Localized Boosting

Ron Meir
Department of Electrical Engineering
Technion, Haifa 32000, Israel
rmeir@ee.technion.ac.il

Ran El-Yaniv[*]
Department of computer Science
Technion, Haifa 32000, Israel
rani@cs.technion.ac.il

Shai Ben-David
Department of computer Science
Technion, Haifa 32000, Israel
shai@cs.technion.ac.il

Abstract

We introduce and analyze *LocBoost*, a new boosting algorithm, which leads to the incremental construction of a mixture of experts type architecture. We provide upper bounds on the expected loss of such models in terms of the smoothness properties of the gating functions appearing in the mixture of experts model. Furthermore, an incremental algorithm is proposed for the construction of the classifier, based on a maximum-likelihood approach and the EM algorithm. Preliminary numerical results appear to be promising.

1 INTRODUCTION

One of the most successful recent approaches to machine learning and pattern classification is based on the idea of adaptively combining 'weak' classifiers, through a procedure that has come to be termed Boosting (see Schapire et al. [19] for a detailed exposition of the practice and theory of this type of algorithm). Work by several authors [3, 18, 7, 9, 16] has provided a rather general approach to Boosting, through the incremental greedy minimization of some empirical cost function. This point of view stresses the relationship to some widely studied algorithms in the statistics, signal processing and neural network communities. For example, the popular approach to nonlinear wavelet approximation through matching pursuit [14] falls in this class of algorithms.

In this work we extend the framework for the construction of composite classifiers by allowing the weights of the different weak classifiers to depend on the input. That is, rather than having constant weights attached to each of the experts (as in previous approaches), we allow weights that are functions over the input domain. Our extension models a scenario in which a learner may base the relative significance of each of his expert advisors on the features of the specific input he has to classify. This extension seems to better model real-world situations where particularly complex tasks are split between experts, each of whose expertise lies in a narrow field (corresponding to a small sub-domain of the input space in our simplified model). The structure of the final classifier produced in our approach is related to the mixture of experts (ME) architecture studied by Jordan and co-workers (e.g., [11]).

In this work we introduce such a "boosting with localization" framework which we call *LocBoost*. We provide analysis of the generalization ability of LocBoost type classifiers and show that under certain smoothness assumptions regarding the expert coefficient functions, uniform convergence bounds hold in our extended framework. One of the most appealing features of the generalization bounds for convex constant coefficients is their independence of the number of participating experts [19]. We show that similar results can be derived for our input-dependent expert mixtures. As far as we aware these are the first generalization results for mixtures of experts with non-constant expert coefficients, which possess the desirable features of previous constant coefficient bounds. We apply this new approach by presenting an incremental greedy learning algorithm based on a maximum-likelihood criterion. The resulting classifier is similar to the final classifier obtained in boosting algorithms, except that it has the greater flexibility of input-dependent weights. Initial experiments with the LocBoost algorithm appear to be promising.

The remainder of the paper is organized as follows. We begin in Section 2 by presenting some generalization bounds for mixture of experts architectures based on recent results for convex classes. Section 3 then proceeds to introduce an incremental algorithm based on maximum-likelihood estimation. Implementation details and Preliminary numerical results are presented in Section 4, and a short discussion is concludes the paper in Section 5.

2 GENERALIZATION BOUNDS FOR MIXTURES OF EXPERTS

We consider two-class classification problems, using *confidence rated* classifiers, which produce real-valued hypotheses $h \in [-1, 1]$, rather than simple binary hypotheses. Such classifiers have recently been shown to be very

[*] Ran El-Yaniv is a Marcella S. Geltman Memorial Academic Lecturer.

effective in boosting algorithms [20, 9], and in general yield greater flexibility (e.g., [1]). Consider a composite classifier formed by linearly combining a set of *base* classifiers $h_t, t = 1, \ldots, T$, where the combination coefficients depend on the input variable. Formally, we have

$$H(x) = \sum_{t=1}^{T} \beta_t(x) h_t(x),$$

where the soft classifiers $h_t(x)$ assume values in the interval $[-1, +1]$, and the input-dependent mixture coefficients obey the condition

$$\beta_t(x) \geq 1 \qquad ; \qquad \sum_{t=1}^{T} \beta_t(x) = 1.$$

In some cases it will be convenient to represent $H(x)$ as $H(x) = \sum_t \alpha_t(x) h_t(x) / \sum_t \alpha_t(x)$, where the normalization is made explicit. We will now show that if the functions $\alpha_t(x)$ are sufficiently smooth, and the class $\{h\}$ is not too large, then the estimation error does not grow too quickly.

First, it is important to understand the need for smoothness assumptions. To see this, consider an arbitrary set of m points $S = \{(x_i, y_i)\}_{i=1}^m \in \left(\mathbb{R}^d \times \{-1, +1\}\right)^m$. Let two constant valued classifiers be given, say $h_+(x) = 1$ and $h_-(x) = -1$ for any x. Let $I(E)$ denote the indicator function for the event E, and let $S_\pm$ be the sub-sets of S corresponding to the positive and negative examples, respectively. Set the weighting function $\beta_+(x) = 1 - I(x \in S_-)$, so that $\beta_+(x)$ equals 0 on all the negative examples and unity otherwise, and similarly $\beta_-(x) = I(x \in S_-)$. Clearly the two-component mixture $\beta_+(x)h_+(x) + \beta_-(x)h_-(x)$ achieves the correct classification on S, and $\beta_+(x) + \beta_-(x) = 1$. Obviously, no generalization can be expected from such a classifier, since *any* classification on a finite set of points can be achieved in this fashion. Thus, it is desirable to envisage the complexity of the classifier being generated by the convex combination itself, rather than by the complexity of the component classifiers $h_t(x)$ and mixture coefficients $\beta_t(x)$.

We begin with a few comments on notation. The probability of an event E under D will be denoted by $\mathbf{P}_D(E)$, while $\mathbf{P}_S(E)$ denotes the probability of the event E with respect to choosing an example uniformly at random from a set S.

Definition 1 *A function class $\hat{\mathcal{F}}$ is an ϵ-*sloppy *η-*cover *of $\mathcal{F}$ w.r.t. data S if, for all $f \in \mathcal{F}$, there exists a $\hat{f} \in \hat{\mathcal{F}}$ with $\mathbf{P}_S \left\{ |\hat{f}(x) - f(x)| > \eta \right\} \leq \epsilon$. Let $\mathcal{N}(F, \eta, \epsilon, m)$ denote the maximum over all subsets S, $|S| = m$, from the domain, of the size of the smallest ϵ-sloppy η-cover of $\mathcal{F}$ w.r.t. S. Note that the standard definition of cover holds when $\epsilon = 0$.*

We now introduce some operations that generate new classes of functions from existing classes. As our goal is to broaden the scope of classes that are known to be learnable, we prove upper bounds on the covering numbers of the generated classes in terms of the covering numbers of the original classes.

Definition 2 *Let $\mathcal{H}$ and $\mathcal{A}$ denote classes of real-valued functions (we assume that $\alpha \geq 0, \forall \alpha \in \mathcal{A}$). We then define the following classes of functions:*

$$\mathcal{HA} = \{fg : h \in \mathcal{H}, g \in \mathcal{A}\}$$

$$\mathrm{const}^M(\mathcal{H}) = \{c \cdot h : 0 < c \leq M, M > 0, h \in \mathcal{H}\}$$

$$\mathrm{co}(\mathcal{H}) = \left\{ \sum_{i=1}^{n} a_i h_i(x) : \sum_{i=1}^{n} a_i = 1, h_i \in \mathcal{H}, n \in \mathbb{N} \right\}$$

$$\mathrm{co}_{\mathcal{A}}(\mathcal{H}) = \left\{ \frac{\sum_{i=1}^n \alpha_i(x) h_i(x)}{\sum_{i=1}^n \alpha_i(x)}, \alpha_i \in \mathcal{A}, h_i \in \mathcal{H}, n \in \mathbb{N} \right\}$$

$$\mathrm{co}_{\mathcal{A}}^k(\mathcal{H}) = \left\{ \sum_{i=1}^{k} \alpha_i(x) h_i(x), \alpha_i \in \mathcal{A}, h_i \in \mathcal{H} \right\}$$

Definition 3 *Let $(\mathcal{F}_1, \ldots, \mathcal{F}_k)$ be classes of functions defined over domains $(\mathcal{X}_1, \ldots, \mathcal{X}_k)$ respectively. Define a class of functions $\prod_{i=1}^k \mathcal{F}_i$ over the domain $\mathcal{X} = \cup_{i=1}^k \mathcal{X}_i$, by*

$$\prod_{i=1}^{k} \mathcal{F}_i = \{f : \exists f_1 \in \mathcal{F}_1, \ldots, \exists f_k \in \mathcal{F}_k, \ \forall i \leq k, \\ \forall x \in \mathcal{X}_i, f(x) = f_i(x)\}.$$

Let G be a partition of the domain set $\mathcal{X}$. That is $G = \{\mathcal{X}_1, \ldots, \mathcal{X}_k\}$ where the $\mathcal{X}_i$'s are pairwise disjoint subsets of $\mathcal{X}$ and $\cup_{i=1}^k \mathcal{X}_i = \mathcal{X}$. Given a class of function $\mathcal{H}$ and a partition G of its domain, the class of case-wise defined $\mathcal{H}$ functions w.r.t. G is defined as

$$CW^G(\mathcal{H}) = \{f : \exists \{h_1, \ldots, h_k\} \in \mathcal{H}, \forall i \leq k, \\ \forall x \in \mathcal{X}_i, f(x) = h_i(x)\}.$$

We present a few simple claims that bound the covering numbers of the classes generated by the above operators. Unless otherwise mentioned, all the above claims are universally quantified with respect to the parameters that are not explicitly mentioned (e.g., unless m is explicitly mentioned, each of the following claims should be read with the prefix 'for all m'). As the proofs of these lemmas are all relatively straightforward, we skip some of the simple proofs.

Lemma 1 *For any pair of classes of functions, $\mathcal{H}$, $\mathcal{A}$, $|h| \leq B, |\alpha| \leq B$, for $h \in \mathcal{H}, \alpha \in \mathcal{A}$, and for every $\eta > 0$ and $\epsilon > 0$,*

$$\mathcal{N}(\mathcal{HA}, \eta, \epsilon, m) \leq \mathcal{N}(\mathcal{H}, \eta/2B, \epsilon, m)\mathcal{N}(\mathcal{A}, \eta/2B, \epsilon, m)$$

Proof Assume that $\hat{\mathcal{H}}$ and $\hat{\mathcal{A}}$ are finite sloppy $\eta/2$-covers of $\mathcal{H}$ and $\mathcal{A}$, respectively. We show that $\hat{\mathcal{H}}\hat{\mathcal{A}}$ is a sloppy

η-cover of $\mathcal{HA}$. To see this simply note that for any $h \in \mathcal{H}$ and $\alpha \in \mathcal{A}$ there exist $\hat{h}$ and $\hat{\alpha}$ such that $\mathbf{P}_S\left\{|\hat{h}(x) - \hat{h}(x)| > \eta/2\right\} \le \epsilon$, and similarly for α. The claim then follows from the observation that $h\alpha - \hat{h}\hat{\alpha} = (h - \hat{h})\alpha + (\alpha - \hat{\alpha})\hat{h}$, and the boundedness of the functions. ■

Lemma 2 *For $\eta > 0$, $M > 0$ and any class $\mathcal{H}$,*

$$\mathcal{N}(\mathrm{const}^M\mathcal{H}, \eta, \epsilon, m) \le \mathcal{N}(\mathcal{H}, \eta/M, \epsilon, M).$$

If $\mathcal{H}$ is closed under multiplication by constants then $const^M\mathcal{H} = \mathcal{H}$.

Lemma 3 *Let $(\mathcal{F}_1, \dots, \mathcal{F}_k)$ be classes of functions defined over domains $(\mathcal{X}_1, \dots, \mathcal{X}_k)$ respectively. Then,*

$$\mathcal{N}\left(\prod_{i=1}^{k}\mathcal{F}_i, \eta, \epsilon, m\right) \le \prod_{i=1}^{k}\mathcal{N}(\mathcal{F}_i, \eta, \epsilon, m).$$

Corollary 1 *For every class $\mathcal{H}$ and every partition G of its domain into k subsets,*

$$\mathcal{N}(CW^G(\mathcal{H}), \eta, \epsilon, m) \le \mathcal{N}(\mathcal{H}, \eta, \epsilon, m)^k.$$

Corollary 2 *Let $\mathcal{H}$ be a class of functions and $G = \{\mathcal{X}_1, \dots, \mathcal{X}_k\}$ a partition of its domain $\mathcal{X}$ into k subsets. Let $\mathcal{B}_G$ denote the class of functions from $\mathcal{X}$ to the unit interval that are constant on each of the $\mathcal{X}_i$'s in G. Then*

$$\mathcal{N}(\mathrm{co}_{\mathcal{B}_G}(\mathcal{H}), \eta, \epsilon, m) \le \mathcal{N}(\mathrm{co}(\mathcal{H}), \eta, \epsilon, m)^k.$$

Proof: Just note that $\mathrm{co}_{\mathcal{B}_G}(\mathcal{H}) \subseteq CW^G(\mathrm{co}(\mathcal{H}))$ and apply Corollary 1.

Lemma 4 *For every $\mathcal{A}$ and $\mathcal{H}$, the class $\mathrm{co}^k_{\mathcal{A}}(\mathcal{H})$ is a subset of the class $\mathrm{const}^k(\mathrm{co}(\mathcal{HA}))$.*

Lemma 5 *For every real valued function f, constant k, distribution D, sample S and accuracy parameters ϵ and η, $\mathbf{P}_D(yf(x) < 0) > \mathbf{P}_S(yf(x) \le \eta) + \epsilon$ if, and only if, $\mathbf{P}_D(ykf(x) < 0) > \mathbf{P}_S(ykf(x) \le k\eta) + \epsilon$*

Proof: Just note that $yf(x) < 0$ iff $ykf(x) < 0$ and $ykf(x) \le k\eta$ iff $yf(x) \le \eta$.

Remark 1 We can now apply the generalization bounds for convex hulls (e.g., [19]), to obtain generalization bounds for $\mathrm{co}^k_{\mathcal{A}}(\mathcal{H})$. This follows since for each $g \in \mathrm{co}^k_{\mathcal{A}}(\mathcal{H})$ there exist some $f \in \mathrm{co}(\mathcal{AH})$ such that $g = kf$.

We recall the definition of the pseudo-dimension of a class of real-valued functions.

Definition 4 *Suppose $\mathcal{H}$ is a class of real-valued functions defined over a domain $\mathcal{X}$. A set of points $\{x_1, \dots, x_m\}$ chosen from $\mathcal{X}$ is pseudo-shattered by $\mathcal{H}$ if there are real numbers $r_1, \dots, r_m$ such that for each $b \in \{0,1\}^m$ there is a function $h_b \in \mathcal{H}$ with $\mathrm{sgn}(h_b(x_i) - r_i) = b_i$ for $1 \le i \le m$. The pseudo-dimension of $\mathcal{H}$, denoted* P-dim$(\mathcal{H})$, *is the maximum cardinality of a pseudo-shattered subset of $\mathcal{X}$.*

The following Lemma, from [20], relates the sloppy covering number of $\mathrm{co}(\mathcal{H})$ to the pseudo-dimension of $\mathcal{H}$.

Lemma 6 ([20], Theorem 8) *For any $\eta > 0$,*

$$\mathcal{N}(\mathrm{co}(\mathcal{H}), \eta, 2e^{-N\eta^2/8}, m) \le (2em/(\eta d))^{dN},$$

where $d =$ P-dim$(\mathcal{H})$.

Corollary 3 *Given any pair of classes of functions, $\mathcal{H}$, $\mathcal{A}$, for every $\epsilon > 0$ and $k \in \mathbb{N}$, the ϵ-sloppy η-covering number of $\mathrm{co}^k_{\mathcal{A}}(\mathcal{H})$ is at most the result of the last lemma applied to the product of the sloppy $\eta/2k$-covering numbers of $\mathcal{H}$ and $\mathcal{A}$.*

We recall an important result from the work of Schapire et al. [19].

Lemma 7 ([19], Theorem 4) *Let $\mathcal{F}$ be a class of real-valued functions defined on the instance space $\mathcal{X}$. Let D be a distribution over $\mathcal{X} \times \{-1, 1\}$, and let S be a sample of m examples drawn independently at random according to D. Let $\epsilon > 0$ and $\eta > 0$. Then for any $f \in \mathcal{F}$, the probability that*

$$\mathbf{P}_D\left[Yf(X) \le 0\right] > \mathbf{P}_S\left[Yf(X) \le \eta\right] + \epsilon$$

is smaller than

$$2\mathcal{N}\left(F, \eta/2, \epsilon/8, 2m\right)\exp(-\epsilon^2 m/32).$$

Using Corollary 2 the following corollary follows, upon using similar arguments to those in Theorem 8 of [20].

Corollary 4 *Let D be a distribution over $\mathcal{X} \times \{-1, +1\}$, and let S be a sample of m points chosen independently at random according to D. Assume that $m \ge d_p \ge 1$, where d_p is the pseudo-dimension of $\mathcal{H}$, and assume that $\mathcal{A}$ is composed of piecewise constant functions, assuming constant values on a partition G of $\mathcal{X}$. Then for any $f \in$* $\mathrm{co}_{\mathcal{A}}(\mathcal{F})$

$$\mathbf{P}_D\left[Yf(X) \le 0\right] \le \mathbf{P}_S\left[Yf(X) \le \eta\right] + O\left(\frac{1}{\sqrt{m}}\left(\frac{|G|d_p\log^2(m/d_p)}{\eta^2} + \log\left(\frac{1}{\delta}\right)\right)^{1/2}\right).$$

Observe that Corollary 4 extends the results of [20] to functions formed by mixtures, where the mixing coefficients are themselves piece-wise constant functions. Note also that only difference between this result and Theorem 8 in [20] is the additional factor of $|G|$, accounting for the number of regions. An essential feature of this bound, as of its precursors in [19], is that it does *not* depend on the number of terms in the convex combination.

We next extend the results of Corollary 4 to smooth functions. The smoothness constraints are enforced through a Lipschitz condition, guaranteeing that the mixture coefficient functions do not change too rapidly. For technical reasons we also require that the domain $\mathcal{X}$ is bounded. The result below applies to the case $\mathcal{X} = [0,1]^d$, but may be easily extended to any bounded subset of $\mathbb{R}^d$.

Theorem 1 *Let D be a distribution over $[0,1]^d \times \{-1,+1\}$, and let S be a sample of m points chosen independently at random according to D. Assume that $m \geq d_p \geq 1$, where d_p is the pseudo-dimension of $\mathcal{H}$. Let the weight functions $\alpha_j(x)$ satisfy the Lipschitz condition $|\alpha_j(x) - \alpha_j(y)| \leq L\|x-y\|_\infty$ as well as the condition $\alpha_j(x) \geq a > 0$ for all j. Then with probability at least $1-\delta$, every function $f \in \mathrm{co}_{\mathcal{A}}(\mathcal{H})$ satisfies the following bound for all $\eta > 0$:*

$$\mathbf{P}_D\left[Yf(X) \leq 0\right] \leq \mathbf{P}_S\left[Yf(X) \leq \eta\right] + O\left(\frac{1}{\sqrt{m}}\left(\frac{d_p(L/\eta)^d \log^2(m/d_p)}{\eta^2} + \log\left(\frac{1}{\delta}\right)\right)^{1/2}\right).$$

Proof The idea of the proof is simple. Since the functions obey a Lipschitz condition, they cannot vary too rapidly over a bounded region. If we then partition the domain $\mathcal{X}$ into sub-regions, and replace the functions $\alpha_j(x)$ over each sub-region by a constant, we obtain a piecewise constant approximation. The proof then proceeds by characterizing the number of regions needed to achieve a given level of accuracy.

Consider a function $f \in \mathrm{co}_{\mathcal{A}}(\mathcal{H})$, namely

$$f(x) = \sum_i \alpha_i(x) h_i(x) \Big/ \sum_i \alpha_i(x).$$

Construct a finite η-cover of $\mathcal{H}$, denoted by $\hat{\mathcal{H}}$, namely for each $h \in \mathcal{H}$ there exists a function $\hat{h} \in \hat{\mathcal{H}}$ such that $|h(x) - \hat{h}(x)| \leq \eta$ for any $x \in S$. Define

$$\hat{f}(x) = \sum_i \alpha_i(x) \hat{h}_i(x) \Big/ \sum_i \alpha_i(x).$$

It is easy to show that $|f(x) - \hat{f}(x)| \leq \eta$ for all $x \in S$. Consider a partition of the hyper-cube $[0,1]^d$ into $K = \gamma^{-d}$ axis-parallel sub-cubes $\{\mathcal{X}_i\}_{i=1}^K$, each of volume γ^d. From the Lipschitz condition it follows that for any function $\alpha_j(x)$, and x, y in a sub-cube, $|\alpha_j(x) - \alpha_j(y)| \leq L\|x-y\|_\infty \leq L\gamma$, since the side-length of each sub-cube is γ. Let $\tilde{\alpha}_j(x)$ be the function obtained from $\alpha_j(x)$ by replacing it within each domain $\mathcal{X}_i$ with its value at ξ_i, the center of $\mathcal{X}_i$. Thus $\{\tilde{\alpha}_j(x)\}_j$ are piecewise constant functions assuming constant values in $\mathcal{X}_i$. We define a quantized approximation to $\hat{f}$ by

$$\tilde{f}_\gamma(x) = \sum_i \tilde{\alpha}_i(x) \hat{h}_i(x) \Big/ \sum_i \tilde{\alpha}_i(x). \qquad (1)$$

We show that $|\hat{f}(x) - \tilde{f}_\gamma(x)|$ is small if γ is small.

$$\begin{aligned}|\hat{f}(x) - \tilde{f}_\gamma(x)| &= \left|\frac{\sum_j \alpha_j(x)\hat{h}_j(x)}{\sum_k \alpha_k(x)} - \frac{\sum_j \tilde{\alpha}_j(x)\hat{h}_j(x)}{\sum_k \tilde{\alpha}_k(x)}\right| \\ &\leq \sum_j \left|\frac{\alpha_j(x)}{\sum_k \alpha_k(x)} - \frac{\tilde{\alpha}_j(x)}{\sum_k \tilde{\alpha}_k(x)}\right| \\ &\leq \frac{2\sum_j |\alpha_j(x) - \tilde{\alpha}_j(x)|}{\sum_j \tilde{\alpha}_j(x)},\end{aligned}$$

where the last line is obtained by simple algebra. Since $\tilde{\alpha}_j(x) \geq a > 0$ for each j, then using $|\alpha_j(x) - \tilde{\alpha}_j(x)| \leq |\alpha_j(x) - \alpha_j(\xi_i)| \leq L\gamma/2$, and setting $\gamma = a\eta/L$ we conclude that $|\hat{f}(x) - \tilde{f}_\gamma(x)| \leq \eta \; \forall x \in [0,1]^d$. Since $|f(x) - \hat{f}(x)| \leq \eta$ and $|\hat{f}(x) - \tilde{f}_\gamma(x)| \leq \eta$ for any $x \in S$, we conclude from the triangle inequality that $|f(x) - \tilde{f}_\gamma(x)| \leq 2\eta$ for all $x \in S$, where $\tilde{f}_\gamma(x) = \sum_j \tilde{\alpha}_j(x)\hat{h}_j(x) / \sum_j \tilde{\alpha}_j(x)$ and $\tilde{\alpha}_j(x)$ are constant over each of the K regions. We may therefore directly utilize the results of Corollary 4 replacing the number of regions K by $(L/a\eta)^d$. ■

Remark 2 It is helpful to understand the trade-off between the two terms appearing in the bound of Theorem 1. For a fixed value η, we have the standard trade-off which appears in all margin-based bounds (e.g., Chapter 13 in [1]), namely the empirical error $\mathbf{P}_S\left[Yf(X) \leq \eta\right]$ is monotonically increasing with η, while the second term is monotonically decreasing, leading to an optimal value for the margin η. A similar behavior can be seen to occur as a function of the smoothness parameter L. First, note that if $L = \eta$, the bound we obtain is of the exact same form as in [19]. On the other hand, as L increases we expect the first empirical term to decrease, while the second term increases. There thus seems to be an optimal value of smoothness, at which a tight bound is attained.

3 THE LOCBOOST ALGORITHM

A rather general approach to boosting, based on incremental greedy optimization, has been recently introduced by Mason et al. [16]. In this procedure a composite classifier is formed by incrementally adding on a weak classifier based on some margin-based empirical loss function. In this work, as in many boosting algorithms, a final composite hypothesis is constructed by a weighted combination of weak (base) classifiers. The coefficients of the combination in the standard approach, however, do not depend on the position of the point x whose label is desired. Since the boosting procedure filters the data sequentially through re-weighting, it is possible that some of the classifiers $h_t(x)$ were not exposed during training to any data in the vicinity of the point x. Moreover, greater flexibility can be achieved by having each classifier operate only in a localized region. It would thus seem more opportune to weight each classifier h_t at point x by a local weight $\beta_t(x)$ depending on x. In order to implement this idea, we recall the mixture of expert model [11], where the posterior class-conditional probability distribution $P(y|x)$ is expressed as a locally weighted mixture of base probabilities, $P(y|x, \Omega^k) = \sum_{j=1}^{k} \beta(x, \phi_j) p(y|x, \theta_j)$, where $(\phi_j \in \Phi, \; \theta_j \in \Theta)$, $\beta(x, \phi_j) \geq 0$, $\sum_{j=1}^{k} \beta(x, \phi_j) = 1$ and

$$\Omega^k = \{\phi_j, \theta_j\}_{j=1}^{k}.$$

In analogy with the boosting literature, we refer to $p(y|x, \theta_j)$ as 'weak' models. This type of structure, was shown to be very effective in problems of regression, classification and time series prediction. It was also shown in [11] that the well-known Expectation-Maximization (EM) algorithm is especially useful for the purpose of learning through maximum likelihood estimation.

In this work, we focus on an sequential approach, whereby experts are added on incrementally as in boosting. Assume that at step t a model $P_{t-1}(y|x, \Omega^{t-1})$ has been constructed, where Ω^{t-1} denotes all the parameters up to time $t-1$. At step t we form the model

$$\begin{aligned} P_t(y|x, \Omega^t) = (1 - \gamma_t(x, \phi_t)) P_{t-1}(y|x, \Omega^{t-1}) \\ + \gamma_t(x, \phi_t) p_t(y|x, \theta_t), \end{aligned} \tag{2}$$

where $P_0(y|x) = 1/2 \forall x, y$. Note, by induction, that for any t we may express $P_t(y|x, \Omega^t)$ in terms of the weak conditional probabilities by

$$P_t(y|x, \Omega^t) = \sum_{\tau=1}^{t} \beta_\tau(x, \phi^\tau) p_\tau(y|x, \theta_\tau), \tag{3}$$

where $\phi^\tau = \{\phi_1, \ldots, \phi_\tau\}$ and $\sum_{\tau=1}^{t} \beta_\tau(x, \phi^\tau) = 1$ for any x. To each of the weak conditional probabilities $p(y|x, \theta)$ we may also assign a soft-classifier given by

$$h(x, \theta) = 2p(y = 1|x, \theta) - 1.$$

The combined soft classifier is then given by

$$H_t(x) = \sum_{\tau=1}^{t} \beta_\tau(x, \phi^\tau) h_\tau(x, \theta_\tau). \tag{4}$$

Within a maximum-likelihood approach, at time t we wish to maximize the function

$$\begin{aligned} \ell(\phi, \theta; \Omega^{t-1}) &= \log \prod_{i=1}^{m} P_t(y_i|x_i, \Omega^t) \\ &= \sum_{i=1}^{m} \log \left[(1 - \gamma_t(x_i, \phi)) P_{t-1}(y_i|x_i, \Omega^{t-1}) \right. \\ &\quad \left. + \gamma_t(x_i, \phi) p_t(y_i|x_i, \theta) \right], \end{aligned} \tag{5}$$

with respect to the parameters ϕ and θ, the parameters $\Omega^{t-1} = \{\theta_1, \ldots, \theta_{t-1}, \phi_1, \ldots, \phi_{t-1}\}$ being fixed. In other words, at step t we set

$$\theta_t, \phi_t = \operatorname*{argmax}_{\theta \in \Theta, \phi \in \Phi} \ell(\phi, \theta; \Omega^{t-1}). \tag{6}$$

Note that the function (5) is related to the generalized cost functions used for boosting in [9] and [16], although it is not directly based on the margin function $yh(x)$.

If the class $\Gamma = \{\gamma(\cdot, \phi) : \phi \in \Phi\}$ contains the zero function, clearly maximization of (5) with respect to θ and ϕ, may only increase the likelihood. Note that for any finite value of m the log-likelihood is bounded from above by zero since $p(y|x, \theta) \leq 1$ and $0 \leq \gamma(x, \theta) \leq 1$. Therefore, if an increase in the likelihood can be guaranteed at each step, the algorithm is guaranteed to converge. However, care must be taken not to induce over-fitting.

It is interesting to observe that recent work by Li [12], in the context of density estimation, has shown that not much is lost by performing the optimization in an incremental fashion. It can be shown that these results may be extended to the present case of estimating conditional probability models. One finds that under appropriate regularity conditions on the conditional probability distribution $p(y|x, \theta)$, the value of the log-likelihood obtained incrementally after T steps of the procedure (6), is at most $O(1/T)$ away from the value of the true maximum log-likelihood obtained by optimizing the full mixture model of order T over all parameters simultaneously. More precisely, let $\hat{P}_T(y|x, \Omega^T)$ denote the conditional probability distribution obtained after T steps of the incremental algorithm (6), and denote by $\hat{P}_T^{\mathtt{MLE}}(y|x)$ the value of the maximum likelihood estimate obtained by simultaneously optimizing all parameters in the mixture model (3). Then one can show, under appropriate regularity conditions, that

$$\frac{1}{m} \sum_{i=1}^{m} \log \hat{P}_T(y_i|x_i, \Omega^T) \geq \frac{1}{m} \sum_{i=1}^{m} \log \hat{P}_T^{\mathtt{MLE}}(y|x) - \frac{c}{T},$$

It should be noted, though, that the result disregards optimization issues and assumes that a global maximum may be attained at each stage of the incremental procedure.

While standard numerical procedures may be used in order to maximize the log-likelihood, it is particularly convenient at this point to cast the problem as an incremental generalized EM (GEM) algorithm. The main motivation for doing this is the explicit connection to boosting through the optimization of a re-weighted version of the likelihood, and the decoupling of the optimization process (see below). Using standard results (see, for example, [11]), one can show that the EM-based maximization of the likelihood may be achieved by successively maximizing the function

$$\begin{aligned} Q_t(\phi, \theta|\phi^c, \theta^c) &= \sum_{i=1}^{m} h_i^{t-1} \log p_t(y_i|x_i, \theta) \\ &+ \sum_{i=1}^{m} \left[h_i^{t-1} \log \gamma_t(x_i, \phi) + (1 - h_i^{t-1}) \log(1 - \gamma_t(x_i, \phi)) \right] \end{aligned} \tag{7}$$

with respect to ϕ and θ and iterating, where

$$\begin{aligned} h_i^{t-1} = \gamma_t(x_i, \phi^c) p(y_i|x_i, \theta^c) \Big/ \\ \left[(1 - \gamma_t(x_i, \phi^c)) P_{t-1}(y_i|x_i, \Omega^{t-1}) + \gamma_t(x_i, \phi^c) p(y_i|x_i, \theta^c) \right] \end{aligned} \tag{8}$$

A brief discussion of the effect of the weights h_i^{t-1} is given below, following the description of the GEM algorithm.

The basic theory of the GEM algorithm [5] guarantees that if ϕ and θ are chosen so that

$$Q_t(\phi, \theta|\phi^c, \theta^c) \geq Q_t(\phi^c, \theta^c|\phi^c, \theta^c),$$

then the likelihood increases, namely

$$\ell(\phi, \theta; \Omega^{t-1}) \geq \ell(\phi^c, \theta^c; \Omega^{t-1}).$$

Observe that the maximization over the parameters ϕ and θ is broken up into two separate procedures. The implementation of the LocBoost algorithm is described in Figure 1 and is further discussed in Section 4.

```
The LocBoost Algorithm

1. Given: $(x_1, y_1), \ldots, (x_m, y_m) \in (\mathbb{R}^d \times \{\pm 1\})^m$
2. For $t = 1, 2, \ldots, T$
    (a) Set $\phi^c$ and $\theta^c$ to appropriate initial values (see Sec. 4).
    (b) Let $\phi_t$ and $\theta_t$ be any parameter values for which $Q_t(\phi, \theta|\phi^c, \theta^c) \geq Q_t(\phi^c, \theta^c|\phi^c, \theta^c)$.
    (c) Compute $P_t(y|x, \Omega^t)$ using (2).
3. Output $P_T(y|x, \Omega^T)$
```

Figure 1: The LocBoost algorithm

Some points are worth noting concerning the LocBoost algorithm. First, if 'past' performance at the point x_i is good, namely $P_{t-1}(y_i|x_i, \Omega^{t-1})$ (8) is large, then this data point is down-weighted, influencing the new weak classifier only marginally as is seen in the expression for h_i^{t-1}. Conversely, the weight h_i^{t-1} of the new weak model is increased near points for which the past prediction $P_{t-1}(y_i|x_i, \Omega^{t-1})$ is poor.

A standard approach to the representation of conditional probabilities is through the logistic function $p(y|x, \theta) = \left(1 + e^{-yf(x)}\right)^{-1} \equiv \sigma(yf(x))$, for some function $f(x)$. Note that this representation directly relates the margin $yf(x)$ to the probability in a very natural way, in that large margins correspond to confident predictions. It is important at this point to contrast our approach with the Logit algorithm of Friedman et al. [9] who consider conditional probabilities of the form $p(y|x) = \sigma\left(y \sum_{t=1}^{T} f_t(x)\right)$, where the functions f_t are estimated in an incremental greedy fashion as in [16]. In our case we form a locally weighted mixture of *probabilities* (rather than functions) and use an incremental EM algorithm for estimation.

The procedure described here is based on maximum likelihood estimation of the posterior class-conditional probability $P(y|x)$. Note that the empirical classification error of the soft classifier

$$H_t(x_i) = 2P_t(y_i = 1|x_i) - 1$$

may be easily bounded by the logarithmic loss using the bound

$$I(y_i H_t(x_i) < 0) \leq -\log_2 P_t(y_i|x_i).$$

Moreover, it is well known that under appropriate regularity conditions (e.g., [21]) the maximum likelihood estimator asymptotically minimizes the Kullback-Leibler divergence between the true underlying distribution and the estimated distribution.

In order to connect the Kullback-Leibler divergence to the classification error, we consider bounding the the latter by an expression depending on the former. Let $Q(y|x)$ be an estimator for the class-conditional probability distribution, and consider the plug-in classifier given by $g(x) = \text{sgn}(2Q(y = 1|x) - 1)$. Denote the true class-conditional probability distribution by by $P(y|x)$. Denoting the Bayes error by

$$P^* = \mathbb{E}\left\{\min(P(y = +1|x), P(y = -1|x))\right\},$$

we have from [6] (p. 93) that

$$\begin{aligned} &\mathbf{P}\{g(X) \neq Y\} - P^* \\ &\leq 2\left(\int_{\mathcal{X}} |P(y = 1|x) - Q(y = 1|x)|^2 \mu(dx)\right)^{1/2}. \end{aligned}$$

Next, note that for any two numbers η and ν such that $0 \leq \eta, \nu \leq 1$, $\eta \log \frac{\eta}{\nu} + (1 - \eta) \log \frac{1-\eta}{1-\nu} \geq (\eta - \nu)^2$ (the result may be easily established by subtracting the l.h.s. from the r.h.s. and showing that the resulting function is convex with a minimum of zero at $\nu = \eta$). We therefore conclude that

$$\begin{aligned} &\mathbf{P}\{g(X) \neq Y\} - P^* \\ &\leq 2\sqrt{\int_{\mathcal{X}} D_{\text{KL}}(P(y|x) \| Q(y|x)) \mu(dx)}. \end{aligned}$$

Although this result relates the probability of misclassification to the Kullback-Leibler divergence between the class-conditional distributions, the bound may not be tight in general (e.g., Theorem 6.5 in [6]). However, it does show that consistency in the sense of the Kullback-Leibler divergence leads to consistency in terms of classification error. It is interesting to note that Yang [22] recently established conditions under which bounds of this type are asymptotically tight in a minimax sense.

Before describing several numerical experiments we have performed, we comment on two recently proposed approaches to localized boosting. First, Maclin [13] has introduced an approach to boosting where the weights of each weak learner are determined by its accuracy on points which are similar to it. In particular, either a k nearest neighbor approach or a neural network are used in order to assess the accuracy. Second, Moerland and Mayoraz [17] introduced *DynaBoost*, which similarly to AdaBoost is based on a margin based exponential cost-function (rather than the log-likelihood function as in our case), except that the mixing coefficients depend on the input location, as in our approach. The experimental results of both these papers on several data-sets from the

UCI repository of Machine Learning demonstrate that they often significantly out-perform boosting, especially in situations where the weak learners are not very powerful. However, both approaches suffer from the potential for overfitting noisy data sets. We comment that the generalization bounds presented in Section 2 should apply to these two algorithms as well.

4 IMPLEMENTATION AND NUMERICAL EXAMPLES

The above description of the LocBoost algorithm (Figure 1) leaves open the choices for the weak probability model, the localizing function, and the method for choosing appropriate initial values for the parameters ϕ^c and θ^c. In our implementation of the LocBoost algorithm we take the weak probability model $p(y|x,\theta)$ to be the logistic function $\sigma(y\theta\cdot x) = (1+\exp(-y\theta\cdot x))^{-1}$, with x and θ being vectors in $\mathbb{R}^{d+1}$, where the extra dimension incorporates the threshold. The localizing function we use is an unnormalized symmetric multivariate normal distribution,
$\gamma(x|\phi) = \gamma(x|\mu,\Sigma) = \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$ with $\Sigma = s^2 I$. Clearly, $0 < \gamma(x|\mu,\Sigma) \leq 1$. Letting $\sigma_i = \sigma(y_i\theta x_i)$ we obtain from a simple calculation that $\nabla_\theta Q_t = \sum_{i=1}^m h_i^{t-1}(1-\sigma_i)y_i x_i$ and $\nabla_\theta^2 Q_t = -\sum_{i=1}^m h_i^{t-1}\sigma_i(1-\sigma_i)x_i x_i^T$ implying that Q_t is concave w.r.t. θ. This fact is important for numerical optimization and means that we can optimize Q_t w.r.t. θ to a global maximum. Nevertheless, the optimization of $\phi = (\mu,\Sigma)$ is non-concave and is therefore highly susceptible to local maxima problems. It is thus extremely important to choose appropriate initial values for these parameters.

We use the following strategy for choosing the initial values of μ and Σ. We identify the set S of training examples that are badly classified by P_{t-1}, and attempt to partition S into "large" clusters containing as many as possible training examples which are badly classified by P_{t-1} and as few as possible training examples that are correctly classified by P_{t-1}. This partition is computed using a standard algorithm for identifying strongly connected components [4] applied to the heuristically constructed graph $G = (V,E)$ whose vertex set V is the set of training examples, and whose edges are constructed as follows. For each $x \in S$, let $x_1, x_2, \ldots, x_m$ be the entire training set sorted by distance to x. Let j be the minimum index such that x_j is not in S. For all $1 \leq i < j$ we let the (undirected) edge (x, x_i) be in E. Intuitively, each connected component of G tends to identify a cluster of training examples which are incorrectly classified by P_{t-1}. We therefore take μ to be the mean of the largest connected component G' of G, and s^2 ($\Sigma = s^2 I$) to be the average of of the main diagonal of the covariance matrix of G'. The routine for computing these initial choices for (μ, s^2) is summarized in Figure 2.

For the GEM optimization step (step (b) in Figure 1) we used trust region Newton and quasi-Newton line search [2]. In order to ease the computational burden of these algorithms one optimizes over a random sub-sample of the training examples (with a new sample chosen for each boosting round). In some of the experimental results reported below we used this strategy.

Compute Initial (μ, s^2) values for round t

```
1. Given: $(x_1,y_1),\ldots,(x_m,y_m) \in (\mathbb{R}^d \times \{\pm 1\})^m$
2. $S = \{x_i : \mathrm{sign}(2P_{t-1}(y|x_i) - 1) \neq y_i\}$
3. Construct $G = (V,E)$
   (a) $V = \{x_i\}$
   (b) For each $x \in S$
        i. Let $x_1, x_2, \ldots, x_m$ be the elements of $V$ sorted by increasing distance to $x$
       ii. Set $j_x = \mathrm{argmin}_j\{x_j \notin S\}$
      iii. For all $1 \leq i < j_x$ let $(x, x_i) \in E$
4. Compute $C_1, \ldots, C_k$, the strongly connected components of $G$
5. Set $i^* = \mathrm{argmax}\{|C_i|\}$
6. Set $\mu = \mathrm{mean}(C_{i^*})$
7. Set $s^2 = \mathrm{mean}(\mathrm{trace}(\mathrm{cov}(C_{i^*})))$
8. Output: $\mu, s^2$
```

Figure 2: A heuristic for computing reasonable initial (μ, s^2) values before round t

The frames in Figure 3 depict a run of the algorithm applied to a toy XOR problem of 400 points drawn from 4 symmetric Gaussians. In this problem the optimal Bayes decision boundary consists of the x and y axes, and the optimal Bayes error is approximately 1.2. The underlying weak classifiers and the localizing functions are depicted in the 4 top frames in Figure 3. In each of these frames, the line corresponds to the weak model and the two concentric ellipsoidal[1] contours depict the localizing function computed after an iteration consisting of 2 EM steps. The 4 bottom frames in the figure correspond to the first 4 boosting iterations of the algorithm where each frame depicts the final (strong) classifier at the end of a round. The optimization performed in each EM step was over a random sample containing 25% of the training examples. The inner ellipsoid correspond to a 0.9 contour line and the outer ellipsoid, to a 0.5 contour line.

As can be seen, the algorithm progressively and systematically identifies poor-performance regions which are treated by new localized weak classifiers. A 10-fold cross-validation run consisting of 5 boosting iterations each of which computes one EM round, resulted in an average error of 0.152 and standard deviation of 0.0067.

An attractive feature of the algorithm is the relatively

[1] These ellipsoids are in fact circles and appear as ellipsoids due to the aspect ratio of the figure.

Dataset	No. attributes	No. examples
Sonar	60	208
Ionosphere	34	351
Pima	8	768

Table 1: Characteristics of three datasets from the UCI repository used in the experiments

Dataset	LocBoost	RealBoost
Sonar	24.3 ± 5.43	20.1 ± 9.2
Ionosphere	12.3 ± 2.8	10.2 ± 5.5
Pima	22.6 ± 2.2	26.7 ± 4.7

Table 2: 5-fold cross-validation test error results obtained by LocBoost and RealBost on three datasets from the UCI repository

smooth[2] decision surfaces it generates. This can also be seen in Figure 4 depicting the decision boundary computed by the algorithm after 8 boosting iterations over a synthetically constructed 800-point "spirals" dataset.

Our initial experiments with real datasets from the UCI repository appear to be promising and comparable to those obtained by standard boosting algorithms. For example, Table 2 summarizes cross-validated test error results obtained by LocBoost and RealBoost (confidence rated AdaBoost) [20] for three datasets from the UCI repository. The figures are the 5-fold cross-validated error results obtain by LocBoost and RealBoost, respectively, where the weak classifier used by RealBoost is the perceptron "pocket algorithm" [10]. The LocBoost algorithm was applied with up to 10 boosting rounds and 5 EM iterations per step. RealBoost was applied with 50 boosting rounds in all runs and in each boosting round the pocket algorithm was trained for 1000 iterations. Note that these preliminary results are for illustrative purposes only and are by no means conclusive. For example, for the first two datasets (Sonar and Ionosphere) standard versions of boosting algorithms like AdaBoost, using different weak classifiers, have obtained better cross-validation errors (see e.g. Freund and Schapire [8]).

5 DISCUSSION

We have presented and analyzed the mixture of expert architecture for the problem of binary classification. Generalization error bounds, in the spirit of [19], have been established. Moreover, we have introduced an incremental procedure based on maximum-likelihood estimation and the EM algorithm, which bears strong affinities to boosting algorithms, in particular to their incarnations as gradient descent in function space [16]. Finally, we have presented some preliminary numerical simulations, demonstrating the potential practical efficacy of the approach.

Several open questions are left for future research. First, an extension of the approach to multi-category classification should be quite straightforward given the probabilistic framework adopted here (see, for example, [9]). Second, as stressed in the text, a possible problem of the approach is the potential for over-fitting. An important objective of our immediate research is the establishment of effective regularization procedures. Third, characterization of good choices of mixing functions $\gamma(x, \phi)$ and class conditional probabilities $p(y|x, \theta)$ are needed, yielding both good representational power and effective optimization. Fourth, an interesting research direction would be the consideration of other cost functions, perhaps in the spirit of the theoretically motivated functions introduced in [15]. It would be interesting to see whether EM type algorithms can be developed in these situations. Finally, a detailed numerical comparison to other boosting-type algorithm is in order.

Acknowledgments: We are grateful to Zeev Litichever for his help with the numerical simulations.

References

[1] M. Anthony and P.L. Bartlett. *Neural Network Learning; Theoretical Foundations*. Cambridge University Press, 1999.

[2] D.P. Bertsekas *Nonlinear Programming*, Second Edition. Athena Scientific 1999

[3] L. Breiman Arcing the edge. Technical report 486, Statistics department, University of California 1997.

[4] T. H. Cormen, C.E. Leiserson and R.L. Rivest. *Introduction to Algorithms.* MIT Press 1990

[5] A.P. Dempster, N.M. Laird and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society B*, 39:1–38, 1977.

[6] L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer Verlag, New York, 1996.

[7] M. Frean and T. Downs. A simple cost function for boosting. Technical report, Dep. of Computer Science and Electrical Engineering, University of Queensland, 1998.

[8] Y. Freund and R.E. Schapire, Experiments with a new boosting algorithm. *Proceeding of the thirteenth international conference on machine learning*, 1996

[9] J. Friedman, T. Hastie and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *The Annals of Statistics*, To appear, 2000.

[10] S. Gallant. Perceptron-based learning algorithms. *IEEE Trans. Neural Networks*, 1(2):179-191, 1990.

[11] M.I. Jordan and R.A. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, 6(2):181–214, 1994.

[12] Q. Li. *Estimation of Mixture Models*. PhD thesis,

[2] The rugged appearance of the decision boundaries in the figures is due to low plot resolution.

Yale University, New Haven, Conneticut, U.S.A, 1999.

[13] R. Maclin Boosting classifiers locally. Proceedings of AAAI, 1998.

[14] S. Mallat and Z. Zhang. Matching pursuit with time-frequencey dictionaries. *IEEE Trans. Signal Processing*, 41(12):3397–3415, December 1993.

[15] L. Mason, J. Baxter, P. Bartlett. Improved generalization through explicit optimization of margins. *Machine Learning*, To appear

[16] L. Mason, J. Baxter, P. Bartlett and M. Frean. Functional gradient techniques for combining hypotheses. In A. Smola, P. Bartlett, B. Schölkopf and D. Schuurmans, (Eds.), *Advances in Large Margin Classifiers,*. MIT Press, 2000.

[17] P. Moerland and E. Mayoraz DynamBoost: combining boosted hypotheses in a dynamic way Technical Report RR 99-09, IDIAP Switzerland, May 1999.

[18] G. Rätsch, T. Onoda and K.R. Müller Regularizing AdaBoost. in Kearns, Solla and Cohn (Eds.) *Advances in Neural Information Processing Systems 11*, pp. 564-570. MIT Press.

[19] R.E. Schapire, Y. Freund, P. Bartlett and W.S. Lee. Boosting the margin: a new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):1651–1686, 1998.

[20] R.E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297-336, 1999.

[21] M.J. Schervich. *Theory of Statistics*. Springer Verlag, New York, 1995.

[22] Y. Yang. Minimax nonparametric classification - Part I: rates of convergence. *IEEE Trans. Inf. Theory*, 45(7):2271-2284, 1999.

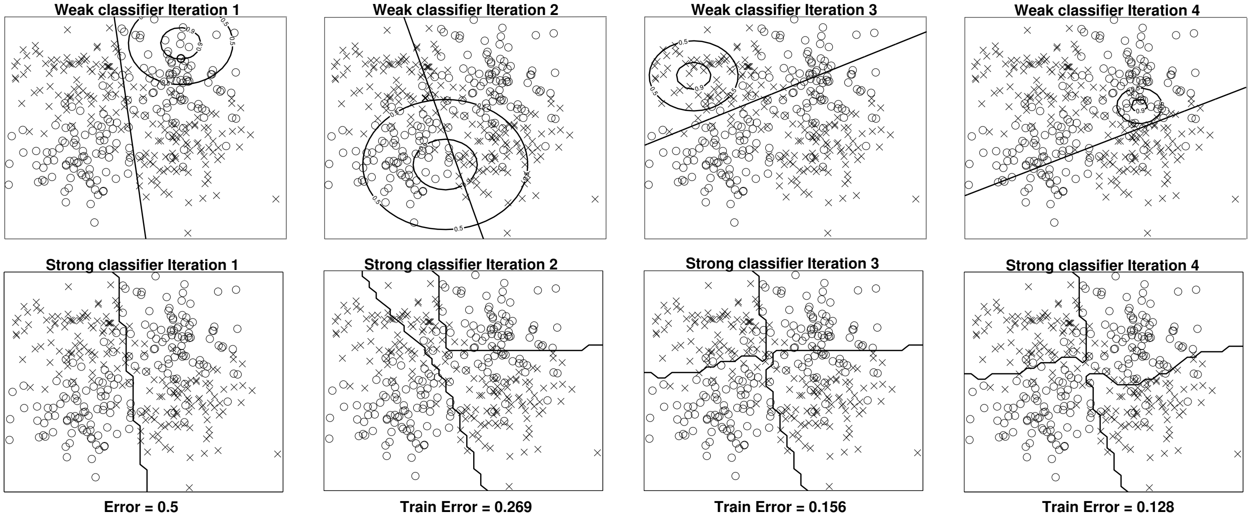

Figure 3: A run of LocBoost over a 400-point XOR problem. The 'x's and 'o's represent training points from two classes. The top 4 frames correspond to the underlying first 4 weak classifiers and the localizing functions. In each of these frames the line represent the weak model and the two ellipsoids depict the activation area of this weak model (the inner ellipsoid corresponds to a 0.9 contour level of this region and the outer ellipsoid corresponds to a 0.5 contour level. The lower 4 frames depict the first 4 strong classifiers constructed by the localized boosting algorithm. In each of these frames the lines represent the decision boundary. The caption on top of each frame specifies the iteration number and bottom caption (of the top frames) specifies the training error achieved.

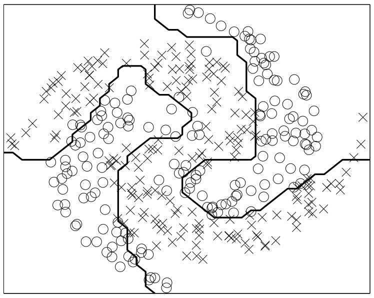

Figure 4: The decision region obtained by LocBoost after 8 boosting rounds applied to a synthetic spirals problem.