Relative Loss Bounds and Polynomial-time Predictions for the K-LMS-NET Algorithm

Mark Herbster[⋆]

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

Abstract. We consider a two-layer network algorithm. The first layer consists of an uncountable number of linear units. Each linear unit is an *LMS* algorithm whose inputs are first "kernelized." Each unit is indexed by the value of a parameter corresponding to a parameterized reproducing kernel. The first-layer outputs are then connected to an *exponential weights* algorithm which combines them to produce the final output. We give loss bounds for this algorithm; and for specific applications to prediction relative to the best convex combination of kernels, and the best width of a Gaussian kernel. The algorithm's predictions require the computation of an expectation which is a quotient of integrals as seen in a variety of Bayesian inference problems. Typically this computational problem is tackled by MCMC, importance sampling, and other sampling techniques for which there are few polynomial time guarantees of the quality of the approximation in general and none for our problem specifically. We develop a novel deterministic polynomial time approximation scheme for the computations of expectations considered in this paper.

1 Introduction

We give performance guarantees and a tractable method of computation for the two-layer network algorithm K-LMS-NET. The performance guarantees measure online performance in a non-statistical learning framework introduced by Littlestone [13, 14]. Here, learning proceeds in trials $t = 1, 2, \ldots, \ell$. In each trial t the algorithm receives a *pattern* x_t. It then gives a prediction denoted $\hat{y}_t \in \mathbb{R}$. The algorithm then receives an *outcome* $y_t \in \mathbb{R}$, and incurs a loss $L(y_t, \hat{y}_t)$ measuring the discrepancy between y_t and $\hat{y}_t$; in this paper $L(y_t, \hat{y}_t) = (y_t - \hat{y}_t)^2$. A *relative loss bound* performance guarantee bounds the cumulative loss of the algorithm with the cumulative loss of any member $c : \mathcal{X} \rightarrow \mathbb{R}$ of a comparison class $\mathcal{C}$ of predictors plus an additional term. These bounds are of the following form, for all data sequences $S = \langle (x_1, y_1), (x_2, y_2), \ldots, (x_\ell, y_\ell) \rangle$,

$$\sum_{t=1}^{\ell} L(y_t, \hat{y}_t) \leq \sum_{t=1}^{\ell} L(y_t, c(x_t)) + O(r(S, \mathcal{C}, c)) \quad \forall c \in \mathcal{C}$$

[⋆] This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778.

where $r(S, \mathcal{C}, c)$ is known as the *regret*, since it measures our "regret" at using our algorithm versus the "best" predictor c in the comparison class. In the ideal case the regret is a slowly growing function of the data sequence, the comparison class, and the particular predictor. Surprisingly, such bounds are possible without probabilistic assumptions on the sequence of examples.

The architecture of the K-LMS-NET algorithm is a simple chaining of two well-known online algorithms. The first layer consists of an uncountable number of linear units. Each unit is an LMS algorithm [4] whose inputs are first "kernelized." Each unit is indexed by the value of a parameter ($\alpha \in [0, 1]$) corresponding to a parameterized reproducing kernel [2], for example a Gaussian kernel and its width $k_\alpha(\mathbf{v}, \mathbf{w}) = e^{-\alpha\|\mathbf{v}-\mathbf{w}\|^2}$. The first-layer outputs are then directed to an *exponential weights* algorithm [20, 14, 12] which combines them to produce the final output (prediction). This topology gives an algorithm whose comparison class (hypothesis space) is a union of linear spaces of functions.

The results of this research are twofold. First, we give a general bound for the K-LMS-NET algorithm. This general bound is then applied to the problem of predicting almost as well as any function i) from the space defined by the "best" convex combination of two kernels and ii) from the space defined by the "best" width of an isotropic Gaussian kernel. Second, though the second layer combines an uncountable number of outputs from the first layer we show that the final prediction may be well-approximated in polynomial time. This prediction is an expectation (5) whose form is a quotient of integrals which does not have an analytic closed form; thus we resort to a novel sampling scheme. The significance of our sampler is that it produces a provably polynomial time approximation to our predictions. The sampler is deterministic, and relies on finding the critical points of the functions to be integrated; this leads to a limitation on the types of parameterized kernels for which we can give predictions in polynomial time. The applications for which we give bounds are among those whose predictions may be approximated in polynomial time by our sampling scheme.

1.1 Related Work

In [7] Freund applied the *exponential weights* algorithm to predicting as well as the best "biased coin" that modeled a data sequence, with an uncountable set of predictors each corresponding to a probability of "heads." In [21] an algorithm similar to ridge regression was given, where the set of predictors corresponded to each linear function on $\mathbb{R}^n$; these were then combined with an *exponential weights* algorithm. For those algorithms exact computation of the prediction was possible; exact computation, however, is not possible for the K-LMS-NET algorithm.

In [4], relative loss bounds are proven for the classical LMS algorithm; the bounds naturally apply to kernel LMS. Some recent relative-loss bounds for variants of kernel LMS have appeared in [11, 9]. In contrast, our kernel function has a free parameter; hence our comparison class (hypothesis space) is not a single kernel space, but a union of kernel spaces.

The problem of learning the best parameters of a kernel function has been modeled as a regularized optimization problem in [16, 15, 23]. Methods based on Gaussian process regression have proven to be practical for learning or predicting with a mixture of kernel parameterizations, for which we cite only a few of the many offline [22, 8] and online algorithms [6, 19] developed. The parameterized kernel function now corresponds to a parameterized covariance function. A key difference in focus is that our free parameter is a one-dimensional scalar, whereas in Gaussian process regression the free parameter vector is often in the hundreds of dimensions. The one-dimensional case we consider is certainly much simpler than the multidimensional case. However, we make no statistical assumptions on the data generation process, we give non-asymptotic relative loss bounds and we observe that even in the "simple" one-dimensional case it is not obvious how to sample so that predictions of *guaranteed* accuracy are produced in polynomial time.

The predictions (5) of K-LMS-NET algorithm are of the following (simplified) form,

$$\hat{y}_t = \frac{\int_0^1 \hat{\mathbf{y}}_t^i(\alpha) \exp(-L_{[1,t]}(\alpha)) d\alpha}{\int_0^1 \exp(-L_{[1,t]}(\alpha)) d\alpha} \quad ; \tag{1}$$

here $\hat{\mathbf{y}}_t^i(\alpha)$ and $L_{[1,t]}(\alpha)$ are the outputs and cumulative losses, respectively, of each of the kernel LMS algorithms at time t. In the applications to be discussed $\hat{\mathbf{y}}_t^i(\alpha)$ and $L_{[1,t]}(\alpha)$ reduce to either polynomials or to polynomials after a change in variables. The problem of estimating such expectations is common in Bayesian statistics. Since we cannot expect to compute $\hat{y}_t$ exactly, we consider that a good polynomial time approximation scheme for $\hat{y}_t$ should have the following property: for every $\epsilon \in (0, 1)$, an *absolute* error approximation $\bar{y}_t$, should satisfy $|\bar{y}_t - \hat{y}_t| \leq \epsilon$ and be computable in time polynomial in $O(\frac{1}{\epsilon})$. It is also natural to extend the previous to a randomized approximations schemes; however the scheme we produce is fully deterministic. Hoeffding bounds [10] allow one to produce an absolute error approximation for the Monte-Carlo integration of $\int f \, d\mu$, with $\bar{y} = \frac{1}{n}\sum_{i=1}^n f(x_i)$ and where x_i is sampled from the probability measure μ and $\epsilon = O(\frac{1}{\sqrt{n}})$. Hoeffding bounds are not applicable to the approximation of (1) as it is a nontrivial problem in itself to produce samples from the distribution $\frac{\exp(-L_{[1,t]}(\alpha))}{\int_0^1 \exp(-L_{[1,t]}(\alpha))d\alpha}$ in polynomial time; nor can we can we apply Hoeffding bounds individually to the integrals in the numerator and the denominator, as absolute error bounds do not "divide" naturally. A variety of other bounds have been proven for numeric integration within the *information-based complexity* framework [18]; however, as these are absolute error bounds they are likewise not applicable. Another approach to the approximation of equations of the form (1) which has proven useful in practice for similar applications is to use one of the many variants of MCMC sampling [1]. We are not aware of any bounds for MCMC sampling methods which give a polynomial time guarantee for randomized approximation to $\hat{y}_t$ which are applicable to this research. Our sampling methodology is discussed in Sect. 3; the key to our method is to produce a sampler for 1-d integrals with a provable *relative* error approximation, which unlike the absolute error approximation, is, loosely speaking, closed under division.

1.2 Preliminaries

The symbol $\mathcal{X}$ denotes an abstract space, for example $\mathcal{X}$ could be a set of strings. Given a vector space $\langle V; +\rangle$, the sum of two subsets F and G of V is defined by $F + G = \{f + g : f \in F, g \in G\}$. A Hilbert space $\mathcal{H}$ denotes a complete inner product space. The inner product between vectors $\mathbf{v}$ and $\mathbf{w}$ in $\mathcal{H}$ is denoted by $\langle \mathbf{v}, \mathbf{w}\rangle$ and the norm by $\|\mathbf{v}\|$. In this paper, we will consider Hilbert spaces determined by a reproducing kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. The prehilbert space induced by kernel k is the set $H_k = \operatorname{span}(\{k(x, \cdot)\}_{\forall x \in \mathcal{X}})$ and the inner product of $f = \sum_{i=1}^{m} \beta_i k(x_i, \cdot)$ and $g = \sum_{j=1}^{n} \beta'_i k(x'_j, \cdot)$ is $\langle f, g\rangle = \sum_{i=1}^{m}\sum_{j=1}^{n} \beta_i \beta'_j k(x_i, x'_j)$. The completion of H_k is denoted $\mathcal{H}_k$. Two kernels $k_0 : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $k_1 : \mathcal{X}' \times \mathcal{X}' \to \mathbb{R}$ are termed *domain compatible* if $\mathcal{X} = \mathcal{X}'$. The *reproducing* property of the kernel is that given any $f \in \mathcal{H}_k$ and any $x \in \mathcal{X}$ then $f(x) = \langle f(\cdot), k(x, \cdot)\rangle$; other useful properties of reproducing kernels, and introductory material may be found in [5]. In this paper we are particularly interested in parameterized kernels k_α with an associated Hilbert Space $\mathcal{H}_\alpha$, inner product $\langle \cdot, \cdot \rangle_\alpha$, and norm $\|\cdot\|_\alpha$, for every $\alpha \in [0, 1]$. We denote the Lebesque measure of a set A by $\mu(A)$.

An *absolute ϵ-approximation* of $y \in \mathbb{R}$ by $\hat{y} \in \mathbb{R}$ satisfies

$$|y - \hat{y}| \leq \epsilon \tag{2}$$

denoted by $\hat{y} \overset{a}{\approx}_\epsilon y$. A *relative ϵ-approximation* of $y \in \mathbb{R}^+$ by $\hat{y} \in \mathbb{R}^+$ satisfies

$$(1 - \epsilon)y \leq \hat{y} \leq (1 + \epsilon)y, \tag{3}$$

which is denoted by $\hat{y} \overset{r}{\approx}_\epsilon y$. A polynomial ϵ-approximation scheme requires for each $\epsilon \in (0, 1)$ that we can compute $\hat{y}$ s.t. $\hat{y} \approx_\epsilon y$ in time $O(\frac{1}{\epsilon})$. For simplicity, we describe the time complexity of our algorithms in terms of a naive real-valued model of computation, where arithmetic operations on real numbers, e.g., addition, exponentiation, kernel evaluation, etc., all require $O(1)$ "steps."

2 The K-LMS-NET Algorithm

The following general bound for the K-LMS-NET algorithm is applied to predicting as well as the convex combination of two kernels and predicting as well as the best width of a Gaussian kernel over a discretized domain.

Theorem 1. *The* K-LMS-NET *algorithm with parameterized kernel function k_α ($\alpha \in [0, 1]$) with any data sequence $\langle (x_1, y_1), (x_2, y_2), \ldots, (x_\ell, y_\ell)\rangle \in (\mathcal{X}, [r_1, r_2])^\ell$ when the algorithm is tuned with constants r_1, r_2, and η, the total square loss of the algorithm will satisfy*

$$\sum_{t=1}^{\ell} L(y_t, \hat{y}_t) \leq \sup_{\alpha \in \mathcal{A}} \left[\sum_{t=1}^{\ell} L(y_t, h_\alpha(x_t))\right] + 2\sqrt{\hat{L}_\mathcal{A}}\hat{H}_\mathcal{A}\hat{X}_\mathcal{A} + \hat{H}_\mathcal{A}^2 \hat{X}_\mathcal{A}^2 + 2(r_2 - r_1)^2 \ln \frac{1}{\mu(\mathcal{A})} \tag{8}$$

for all measurable sets $\mathcal{A} \subseteq [0, 1]$ for all tuples of functions $(h_\alpha)_{\alpha \in \mathcal{A}} \in \prod_{\alpha \in \mathcal{A}} \mathcal{H}_\alpha$ and for all constants $\hat{L}_\mathcal{A}, \hat{H}_\mathcal{A}$, and, $\hat{X}_\mathcal{A}$, where for all $\alpha \in \mathcal{A}$ the following four conditions must hold: $\sum_{t=1}^{\ell} L(y_t, h_\alpha(x_t)) \leq \hat{L}_\mathcal{A}$, $\|h_\alpha\|_\alpha^2 \leq \hat{H}_\mathcal{A}^2$, $\forall t : k_\alpha(x_t, x_t) \leq \hat{X}_\mathcal{A}^2$, and $\eta = [1 + \frac{\sqrt{\hat{L}_\mathcal{A}}}{\hat{H}_\mathcal{A}\hat{X}_\mathcal{A}})\hat{X}_\mathcal{A}^2]^{-1}$.

```
Parameters: $\mathcal{X}$: a pattern space;
            $k_\alpha : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$: a parameterized kernel function ($\alpha \in [0,1]$);
            $\{\mathcal{H}_\alpha\}$ : a set of Hilbert spaces induced by $k_\alpha$;
            $\eta$ : a learning rate; $[r_1, r_2]$ : an outcome range.
Data: An online sequence $\langle (x_1, y_1), (x_2, y_2), \ldots, (x_\ell, y_\ell) \rangle \in (\mathcal{X}, [r_1, r_2])^\ell$.
Initialization: $r = (r_2 - r_1)$, $\mathbf{w}^i_{\alpha,1}(x) = \mathbf{0}$, $\mathbf{w}^{ii}_1(\alpha) = 1$,
                $\Phi^i(x) = \max(r_1, \min(r_2, x))$ ; $\Phi^{ii}_t(w) = \max(\exp(-\frac{t}{2}), w)$.
for $t = 1, \ldots, \ell$ do
    Predict: receive $x_t$,
        $\hat{\mathbf{y}}^i_t(\alpha) = \mathbf{w}^i_{\alpha,t}(x_t) = \eta \sum_{j=1}^{t-1} (y_j - \hat{\mathbf{y}}^i_j(\alpha)) k_\alpha(x_j, x_t)$    (4)
        $\hat{y}_t = \frac{\int_0^1 \mathbf{w}^{ii}_t(\alpha) \Phi^i(\hat{\mathbf{y}}^i_t(\alpha)) d\alpha}{\int_0^1 \mathbf{w}^{ii}_t(\alpha) d\alpha}$    (5)
    Update: receive $y_t$,
        $\mathbf{w}^i_{\alpha,t+1}(x) = \mathbf{w}^i_{\alpha,t}(x) + \eta(y_t - \hat{\mathbf{y}}^i_t(\alpha)) k_\alpha(x_t, x)$    (6)
        $L_{[1,t]}(\alpha) = L_{[1,t-1]}(\alpha) + (y_t - \hat{\mathbf{y}}^i_t(\alpha))^2$
        $\mathbf{w}^{ii}_{t+1}(\alpha) = \Phi^{ii}_t(\exp(\frac{1}{2r^2} L_{[1,t]}(\alpha)))$    (7)
end
```

Algorithm 1: K-LMS-NET algorithm

The bound is a straightforward chaining of the well known loss bounds [4] of the LMS (GD) algorithm and a variant of the *exponential weights* algorithm [20, 14, 12] that implements direct clipping of the inputs to guarantee a loss bound and an amortized clipping of the cumulative loss to enable efficient sampling.

The generic bound given is neither a pure relative loss bound nor does it give an indication of whether the K-LMS-NET is polynomially tractable for a particular parameterized kernel. The bound is not a "pure" relative loss bound insofar as the regret (the final term of (8)) for any particular predictor is infinite (since $\mathcal{A}$ is then a point set thus $\frac{1}{\mu(\mathcal{A})} = \infty$). A pure relative loss bound may be given if we can determine how the loss and the norm of a particular predictor in $\mathcal{H}_{\alpha'}$ is related to near comparable predictors in $\mathcal{H}_{\alpha''}$ when $|\alpha' - \alpha''|$ is small. In the following we "flesh out" the generic bound of Theorem 1 by giving pure relative loss bounds for two particular parameterized kernels; then in Sect. 3 we sketch how the prediction with these kernels is computable by a polynomial-time approximation scheme.

2.1 Applications to Specific Parameterized Kernels

Relative loss bounds are given in Theorems 4 and 5 for a parameterized kernel which is a parameterized convex combination of kernels and for a Gaussian kernel with a parameterized width, respectively. Each of these bounds given are in terms of adjunct norms $\mathcal{C}(\cdot)$ and $\mathcal{S}(\cdot)$ on the kernel spaces rather than the norms inherited from the underlying kernel space. The norms $\mathcal{C}(\cdot)$ and $\mathcal{S}(\cdot)$ are tighter and weaker, respectively, than their inherited norm $\|\cdot\|_\alpha$. The proofs of the theorems follow directly from Theorem 1 in conjunction with Lemmas 5 and 6 which appear in Appendix A.

Predicting Almost as Well as the Best Convex Combination of Two Kernels We consider the convex combination of two domain-compatible kernels. Hence our parameterized kernel function $k_\alpha = (1-\alpha)k_0 + \alpha k_1$ where k_0 and k_1 are two distinct kernel functions. We further require in this abstract that the corresponding Hilbert spaces $\mathcal{H}_0$ and $\mathcal{H}_1$ be disjoint except for the zero function, i.e., $\mathcal{H}_0 \bigcap \mathcal{H}_1 = \{\mathbf{0}\}$. Typical kernel spaces that are disjoint except for the zero, include spaces derived from polynomial kernels of differing degree and wavelet kernels at two distinct levels of resolution. The following useful theorem from Aronszajn [2] gives a basis for our following observations.

Theorem 2 ([2]). *If k_i are the domain-compatible kernels of Hilbert spaces $\mathcal{H}_i$ with the norms $\|\cdot\|_i$, then $k = k_0 + k_1$ is the kernel of Hilbert space $\mathcal{H} = \mathcal{H}_0 + \mathcal{H}_1$ of all functions $f = f_0 + f_1$ with $f_i \in \mathcal{H}_i$, and with the norm defined by*

$$\|f\|^2 = \inf\left[\|f_0\|_0^2 + \|f_1\|_1^2\right],$$

which is the infimum taken for all decompositions $f = f_0 + f_1$ with $f_i \in \mathcal{H}_i$.

Therefore given $\alpha', \alpha'' \in (0,1)$ the three sets $\mathcal{H}_{\alpha'}$, $\mathcal{H}_{\alpha''}$, and $\bigcup_{\alpha\in[0,1]} \mathcal{H}_\alpha$ contain exactly the same functions; however in general $\|f\|_{\alpha'} \neq \|f\|_{\alpha''}$. Observe that with the assumption $\mathcal{H}_0 \bigcap \mathcal{H}_1 = \{\mathbf{0}\}$ any function $f \in \bigcup_{\alpha\in[0,1]} \mathcal{H}_\alpha$ has a unique decomposition $f = f_0 + f_1$ with $f_i \in \mathcal{H}_i$. Given the decomposition we can compute the norm of f in any particular $\mathcal{H}_\alpha$ via

$$\|f\|_\alpha^2 = \frac{1}{1-\alpha}\|f_0\|_0^2 + \frac{1}{\alpha}\|f_1\|_1^2, \quad \alpha \in (0,1) \ , \tag{9}$$

since for a scaled kernel $k' = \beta k$ the norm is rescaled as $\|f\|_{k'}^2 = \frac{1}{\beta}\|f\|_k^2$. We may define the following norm over $\mathcal{H}_0 + \mathcal{H}_1$.

Definition 1. *Given domain-compatible kernels k_0 and k_1 such that $\mathcal{H}_0 \bigcap \mathcal{H}_1 = \{\mathbf{0}\}$ let $k_\alpha = (1-\alpha)k_0 + \alpha k_1$ then given $f \in \mathcal{H}_0 + \mathcal{H}_1$. Define $\mathcal{C}(f)$ by*

$$\mathcal{C}^2(f) = \inf_{\alpha\in[0,1]} \|f\|_\alpha^2 \ . \tag{10}$$

The following theorem gives a canonical form for $\mathcal{C}(f)$.

Theorem 3. *Given $f = f_0 + f_1$ such that $f_i \in \mathcal{H}_i$ and $\mathcal{H}_0 \bigcap \mathcal{H}_1 = \{\mathbf{0}\}$ then*

$$\mathcal{C}^2(f) = \left(\|f_0\|_0 + \|f_1\|_1\right)^2 \ . \tag{11}$$

Proof. The theorem immediately follows from the substitution of the minimizer $\alpha = \frac{\|f_1\|}{\|f_0\|+\|f_1\|}$ into (9). □

A recent generalization of this canonical form is given in [15, Lemma A.2].

Theorem 4. *Given the* K-LMS-NET *algorithm tuned with learning rate η, an outcome range $[r_1, r_2]$, parameterized kernel $k_\alpha = (1-\alpha)k_0 + \alpha k_1$ constructed from two domain-compatible kernels k_0 and k_1 such that $\mathcal{H}_0 \bigcap \mathcal{H}_1 = \{\mathbf{0}\}$, a data sequence $\langle (x_1, y_1), (x_2, y_2), \ldots, (x_\ell, y_\ell) \rangle \in (\mathcal{X}, [r_1, r_2])^\ell$ then the total loss of the algorithm satisfies*

$$\sum_{t=1}^{\ell} L(y_t, \hat{y}_t) \leq \sum_{t=1}^{\ell} L(y_t, h(x_t)) + 2\sqrt{\hat{L}}\hat{H}\hat{X} + \hat{H}^2\hat{X}^2 + 2(r_2 - r_1)^2 \max(2\ln \mathcal{C}(h) + \ln \frac{1}{c} + \ln \frac{4}{3}, \ln 4) \quad (12)$$

for all $h \in \mathcal{H}_0 + \mathcal{H}_1$ and all constants $\hat{L}, \hat{H}, \hat{X}$, and $c \in (0, 1]$ such that the following four conditions hold: $\eta = [(1 + \frac{\sqrt{\hat{L}}}{\hat{H}\hat{X}})]^{-1}$, and

$$\mathcal{C}^2(h) + c \leq \hat{H}^2, \quad \sum_{t=1}^{\ell} L(y_t, h_\alpha(\mathbf{x}_t)) \leq \hat{L}, \text{ and } \sup_{\{t \in [1,\ldots,\ell], \alpha \in [0,1]\}} k_\alpha(x_t, x_t) \leq \hat{X}^2. \quad (13)$$

Predicting Almost as Well as the Best Width of a Gaussian Kernel In the following we define the surfeit of a function. In this abstract we avoid the technicalities of defining the surfeit for the complete Hilbert space $\mathcal{H}_k$; we consider the definition only on the prehilbert space H_k.

Definition 2. *Given a positive kernel ($\forall x, y \in \mathcal{X}^2 : k(x, y) \geq 0$), let $f \in H_k$; then define the* surfeit *by*

$$\mathcal{S}^2(f) = \inf \left[\|f^+\|^2 + \|f^-\|^2 \right] \; . \quad (14)$$

The infimum is taken over all decompositions $f^+ + f^- = f$, where $f^+ = \sum_{i:\beta_i > 0} \beta_i k(x_i, \cdot)$ and $f^- = \sum_{i:\beta_i < 0} \beta_i k(x_i, \cdot)$ are a positive linear and negative linear combination of kernel functions, respectively, such that $f = f^+ + f^- = \sum_{i=1}^{m} \beta_i k(x_i, \cdot)$.

The infimum exists since $0 \leq \|f\|^2 \leq \mathcal{S}^2(f)$.

Theorem 5. *Given the* K-LMS-NET *algorithm with learning rate η, an outcome range $[r_1, r_2]$ with $\max(|r_1|, |r_2|) \geq 1$, a parameterized ($\alpha \in [0, 1]$) Gaussian kernel, $k_\alpha(\mathbf{v}_1, \mathbf{v}_2) = \exp(-s_0\alpha\|\mathbf{v}_1 - \mathbf{v}_2\|^2)$ with fixed scale constant $s_0 \geq 1$ over the domain $[x_1, x_2]^n \times [x_1, x_2]^n$ with associated prehilbert spaces H_α a data sequence $\langle (\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_\ell, y_\ell) \rangle \in ([x_1, x_2]^n, [r_1, r_2])^\ell$, and the constants $c \in (0, 1]$, $s_0 \geq 1$, $\hat{L} \geq 0$, $\hat{H} \geq 1 + c$ and with $\eta = [(1 + \frac{\sqrt{\hat{L}}}{\hat{H}})]^{-1}$ then the total loss of the algorithm satisfies*

$$\sum_{t=1}^{\ell} L(y_t, \hat{y}_t) \leq \sum_{t=1}^{\ell} L(y_t, h_\alpha(\mathbf{x}_t)) + 2\sqrt{\hat{L}}\hat{H} + \hat{H}^2 + c + 2(r_2 - r_1)^2 [\ln \ell + 2\ln \mathcal{S}(h_\alpha) + \ln s_0 + \ln n + 2\ln(x_2 - x_1) + \ln \max(|r_1|, |r_2|) + \ln \frac{1}{c} + \ln 5] \quad (15)$$

for all $h_\alpha \in \bigcup_{\alpha\in[0,1]} H_\alpha$ *such that* $\|h_\alpha\|_\alpha^2 + c \leq \hat{H}^2, \;\; \sum_{t=1}^{\ell} L(y_t, h_\alpha(\mathbf{x}_t)) + c \leq \hat{L}$,

$$\text{and } \alpha \in \left[0, 1 - \frac{c}{5 s_0 \ell \max(|r_1|, |r_2|) n (x_2 - x_1)^2 \mathcal{S}^2(h_\alpha)}\right]. \tag{16}$$

The previous bound is given without regard of the computability of the predictions. If we restrict the data sequence to a discretization of the unit interval we can then apply the methods of Sect. 3 (in particular see Claim 2) to obtain polynomially tractable approximate predictions.

3 Computing the Predictions of K-LMS-NET

In the previous section we gave both a generic bound, and bounds for two specific applications of the K-LMS-NET algorithm. Here we consider how the predictions may be computed. Rather than computing the predictions exactly we give a polynomial-time absolute ϵ-approximation scheme to the predictions (see (5)) of the K-LMS-NET algorithm. The scheme is a deterministic sampling algorithm that separately approximates the numerator and denominator of the quotient of integrals that define the predictions of the K-LMS-NET algorithm.

The sampling methodology builds on the following three ideas. First, by obtaining a relative ϵ-approximation on an integral; this automatically gives a relative ϵ-approximation for a quotient of integrals (cf. (5)) since relative error approximations (aka “significant digits”) are closed under division. A good relative error approximation is also a good absolute error approximation up to a magnitude scaling constant. Second, to minimize the number of required samples we must concentrate samples in areas of large magnitude. Third, since the areas of large magnitude are co-determined by the critical points of the function to be approximated, the inspection of the analytic form gives both a bound on the number of the critical points and a method to find the critical points (areas of large magnitude).

In the following we first consider the cost in terms of relative loss bounds for using approximate predictions. We then consider the analytic forms of the functions to be integrated, both in general and then for our particular applications. Finally we give the details of our sampling methodology.

3.1 Additional Regret for Approximate Predictions

Rather than fixing the quality of the absolute ϵ accuracy of our approximate predictions, we advocate using a schedule $\{\epsilon_t\}$. In the following we see that a schedule which gradually increases the accuracy of our approximate predictions allows the additional cumulative regret incurred to be bounded by an $O(1)$ term.

When an exact prediction of the K-LMS-NET algorithm is replaced by absolute ϵ_t-approximate prediction on trial t the additional “approximation regret” incurred on that trial may be bounded by $2\epsilon_t|y_t - \hat{y}_t| + \epsilon_t^2$. Therefore we may bound the additional cumulative regret for using approximate predictions by $\epsilon_0 \in (0,1)$

with a decreasing schedule for $\{\epsilon_t\}$ of $\epsilon_t = \frac{\epsilon_0}{6.5\max([r_2-r_1],1)(t+1)\ln^2(t+1)}$, recalling that the outcomes and predictions are contained in $[r_1, r_2]$. Thus with the above schedule of $\{\epsilon_t\}$, and given that the approximate predictions are obtainable in time polynomial in $O(\frac{1}{\epsilon_t})$ on trial t, then the additional cumulative regret is bounded by $0 < \epsilon_0 < 1$ and the cumulative running time of the algorithm is polynomial in the number of trials.

3.2 Analytic Forms of the Prediction and Loss Lunctions

We compute the predictions $\hat{y}_t$ of the K-LMS-NET algorithm (cf. (5)) by maintaining explicit symbolic representations of $\mathbf{w}^i_{\alpha,t}(x)$, $\hat{\mathbf{y}}^i_t(\alpha)$ and $L_{[1,t]}(\alpha)$; the symbolic representations may then be exploited to find the critical points.

Below we give an explicit representation of $\mathbf{w}^i_{\alpha,t+1}(x)$ (omitting $\hat{\mathbf{y}}^i_t(\alpha)$ and $L_{[1,t]}(\alpha)$ as they follow directly) by expanding the recurrence (6) giving the 2^t-1 terms below

$$\mathbf{w}^i_{\alpha,t+1}(x) = \sum_{k=1}^{t}(-1)^{k+1}\eta^k T_{t,k}(x) \text{ where } T_{t',k}(x) = \sum_{\{(i_1,i_2,\ldots,i_k)|1\le i_1<\cdots<i_k\le t'\}} S_{(i_1,i_2,\ldots,i_k)}(x),$$
$$\text{and } S_{(i_1,i_2,\ldots,i_k)}(x) = y_{i_1}k_\alpha(x_{i_1},x_{i_2}) \times \cdots \times k_\alpha(x_{i_{k-1}},x_{i_k})k_\alpha(x_{i_k},x) \ . \quad (17)$$

For example, $\mathbf{w}^i_{\alpha,4}(x) = \eta\sum_{i=1}^3 y_i k_\alpha(x_i,x) + \eta^3 y_1 k_\alpha(x_1,x_2)k_\alpha(x_2,x_3)k_\alpha(x_3,x) - \eta^2[y_1k_\alpha(x_1,x_2)k_\alpha(x_2,x) + y_1k_\alpha(x_1,x_3)k_\alpha(x_3,x) + y_2k_\alpha(x_2,x_3)k_\alpha(x_3,x)]$. Clearly we cannot expect to give a polynomial time algorithm if we manipulate this representation directly, thus the applications we consider are cases where (17) algebraically collapses to a polynomial-sized representation.

In the following claims we give the representations of the functions needed to compute the predictions of the K-LMS-NET algorithm from the applications of Theorems 4 and 5. The proofs of these claims are straightforward and are omitted for reasons of brevity.

Claim 1 *In the* K-LMS-NET *algorithm with a kernel $k_\alpha = (1-\alpha)k_0 + \alpha k_1$, the first layer weight function may be expressed as*

$$\mathbf{w}^i_{\alpha,t}(x) = \sum_{i=1}^{t-1} p_{t,i}(\alpha)k_0(x_i,x) + q_{t,i}(\alpha)k_1(x_i,x)$$

where $p_{t,i}(\alpha)$ and $q_{t,i}(\alpha)$ are polynomials of degree i in α. Therefore, the functions $\hat{\mathbf{y}}^i_t(\alpha)$ and $L_{[1,t]}(\alpha)$ may be expressed as polynomials in α of degree $t-1$ and $2t-2$ respectively.

We discretize the input data to obtain a tractable method to predict with a Gaussian kernel. The following claim quantifies the size of the representation of the functions to be sampled by the degree of discretization of the input data.

Claim 2 *In the* K-LMS-NET *algorithm with the parameterized Gaussian kernel $k_\alpha(\mathbf{v},\mathbf{x}) = e^{-s_0\alpha\|\mathbf{v}-\mathbf{x}\|^2}$ with fixed scale constant $s_0 \in \mathbb{R}^+$ and with the discretized interval $\mathcal{X} = \{0, \frac{1}{m}, \ldots, \frac{m-1}{m}, 1\}^n$, the first layer weight function may be expressed as*

$$\mathbf{w}^i_{\alpha,t}(\mathbf{x}) = \sum_{i=1}^{t-1} p_{t,i}(\alpha) e^{-s_0\alpha\|\mathbf{x}_i-\mathbf{x}\|^2} \textit{ where } p_{t,i}(\alpha) = \sum_{j=0}^{nm^2(i-1)} c_{t,i,j}\left[e^{\frac{-s_0\alpha}{m^2}}\right]^j \quad ,$$

with each $c_{t,i,j} \in \mathbb{R}$. Applying the change in variable $\sigma = e^{\frac{-s_0\alpha}{m^2}}$ to the functions $\hat{\mathbf{y}}^i_t(\alpha)$ and $L_{[1,t]}(\alpha)$ gives polynomials in σ of degree $nm^2(t-1)$ and $2nm^2(t-1)$ respectively.

3.3 Finding Critical Points

We proceed by dividing functions into piecewise monotonic intervals. This means finding their critical points, or the zeros of the derivative. The problem of finding zeros of a polynomial has been called the *Fundamental Computational Problem of Algebra* [24]. There is a vast literature regarding this problem some general references are [17, 24, 3]. We do not need to actually find the zeros, we only need to *isolate* them as defined below.

Definition 3. *The k-*isolation *of measure δ of the zeros of a function $f : [a, b] \rightarrow \mathbb{R}$ is a list of $j \leq k$ intervals $\{[a_1, b_1], \ldots, [a_j, b_j]\}$ such that if $f(r) = 0$ then there exists an i s.t. $r \in [a_i, b_i]$, and also the sum total measure of the intervals is δ.*

Observe that in the above we do not precisely find roots but *isolate* them as some intervals could have multiple roots and others none.

Definition 4. *The composition of functions $f(\sigma(\cdot))$ is called a σ-*polynomial *(polynomial after a change in variable to σ) if f is a polynomial and $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ is continuously differentiable and $\forall x \in \mathbb{R} : \sigma'(x) \neq 0$ (σ is then strictly monotone without inflection). The degree of a σ-polynomial $f(\sigma(\cdot))$ is the degree of f.*

Every polynomial is a σ-polynomial where σ is identity function.

Claim 3 *Given a σ-polynomial $f(\sigma(\cdot)) : [0, 1] \rightarrow \mathbb{R}$ of degree s, there exists an s-isolation of the zeros of measure 2^{-p}, the isolation is computable in time polynomial in p and in s.*

Any algorithm that can efficiently act as a *root-existence* oracle for an interval $[a, b]$ of a polynomial which returns TRUE if there exists roots in $[a, b]$ and FALSE otherwise can be subordinated within a bisection algorithm to compute an s-isolation. In this abstract, we do not actually give an algorithm to compute an s-isolation efficiently (see [17, 24, 3] for algorithms where, e.g., the Euclidean Algorithm may efficiently serve as an oracle), for reasons of brevity.

Corollary 1. *Given σ-polynomials $f(\sigma(\cdot)) : [0, 1] \rightarrow \mathbb{R}$ and $g(\sigma(\cdot)) : [0, 1] \rightarrow \mathbb{R}$ of degree s_1 and s_2 respectively, and letting $l = e^{f(\sigma)}$ and $m = g(\sigma)e^{f(\sigma)}$, then there exists a $s_1 - 1$ and $s_2(s_1 - 1)$ isolation of l' and m' respectively of measure 2^{-p} computable in time polynomial in s_1, s_2 and p.*

Proof. Omitted for the sake of brevity.

3.4 Deterministic Piecewise Monotone Sampling

Our sampler functions as follows. The quantity we wish to estimate is a quotient of integrals. Relative error approximations (i.e., "significant digits") are closed under division. Hence we develop a sampler for a single integral for which we can give a relative ϵ-approximation scheme. Our intuition from absolute error approximations may suggest that a quantity to bound is the maximal slope of the function to be integrated; this quantity is of less use than the *relative variation* of a positive function, i.e., $\frac{\max_x f(x)}{\min_x f(x)}$. For a positive monotone function we will require a quantity of samples logarithmic in the relative variation. With a bound for monotone functions we can generalize to piecewise monotone, this requires that we isolate the critical points of our function. In the previous section, the applications chosen lead to functions for which it is easy to find the critical points. This leads to a method that samples exponentially more often in areas of large volume. We note that the sampler here has been designed to directly "prove a bound"; using the techniques here it is possible to design a sampler that also proves the bound, but which is considerable more adaptive (uses fewer samples), and hence more useful in practice.

In the following three lemmas we give simple algebraic results about ϵ-approximations.

Lemma 1. *Suppose* $\hat{a}\overset{r}{\approx}_{\epsilon}a$ *and* $\hat{b}\overset{r}{\approx}_{\epsilon}b$ *then* $(\hat{a}+\hat{b})\overset{r}{\approx}_{\epsilon}(a+b)$.

Lemma 2. *Suppose* $\hat{b}\overset{r}{\approx}_{\epsilon}b$ *and* $b \le B$ *then* $\hat{b}\overset{a}{\approx}_{2B\epsilon}b$.

Lemma 3. *Suppose* $\hat{a}\overset{r}{\approx}_{\epsilon}a$ *and* $\hat{b}\overset{r}{\approx}_{\epsilon}b$ *then* $\frac{\hat{a}}{\hat{b}}\overset{r}{\approx}_{3\epsilon}\frac{a}{b}$ *for all* $\epsilon \in (0, \frac{1}{3})$.

The following scale-invariant theorem is the key to our sampling methodology.

Theorem 6. *Given a continuous nondecreasing function* $f : [a, b] \rightarrow \mathbb{R}^+$*, let* $y = \int_a^b f(\alpha)d\alpha$*. Define* $z = \frac{f(b)}{f(a)}$*; then there exists a relative* ϵ*-approximation for* y *which requires* $\left\lceil \frac{1}{2}\left(\frac{1}{\epsilon}+1\right)\ln z\right\rceil + 2$ *samples (evaluations) of* f.

The proof in Appendix A details how the samples are chosen.

The following lemma demonstrates that a good relative ϵ-approximation may be obtained by well-approximating a function on a subset of its domain if the measure of the non-approximated subset times the function's relative variation is sufficiently small.

Lemma 4. *Given measurable sets* $E' \subset E$ *with* $\mu(E) = 1$ *and a continuous function* f*, such that* $\forall x \in E : 0 < a \le f(x) \le b$*, define* $\Delta = \mu(E - E')$ *and* $z = \frac{b}{a}$*. Then if* $\hat{y}\overset{r}{\approx}_{\epsilon'} \int_{E'} f d\mu$ *it is also case that* $\hat{y}\overset{r}{\approx}_{\epsilon} \int_{E} f d\mu$ *when* $\epsilon' + \Delta z \le \epsilon$.

Proof. Omitted for the sake of brevity.

We now summarize the process for approximating $\int_E f d\mu$: i) we divide f into monotonic regions by isolating the critical points in sufficiently small intervals (cf. Claim 3 and Corollary 1); ii) the integral of each monotonic regions is then separately approximated (cf. Theorem 6); and iii) the separate approximations are then summed without the isolated intervals (cf. Lemma 4) to obtain a $\hat{y} \overset{r}{\approx}_\epsilon \int_E f d\mu$. In Appendix A Lemmas 8 and 9 are given. These detail the separate approximation of the denominator and numerator of (5). Their proofs follow the basic sketch above except that additional points of the functions need to be isolated in order to properly clip the functions. The following theorem combines Lemmas 8 and 9 to demonstrate the computation of an absolute ϵ-approximation to a prediction of K-LMS-NET.

Theorem 7. *Given a σ-polynomial $f(\sigma(\cdot)) : [0,1] \rightarrow [0,\infty)$ of degree s and $z \in (1,\infty)$, and a σ-polynomial $g(\sigma(\cdot)) : [0,1] \rightarrow (-\infty,\infty)$ of degree t we may compute an absolute ϵ-approximation*

$$\bar{y} \overset{a}{\approx}_\epsilon \frac{\int_0^1 \max(r_1, \min(g(\sigma(\alpha)), r_2)) \max(e^{-f(\sigma(\alpha))}, z^{-1}) d\alpha}{\int_0^1 \max(e^{-f(\sigma(\alpha))}, z^{-1}) d\alpha} \tag{18}$$

in time polynomial in s, t, $\ln(z(1 + r_2 - r_1))$, and ϵ^{-1} .

Acknowledgments: I thank Massimiliano Pontil for valuable discussions and for pointing out the inspirational kernel sum theorem from Aronszajn [2].

References

1. C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50(1-2):5–43, 2003.
2. N. Aronszajn. Theory of reproducing kernels. *Trans. Amer. Math. Soc.*, 68:337–404, 1950.
3. L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and real computation.* Springer-Verlag, 1998.
4. N. Cesa-Bianchi, P. Long, and M. Warmuth. Worst-case quadratic loss bounds for on-line prediction of linear functions by gradient descent. *IEEE Transactions on Neural Networks*, 7(2):604–619, May 1996.
5. N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines.* Cambridge University Press, Cambridge, UK, 2000.
6. L. Csató and M. Opper. Sparse on-line gaussian processes. *Neural Computation*, 14(3):641–668, 2002.
7. Y. Freund. Predicting a binary sequence almost as well as the optimal biased coin. In *Proc. of COLT.*, pages 89–98. ACM Press, New York, NY, 1996.
8. M. Gibbs and D. MacKay. Efficient implementation of gaussian processes (draft manuscript), 1996.
9. M. Herbster. Learning additive models online with fast evaluating kernels. In *COLT 2001, Proceedings*, volume 2111 of *LNAI*, pages 444–460. Springer, 2001.
10. W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, Mar. 1963.
11. J. Kivinen, A. J. Smola, and R. C. Williamson. Online learning with kernels. In *NIPS 14*, Cambridge, MA, 2002. MIT Press.
12. J. Kivinen and M. K. Warmuth. Averaging expert predictions. *Lecture Notes in Computer Science (EUROCOLT)*, 1572:153–167, 1999.

13. N. Littlestone. Learning when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1988.
14. N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.
15. C. Micchelli and M. Pontil. Learning the kernel function via regularization, Dept. of Computer Science, University College London, Research Note: RN/04/12, 2004.
16. C. S. Ong, A. J. Smola, and R. C. Williamson. Hyperkernels. In *Neural Information Processing Systems*, volume 15. MIT Press, 2002.
17. V. Y. Pan. Solving a polynomial equation: Some history and recent progress. *SIAM Review*, 39(2):187–220, 1997.
18. J. F. Traub and A. G. Werschulz. *Complexity and Information.* Cambridge University Press, Cambridge, 1998.
19. J. Vermaak, S. J. Godsill, and A. Doucet. Sequential bayesian kernel regression. In *NIPS 16.* MIT Press, Cambridge, MA, 2004.
20. V. Vovk. Aggregating strategies. In *Proc. 3rd Annu. Workshop on Comput. Learning Theory*, pages 371–383. Morgan Kaufmann, 1990.
21. V. Vovk. Competitive on-line statistics. *Bull. of the International Stat. Inst.*, 1999.
22. C. K. I. Williams and C. E. Rasmussen. Gaussian processes for regression. In *NIPS 1995*, Cambridge, Massachusetts, 1996. MIT Press.
23. Q. Wu, Y. Ying, and D.-X. Zhou. Multi-kernel regularized classifiers. Submitted to *Journ. of Complexity*, 2004.
24. C. K. Yap. *Fundamental problems of algorithmic algebra.* Oxford Uni. Press, 2000.

A Additional Proofs

Lemma 5. *Given domain-compatible kernels k_0 and k_1 such that $\mathcal{H}_0 \bigcap \mathcal{H}_1 = \{\mathbf{0}\}$, let $k_\alpha = (1-\alpha)k_0 + \alpha k_1$. Then given $f \in \mathcal{H}_0 + \mathcal{H}_1$, we have $\forall c \in (0,1]$: $\forall \delta \in [0, \min(\frac{1}{4}, \frac{3c}{4\mathcal{C}^2(f)})]$ that there exists $0 \le \alpha' < \alpha'' \le 1$ with $\alpha'' - \alpha' = \delta$ such that $\forall \alpha \in [\alpha', \alpha'']$, it is the case that $\|f\|_\alpha^2 \le \mathcal{C}^2(f) + c$.*

Proof. Let $f_0 + f_1$ with $f_i \in \mathcal{H}_i$. Without loss of generality assume that $\|f_1\|_1 \le \|f_0\|_0$. Set $\alpha' = \frac{\|f_1\|_1}{\|f_1\|_1 + \|f_0\|_0}$, recalling that $\|f\|_{\alpha'}^2 = \inf_{\alpha \in [0,1]} \|f\|_\alpha^2 = \mathcal{C}^2(f)$. Let $x = \|f\|_{\alpha'+\delta}^2 - \|f\|_{\alpha'}^2$; by substituting $\alpha' = \frac{\|f_1\|_1}{\|f_1\|_1 + \|f_0\|_0}$ into (9) we have that

$$x = \delta(\|f_1\|_1 + \|f_0\|_0)^2 \left[\frac{\frac{\|f_0\|_0 \|f_1\|_1}{\delta} + \|f_0\|_0^2 - \|f_1\|_1^2}{(\|f_0\|_0 + \|f_1\|_1)^2} - \delta \right]^{-1} . \tag{19}$$

As we are upper bounding x let us separately upper bound

$$p(\|f_0\|_0, \|f_1\|_1, \delta) = \left[\frac{\frac{\|f_0\|_0 \|f_1\|_1}{\delta} + \|f_0\|_0^2 - \|f_1\|_1^2}{(\|f_0\|_0 + \|f_1\|_1)^2} - \delta \right]^{-1} . \tag{20}$$

As a function of $\|f_1\|_1$ through routine calculations it can be shown that p obtains its maximum (for $\delta \in [0, 1/4]$) on either the boundary $\|f_1\|_1 = 0$ or $\|f_1\|_1 = \|f_0\|_0$. Thus substituting, $p(\|f_0\|_0, 0, \delta) = \frac{1}{1-\delta}$ and $p(\|f_0\|_0, \|f_0\|_0, \delta) = \frac{4\delta}{1-4\delta^2}$. Therefore, for all $\delta \in [0, 1/4]$, we have $p(\|f_0\|_0, \|f_0\|_0, \delta) \le p(\|f_0\|_0, 0, \delta) \le \frac{4}{3}$. By combining the upper bound of $\frac{4}{3}$ with (19) we have that for $\delta \in [0, 1/4]$, $\|f\|_{\alpha'+\delta}^2 \le \mathcal{C}^2(f) + \frac{4}{3}\delta\mathcal{C}^2(f)$; therefore with $\alpha'' = \alpha' + \delta$ we are done. □

Lemma 6. *Let $k_\alpha(\mathbf{v}_1, \mathbf{v}_2) = \exp(-s_0\alpha\|\mathbf{v}_1 - \mathbf{v}_2\|^2)$ denote a parameterized ($\alpha \in [0,1]$) Gaussian kernel with fixed scale constant $s_0 \geq 1$ over the domain $[x_1, x_2]^n \times [x_1, x_2]^n$ with associated prehilbert spaces H_α. Given a function $h_{\alpha'} \in H_{\alpha'}$ such that $\|h_{\alpha'}\|_{\alpha'} \geq 1$ with representation $h_{\alpha'}(\cdot) = \sum_{i=1}^{m}\beta_i k_{\alpha'}(\mathbf{v}_i, \cdot)$ then set $h_{\alpha'+\delta}(\cdot) = \sum_{i=1}^{m}\beta_i k_{\alpha'+\delta}(\mathbf{v}_i, \cdot)$. Then the square loss and squared norm of $h_{\alpha'+\delta}$ may be bounded by those of $h_{\alpha'}$ plus any constant $0 < c < 1$ for all sequences $\langle(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_\ell, y_\ell)\rangle \in ([x_1, x_2]^n, [r_1, r_2])^\ell$ where $\max(|r_1|, |r_2|) \geq 1$. Hence $\sum_{t=1}^{\ell}(y_t - h_{\alpha'+\delta}(\mathbf{x}_t))^2 \leq \sum_{t=1}^{\ell}(y_t - h_{\alpha'}(\mathbf{x}_t))^2 + c$ and $\|h_{\alpha'+\delta}\|_{\alpha'+\delta}^2 \leq \|h_{\alpha'}\|_{\alpha'}^2 + c$ for all $\delta \in [0, \frac{c}{5s_0\ell\max(|r_1|,|r_2|)n(x_2-x_1)^2\mathcal{S}^2(h_{\alpha'})}]$.*

Proof. Omitted in this abstract; see full version.

Proof (sketches of Theorems 4 and 5). The theorems follow directly from Theorem 1 with Lemmas 5 and 6 with $\mathcal{A}$ chosen so that in each $\mu(\mathcal{A}) = \delta$.

The following inequality is needed for Theorem 6.

Lemma 7. *Suppose $r \geq 1$, then $r + \frac{1}{2} \geq \ln(1 + \frac{1}{r})^{-1}$.*

Proof (of Theorem 6). Let $r = \frac{1}{2\epsilon}$ and choose n s.t.

$$\left\lceil (r + \frac{1}{2}) \ln z \right\rceil + 1 \leq n \tag{21}$$

where $n+1$ is the total number of function samples. The function is sampled over n intervals with widths $\mathit{\Delta}_i = \frac{(b-a)\left(\frac{r}{r+1}\right)^{i-1}}{(1+r)(1-\left(\frac{r}{r+1}\right)^n)}$; the samples are denoted $f(a) = f_0, f_1, \ldots, f_n = f(b)$ where $f_i = f(a + \sum_{j=1}^{i}\mathit{\Delta}_i)$. We also need the following inequality relating r, n and z:

$$\left(\frac{r}{r+1}\right)^{n-1} \leq \frac{1}{z}. \tag{22}$$

From (21) and Lemma 7 it follows that, $\ln z \leq (n-1)\ln\frac{r+1}{r}$ which implies (22).

Define $M = \sum_{i=1}^{n-1} f_i\mathit{\Delta}_i$; now define lower and upper bounds of y, L and U by $L = f_0\mathit{\Delta}_1 + \frac{r}{r+1}M \leq y \leq M + f_n\mathit{\Delta}_n = U$. We proceed to show that $\hat{y} = \frac{1}{2}(L+U)$ is a relative ϵ-approximation of y, since U and L are upper and lower bounds if we can show $U(1-\epsilon) \leq \hat{y} \leq L(1+\epsilon)$ which is equivalent to the conjunction of conditions $\frac{1}{2}\frac{U-L}{L} \leq \epsilon$, and $\frac{1}{2}\frac{U-L}{U} \leq \epsilon$. This will prove that $\hat{y}$ is an ϵ-approximation of y. However, since $L \leq U$ we need only show $\frac{1}{2}\frac{U-L}{L} \leq \epsilon$. Thus,

$$\frac{1}{2}\frac{U-L}{L} = \frac{1}{2}\frac{M + f_n\mathit{\Delta}_n - f_0\mathit{\Delta}_1 - \frac{r}{r+1}M}{f_0\mathit{\Delta}_1 + \frac{r}{r+1}M} \tag{23}$$

$$\leq \frac{1}{2}\frac{\frac{1}{r+1}\sum_{i=1}^{n-1}\left(\frac{r}{r+1}\right)^i f_i}{f_0 + \frac{r}{r+1}\sum_{i=1}^{n-1}\left(\frac{r}{r+1}\right)^i f_i} \leq \frac{1}{2}\frac{1}{r} \tag{24}$$

where (24) follows from (22). Hence $\hat{y}$ is an ϵ-approximation of y with the requisite number of samples. □

The following two lemmas give a method to obtain relative ϵ-approximations of the denominator and the numerator of the predictions (5) K-LMS-NET.

Lemma 8. *Given a σ-polynomial $f(\sigma(\cdot)):[0,1]\to[0,\infty)$ of degree s and a $z \in (1,\infty)$, we may compute a relative ϵ-approximation $\hat{y}\overset{r}{\approx}_\epsilon \int_0^1 \max(e^{-f(\sigma(\alpha))}, z^{-1})d\alpha$ in time polynomial in s, $\ln(z)$ and ϵ^{-1} .*

Proof. Let $l = e^{-f(\sigma(\alpha))}$. By Corollary 1, we may puncture the interval $[0,1]$ into no more than $r \leq s$ regions with the (up to) $s-1$ of the critical points l isolated into a total measure of no more than 2^{-p}. By construction in each of the r regions $\min(l, z^{-1})$ is monotonic; thus we may apply Theorem 6 to each of the r regions (with $\epsilon = \epsilon'/2$) since relative ϵ-approximations add (cf Lemma 1). The estimator formed by adding the r estimators is a relative $\epsilon'/2$-approximation to $\int_{E'} \max(l, z^{-1})d\alpha$ where E' is the interval $[0,1]$ minus the isolates. However if we set $p = 1 + \log(z/\epsilon')$ by Lemma 4 we have a relative ϵ'-approximation to $\int_0^1 \max(l, z^{-1})d\alpha$. □

Lemma 9. *Given a σ-polynomial $f(\sigma(\cdot)) : [0,1]\to[0,\infty)$ of degree s and $z \in (1,\infty)$, and σ-polynomial $g(\sigma(\cdot)) : [0,1]\to(-\infty,\infty)$ of degree t we may compute a relative ϵ-approximation $\hat{y}\overset{r}{\approx}_\epsilon \int_0^1 \max(1, \min(g(\sigma(\alpha)), 1+r)) \max(e^{-f(\sigma(\alpha))}, \frac{1}{z})d\alpha$ in time polynomial in s, t, $\ln(z(1+r))$, and ϵ^{-1} .*

Proof. We sketch the proof for reasons of brevity and the fact that it closely follows the proof of Lemma 8. The key difference is the need to create additional isolates since it is subtle to clip $g(\sigma(\alpha))$ and $e^{-f(\sigma(\alpha))}$ independently. In fact we need to isolate the critical points of $g(\sigma(\alpha))e^{-f(\sigma(\alpha))}$, $g(\sigma(\alpha))$, and $e^{-f(\sigma(\alpha))}$; and the zeroes of $g(\sigma(\alpha)) = 1$, $g(\sigma(\alpha)) = 1 + r$, and $e^{-f(\sigma(\alpha))} = z^{-1}$. Now we can ensure that we can correctly clip and also for each region between isolates that the function $\max(1, \min(g(\sigma(\alpha)), 1 + r)) \max(e^{-f(\sigma(\alpha))}, z^{-1})$ is monotonic. We observe that the number of isolates is polynomial in s and t. □

We compute a shifted version of the quotient (5) so that the predictions and outcomes may have both positive and negative values.

Proof (of Theorem 7). Apply Lemmas 9 and 8 to give relative ϵ'-approximations $\hat{n}\overset{r}{\approx}_{\epsilon'} \int_0^1 \max(1, \min(g(\sigma(\alpha)) + 1 - r_1, 1 + r_2 - r_1)) \max(e^{-f(\sigma(\alpha))}, z^{-1})d\alpha$, and $\hat{d}\overset{r}{\approx}_{\epsilon'} \int_0^1 \max(e^{-f(\sigma(\alpha))}, z^{-1})d\alpha$ with $\epsilon' = \frac{\epsilon}{6(1+r_2-r_1)}$. Observe that

$$y' = \frac{\int_0^1 \max(1, \min(g(\sigma(\alpha)) + 1 - r_1, 1 + r_2 - r_1)) \max(e^{-f(\sigma(\alpha))}, z^{-1})d\alpha}{\int_0^1 \max(e^{-f(\sigma(\alpha))}, z^{-1})d\alpha} \tag{25}$$

is an expectation of a quantity bounded by 1 and $1+r_2-r_1$, hence $y' \leq 1+r_2-r_1$. Therefore by Lemmas 2 and 3 $\frac{\hat{n}}{\hat{d}}\overset{a}{\approx}_\epsilon y'$. Since $y' = \bar{y} + 1 - r_1$, we conclude that $\frac{\hat{n}}{\hat{d}} - (1 - r_1)\overset{a}{\approx}_\epsilon \bar{y}$. □