separating-cover. If the data is not well-aligned a less severe tuning may be more useful in practice.

Theorem 8. *If $X \subseteq \mathcal{X}$ is a finite set, $d : \mathcal{X} \times \mathcal{X} \to [0, \infty]$ is a discrepancy function and X is connected then for any label function $\mathcal{Y} : \mathcal{X} \to \{-1, 1\}$ and any sequence of labeled examples $\{(x_{i_t}, \mathcal{Y}(x_{i_t}))\}_{t=1}^{\ell} \in (X \times \{-1, 1\})^\ell$ there exists an $a' > 0$ such that for all $a > a'$ the cumulative mistakes $|\mathcal{M}|$ of the* POUNCE *algorithm on the embedded sequence $\{(F_{(X,d,a)}(x_{i_t}), \mathcal{Y}(x_{i_t}))\}_{t=1}^{\ell}$ are bounded by*

$$|\mathcal{M}| \le \mathcal{C}^\circ(X, \mathcal{Y}, d) + 1 \le \mathcal{N}^\circ(\mathcal{X}, \mathcal{Y}, d) + 1\,. \tag{28}$$

Proof. As X is finite there exists $\rho, \epsilon > 0$ and a component separating cover $\mathcal{C}^\circ(X, \mathcal{Y}, d)$ of X such that $\mathcal{Y}(x) \neq \mathcal{Y}(x') \to d(x, x') > \rho + \epsilon$. Define $\mathcal{H}(\mathbf{G}^a)$ from X via (23). Since X is connected, $\mathbf{G}^a$ is irreducible. Given $x_p, x_q \in X$ in the same ρ-path-connected set, there exists a path P from x_p to x_q along fewer than $|X|$ edges such that the discrepancy on each edge is no more than ρ. Therefore the resistance $1/G^a_{ij}$ on each edge (i, j) of the embedding of path P into $\mathcal{H}(\mathbf{G}^a)$ is smaller than $e^{a\rho}$; thus the effective resistance (recalling (3)) between vertex p and q is upper bounded by $\|\mathbf{v}_p - \mathbf{v}_q\|^2_{\mathbf{G}^a} \le |X| e^{a\rho}$, as follows from Rayleigh's monotonicity law (see [15, Corollary 3.1] also [7]). Thus we can cover $\mathcal{V}_{\mathbf{G}^a}$ such that

$$\mathcal{N}(\mathcal{V}_{\mathbf{G}^a}, |X| e^{a\rho}) \le \mathcal{C}^\circ(X, \mathcal{Y}, d)\,. \tag{29}$$

Therefore from Theorem 2 we may bound the mistakes of the algorithm by

$$|\mathcal{M}| \le \mathcal{N}(\mathcal{V}_{\mathbf{G}^a}, |X| e^{a\rho}) + \|\mathbf{u}^*\|^2 |X| e^{a\rho} + 1\,, \tag{30}$$

with $\mathbf{u}^*(i) = \mathcal{Y}(x_i)$ for $i = 1, \ldots, |X|$. We upper bound $\|\mathbf{u}^*\|^2$ using the fact that x and x' in the same ρ-path-connected set have the same label,

$$\|\mathbf{u}^*\|^2 = \sum_{1 \le i \le j \le n} (u^*_i - u^*_j)^2 e^{-a d(x_i, x_j)} = \sum_{x_i, x_j \in X : \mathcal{Y}(x_i) \neq \mathcal{Y}(x_j)} 4 e^{-a d(x_i, x_j)} \le 4 |X|^2 e^{-a(\rho + \epsilon)}. \tag{31}$$

We proceed by substituting the upper bounds (29) and (31) into (30) to give $|\mathcal{M}| \leq \mathcal{C}^\circ(X, \mathcal{Y}, d)+1$, for all $a > \frac{3 \ln 4|X|}{\epsilon}$ as there cannot be a fractional mistake. Finally the assumption $X \subseteq \mathcal{X}$ implies that $\mathcal{C}^\circ(X, \mathcal{Y}, d) \leq \mathcal{N}^\circ(\mathcal{X}, \mathcal{Y}, d)$. □

An interpretation of the above result is that the separating-cover number $\mathcal{N}^\circ(\mathcal{X}, \mathcal{Y}, d)$ is an upper bound which is independent of prior knowledge of a particular set X' to be predicted. However, a supersample $X \supseteq X'$ may potentially induce a component separating-cover (e.g., see Figure 1) such that $\mathcal{C}^\circ(X, \mathcal{Y}, d) \ll \mathcal{N}^\circ(\mathcal{X}, \mathcal{Y}, d)$. Therefore the prior knowledge of X reduces the mistake bound by $\mathcal{N}^\circ(\mathcal{X}, \mathcal{Y}, d) - \mathcal{C}^\circ(X, \mathcal{Y}, d)$ a possibly significant gain from prior knowledge of the input space in this idealized learning scenario.

We observe that the separating-cover number $\mathcal{N}^\circ$ is also an upper bound on the number of mistakes incurred by the online 1-"nearest neighbors" algorithm, because once a mistake is made in a given ball, that mistaken point is always nearer to any other point in that ball than to a point of an opposite label. Thus there can be no more mistakes than balls in the cover. Given a discrepancy function d, there then exists an $a' > 0$ such that the component separating-cover number $\mathcal{C}^\circ$ is the mistake bound of graph geodesic 1-"nearest neighbors" for every discrepancy $d'_a = e^{ad}$ with $a > a'$. These upper bounds are tight in so far as an adversary may select a discrepancy such that a mistake is forced for every ball (component). Thus the bounds in Theorem 8 are tight up to a single additional mistake.

The preceding analysis connects the mistake bound analysis of the exponential embedding to graph geodesic nearest neighbors as $a \to \infty$. For small a the comparison may mislead as there exists a family of unweighted graphs such that the mistakes of POUNCE is upper bounded by a constant, while the mistakes of geodesic nearest neighbors is linear in the size of the graph as follows from [15, Section 5.1].

The Value of Unlabeled Data

Does unlabeled data help the learner in our framework? In the framework of this section the *initially* unlabeled data is just the input set X and the discrepancy d given to the learner before prediction. Can we obtain similar mistake bounds for predicting in X if instead it is revealed to the learner sequentially as we predict? In the following example we construct a problem for which any algorithm which does not preview the unlabeled input set will incur mistakes linear in the data set size in expectation whereas POUNCE will make no more than three mistakes.

Consider the following learning task illustrated in Figure 4. The task is generated at random as follows. The n points from $\mathbb{R}^2$ to be predicted are at $\{(1,1),(2,1),\ldots,(n,1)\}$. An additional, $4n$ points are then situated at the loci $\{(1,0),(3/2,0),\ldots,(n,0)\}$ and at $\{(1,2),(3/2,2),\ldots,(n,2)\}$. Each of the initial n points is labeled independently $+1$ or -1 with equal probability. We denote the first-coordinates of the positively (negatively) labeled points $\{a_1^+,\ldots,a_k^+\}$ ($\{a_1^-,\ldots,a_{n-k}^-\}$) and then generate points at $\{(a_1^+,1/2),\ldots,(a_k^+,1/2)\}$ and also generate the points $\{(a_1^-,3/2),\ldots,(a_{n-k}^-,3/2)\}$ which path-connect to the their labels. Thus our task has $6n$ points in total and we only consider prediction of

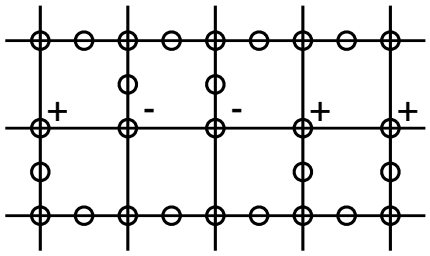

Fig. 4. Two (1/2)-Path-Connected Sets

the initial n. Any algorithm which does not preview the initially unlabeled data must incur $n/2$ expected mistakes whereas using Euclidean distance as a discrepancy the positively labeled and negatively labeled points are separated into two (1/2)-path-connected components with no points of differing labels less than a unit distant. Thus from Theorem 8, POUNCE incurs no more than 3 mistakes. Consequently, we observe that although there exists an algorithm that obtains the bound $|\mathcal{M}| \leq \mathcal{N}^{\circ}(X, \mathcal{Y}, d)$ without prior knowledge of its input set X, no algorithm may exist that obtains either $|\mathcal{M}| \leq \mathcal{C}^{\circ}(X, \mathcal{Y}, d) + 1$ or (10) without a preview of X or equivalently $\mathbf{M}$, respectively.

6 Discussion

We have presented a novel perceptron-like algorithm POUNCE. We've given a mistake bound analysis of POUNCE which builds on the classic Novikoff analysis via a cover number to provide a finer measure of the structure of the input space. When the input space corresponds to an embedding via a signed Laplacian and its cover is relatively "small," we may significantly improve over the traditional analysis. This work is a continuation of the researches begun in [16, 15], and as such it improves the previous bound [15, Theorem 4.2] at a minimum by a factor two[1] except for an additive constant of "1" even when we cover the space with a single "ball." The improvement in bound may be arbitrarily large as shown by the three-cluster example in Section 2.2. Furthermore this improvement cannot be obtained by the perceptron [14].

Although the bounds for predicting the online labeling of "small" diameter graphs improve on those given by a straightforward application of the classical halving algorithm, the bounds presented here are weaker than the halving algorithm for "large" diameter graphs as exemplified by an n-vertex line graph (a simple path) [15, p. 8]. Here we can see that a straightforward application of the halving algorithm to the concept class of labelings of a line graph with a cut-size of one leads to a mistake bound of $O(\log n)$. In contrast, the application of Theorem 2 using a cover of $O(\sqrt{n})$ line segments each of diameter $O(\sqrt{n})$ gives the suboptimal mistake bound for POUNCE of $O(\sqrt{n})$; this however improves on the bound for the perceptron of $O(n)$ in [15]. Thus this leaves as an open

[1] There are subtleties in an exact comparison, one of which is the issue of resistance "radius" versus resistance diameter. Surprisingly these may be asymptotically equivalent even in unweighted graphs (see the "flower" graph example of [15, p. 5]).

question whether there is an efficient algorithm which incorporates the strengths of a halving algorithm based analysis along with the analysis presented here.

Acknowledgments. I would like to thank Massimiliano Pontil for useful discussions and anonymous referees for valuable comments.

References

1. N. Aronszajn. Theory of reproducing kernels. *Trans. Amer. Math. Soc.*, 68:337–404, 1950.
2. M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *J. Mach. Learn. Res.*, 7:2399–2434, 2006.
3. A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *Proc. 18th International Conf. on Machine Learning*, 2001.
4. N. Cesa-Bianchi, A. Conconi, and C. Gentile. A second-order perceptron algorithm. *SIAM J. Comput.*, 34(3):640–668, 2005.
5. O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.
6. K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *J. Mach. Learn. Res.*, 7:551–585, 2006.
7. P. Doyle and J. Snell. *Random walks and electric networks*. MAA, 1984.
8. Y. Freund and R. E. Schapire. Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277–296, 1999.
9. S. R. Galeano and M. Herbster. A fast method to predict the labeling of a tree. In *ECML 2007 Workshop on Graph Labeling*, 2007.
10. C. Gentile. The robustness of the p-norm algorithms. *Machine Learning*, 53(3):265–299, 2003.
11. A. Goldberg, X. Zhu, and S. Wright. Dissimilarity in graph-based semi-supervised classification. In *11th Intl. Conf. on Artificial Intelligence and Statistics*, 2007.
12. R. Herbrich and R. C. Williamson. Algorithmic luckiness. *J. Mach. Learn. Res.*, 3:175–212, 2003.
13. M. Herbster. Learning additive models online with fast evaluating kernels. In Proc. of the 14th Annual Conf. on Computational Learning Theory, 2001.
14. M. Herbster. A linear lower bound for the perceptron for input sets of constant cardinality. Research Note RN/08/03, University College London, London, 2008.
15. M. Herbster and M. Pontil. Prediction on a graph with a perceptron. In *Advances in Neural Information Processing Systems 19*, pages 577–584. MIT Press, 2007.
16. M. Herbster, M. Pontil, and L. Wainer. Online learning over graphs. In *Proc. 22nd Intl Conf. on Machine Learning*, pages 305–312. ACM Press, New York, NY, 2005.
17. R. Johnson and T. Zhang. On the effectiveness of laplacian normalization for graph semi-supervised learning. *J. Mach. Learn. Res.*, 8:1489–1517, 2007.
18. J. Kivinen and M. Warmuth. Additive versus exponentiated gradient updates for linear prediction. In *Proc. 27th Annu. ACM Symp. on Theory of Computing*, 1995.
19. D. Klein and M. Randić. Resistance distance. *J. of Math. Chem.*, 12(1):81–95, 1993.
20. A. Novikoff. On convergence proofs for perceptrons. In *Proc. Sympos. Math. Theory of Automata*, Polytechnic Press of Polytechnic Inst. of Brooklyn, NY, 1963.
21. R. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.
22. X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *20th Intl. Conf. on Machine Learning*, 2003.