JMLR: Workshop and Conference Proceedings 29:451–466, 2013 ACML 2013

Locally-Linear Learning Machines (L3M)

Joseph Wang JOEWANG@BU.EDU and
Venkatesh Saligrama SRV@BU.EDU
Department of Electrical & Computer Engineering
Boston University
Boston, MA 02215

Editor: Cheng Soon Ong and Tu Bao Ho

Abstract

We present locally-linear learning machines (L3M) for multi-class classification. We formulate a global convex risk function to jointly learn linear feature space partitions and region-specific linear classifiers. L3M's features such as: (1) discriminative power similar to Kernel SVMs and Adaboost; (2) tight control on generalization error; (3) low training time cost due to on-line training; (4) low test-time costs due to local linearity; are all potentially well-suited for "big-data" applications. We derive tight convex surrogates for the empirical risk function associated with space partitioning classifiers. These empirical risk functions are non-convex since they involve products of indicator functions. We obtain a global convex surrogate by first embedding empirical risk loss as an extremal point of an optimization problem and then convexifying this resulting problem. Using the proposed convex formulation, we demonstrate improvement in classification performance, test and training time relative to common discriminative learning methods on challenging multiclass data sets.

1. Introduction

We present a convex parameterization for learning locally linear decision boundaries for multi-class classification. The proposed classifier partitions the features space into local regions, and within each region applies an independent classifier, as shown in Fig. 1, and we refer to them as space partitioning classifiers (SPC). Our approach jointly learns both feature space partitions and independent classifiers in each partition by optimizing a globally convex risk function. A typical output of L3M is illustrated in Fig. 1. Although the proposed method admits local kernelization (i.e, locally non-linear boundaries), practically, locally linear boundaries are desirable.

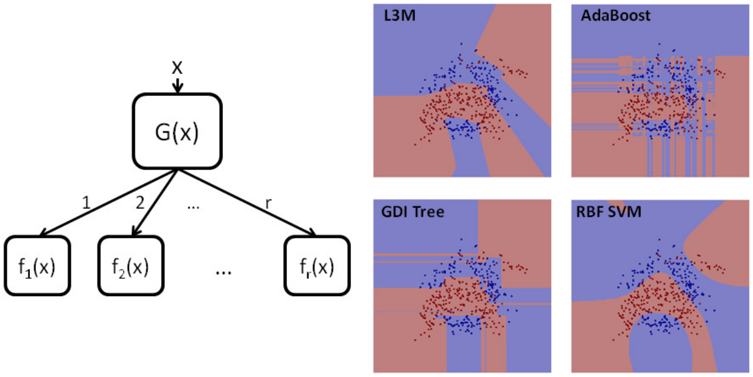

Figure 1: **Left:** L3M architecture. **Right:** Decision regions on the banana data set for local linear classification, AdaBoost using stumps as weak learners, GDI decision tree, and Gaussian RBF SVM.

© 2013 J. Wang & V. Saligrama.

Locally Linear Learning Machine($L3M$): Fig. 1 depicts some of the practical advantages of $L3M$s that make them suitable for "big-data" applications. $L3M$s are flexible and have good *discriminative* power–similar to other powerful methods such as Adaboost and Kernel-SVMs–since for many real-world datasets, class boundaries are well-behaved and approximable by a few locally linear functions & partitions. $L3M$s have low VC dimension and have predictable generalization performance. L3M's have low training time cost in contrast to Adaboost & Kernel-SVMs. Indeed, on account of local-linearity and convex parameterization they can be trained on-line using stochastic gradient descent algorithms and are guaranteed to converge. Finally, $L3M$s have low test-time computational cost in contrast to Kernel-SVMs which typically require large number of support vectors for many real-world datasets.

We derive a tight convex surrogate for the empirical risk function associated with SPCs. To motivate the problem consider the simple case of binary classification using 2-region local-linear learning (a setup well suited for XORs). The empirical risk function can be written as:

$$R(G, f_1, f_2) = \sum_{i=1}^{n} \left[\mathbb{1}_{G(x_i)<0} \mathbb{1}_{y_i f_1(x_i)\leq 0} + \mathbb{1}_{G(x_i)\geq 0} \mathbb{1}_{y_i f_2(x_i)\leq 0} \right]. \tag{1}$$

where, $(x_1, y_1), \ldots, (x_n, y_n)$, are labeled training set examples with $x_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$ for $i = 1, \ldots, n$, $G(\cdot) \in \{-1, 1\}$, the partitioning map and $f_1(\cdot)$ and $f_2(\cdot)$, the classifiers associated with the two regions. The function $G(x)$ partitions the feature space into two regions, and in each region, a local classifier, $f_1(x)$ or $f_2(x)$, predicts a label for the observation. The goal is to learn $G(x), f_1(x), f_2(x)$ jointly that minimizes the empirical loss. Such product of indicator functions have been considered before and whose origins can be traced to Perceptron Decision Trees Bennett et al. (2000), in particular the work of Bennett & Mangasarian Bennett and Mangasarian (1993). In their work they proposed to bound each indicator function with hinge-loss. The resulting objective, leads to a bilinear optimization problem, which is known to be NP-complete Megiddo; Blum and Rivest (1992), and so they propose branch & bound algorithms in this context. More recently, Wang & Saligrama Wang and Saligrama (2012) proposed a heuristic method based on alternative minimization (AM) for solving Eq. 1 but their AM method lacks convergence guarantees. The issue is that while the product of indicators can be convexified, for instance,

$$\mathbb{1}_{G(x)<0} \mathbb{1}_{f(x)y<0} \leq \max(1 - G(x) - f(x)y, 0),$$

this does not turn out to be useful as it evidently dilutes the separation between partitioning function, $G(x)$ and classifier function $f_1(x), f_2(x)$. To address this point we obtain a global convex surrogate by first embedding empirical risk loss as an extremal point of an optimization problem. The key aspect of this embedding is that the resulting optimization problem is composed of single indicator functions and so can be tightly convexified using hinge-losses. We can generalize this approach to derive convex parameterizations not only for losses composed of product of multiple indicator loss functions (cascade classifiers) but also multi-region partitioning functions. Using the proposed convex formulation, we demonstrate improvement in classification performance, test and training time relative to common discriminative learning methods on challenging multiclass data sets.

Issues with Convex Losses: Naive convexification of SPCs induces classifier symmetry, which leads to fundamental problems. To demonstrate this, consider the two classifiers shown in Fig. 2. Both classifiers induce the same decision boundaries and empirical error over the training set, and

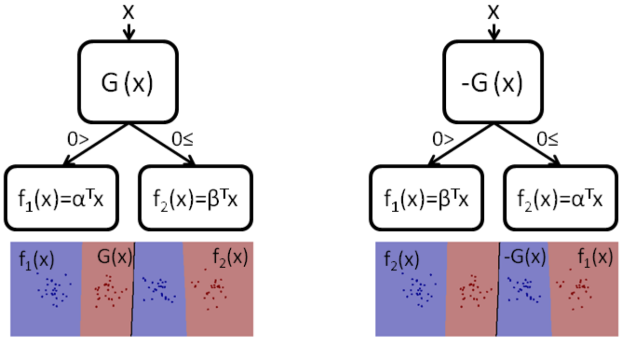

Figure 2: Two classifiers that induce identical decision boundaries. This symmetry of solutions is a fundamental obstacle for convex relaxations of the empirical error.

both classifiers minimize the empirical error. Unfortunately, minimizing any convex relaxation of the empirical loss, will yield a single global classifier instead of either of these optimal classifiers:

Proposition 1.1 *For any convex relaxation of* (1)*, the solution* $f_1 = f_2$ *and* $G = 0$ *is globally optimal.*

Proof Consider the solution $f_1 = f_1^*$, $f_2 = f_2^*$, and $G = G^*$ that minimizes the empirical error. The solution $f_1 = f_2^*$, $f_2 = f_1^*$, and $G = -G^*$ will induce the same decision boundaries and identical loss. For any convex relaxation of the empirical risk, by the definition of convexity, the solution $f_1 = \frac{f_1^*+f_2^*}{2}$, $f_2 = \frac{f_1^*+f_2^*}{2}$, and $G = 0$ will at least match the loss of these solutions. ■

While the convex relaxation of the solution yields a single global classifier, this is not the optimal solution with respect to the indicator loss function, such as the case in Fig. 2. The symmetry of the loss function around the point $G = 0$ presents a fundamental limitation for all convex relaxations. Similar issues have previously been raised when convexifying latent variable models, as noted by Guo and Schuurmans Guo and Schuurmans (2008). A simple way to overcoming this issue is to break the symmetry. Specifically, we can remove the solutions $G < G^*$ from the set of feasible solutions to the optimization problem by imposing a constraint. In particular, we accomplish this by choosing a random point, x_k, and constraining $G(x_k) \geq \beta$, which immediately removes the symmetric part of solution.

1.1. Related Work

Apart from the closely related work of Bennett and Mangasarian (1993); Wang and Saligrama (2012) described above, our approach is also loosely related to mixture of experts framework Lima et al. (2007); svm. The mixture of experts framework hybridizes generative and discriminative approaches by replacing the partitioning classifier, G, with a ”latent” probability distribution. Alternating minimization is used, switching between learning the parameters of the ”latent” distribution and training local classifiers using standard learning methods.

The SPC architecture appears to resemble decision trees L. Breiman and Stone (1984) and in particular decision trees with multivariate splits Brodley and Utgoff (1995). Nevertheless, to the best of our knowledge multivariate splitting techniques have focused on either optimizing a heuristic, such as split purity or entropy, or greedily attempting to discriminate the data at each stage without

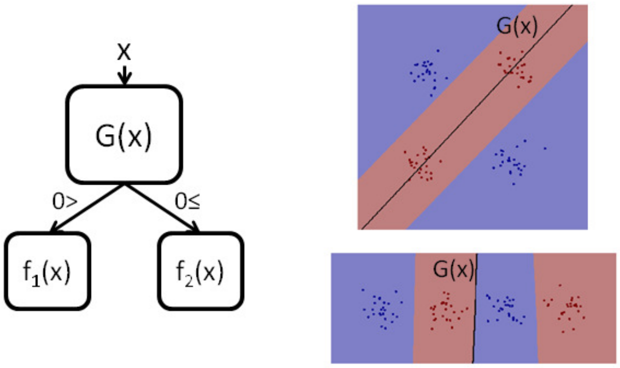

Figure 3: **Left:** 2-region L3M architecture. **Right:** L3m decision boundaries for two synthetic 2-dimensional examples.

regard to proceeding splits. Furthermore, the entropy or purity heuristics employed by decision trees are often difficult to optimize, limiting each decision to single feature splits which can be optimized by brute force search. In contrast, our approach directly minimizes a surrogate on the global empirical risk. Another crucial difference is that L3M is a single level multi-partition split that directly assigns the observation to a local classifier. In contrast decision trees are generally constructed using binary splits with relatively large depth.

Approximating decision boundaries with piecewise simple functions has also been proposed in generative learning schemes, such as Mixture Discriminant Analysis (MDA), proposed by Hastie *et al.* Hastie and Tibshirani (1996), where each class is modeled as by mixture of Gaussian distributions. Additional piecewise linear techniques have been proposed in the past Dekel and Shamir; Dai et al. (2006); Toussaint and Vijayakumar (2005), however these approaches do not learn decision boundaries based on minimizing global empirical risk.

2. Convex Parametrization for Binary Partitioning & Binary Classification

To build intuition we first consider the 2-region binary L3M described by the empirical loss in Eq. 1. The classification output $F(x)$ as shown in Fig. 3 associated with this empirical loss can be expressed as a function of the partitioning and local classifiers:

$$F(x) = \mathbb{1}_{G(x)<0} f_1(x) + \mathbb{1}_{G(x)\geq 0} f_2(x). \tag{2}$$

The function $G(x)$ partitions the feature space into two regions, and in each region, a local classifier, $f_1(x)$ or $f_2(x)$, predicts a label for the observation.

We recast the empirical risk as an optimization problem over introduced variables, transforming the problem from a fundamentally difficult bilinear optimization problem to a convex optimization problem. To accomplish this, we first make the following observation:

Proposition 2.1 *The product of indicators can be expressed as a minimization:*

$$\mathbb{1}_{a<0}\mathbb{1}_{b<0} = \min_{\lambda\in[0,1]} \lambda \mathbb{1}_{a<0} + (1-\lambda)\mathbb{1}_{b<0}$$

This observation allows the product of indicators to be separated into a linear combination of indicators. One natural approach is to replace the products of indicators in Eqn. 1 with this transformation, transforming the empirical error to:

$$R(G, f_1, f_2) = \sum_{i=1}^{n} \min_{\lambda_1,\lambda_2\in[0,1]} \left[\lambda_1 \mathbb{1}_{G(x_i)<0} + (1-\lambda_1)\mathbb{1}_{y_i f_1(x_i)\leq 0} + \lambda_2 \mathbb{1}_{G(x_i)\geq 0} + (1-\lambda_2)\mathbb{1}_{y_i f_2(x_i)\leq 0}\right].$$

While replacing the indicator functions with upper-bounding surrogates yields a tighter surrogate function on the empirical error than previously proposed bilinear surrogates Bennett and Mangasarian (1993), the problem is still a bilinear optimization problem whose global optimum is computationally intractable to find. Instead, we make one more basic observation that allows us to convert the problem from a bilinear optimization problem to a convex optimization problem:

Proposition 2.2 *The empirical error can be expressed by the event of a correctly classified observation:*

$$\mathbb{1}_{F(x_i)\neq y_i} = 1 - \mathbb{1}_{F(x_i)=y_i}$$

. Using these observations, we reformulate the empirical error by expressing the empirical error with respect to the event of a correctly classified observation:

Theorem 2.3 *The empirical risk* (1) *can equivalently be expressed:*

$$R(G, f_1, f_2) = \sum_{i=1}^{n} \max\left[\mathbb{1}_{G(x_i)\geq 0} + \mathbb{1}_{y_i f_2(x_i)\leq 0}, \mathbb{1}_{y_i f_1(x_i)\leq 0} + \mathbb{1}_{G(x_i)<0}\right] - 1. \tag{3}$$

Proof From Proposition 2.2, we express the empirical error with respect to the event of a correct classification:

$$R(G, f_1, f_2) = \sum_{i=1}^{n} \left[1 - \mathbb{1}_{G(x_i)<0}\mathbb{1}_{y_i f_1(x_i)>0} - \mathbb{1}_{G(x_i)\geq 0}\mathbb{1}_{y_i f_2(x_i)>0}\right]$$

From Proposition 2.1, we convert the empirical risk to a minimization over two introduced variables, λ_1 and λ_2:

$$\begin{aligned} R(G, f_1, f_2) &= \sum_{i=1}^{n} \left[1 - \min_{\lambda_1\in[0,1],\lambda_2\in[0,1]} \left(\lambda_1 \mathbb{1}_{G(x_i)<0} + (1-\lambda_1)\mathbb{1}_{y_i f_1(x_i)>0} + \lambda_2 \mathbb{1}_{G(x_i)\geq 0} + (1-\lambda_2)\mathbb{1}_{y_i f_2(x_i)>0}\right)\right] \\ &= \sum_{i=1}^{n} \max_{\lambda_1\in[0,1],\lambda_2\in[0,1]} \left[1 - \lambda_1 \mathbb{1}_{G(x_i)<0} - (1-\lambda_1)\mathbb{1}_{y_i f_1(x_i)>0} - \lambda_2 \mathbb{1}_{G(x_i)\geq 0} - (1-\lambda_2)\mathbb{1}_{y_i f_2(x_i)>0}\right]. \end{aligned}$$

By definition, $\mathbb{1}_{z<0} = 1 - \mathbb{1}_{z\geq 0}$, so we substitute for the indicator functions:

$$R(G, f_1, f_2) = \sum_{i=1}^{n} \max_{\lambda_1\in[0,1],\lambda_2\in[0,1]} \left[\lambda_1 \mathbb{1}_{G(x_i)\geq 0} + (1-\lambda_1)\mathbb{1}_{y_i f_1(x_i)\leq 0} + \lambda_2 \mathbb{1}_{G(x_i)<0} + (1-\lambda_2)\mathbb{1}_{y_i f_2(x_i)\leq 0} - 1\right].$$

The optimal values of λ_1 and λ_2 may not be unique. However, given that $\mathbb{1}_{G(x_i)\geq 0} = 1 - \mathbb{1}_{G(x_i)<0}$, one optimal solution always lies on the line $\lambda_1 = 1 - \lambda_2$. We substitute $\lambda = \lambda_1$ and $\lambda = 1 - \lambda_2$:

$$R(g, f_1, f_2) = \sum_{i=1}^{n} \max_{\lambda\in[0,1]} \left[\lambda \mathbb{1}_{G(x_i)\geq 0} + (1-\lambda)\mathbb{1}_{y_i f_1(x_i)\leq 0} + (1-\lambda)\mathbb{1}_{G(x_i)<0} + \lambda \mathbb{1}_{y_i f_2(x_i)\leq 0} - 1\right].$$

The variable λ acts as a maximization function and can be removed, yielding the expression in (3). ■

Convex Surrogate: A key advantage of the empirical risk as formulated in Eqn. 3 is that convexity is preserved when replacing the indicator functions with convex surrogate functions, whereas introducing convex surrogate functions in the empirical risk proposed in (1) does not generally yield a convex objective. From the empirical risk as formulated in Eqn. 3, we construct a convex, upper-bounding surrogate function by replacing the indicator functions with hinge losses:

$$\hat{R}(G, f_1, f_2) = \sum_{i=1}^{n} \max \Big[(1 - y_i f_1(x_i))_+ + (1 - G(x_i))_+, (1 + G(x_i))_+ + (1 - y_i f_2(x_i))_+\Big] - 1, \quad (4)$$

where the hinge loss is defined $(1 - z)_+ = \max(1 - z, 0)$. This relaxation of the empirical risk is not only convex, but additionally is the tightest convex relaxation as stated below. Proof appears in supplementary section.

Proposition 2.4 *For a function of the form* $\max\Big[\mathbb{1}_{a\geq 0} + \mathbb{1}_{b\leq 0}, \mathbb{1}_{c\leq 0} + \mathbb{1}_{d\leq 0}\Big] - 1$, *the tightest upper-bounding convex surrogate is given by* $\max\Big[(1+a)_+ + (1-b)_+, (1-c)_+ + (1-d)_+\Big] - 1.$

The final optimization problem, including the linear constraint arising from Prop. 1.1 and a regularization term to maximize margins, can be formulated:

$$\min_{G, f_1, f_2, G(x_k)\geq\beta} \sum_{i=1}^{n} \max \Big[(1-y_i f_1(x_i))_+ + (1-G(x_i))_+, (1+G(x_i))_+ + (1-y_i f_2(x_i))_+\Big] + \lambda\left(\|f_1\|_2^2 + \|f_2\|_2^2\right) \quad (5)$$

where x_k is a randomly chosen test observation and $\lambda > 0$ and $\beta > 1$ are user chosen parameters that minimize the empirical training error.

Fig. 3 shows the decision boundaries on two synthetic 2-dimensional examples on the right. On the top right of Fig. 3, data is generated from four symmetric Gaussian distributions, with means at $(-1,-1), (-1,1), (1,-1), (1,1)$, with data generated from Gaussians centered at $(1,1)$ and $(-1,-1)$ having positive labels and data centered at $(1,-1)$ and $(-1,1)$ having negative labels. On the bottom right, the data is drawn from 4 symmetric Gaussians with means on the x-axis at $(0,0), (1,0), (2,0), (3,0)$ and alternating positive and negative labels associated with each Gaussian. As seen in Fig. 3, the proposed convex formulation correctly learns both a partitioning and and local classification functions, resulting in perfect classification of the training sets.

Qualitative Behavior of Indicator & Convex Risks:
To examine the behavior of the reformulated loss and convex relaxation, we consider two cases. In the first case, we assume that the partitioning function G is fixed and examine the effect of local classifiers f_1 and f_2 on the loss. In the second case we examine the opposite situation, where the local classifiers f_1 and f_2 are fixed and observe the behavior of G on the loss.

In the first case where G is fixed, $\mathbb{1}_{y_i f_1(x_i)\leq 0} + \mathbb{1}_{G(x_i)<0} - 1 \geq \mathbb{1}_{G(x_i)\geq 0} + \mathbb{1}_{y_i f_2(x_i)\leq 0} - 1$, so the empirical error simplifies to $\mathbb{1}_{y_i f_1(x_i)\leq 0}$. The empirical risk on the observation x_i has the desired behavior, as it is independent of $f_2(x_i)$, with a value of 1 if $f_1(x_i) \neq y_i$ and a value of 0 if $f_1(x_i) = y_i$.

In the second case where the local classifiers are fixed, $\mathbb{1}_{y_j f_1(x_j)\leq 0} + \mathbb{1}_{G(x_j)<0} - 1 \geq \mathbb{1}_{G(x_j)\geq 0} + \mathbb{1}_{y_j f_2(x_j)\leq 0} - 1$, so the loss can be simplified to $\mathbb{1}_{G(x_j)<0}$. This can be viewed as a pseudo-label for the classifier G on observation x_j, with a "correct" label of $G(x_j) = 1$, such that the observation x_j is partitioned into the second region and correctly classified by f_2, and an "incorrect" label of $G(x_j) = -1$, where the observation is partitioned into the first region and incorrectly classified by f_1. In this manner, the pseudo-label partitions training examples into the regions where the local classifier correctly estimates the label, as described in Wang and Saligrama (2012).

Similarly, we examine the same cases for the convex loss function in Eqn. 5. For the first case, where the partitioning function is fixed $G(x_i) = -1$, the objective of (5) is independent of $f_2(x_i)$ unless the hinge loss of $y_i f_2(x_i)$ is larger than the partitioning margin and the hinge loss of $y_i f_1(x_i)$.

In the case where the local classifier margins are roughly the same magnitude, $|f_1(x_i)| \approx |f_2(x_i)|$, the surrogate loss function is independent of $f_2(x_i)$, that is not dependent on the classifier in the region the observation is not assigned.

Alternatively, consider the second case with fixed local classifiers such that for an observation x_j, $y_j f_1(x_j) = 1$ and $y_j f_2(x_j) = -1$. For the observation x_j, the classifier G minimizes the loss function is $G(x_j) = -1$, that is observation x_j is partitioned into region 1, with the loss equalling 1. In the event that $G(x_j) = 1$, the observation is partitioned into region 2, where a mistake is made, resulting in a loss equalling 3. As in the indicator case, the optimal solution of the convex loss function is to partition observations into the region where the local classifier makes a correct classification.

L3M Summary: We summarize some of unique features of L3M below.

Global Minimum: The global minimum of the objective function can be efficiently found using existing convex optimization tools. This allows for reliable and repeatable performance compared to finding a local minimum of the non-convex formulations as done in past work Bennett and Mangasarian (1993); Wang and Saligrama (2012).

Outlier Robustness: The proposed surrogate function is more robust to outliers, as the margins of the partitioning and classifying functions add as compared to the multiplicative behavior exhibited by the bilinear loss formulation Bennett and Mangasarian (1993). As a result, outlier observations far away from both the partitioning boundary and the local classification boundary have a significantly smaller effect on the empirical risk minimization problem.

Learning with Big Data: Since the empirical risk formulation is convex, established approaches to optimizing over large training sets can be applied. In particular, the convex formulation can be trained directly using stochastic gradient descent techniques, allowing training using streaming observations and batch processing approaches that still converge to a global minimum Shalev-Shwartz (2012); Zinkevich (2003). We demonstrate the ability to train in an online fashion in Section 5.

3. L3M for Multiple Regions and Multiclass Data

Multiple Regions: A natural extension of the 2-region L3M is to partition the feature space into multiple ($r > 2$) regions using the structure shown in Fig. 1. In this structure, G is a multiclass partitioning function that partitions the space into r regions, with the associated local classifiers, $f_1, \ldots, f_r$, applied independently in each region.

The key observations from the 2-region case can be applied to the case of multiple regions, allowing the empirical risk to be reformulated as a maximization over sums of indicator functions:

Theorem 3.1 *For a classifier of the form shown in Fig. 1, the empirical risk can be expressed:*

$$R(G, f_1, \ldots, f_r) = \sum_{i=1}^{n} \max_{k \in \{1,\ldots,r\}} \left[\mathbb{1}_{f_k(x_i) \neq y_i} + \mathbb{1}_{G(x_i) = k} - 1 \right] . \tag{6}$$

Replacing the indicator functions with upper-bounding convex surrogates yields a globally convex, upper-bounding surrogate function on the empirical risk.

The proof of Thm. 3.1 follows closely from the proof of Thm. 2.3 and is included in the supplementary material.

As in the case of 2 regions, the empirical loss proposed in Thm. 3.1 is a maximization over sums of indicator functions, and as a result, preserves convexity of the global objective when the indicator functions are replaced with convex surrogate functions. In the case of more than 2 regions, the partitioning function $G(x)$ can be viewed as a multiclass classification function, with an appropriate convex upper-bounding surrogate function required for the function $\mathbb{1}_{G(x_i)=k}$. To handle this multiclass problem, we define the partitioning function as a one-vs-all maximum margin approach, with the partitioning function defined $G(x) = \text{argmax}_{k\in\{1,\ldots,r\}}\, g_k(x)$. While multiple alternative multiclass coding schemes and surrogate functions are valid, such as the multicategory SVM Lee et al. (2004) or simplex SVM Mroueh et al. (2012) approaches, for technical simplicity we implement the one-vs-all scheme in constructing L3M's with multiple regions.

For a partitioning function of this form, we upper bound the indicator function $\mathbb{1}_{G(x_i)=k} \leq \phi(G, k, x_i)$, where the function $\phi(G, k, x_i)$ is defined:

$$\phi(G, k, x_i) = \max\left[(1 + g_k(x_i))_+\, , \max_{j\neq k}\,(1 - g_j(x_i))_+\right] . \tag{7}$$

This is equal to the maximum hinge-loss over the one-vs-all classifiers. For this surrogate function, the only case where $\phi(G, k, x_i)$ is equal to zero is the case where the k^{th} classifier has a large negative margin ($g_k(x_i) < -1$) and all other classifiers have a large positive margin ($g_j(x_i) > 1$, $\forall j \neq k$).

Using the surrogate function ϕ, we construct an upper-bounding convex function for the empirical risk:

$$R(G, f_1, \ldots, f_r) = \sum_{i=1}^{n} \max_{k\in\{1,\ldots,r\}} \left[(1 - y_i f_k(x_i))_+ + \phi(G, k, x_i) - 1\right] . \tag{8}$$

This surrogate function is convex and can in fact be expressed as a maximization of a linear function over a set of linear inequality constraints. In practice, quadratic regularization constraints are added on the functions $f_1, \ldots, f_r$ based on the maximum margin principle. Note that the result from Prop. 2.4 can be generalized to the multiclass case using tight multiclass convex surrogates such as the simplex coding SVM Mroueh et al. (2012).

The symmetry issue noted in Prop. 1.1 arises in the case of multiple regions. To overcome this issue, we assign $r - 1$ randomly selected observations to different regions and enforce positive margins for these points within these regions. Comparing the empirical error between multiple random assignments allows for verification of poorly selected constraints. In practice, few random assignments are necessary to find a suitable solution as r is small (see Experimental section for more details).

Multiclass Classification: The convex formulations proposed in Eqns. (4) and (8) can also be naturally extended to multiclass data. In practice, we use the same maximum margin one-vs-all scheme as used in the partitioning function to define the functions $f_1, \ldots, f_r$. In order to upper bound the indicator function, the binary hinge losses in Eqns. (4) and (8) associated with the local classifiers is replace with a multiclass hinge loss similar to the one proposed in Eqn. (7). As in the multi-region partitioning case, multiple alternative multiclass coding schemes and surrogate functions can be substituted in place of the proposed one-vs-all scheme.

4. Properties of L3M

Generalization Error: One important consideration in constructing L3M's is choosing the parameter r, which dictates the number of linear functions used to approximate the decision boundary. Increasing the number of partitioned regions (r) allows for the empirical error to be made small by using many local linear classifiers. Conversely, the variance error introduced by complex classifiers can be controlled by limiting the number of partitioned regions. In this sense, the parameter r can be viewed as a tradeoff parameter between bias and variance error, with behavior in the binary case characterized by the VC-dimension:

Theorem 4.1 *The VC-dimension of a local linear classifier with r regions can be bounded:*

$$2(\frac{(r-1)^2+2}{2})\log\left(e(\frac{(r-1)^2+2}{2})\right)(d+1).$$

The proof of this theorem is based on decomposing the classifier into a boolean function of binary classifiers Sontag (1998) (see Supplementary for details).

The VC-dimension of L3M's grows linearly with dimension and polynomially with the number of partitions. In practice, few regions ($r << d$) are necessary to sufficiently reduce empirical error, implying that complex non-linear boundaries are often well approximated with piecewise linear functions. Additionally, the VC-dimension yields a direct approach to finding the number of partitions by choosing the parameter r to minimize a high-probability bound on the generalization error.

Test Time Computational Efficiency: Test time computational efficiency is a major advantage of L3M. In the case of binary labels, the cost of predicting a test label scales linearly with the dimension of the data, as label estimation requires $O(dr + d)$ computations, where d is the dimension of the data and r is the number of regions.

Significant test time computational savings occur in the multiclass setting as label predictions can be accomplished in $O(dr + dc)$ computations, where c is the number of classes. In comparison, prediction using a one-vs-all kernel SVM scales $O(dsc)$, where s is the sparsity of each one-vs-all SVM. Similarly, in the case of one-vs-all AdaBoost, prediction requires $O(Nc)$ computations, where N is the number of weak learners, which is typically significantly larger than d. Of note, the computational cost of partitioning the space ($O(dr)$) scales independently of the number of classes, whereas approximating the decision boundary more accurately in the standard one-vs-all approach scales linearly with respect to the number of classes. Experimental results validate that L3M's allow accurate approximations of highly complex decision boundaries while still maintaining low computational cost.

5. Online Training of L3M's

Algorithm 1 Online Update

```
Input: Observation and label, $x_t, y_t$, current partitioning classifier, $\alpha$, and local classifiers $\beta_1, \beta_2$

Output: Updated partitioning classifier, $\alpha$, updated local classifiers $\beta_1, \beta_2$
1. Find active region
    $$r_t = \begin{cases} 1 & \text{if } \log(1 + e^{\alpha^T x_t}) + \log(1 + e^{-y_t \beta_1^T x_t}) > \\ & \quad \log(1 + e^{-\alpha^T x_t}) + \log(1 + e^{-y_t \beta_2^T x_t}) \\ 2 & \text{otherwise} \end{cases}$$
2. Calculate the subgradient for the partitioning classification functions:
    $$\bigtriangledown\alpha = \begin{cases} \frac{-x_t}{1+e^{-\alpha^T x_t}} & \text{if } r = 1 \\ \frac{x_t}{1+e^{\alpha^T x_t}} & \text{if } r = 2 \end{cases}, \quad \bigtriangledown\beta_1 = \begin{cases} \frac{-y_t x_t}{1+e^{y_t \beta_1^T x_t}} & \text{if } r = 1 \\ 0 & \text{if } r = 2 \end{cases}, \quad \bigtriangledown\beta_2 = \begin{cases} 0 & \text{if } r = 1 \\ \frac{-y_t x_t}{1+e^{y_t \beta_2^T x_t}} & \text{if } r = 2 \end{cases}$$
3. Return updated functions:
    $$\alpha = \alpha - \frac{\bigtriangledown\alpha}{\sqrt{t}}, \quad \beta_1 = \beta_1 - \frac{\bigtriangledown\beta_1}{\sqrt{t}}, \quad \beta_2 = \beta_2 - \frac{\bigtriangledown\beta_2}{\sqrt{t}}$$
```

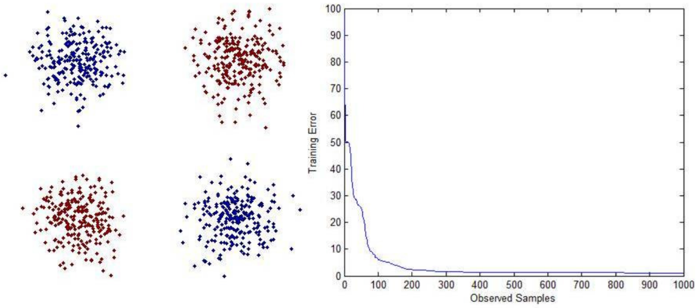

Figure 4: **Left: Synthetic gaussian XOR data. Right: Average training error over the entire training set vs. observed training observations.**

To demonstrate online training of L3M's, we upper bound the indicator losses in (3) using logistic loss functions. The logistic loss function is an ideal choice when training local linear classifiers using streaming data, as it is smooth continuously differentiable while asymptotically approximating the tightest convex surrogate functions (hinge losses as shown in Lee et al. (2004)). Although hinge losses produce a tighter convex surrogate, we find that training in an online setting converges noticeably faster when using smooth loss functions. Starting with a random set of functions, we use a stochastic subgradient descent algorithm shown in Alg. 1 to find the local linear classifier that minimizes the objective function Zinkevich (2003). Using a descent rate of $t^{\frac{-1}{2}}$, the average regret between the stochastic subgradient descent solution and the global optimal solution has been shown to approach zero Zinkevich (2003).

Performance of this online algorithm is shown on a synthetic dataset in Fig. 4. The synthetic dataset, shown in Fig. 4, was generated from a mixture of Gaussians, with a single gaussian distribution centered in each quadrant and labels corresponding to each Gaussian equal to the XOR of the mean coordinates. A randomly initialized local linear classifier is updated by randomly generated training examples. The average training error on the entire training dataset is shown on the right of Fig. 4. On the Gaussian XOR data set, the local linear classifier converges at an extremely fast rate, with convergence approximately after 200 updates.

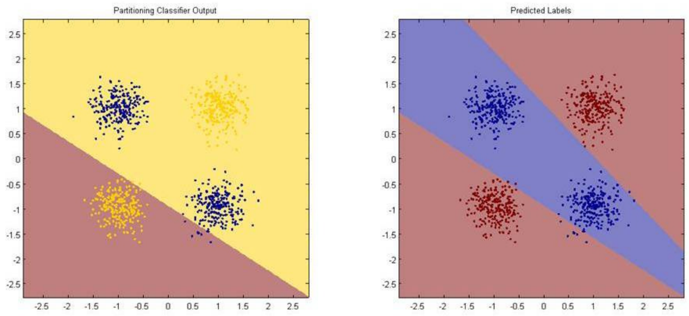

Figure 5: **Left: Partitioned regions learned via online training. Right: Decision boundaries learned by online training.**

6. Experimental Results

6.1. Multiclass Classification Performance

Experimental results are reported in Table 2 for seven benchmark datasets from the Statlog Project Michie et al. (1994) and UCI repository Frank and Asuncion (2010)[1]. These datasets have been previously experimented on to demonstrate multiclass performance Wang and Saligrama (2012); Mroueh et al. (2012); Hsu and Lin (2002).

Dataset	Dimension	Classes	Training Set	Test Set
Banana	2	2	400	4900
DNA	180	3	2000	1186
Landsat	36	7	4435	2000
Vowel	10	11	528	462
Optdigit	64	10	3823	1797
Pendigit	16	10	7494	3498
Image Seg.	19	7	210	2100

Table 1: Multiclass dataset properties. Benchmark training and test splits are used.

The L3M's were constrained to 6 regions ($r = 6$) for all examples, and for each dataset, a sweep was performed over the parameters $\lambda \in \left[10^2, 10^3, 10^4, 10^5, 10^6\right]$ and $\beta \in \left[10^2, 10^3, 10^4, 10^5, 10^6\right]$,

[1] Note that confidence intervals are not possible with the results, as the predefined training and test splits were used. Although fixed training and test splits are used, test set error bounds Langford (2006) show that with high probability the difference between true error and empirical error is small.

with 30 random sets of constraints tested for each parameter pair. The resulting quadratic program was solved using the CVX convex optimization package CVX Research (2012). The pair of parameters and linear constraints producing the smallest empirical error on the training set were used to select the final classifier. Indeed it is easy to show using basic probability that if there exists a good partition with similar number of data points in each region then for a 6 region split about 30 random initializations ensures with 95% confidence that we get the right one.

For comparison, we implemented a variety of non-linear supervised learning methods. A one-vs-all AdaBoost classifier using stumps as weak learners was constructed, with weak learners added until the training error rate ceased to improve Freund and Schapire (1997) . The AdaBoost training error was generally small, with four of the six datasets having zero training error, and the remaining datasets having training errors bounded by 3%, implying that weak learnability issues do not arise on these datasets when using stumps. Decision trees were trained using the Gini Diversity Index (GDI) as a splitting criteria, with optimal pruning performed and a minimum of 5 training examples in each leaf of the tree L. Breiman and Stone (1984). MDA, a generative local linear approach, was trained using 6 Gaussian clusters to represent each class, producing a decision boundary of equal complexity to L3M Hastie and Tibshirani (1996). A Gaussian RBF SVM was trained for each dataset, using the heuristic of setting σ equal to the median of the pairwise distances between distinct points, and the regularization parameter chosen using 4-fold cross-validation over a logarithmic sweep $\lambda \in \left[10^{-3}, 10^{4}\right]$. The resulting error rates are comparable to previously reported error rates Mroueh et al. (2012).

Table 2: Multiclass learning algorithm test errors on Statlog and UCI datasets using benchmark training and test sets.

Algorithm	Banana	DNA	Landsat	Vowel	Optdigit	Pendigit	Image Segmentation
One vs All Linear SVM	39.55%	7.08%	17.90%	59.09%	7.63%	10.92%	8.24%
One vs All RBF SVM	11.86%	5.48%	9.70%	37.23%	2.34%	1.86%	11.30%
One vs All AdaBoost	32.98%	8.35%	16.10%	69.70%	12.24%	11.29%	10.38%
GDI Tree	14.33%	9.36%	14.45%	56.93%	14.58%	8.78%	9.71%
MDA	20.45%	12.14%	36.45%	67.32%	9.79%	7.75%	15.43%
L3M	11.84%	5.31%	17.50%	40.69%	7.12%	10.52%	10.76%

As shown in Table 2, L3M generally outperforms AdaBoost, MDA, and GDI decision trees and is only moderately outperformed by Gaussian RBF SVM. While Gaussian RBF SVM generally outperforms L3M, L3M has multiple computational advantages over RBF SVM. Computationally, the quadratic program for L3M scales in the same fashion as a standard linear SVM. Additionally, L3M's can be learned using streaming training data without the need to store the full set of training data in memory, whereas the RBF kernel cannot even be formed without storing the entire training set. Finally, during test time, the computational cost of evaluating a L3M is extremely small, whereas evaluating the the Gaussian RBF kernel can be computationally expensive, especially in the case of multiclass data.

6.2. Test Time Cost Comparison

An important aspect of handling large sets of data is computational cost for predicting labels. To compare the test time computational cost, error rates were computed for classifiers of varying test time computational cost, as shown in Fig. 6.

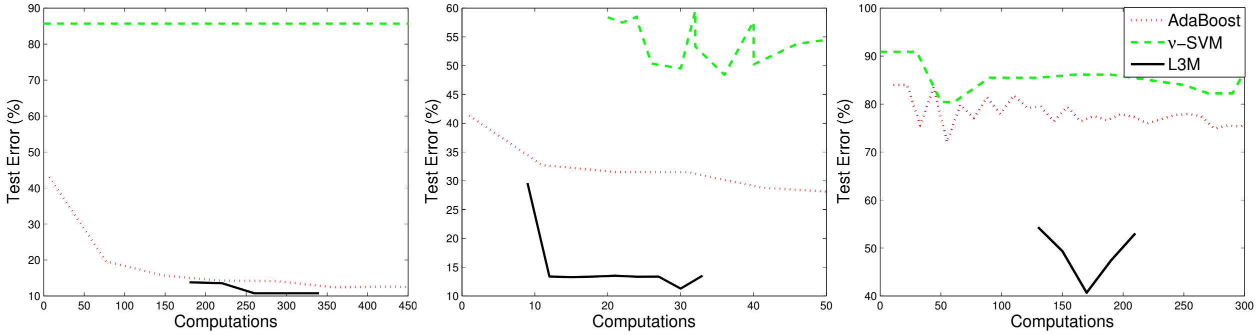

Figure 6: Test error vs. number of test time computations. The range of test time computational costs for L3M is limited even for complex decision boundaries due to linear scaling with respect to number of regions. We attribute V-shaped curve for Vowel dataset to overtraining. **Left:** Image Segmentation data, **Middle:** Banana data, and **Right:** Vowel data.

For comparison, we compare performance with one vs. all AdaBoost, and one vs. all RBF kernel ν-SVM Schölkopf et al. (2000). To construct AdaBoost classifiers under different test time computation constraints, the number of stumps used to construct each binary classifier were limited. To control the test time cost of the kernel ν-SVM, the parameter ν was varied to construct classifiers with varying support sizes. ν-SVM was chosen due to the ability to the direct trade-off between empirical error and classifier sparsity through the parameter ν. In general, non-additive kernel SVM's perform poorly when limited to small test time costs, as observed by the instability of low test time cost ν-SVM. This is due to the highly sparse kernel classifiers constructed, which change dramatically when individual support vectors are added. Also, note that in the Image Segmentation data, ν-SVM cannot be driven sparse enough to produce an output apart from a constant label. L3M's were constructed in the same manner as in the multiclass experimental results, with test time cost controlled by varying the number of partitioned regions $r \in \{2, \ldots, 16\}$. Due to the slow growth of test time cost, a significantly smaller range of test time costs are possible with L3M, as the cost scales linearly with r and independently of the number of classes. In the case of the Image Segmentation dataset, L3M appears to overtrain even for extremely low computational costs ($r > 6$), as the test error increases whereas the training error did not increase when adding more regions.

For a fixed test time computational cost, L3M outperforms both AdaBoost and kernel ν-SVM by a sizable margin. L3M offers performance comparable or better than significantly more computationally expensive approaches. Furthermore, of these methods, L3M is the only approach that can naturally be trained in an online fashion, providing computational savings both in training and test time compared to alternative approaches.

Acknowledgments

This research was partially supported by NSF Grants 0932114 and 1330008.

References

Support vector machines experts for time series forecasting. *Neurocomputing*, 51(0).

Kristin P. Bennett and O. L. Mangasarian. Bilinear separation of two sets in n-space. *Computational Optimization and Applications*, 2, 1993.

Kristin P Bennett, Nello Cristianini, John Shawe-Taylor, and Donghui Wu. Enlarging the margins in perceptron decision trees. *Machine Learning*, 41(3):295–313, 2000.

Avrim L Blum and Ronald L Rivest. Training a 3-node neural network is np-complete. *Neural Networks*, 5(1):117–127, 1992.

Carla E Brodley and Paul E Utgoff. Multivariate decision trees. *Machine learning*, 19(1):45–77, 1995.

Inc. CVX Research. CVX: Matlab software for disciplined convex programming, version 2.0. http://cvxr.com/cvx, August 2012.

Juan Dai, Shuicheng Yan, Xiaoou Tang, and James T. Kwok. Locally adaptive classification piloted by uncertainty. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, pages 225–232, New York, NY, USA, 2006. ACM. ISBN 1-59593-383-2. doi: 10.1145/1143844.1143873. URL http://doi.acm.org/10.1145/1143844.1143873.

Ofer Dekel and Ohad Shamir. There's a hole in my data space: Piecewise predictors for heterogeneous learning problems. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*.

A. Frank and A. Asuncion. UCI machine learning repository, 2010. URL http://archive.ics.uci.edu/ml.

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119 – 139, 1997. ISSN 0022-0000. doi: 10.1006/jcss.1997.1504.

Yuhong Guo and Dale Schuurmans. Convex relaxations of latent variable training. *Advances in Neural Information Processing Systems*, 20:601–608, 2008.

Trevor Hastie and Robert Tibshirani. Discriminant analysis by gaussian mixtures. *Journal of the Royal Statistical Society, Series B*, 58:155–176, 1996.

Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *Neural Networks, IEEE Transactions on*, 13(2):415–425, 2002.

R. A. Olshen L. Breiman, J. H. Friedman and C. J. Stone. *Classification and Regression Trees.* Wadsworth, 1984. ISBN 0-534-98053-8.

J. Langford. Tutorial on practical prediction theory for classification. *Journal of Machine Learning Research*, 6(1):273, 2006.

Y. Lee, Y. Lin, and G. Wahba. Multicategory support vector machines. *Journal of the American Statistical Association*, 99(465):67–81, 2004.

Clodoaldo A.M. Lima, Andre L.V. Coelho, and Fernando J. Von Zuben. Hybridizing mixtures of experts with support vector machines: Investigation into nonlinear dynamic systems identification. *Information Sciences*, 177(10):2049 – 2074, 2007. ISSN 0020-0255. doi: 10.1016/j.ins.2007.01.009. URL http://www.sciencedirect.com/science/article/pii/S0020025507000382.

Nimrod Megiddo. On the complexity of polyhedral separability. *Discrete & Computational Geometry*, 3(1).

Donald Michie, D. J. Spiegelhalter, C. C. Taylor, and John Campbell, editors. *Machine learning, neural and statistical classification.* Ellis Horwood, Upper Saddle River, NJ, USA, 1994. ISBN 0-13-106360-X.

Y. Mroueh, T. Poggio, L. Rosasco, and J.-J. Slotine. Multiclass Learning with Simplex Coding. *ArXiv e-prints*, September 2012.

Bernhard Schölkopf, Alex J Smola, Robert C Williamson, and Peter L Bartlett. New support vector algorithms. *Neural computation*, 12(5):1207–1245, 2000.

Shai Shalev-Shwartz. Online learning and online convex optimization. *Found. Trends Mach. Learn.*, 4(2):107–194, February 2012. ISSN 1935-8237. doi: 10.1561/2200000018. URL http://dx.doi.org/10.1561/2200000018.

Eduardo D. Sontag. Vc dimension of neural networks. In *Neural Networks and Machine Learning*, 1998.

Marc Toussaint and Sethu Vijayakumar. Learning discontinuities with products-of-sigmoids for switching between local models. In *Proceedings of the 22nd international conference on Machine Learning*, pages 904–911. ACM Press, 2005.

Joseph Wang and Venkatesh Saligrama. Local supervised learning through space partitioning. In *Advances in Neural Information Processing Systems 25*. 2012.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, pages 928–936, 2003.

Appendix A. Proof of Theorem 2.4

As previously shown in [17], the tightest convex surrogate for the indicator function is the hinge loss in the sense that any convex upper-bounding function can be lower bounded by a scaled indicator function. Given that taking sums and maxima preserves tightness of convexity, replacing the indicators in the function:

$$\max\left[\mathbb{1}_{a\geq 0}+\mathbb{1}_{b\leq 0},\mathbb{1}_{c\leq 0}+\mathbb{1}_{d\leq 0}\right]-1$$

with hinge losses produces the tightest convex surrogate in the previously described sense.

Appendix B. Proof of Theorem 3.1

As in the 2 region case, the empirical risk can be formulated with respect to the event of correct classification:

$$R(G, f_1, \ldots, f_r) = \sum_{i=1}^{n} \left[1 - \sum_{k=1}^{r} \mathbb{1}_{G(x_i)=k} \mathbb{1}_{f_k(x_i)=y_i} \right] .$$

As in the two region case, the product of indicators can be expressed as a maximum of the two indicators:

$$\begin{aligned} R(G, f_1, \ldots, f_r) =& \sum_{i=1}^{n} \max_{\lambda_i^1, \ldots, \lambda_i^r \in [0,1]} \left[1 - \sum_{k=1}^{r} \left[\lambda_i^k \mathbb{1}_{G(x_i)=k} + (1 - \lambda_i^k) \mathbb{1}_{f_k(x_i)=y_i} \right] \right] \\ =& \sum_{i=1}^{n} \left[\max_{\lambda_i^1, \ldots, \lambda_i^r \in [0,1]} \sum_{k=1}^{r} \left(\lambda_i^k \mathbb{1}_{G(x_i)\neq k} + (1 - \lambda_i^k) \mathbb{1}_{f_k(x_i)\neq y_i} \right) - (r-1) \right] . \end{aligned}$$

The variables $\lambda_i^1, \ldots, \lambda_i^r$ do not necessarily have unique solutions, however there always exists an optimal solution such that $\lambda_i^1 + \lambda_i^2 + \ldots + \lambda_i^r = r - 1$. By enforcing this constraint, the variables $\lambda_i^1, \ldots, \lambda_i^r$ can be removed:

$$R(G, f_1, \ldots, f_r) = \sum_{i=1}^{n} \max_{k \in \{1 \ldots, r\}} \left[\mathbb{1}_{f_k(x_i)\neq y_i} + \sum_{j \neq k} \mathbb{1}_{G(x_i)\neq j} - (r-1) \right] .$$

Note that the term $\sum_{j \neq k} \mathbb{1}_{G(x_i)\neq k} - (r-1)$ can be replaced with the equivalent expression $\mathbb{1}_{G(x_i)=k} - 1$, yielding the empirical risk as expressed in (6).

Appendix C. Proof of Theorem 4.1

The L3M is composed of the partitioning classifier, G, and the local classifiers, $f_1, f_2, \ldots, f_r$. The maximum margin rejection classifier G can be viewed as a boolean function of $\frac{(r-1)^2}{2}$ linear functions, and each of the local classifiers is a linear function. Therefore, the output of the L3M can be viewed as a boolean function of $\frac{(r-1)^2}{2} + 1$ functions, each with a VC-dimension of $d + 1$, where d is the observation dimension. From Lemma 2 of [19], the VC-dimension of the L3M can be bounded:

$$VC\,(F) \leq 2(\frac{(r-1)^2}{2} + 1) \log(e(\frac{(r-1)^2}{2} + 1))(d + 1). \qquad (9)$$