Journal of Machine Learning Research 19 (2018) 1-30 Submitted 7/17; Revised 9/18; Published 10/18

Hinge-Minimax Learner for the Ensemble of Hyperplanes

Dolev Raviv DOLEV.RAVIV@GMAIL.COM
Department of Computer Science
University of haifa
Haifa, 31905, Israel

Tamir Hazan TAMIR.HAZAN@TECHNION.AC.IL
Faculty of Industrial Engineering and Management
Technion - Israel Institute of Technology
Haifa, 32000, Israel

Margarita Osadchy RITA@CS.HAIFA.AC.IL
Department of Computer Science
University of haifa
Haifa, 31905, Israel

Editor: David Sontag

Abstract

In this work we consider non-linear classifiers that comprise intersections of hyperplanes. We learn these classifiers by minimizing the "minimax" bound over the negative training examples and the hinge type loss of the positive training examples. These classifiers fit typical real-life datasets that consist of a small number of positive data points and a large number of negative data points. Such an approach is computationally appealing since the majority of training examples (belonging to the negative class) are represented by the statistics of their distribution, which is used in a single constraint on the empirical risk, as opposed to SVM, in which the number of variables is equal to the size of the training set. We first focus on intersection of K hyperplanes, for which we provide empirical risk bounds. We show that these bounds are dimensionally independent and decay as $K/\sqrt{m}$ for m samples. We then extend the K-hyperplane mixed risk to the latent mixed risk for training a union of C K-hyperplane models, which can form an arbitrary complex, piecewise linear boundaries. We propose efficient algorithms for training the proposed models. Finally, we show how to combine hinge-minimax training with deep architectures and extend it to multi-class settings using transfer learning. The empirical evaluation of the proposed models shows their advantage over the existing methods in a small training labeled data regime.

Keywords: Minimiax, Imbalanced Classification, Intersection of K Hyperplanes, Transfer Learning

1. Introduction

Many real-life binary classification problems involve imbalanced classes, for example object detection in vision and fraud detection in security. In such problems it is easy to collect background data (the negative class), while data representing the target class (the positive class) is rare or hard (expensive) to obtain. The majority of existing classifiers (e.g., SVM, Neural Networks, including deep ones) assume balanced training sets and when trained on imbalanced sets show degraded

classification performance or require a long and tedious bootstrapping process of mining negative examples (e.g. Malisiewicz et al. (2011); Girshick et al. (2014)) out of millions.

When there are (infinitely) many training examples, instead of minimizing the average sample loss, it is more computationally appealing to apply minimax setting (Lanckriet et al. (2003); Honorio and Jaakkola (2014)), which upper bounds the expected risk of a classifier assuming only the knowledge of mean and covariance of the data distribution. The "minimax" bound provides an upper bound for *every* distribution with a given mean and covariance. Applying the minimax learning (Lanckriet et al. (2003)) to the negative class (the majority class) allows to avoid bootstrapping procedure and makes learning more efficient, as it replaces loss evaluation on all negative samples with a single "minimax" bound.

Due to the assumption that the positive class is rare, we cannot apply the minimax learning to the positive class, as it completely relies on the mean and covariance of the data. Estimating the covariance matrix in high-dimensional space from a small number of positive training samples is problematic. Alternatively, we can use the hinge loss (Vapnik (2000); Zhang (2002); Bartlett and Mendelson (2003); Bousquet et al. (2004); Kakade et al. (2008)) for the positive class as it is computationally appealing when there are fairly small number of training samples.

We suggest to combine the hinge-like loss for the positive samples with the "minimax" bound applied to the statistics of the negative samples to enjoy the best of both worlds and we call these classifiers *Hinge-Minimax* classifiers.

The idea of combining "minimax" bound for the negative class and svm-like formulation for the positive samples was introduced in Osadchy et al. (2012, 2016) for computing linear and kernel classifiers. Kernel classifiers could be quite slow, as they require evaluating kernel on many support vectors. In this work we focus on more efficient non-linear classifiers – ensembles of hyperplanes. We first consider an intersection of hyperplanes and then extended it to more general ensembles of hyperplanes.

Previous algorithms for intersection of hyperplanes are computationally costly when considering large sets of negative data points (Klivans and Sherstov (2009); Daniely et al. (2014)). To deal with this computational difficulty we use the mixed risk. Namely, we extend the "minimax" bound to deal with intersection of hyperplanes over (infinitely many) negative examples. For the positive samples, we define a K-hyperplane hinge loss. We derive an empirical mixed-risk bound, that uses the Rademacher complexities to bound the risk of the positive class and vector Bernstein's inequalities to bound the risk associated with the negative class. Note that we treat the positive and negative samples differently because of the computational gain such separation provides.

Recently, Honorio and Jaakkola (2014) derived a generalization bound for the minimax setting using PAC-Bayesian approach, which bounds the expected loss with respect to a posterior distribution over all possible classifiers. Our work differs as we use stronger assumptions - that the norm of the data points is bounded by a constant, an assumption that is natural in many applications.[1] Thus we are able to avoid the PAC-Bayesian approach that considers generalization bounds over randomized predictors.

Intersection of positive half-spaces is a convex set. We generalize the mixed risk for a non-convex classifier. We learn an ensemble of K-hyperplane models, that can form arbitrary, piece-wise linear boundaries. We propose a training algorithm that minimizes this risk by simultaneously discovering the convex components in the positive class and building K-hyperplane models to separate

1. Input normalization is a standard procedure, applied for faster learning.

each component from the negative class. The learning is done by alternating between finding the best partition of the data into hidden components and updating the model over this partition. We call this novel classifier the *Latent Hinge Minimax* (LHM) classifier, as it discovers the latent structure in the data and employs the Hinge-Minimax paradigm.

We show that the LHM model has an equivalent Neural Networks (NN) architecture. This allows us 1) to use deep learning features via transfer learning and 2) to extend the proposed model to the multi-class setting. For the multi-class problems, we build one-against-all classifiers for all classes and combine them in a single model by mapping class specific LHM models to a multi-class NN with a matching architecture. We then use the cross-entropy loss to adjust the weights in the resulting *LHM-NN* combination.

We show that using LHM-NN in the transfer learning settings has significant benefits compared to NN (standard settings), in both classification accuracy and training efficiency. The improved accuracy stems from the ability of LHM model to learn from unlabeled data. The fast convergence of the LHM-NN (just a handful of epochs) is due to a very good initialization of the upper layers with class specific LHM classifiers. Note that class specific LHM models can be trained in parallel. Moreover, adding a new class to LHM-NN is fast and easy: train a classifier for the new class, map it to the corresponding LHM-NN architecture and run a very fast fine-tuning. Similarly to Kuzborskij et al. (2013), which considered the transfer learning for the $n+1$ category from a fully trained n-category classifier, we use only a handful of training samples for tuning it. In contrast to Kuzborskij et al. (2013), we do not restrict the new classifier to belong to the span of the previously learned n classifiers. This allows us greater flexibility in adding a new, non-related class to the multi-class model.

We performed empirical evaluation of the proposed models: the K-hyperplane, the LHM, and the multi-class models. In all cases, the proposed models outperformed their counterparts in small and imbalanced training data regime.

The rest of the paper is organized as follows. Section 2 introduces the K hyperplane Hinge-Minimax classifier (KHHM). Specifically, we extend the "minimax" bound for the intersection of K positive half-spaces in Section 2.1.1. Then in Section 2.2, we propose a novel training algorithm for the KHHM classifier. We prove the uniform generalization bounds for the KHHM model in Section 2.3. Section 3 focuses on a non-convex generalization of the KHHM model – LHM classifier. We introduce the latent mixed risk in Section 3.1, the training algorithm in Section 3.2, and we prove a uniform generalization bound for the LHM classifier with a fixed assignment of the positive labeled training set in Section 3.3. Section 4 discusses the mapping of the LHM classifier to a neural network. Section 5 reports the experiments with KHHM, LHM, and LHM-NN classifiers. Section 6 discusses the efficiency of the proposed models and Section 7 concludes the paper.

2. K-Hyperplane Hinge-Minimax Classifier

In the following, for simplicity we assume that for a linear classifier which predicts $y = sign(w^T x)$, $b = 0$ (or absorbed by w).

Definition 1 *Let w_i, $i = 1, .., K$ denote K hyperplanes. Let W be a matrix with w_i as its ith column. We define K-hyperplane classifier $f_W(x)$ as an intersection of these K half-spaces:*

$$f_W(x) = \begin{cases} 1 & \text{if } W^\top x \geq \vec{0} \\ -1 & \text{otherwise} \end{cases} \tag{1}$$

where $\vec{0}$ denotes a vector of zeros.

2.1. Mixed Risk for K-Hyperplane Model

We are interested in a classification problem in which the positive class corresponds to a single concept and the negative class is its complement and we refer to it as a *background*. We assume that the sample of the positive class is relatively small while the negative sample is very large (it can be represented by an unlabeled data as well, thus is easy to collect). Due to the specifics of the problem we propose a mixed risk for the non-linear classifier in eq. 1. We first define its parts in Definitions 2 and 3 and then define the mixed risk in Definition 4.

Definition 2 *Let $(x, y) \sim D$ be a joint distribution of samples $x \in \mathbb{R}^n$ and labels $y \in \{-1, 1\}$. We define the* hinge risk *of $f_W(x)$ as follows,*

$$L_D^H(W) = \mathbb{E}_D \left[\ell(W, x, y)\mathbb{1}[y = 1] + 0 \cdot \mathbb{1}[y = -1]\right] \tag{2}$$

where $\ell(W, x, y) = \max_{j\in\{1,\dots,K\}}\{\max\{0, 1 - yw_j^\top x\}\}$.

Definition 3 *Under the assumptions of Definition 2, let D_{neg} be a marginal distribution of samples from a ball of radius C over the negative labels with mean μ and covariance matrix Σ. Let $\Omega(\mu, \Sigma)$ be a family of all distributions with mean μ and covariance matrix Σ. We assume that $D_{neg} \in \Omega(\mu, \Sigma)$.*

We define the background risk[2] *of $f_W(x)$ as follows,*

$$L_{\mu,\Sigma}^B(W) = \left[\sup_{Z\in\Omega(\mu,\Sigma)} Pr_{z\sim Z}(W^\top z \geq \vec{0})\right] \mathbb{1}[y = -1] + 0 \cdot \mathbb{1}[y = 1] \tag{3}$$

We derive the expression for the background risk in the next section.

According to the Definitions 2 and 3, the hinge risk is defined over the samples of the positive class only and the background risk is defined over the distribution of the negative class only. Thus we can sum the two to form the mixed risk over D.

Definition 4 *Under the assumptions of Definitions 2 and 3, we define the mixed risk for the K-hyperplane classifier as:*

$$L_D^{HB}(W) = L_D^H(W) + L_{\mu,\Sigma}^B(W), \tag{4}$$

2.1.1. THE EXPECTED RISK OF THE NEGATIVE CLASS

The extension of Theorem 3.1 from Marshall and Olkin (1960) to a nonzero mean variable (as shown in Marshall and Olkin (1960) Eq. 7.7–7.8) states that for a random vector x with mean M and covariance Γ,

$$\sup_{x\sim(M,\Gamma)} \Pr(x \in S) = \frac{1}{1 + d^2},$$

with $d^2 = \inf_{x\in S}(x - M)^\top \Gamma^{-1}(x - M)$, where S is a given convex set.

The intersection of K hyperplanes is a convex set, thus we can bound the probability of a negative sample falling into the intersection of K hyperplanes using the above result and derive the expression for $L_{\mu,\Sigma}^B(W)$ as shown below in Theorem 1.

2. the name "background" is chosen to emphasize the fact that the negative class is the majority class, while the positive class is rare.

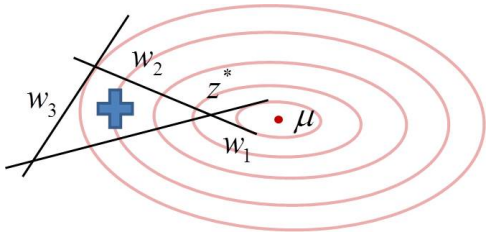

Figure 1: A 2D illustrative example of Theorem 1. z^* is the closest point to the mean of the negative distribution.

Theorem 1 *For any finite number of hyperplanes w_j,*

$$\sup_{Z \in \Omega(\mu,\Sigma)} Pr_{z\sim Z}(W^\top z \geq \vec{0}) = \frac{1}{1+d^2}$$

with $d^2 = \mu^\top U(U^\top \Sigma U)^{-1} U^\top \mu$, where U is a subset of columns of W that satisfy $w^\top z^* = 0$, where $z^* = \arg\min_z (z-\mu)^\top \Sigma^{-1}(z-\mu)$.

Before we proceed with the proof, let us consider the following 2D toy example to gain some intuition into Theorem 1 (see Figure 2.1.1). Assume that the positive class lies inside an intersection of three hyperplanes $W = [w_1, w_2, w_3]$ and the negative class is described by the normal distribution with mean μ and covariance Σ. d^2 is a square distance between the mean of the negative distribution and the point on the boundary of the positive region of the classifier, which is closest to μ. In the example, depicted in Figure 2.1.1, the closest point is denoted by z^* and it's an intersection of w_1 and w_2, thus $U = [w_1, w_2]$.

Proof Let $z \sim D_{neg} \in \Omega(\mu, \Sigma)$ be a sample from the negative class. $W^\top z \geq \vec{0}$ defines a convex set, thus we can apply the result due to Marshall and Olkin (1960) to obtain:

$$\sup_{Z \in \Omega(\mu,\Sigma)} Pr_{z\sim Z}(W^\top z \geq \vec{0}) = \frac{1}{1+d^2},$$

with $d^2 = \inf_{W^\top z \geq \vec{0}} (z-\mu)^\top \Sigma^{-1}(z-\mu)$, where the supremum is taken over all distributions in $\Omega(\mu, \Sigma)$.

Next, we want to derive a closed-form expression for d^2. We seek the solution for the primal problem

$$\min_z (z-\mu)^\top \Sigma^{-1}(z-\mu)$$

s.t. $w_i^\top z \geq 0$ for $i = 1, .., K$. We construct the Lagrangian:

$$L(z, \lambda_i) = (z-\mu)^\top \Sigma^{-1}(z-\mu) + \sum_i \lambda_i w_i^\top z, \quad \lambda_i \geq 0.$$

The optimality condition:

$$\frac{\partial L}{\partial z} = 2\Sigma^{-1} z - 2\Sigma^{-1}\mu + \sum_i \lambda_i w_i = 0,$$

gives us $z^* = \mu - \frac{1}{2}\sum_i \lambda_i \Sigma w_i$. The Lagrange dual function is as follows,

$$L(z^*, \lambda) = \left(\frac{1}{2}\sum_i \lambda_i \Sigma w_i\right)^\top \Sigma^{-1} \left(\frac{1}{2}\sum_j \lambda_j \Sigma w_j\right) + \sum_i \lambda_i w_i^\top \left(\mu - \frac{1}{2}\sum_j \lambda_j \Sigma w_j\right) \quad (5)$$

The optimality conditions are:

$$\frac{\partial L(z^*, \lambda)}{\partial \lambda_t} = -\frac{1}{2}\sum_i \lambda_i w_t^\top \Sigma w_i + w_t^\top \mu = 0$$

for t such that $\lambda_t > 0$.
The function is optimized at

$$\lambda^* = 2(U\Sigma U)^{-1} U^\top \mu, \quad (6)$$

where U is formed by a subset of columns of W for which $\lambda_t > 0$, and thus $w_t^\top z^* = 0$.

For the last step we substitute the optimal λ, given in eq. 6 into the dual function in eq. 5 and after simple algebraic manipulations we get:

$$d^2 = \max_{\lambda \geq 0}(L(z^*, \lambda^*)) = \mu^\top U (U^\top \Sigma U)^{-1} U^\top \mu$$

■

Given the result of Theorem 1, we can express the background part of the mixed risk of the K-hyperplane classifier as follows,

$$L^B_{\mu,\Sigma}(W) = \sup_{Z \in \Omega(\mu,\Sigma)} Pr_{z \sim Z}(W^\top z \geq \vec{0}) = \frac{1}{1 + \mu^\top U (U^\top \Sigma U)^{-1} U^\top \mu}$$

2.2. K-Hyperplane Hinge-Minimax (KHHM) Training

We aim to minimize the mixed risk in eq. 4. To this end, we minimize the empirical risk

$$L^{HB}_S = L^B_S(W) + L^H_S(W) \quad (7)$$

regularized by the sum of L_2 norms of the K hyperplanes: $\frac{C}{2}\sum_i \|w_i\|^2 + L^{HB}_S$. This empirical risk is a non convex and non smooth function, hence a gradient based optimization of it is difficult. However, Osadchy et al. (2016) showed an algorithm for approximating this problem for a single hyperplane $\mathbf{w}$ using the following convex formulation:

$$\begin{aligned} &\underset{\mathbf{w}}{\text{minimize}} \;\; \lambda\|\mathbf{w}\|^2 + \sum_i \max\{0, 1 - \mathbf{w}^\top x_i\} \\ &\text{subject to} \;\; \gamma\sqrt{\mathbf{w}^\top \Sigma \mathbf{w}} + \mathbf{w}^\top \mu \leq 0. \end{aligned} \quad (8)$$

where $\gamma = \sqrt{\frac{1-\delta}{\delta}}$. This formulation minimizes the regularized hinge loss on the positive samples while constraining the probability of "background" data misclassified by the classifier $\mathbf{w}$ to be less than a small threshold δ.

We approximate the solution to the problem in eq. 7 by finding K hyperplanes W, which minimize the regularized hinge loss $L_S^H(W)$ on the positive samples while constraining the probability of "background" data misclassified by the intersection of these hyperplanes to be less than a small threshold ϵ.

We propose an algorithm (Algorithm 1) that computes the hyperplane iteratively, each hyperplane at a time using Osadchy et al. (2016). Note that since the algorithm in Osadchy et al. (2016) constraints the supremum over all distributions with given μ and Σ, it constraints an upper bound on the true distribution of the negative class. However, for a Gaussian distribution the bound is tight.

The Algorithm 1 starts by training K hyperplanes in a greedy manner and then iteratively adjusts each hyperplane to further reduce the loss.

Algorithm 1 KHHM Training

Input: $\{x_i\}, i = 1, .., m^+$ a set of positive examples; $\{z_i\}, i = 1, .., m^-$ a set of negative examples.

Initialization:

Estimate μ and Σ using $\{z_i\}_i^{m^-}$. Find w_1 using Osadchy et al. (2016) with μ, Σ.
for t=2 to k **do**
Estimate μ_t and Σ_t using $\{z_i \mid w_j^\top z_i > 0, j = 1, .., t-1\}$
Find w_t using Osadchy et al. (2016) with μ_t and Σ_t.
end for

Training:

for t=$1, 2, 3...$ **do**
Let P_t be the probability $Pr(W^\top z > 0)$ in iteration t
if $(P_{t-1} - P_t > \epsilon)$
Estimate μ_t and Σ_t using $\{z_i \mid w_j^\top z_i > 0, j = 1, .., K; j \neq t\}$.
Find w_t using Osadchy et al. (2016) with μ_t and Σ_t.
else
Output W (K hyperplanes)
end if
end for

Lemma 2 *Algorithm 1 minimizes the regularized hinge loss $L_S^H(W)$ on the positive samples while keeping $Pr(W^\top z \geq \vec{0}) \leq \epsilon$ (for a small ϵ).*

Proof Let Z denote the distribution of the negative class and $z \sim Z$ denote a sample from this distribution. Let S^t denote the part of negative class that falls inside the intersection of $K-1$ hyperplanes (w_t is not included):

$$S^t = \{z | w_i^\top z \geq 0, \forall i \in \{1, ..., K\} \setminus \{t\}\}.$$

In step i, Algorithm 1 finds w_{t_i} that minimizes the hinge loss (which is always positive) of w_{t_i} over positive labels and constrains $Pr(w_{t_i}^\top z \geq 0) | z \in S^{t_i}) \leq \delta$, while keeping the rest of the hyperplanes fixed.

The empirical risk of the intersection of K hyperplanes over positive labels is the maximum over K of hinge losses. Thus, the hinge loss of W is decreased at the iterations, in which the hyperplane with the maximal loss is updated, and it remains unchanged otherwise. Consequently, Algorithm 1 minimizes the hinge loss of W.

We can write

$$Pr(W^\top z \geq 0) = Pr(w_{t_i}^\top z \geq 0)|z \in S^{t_i})Pr(z \in S^{t_i}) + 0 \cdot Pr(z \notin S^{t_i}).$$

$Pr(z \in S^{t_i}) = a$ which is constant (does not depend on w_{t_i}) and $Pr(w_{t_i}^\top z \geq 0)|z \in S^{t_i}) \leq \delta$. Thus, $Pr(W^\top z \geq 0) \leq a\delta$. Setting $\epsilon = a\delta$ concludes the proof. ■

2.3. Generalization Bound for KHHM Model

In the following we bound the mixed risk of the KHHM classifier by its finite sample. We show that the discrepancy between the risk $L_D^{HB}(W)$ and its empirical estimation $L_S^{HB}(W)$ decays at the rate of $O(K\sqrt{\frac{\log(1/\delta)}{m}})$ where δ is the confidence over the samples of the training data and m is the training data size. The main difficulty in deriving a generalization bound arises from mixing the hinge risk for the positive examples and the background risk for the negative examples. We approach this problem by deriving the uniform generalization bounds separately for the positive and negative classes.

2.3.1. UNIFORM GENERALIZATION BOUND FOR THE EMPIRICAL BACKGROUND RISK

Recall that D_{neg} is the distribution of the negative data points, and μ and Σ are its mean and covariance respectively. Let $\hat{\mu}$ and $\hat{\Sigma}$ be the mean and covariance estimates from the training data points that are associated with negative labels. We bound the background risk by its training sample estimation. The generalization bound is dominated by the discrepancy

$$\Delta = L_{\mu,\Sigma}^B(w) - L_{\hat{\mu},\hat{\Sigma}}^B(w)$$

To provide uniform generalization bound to the background risk, we show that the discrepancy Δ decreases when the size of the training sample increases. Therefore we represent the discrepancy with $\|\hat{\mu} - \mu\|$ and $\|\hat{\Sigma} - \Sigma\|$ that decrease as a function of the training sample.

Let U denote a subset of columns of W that satisfy $w^\top z^* = 0$, where $z^* = \arg\min_z (z - \mu)^\top \Sigma^{-1}(z - \mu)$. We make two additional assumptions on U: First, the number of hyperplanes K_U comprising U is smaller than the dimension of the features and second, the hyperplanes in U are linearly independent. Both assumptions hold in practice. The number of hypeplanes must be small to make the classifier computationally efficient and $K_U \leq K$ (while the dimension of the feature space is usually large). The same reason justifies the independence assumption, as linearly dependent hyperplanes are redundant and do not contribute to the classifier, thus should be removed/avoided.

Using the result of Theorem 1, we can write the discrepancy Δ as follows:

$$\Delta = \frac{1}{1 + \mu^\top U(U^\top \Sigma U)^{-1} U^\top \mu} - \frac{1}{1 + \hat{\mu}^\top U(U^\top \hat{\Sigma} U)^{-1} U^\top \hat{\mu}}$$

By noting that the denominator of both terms is greater than 1 we can upper bound Δ by omitting the denominator. Then,

$$\Delta \leq \hat{\mu}U(U^\top \hat{\Sigma}U)^{-1}U^\top \hat{\mu} - \mu^\top U(U^\top \Sigma U)^{-1}U^\top \mu$$

Next, we denote $A \triangleq U(U^\top \Sigma U)^{-1}U^\top$ and $\hat{A} \triangleq U(U^\top \hat{\Sigma} U)^{-1}U^\top$. By adding and subtracting $\mu^\top A\hat{\mu}$ and rearranging the terms, we obtain

$$\Delta \leq \mu^\top A(\hat{\mu} - \mu) + \hat{\mu}^\top(\hat{A} - A)\hat{\mu} + (\hat{\mu} - \mu)^\top A\hat{\mu}. \tag{9}$$

Denote

$$\Delta_1 \triangleq \mu^\top A(\hat{\mu} - \mu) + (\hat{\mu} - \mu)^\top A\hat{\mu}$$

and

$$\Delta_2 \triangleq \hat{\mu}^\top(\hat{A} - A)\hat{\mu}.$$

Going back to the original notations, we obtain

$$\Delta_1 = \mu^\top U(U^\top \Sigma U)^{-1}U^\top(\hat{\mu} - \mu) + (\hat{\mu} - \mu)^\top U(U^\top \Sigma U)^{-1}U^\top \hat{\mu} \tag{10}$$

$$\Delta_2 = \hat{\mu}^\top(U(U^\top \hat{\Sigma} U)^{-1}U^\top - U(U^\top \Sigma U)^{-1}U^\top)\hat{\mu} \tag{11}$$

In the following, let $\|\cdot\|_F$ denote the Frobenius norm of a matrix (the ℓ_2- norm of matrix vectorized form).

Lemma 3 *Assume $x \sim D_{neg}$ is a distribution over data points x with negative labels such that $\|x\| \leq C$ holds with probability 1. Denote by μ its mean and by Σ its covariance. Let S_1 denote a training sample of size m_1 and let $\hat{\mu} = \frac{1}{m_1}\sum_{x\in S_1} x$ be its sampled mean and $\hat{\Sigma} = \frac{1}{m_1}\sum_{x\in S_1}(x-\hat{\mu})(x-\hat{\mu})^\top$ be its sampled covariance. Define Δ_1 as in eq. 10, where matrix U has K_U linearly independent columns ($K_U \leq n$) . Assume that the minimal eigenvalues of $\Sigma, \hat{\Sigma}$ are lower bounded by $\alpha = \lambda_{min}(\Sigma), \hat{\alpha} = \lambda_{min}(\hat{\Sigma})$, respectively. Then, with probability at least $1-\delta$ over the draws of the training set S_1, the following holds uniformly for all W*

$$\Delta_1 \leq \frac{2C}{\alpha}\sqrt{\frac{32C^4(\log(1/\delta) + 1/4)}{m_1}}$$

Proof First, we show the upper bound

$$\|U(U^\top \Sigma U)^{-1}U^\top\| \leq \frac{1}{\alpha} \tag{12}$$

Following the assumption that the columns of U are linearly independent and that their number is smaller than the dimension of the feature space, we can derive that $U = \sum_{i=1}^{K_U} s_i v_i t_i^\top$, where $v_1, ..., v_{K_U}$ are left singular vectors, $t_1, ..., t_{K_U}$ are right singular vectors, and $s_1, ..., s_{K_U}$ are singular values of U which are all non-zero. Let V be $n \times K_U$ matrix of left singular vectors $v_1, ..., v_{K_U}$, T be a $K_U \times K_U$ matrix of right singular vectors $t_1, ..., t_{K_U}$, and S be a $K_U \times K_U$ diagonal matrix of singular values $s_1, ..., s_{K_U}$, then

$$U(U^\top \Sigma U)^{-1}U^\top = VST^\top(TSV^\top \Sigma VST^\top)^{-1}TSV^\top = V(V^\top \Sigma V)^{-1}V^\top \tag{13}$$

$$\begin{aligned}\|V(V^\top \Sigma V)^{-1}V^\top\|^2 &= \max_{\|\beta\|\leq 1}\|V(V^\top \Sigma V)^{-1}V^\top \beta\|^2 \\ &= \max_{\|\beta\|\leq 1}(\beta^\top V(V^\top \Sigma V)^{-1}V^\top V(V^\top \Sigma V)^{-1}V^\top \beta) \\ &= \max_{\|\beta\|\leq 1}((V^\top \beta)^\top (V^\top \Sigma V)^{-2}(V^\top \beta)) \leq \|V^\top \Sigma V)^{-1}\|^2\end{aligned} \tag{14}$$

Since V is a projection operator, the last inequality in Eq. 14 is due to $\|V\beta\| \leq \|\beta\| \leq 1$.

$$\|V^\top \Sigma V)^{-1}\|^2 = \frac{1}{\lambda_{min}(V^\top \Sigma V)^2} \leq \frac{1}{\lambda_{min}(\Sigma)^2} \tag{15}$$

The last inequality in Eq. 15 follows from Poincaré Separation Theorem (Bellman (1970)).

Combining equations 13, 14, and 15, we obtain

$$\|U(U^\top \Sigma U)^{-1}U\| = \|V(V^\top \Sigma V)^{-1}V^\top\| \leq \|(V^\top \Sigma V)^{-1}\| \leq \frac{1}{\alpha}$$

Then applying the above upper bound and the Cauchy-Schwarz inequality, we obtain:

$$\Delta_1 \leq \frac{(\|\hat{\mu}\| + \|\mu\|)\|\hat{\mu} - \mu\|}{\alpha}.$$

Since $\|x\| \leq C$, then $\|\hat{\mu}\| + \|\mu\| \leq 2C$ and we get:

$$\Delta_1 \leq \frac{2C}{\alpha}\|\mu - \hat{\mu}\|.$$

Finally, we use the Bernstein inequality for vectors (cf. Candes and Plan (2011) Theorem 2.6) which states that for vectors $v_1, ..., v_{m_1}$ with $Ev_k = 0$ and $\|v_k\| \leq B$ and $\sum_k E\|v_k\|^2 \leq \sigma^2$ it holds that for all $0 \leq t \leq \sigma^2/B$ that $P[\|\sum_k v_k\| \geq t] \leq \exp(-\frac{t^2}{8\sigma^2} + \frac{1}{4})$. Here $\|v\|^2 = \sum_i v_i^2$.

To fit the Bernstein inequality for vectors to our setting, we set $v_k = \frac{1}{m_1}(x_k - E_{x\sim D_{neg}}x)$ for any $x_k \in S_1$. To see that the conditions of Candes and Plan (2011) Theorem 2.6 hold, we note that since $\|x\| \leq C$ then $\|v_k\| \leq 2C/m_1 \stackrel{def}{=} B$ and therefore $\|v_k\|^2 \leq \frac{4C^2}{m_1^2}$ and $\sum_{k=1}^{m_1} E\|v_k\|^2 \leq \frac{4C^2}{m_1} \stackrel{def}{=} \sigma^2$. Consequently it holds for $0 \leq t \leq 2C$ that

$$P[\|\hat{\mu} - \mu\| \geq t] \leq \exp\left(-\frac{m_1 t^2}{32C^2} + \frac{1}{4}\right).$$

The result follows when setting $\delta = \exp(-\frac{m_1 t^2}{32C^2} + \frac{1}{4})$, or equivalently $t = \sqrt{\frac{32C^2(\log(1/\delta)+1/4)}{m_1}}$. ■

We turn to handle the second term Δ_2 of the discrepancy.

Lemma 4 *Under the conditions of Lemma 3, define Δ_2 as in eq. 11. Then, with probability at least $1-\delta$ over the draws of the training set S_1, the following holds uniformly for all W*

$$\Delta_2 \leq \frac{C^2}{\alpha\hat{\alpha}}\Big(2C\|\hat{\mu} - \mu\| + \sqrt{\frac{32C^4(\log(1/\delta) + 1/4)}{m_1}}\Big)$$

Proof

$$\begin{aligned}
\Delta_2 &= \hat{\mu}^\top U\left((U^\top \hat{\Sigma} U)^{-1} - (U^\top \Sigma U)^{-1}\right)U^\top \hat{\mu} \qquad (16)\\
&\leq \left|\hat{\mu}^\top U(U^\top \hat{\Sigma} U)^{-1}(U^\top \Sigma U - U^\top \hat{\Sigma} U)(U^\top \Sigma U)^{-1}U^\top \hat{\mu}\right|\\
&= \left|\hat{\mu}^\top U(U^\top \hat{\Sigma} U)^{-1}U^\top(\Sigma - \hat{\Sigma})U(U^\top \Sigma U)^{-1}U^\top \hat{\mu}\right|
\end{aligned}$$

Applying the Cauchy-Schwarz inequality and the upper bound in eq. 12, we obtain:

$$\Delta_2 \leq \frac{\|\hat{\mu}\|^2\|\hat{\Sigma} - \Sigma\|}{\alpha\hat{\alpha}}.$$

We now consider $\|\Sigma - \hat{\Sigma}\|$:

$$\begin{aligned}
\hat{\Sigma} &= \frac{1}{m_1}\sum_{k=1}^{m_1}(x_k - \hat{\mu})(x_k - \hat{\mu})^\top = \frac{1}{m_1}\sum_{k=1}^{m_1} x_k x_k - \hat{\mu}\hat{\mu}^\top\\
\Sigma &= E_{x\sim D_{neg}}(x - \mu)(x - \mu)^\top = E_{x\sim D_{neg}}xx^\top - \mu\mu^\top\\
\|\Sigma - \hat{\Sigma}\| \leq \|\hat{\Sigma} - \Sigma\|_F &\leq \|\frac{1}{m_1}\sum_{k=1}^{m_1} x_k x_k^\top - E_{x\sim D_{neg}}xx^\top\|_F + \|\hat{\mu}\hat{\mu}^\top - \mu\mu^\top\|_F
\end{aligned}$$

For the second component in the bound: $\hat{\mu}\hat{\mu}^\top - \mu\mu^\top = \hat{\mu}(\hat{\mu}-\mu)^\top + (\hat{\mu}-\mu)\mu^\top$ therefore $\|\hat{\mu}\hat{\mu}^\top - \mu\mu^\top\|_F \leq \|\hat{\mu}(\hat{\mu}-\mu)^\top\|_F + \|(\hat{\mu}-\mu)\mu^\top\|_F \leq 2C\|\hat{\mu}-\mu\|$.

For the first component in the bound, we use the Bernstein inequality for vectors (Candes and Plan (2011) Theorem 2.6) in the same manner it is applied in Lemma 3. We set v_k to be the vectorization of $\frac{1}{m_1}x_k x_k^\top - E_{x\sim D_{neg}} xx^\top$ for any $x_k \in S_1$. To see that the conditions of Candes and Plan (2011) in Theorem 2.6 hold, we note that since $\|x\| \leq C$ then $\|v_k\| \leq \frac{2C^2}{m_1} \stackrel{def}{=} B$ and therefore $\|v_k\|^2 \leq \frac{4C^4}{m_1^2}$ and $\sum_{k=1}^{m_1} E\|v_k\|^2 \leq \frac{4C^4}{m_1} \stackrel{def}{=} \sigma^2$. Consequently it holds for $0 \leq t \leq 2C^2$ that

$$P\Big[\Big\|\frac{1}{m_1}\sum_{k=1}^{m_1} x_k x_k^\top - E_{x\sim D_{neg}} xx^\top\Big\| \geq t\Big] \leq \exp(-\frac{m_1 t^2}{32C^4} + \frac{1}{4})$$

The result follows when setting $\delta = \exp(-\frac{m_1 t^2}{32C^4} + \frac{1}{4})$ or equivalently $t = \sqrt{\frac{32C^4(\log(1/\delta)+1/4)}{m_1}}$. ■

The upper bound for Δ relies on the lower bound on the minimal eigenvalue of covariance $\Sigma = E_{x\sim D_{neg}}(x-\mu)(x-\mu)^\top$ and of the sampled covariance $\hat{\Sigma} = \frac{1}{m_1}\sum_{x\in S_1}(x-\hat{\mu})(x-\hat{\mu})^\top$. While the minimal eigenvalue of Σ, say $\alpha = \lambda_{min}(\Sigma)$, can be set to be away from zero, the minimal eigenvalue of the sampled covariance $\hat{\alpha} = \lambda_{min}(\hat{\Sigma})$ is a random variable. We show that $\hat{\alpha}$ is close to α with high probability:

Lemma 5 *Assume the conditions of Lemma 3 hold. Assume that the minimal eigenvalue of $\alpha = \lambda_{min}(\Sigma)$ is positive. Let $\hat{\alpha} = \lambda_{min}(\hat{\Sigma})$ be the minimal eigenvalue of the random covariance matrix $\hat{\Sigma}$. Then $\hat{\alpha} \geq \alpha/2$ with probability at least $1 - 2\exp(-\frac{m_1\alpha^2}{32\cdot 36C^4} + \frac{1}{4})$ over the draws of the training set S_1.*

Proof Using Cauchy-Schwartz inequality we obtain

$$\|U^\top\Sigma U\| - \|U^\top\hat{\Sigma}U\| \leq \Big|\|U^\top\Sigma U\| - \|U^\top\hat{\Sigma}U\|\Big| \leq \|U^\top\Sigma U - U^\top\hat{\Sigma}U\| \leq \|\Sigma-\hat{\Sigma}\|\|U\|^2.$$

Therefore,

$$\|U^\top\hat{\Sigma}U\| \geq \|U^\top\Sigma U\| - \|\Sigma-\hat{\Sigma}\|\|U\|^2.$$

Following Lemma 4 we note that

$$\|\hat{\Sigma}-\Sigma\| \leq \|\hat{\Sigma}-\Sigma\|_F \leq \|\frac{1}{m_1}\sum_{k=1}^{m_1} x_k x_k^\top - E_{x\sim D_{neg}} xx^\top\|_F + \|\hat{\mu}\hat{\mu}^\top - \mu\mu^\top\|_F$$

We use Bernstein inequality for vectors with $t = \alpha/6$ to bound

$$P[\|\frac{1}{m_1}\sum_{k=1}^{m_1} x_k x_k^\top - E_{x\sim D_{neg}} xx^\top\| \geq \alpha/6] \leq \exp(-m_1\alpha^2/(36\cdot 32) + 1/4)$$

for any $\alpha/6 \leq 4m_1$. We also use Bernstein inequality for vectors with $t = \alpha/(2C\cdot 3)$ to bound

$$P[\|\hat{\mu}-\mu\| \geq \alpha/(2C\cdot 3)] \leq \exp\left(-\frac{m_1\alpha^2}{36\cdot 32C^4} + \frac{1}{4}\right).$$

Thus with error probability of $2\exp\left(-\frac{m_1\alpha^2}{36\cdot 32C^4} + \frac{1}{4}\right)$ there hold $\|\frac{1}{m_1}\sum_{k=1}^{m_1} x_k x_k^\top - E_{x\sim D_{neg}} xx^\top\| \leq \alpha/6$ and $2C\|\hat{\mu}-\mu\| \leq \alpha/3$. In particular, the sum of both is upper bounded by $\alpha/2$, hence $\|\hat{\Sigma}-\Sigma\| \leq \alpha/2$, resulting in $\|U^\top\hat{\Sigma}U\| \geq (\alpha-\alpha/2)\|U\|^2$. Therefore $\hat{\alpha}\|U\|^2 \geq \alpha/2\|U\|^2$. ■

Bounds on the discrepancy between the expected and the empirical background risks that are uniform for any U guarantee generalization. The above lemmas suggest that the penalty of observing a finite sample space decreases as $1/m_1$. This is summarized in the following theorem.

Theorem 6 *Under the conditions of Lemma 3, for* $\alpha \geq \sqrt{\frac{1152(\log(2/\delta)+1/4)}{m_1}}$ *with probability at least* $1-3\delta$ *over* m_1 *the i.i.d. samples from* D_{neg} *the following holds uniformly for all* W *with* K *linearly independent hyperplanes:*

$$L^B_{\mu,\Sigma}(U) \leq L^B_{\hat{\mu},\hat{\Sigma}}(U) + \Big(\frac{2C^2}{\alpha} + \frac{6C^4}{\alpha^2}\Big)\sqrt{\frac{32(\log(1/\delta)+1/4)}{m_1}}.$$

Proof Following Eq. 9, we note that $L^B_{\mu,\Sigma}(U) - L^B_{\hat{\mu},\hat{\Sigma}}(U) \leq \Delta_1 + \Delta_2$, for Δ_1, Δ_2 defined in Lemmas 3, 4. The proof applies the bounds on Δ_1, Δ_2, where each of these bounds holds with an error probability at most δ. The proof is concluded by bounding $\hat{\alpha} \geq \alpha/2$ with an error probability at most δ. Formally, with error probability at most 3δ:

$$\begin{aligned}
\Delta_1 &\leq \frac{2C^2}{\alpha}\sqrt{\frac{32\log(1/\delta)+1/4}{m_1}} \\
\Delta_2 &\leq \frac{3C^4}{\alpha\hat{\alpha}}\sqrt{\frac{32(\log(1/\delta)+1/4)}{m_1}} \leq \frac{6C^4}{\alpha^2}\sqrt{\frac{32(\log(1/\delta)+1/4)}{m_1}}
\end{aligned}$$

■

The generalization guarantees of the background risk penalize a finite sample size m by $\sqrt{1/m_1}$. It decays to zero when the number of the negative labels in the training sample tends to infinity. In our setting, we assume that $m \approx m_1$, thus we get favorable guarantees with respect to the training size.

2.3.2. UNIFORM GENERALIZATION BOUND FOR THE EMPIRICAL RISK OF THE HINGE-LOSS

We derive a uniform generalization bound for the expected risk over the positive examples using Rademacher complexity. The Rademacher complexity of a bounded set $A \subset R^K$ is

$$R(A) = \frac{1}{m}E_\sigma\left[\max_{a\in A}\sum_{i=1}^{m}\sigma_i a_i\right],$$

while $\sigma_i \in \{-1,+1\}$ are i.i.d. and equally probable random variables.

Let $\mathcal{F}$ denote a family of functions:

$$\mathcal{F} \triangleq \{(x,y) \to \ell(W,x,y) : W = [w_1,...,w_k], w_j \in \mathbb{R}^d, \forall j\}.$$

Let $S = \{(x_1,y_1),...,(x_m,y_m)\}$ be a training sample. Let $\mathcal{F}\circ S$ be the set of all possible evaluations a function $f \in \mathcal{F}$ can achieve on a sample S:

$$\mathcal{F}\circ S = \{f(x_1,y_1),...,f(x_m,y_m)\}.$$

The Rademacher complexity of $\mathcal{F}$ with respect to S is defined as follows:

$$R(\mathcal{F}\circ S) \triangleq \frac{1}{m}\mathbb{E}_{\sigma\sim\{\pm1\}^m}\left[\sup_{f\in\mathcal{F}}\sum_{i=1}^{m}\sigma_i f(x_i,y_i)\right]$$

Since L^H_D is zero for $y=-1$, we consider only the positive subset of S, denoted as, $S_2 \triangleq \{(x_i,1),...(x_{m_2},1)\}$.

Theorem 7 [3] *Consider a* $K-$*hyperplanes loss function*

$$\ell(W,x,y) = \max_{j\in\{1,...,K\}}\{\max\{0,1-yw_j^\top x\}\}$$

3. Theorem 4 in the ICML'15 version of the paper had a typo. Here we present a corrected version of the theorem with a detailed proof.

for which each hyperplane satisfies $\|w_j\| \leq 1$ *and each data point satisfies* $\|x\| \leq 1$. *Then,*

$$R(\mathcal{F} \circ S_2) \leq \frac{K}{\sqrt{m_2}},$$

where m_2 *is the number of positive examples.*

Proof

$$m_2 R(\mathcal{F} \circ S_2) =$$

$$\mathbb{E}_{\sigma \sim \{\pm 1\}^m} \left[\max_{\substack{\|w_1\| \leq 1 \\ \dots \\ \|w_k\| \leq 1}} \sum_{i=1}^{m} \sigma_i \max_{j \in \{1,\dots,K\}} \{\max(0, 1 - w_j^\top x_i)\} \right] \leq$$

$$\mathbb{E}_{\sigma \sim \{\pm 1\}^m} \left[\max_{\substack{\|w_1\| \leq 1 \\ \dots \\ \|w_k\| \leq 1}} \sum_{i=1}^{m} \sigma_i \sum_{j=1}^{K} \max(0, 1 - w_j^\top x_i) \right] \leq$$

$$\sum_{j=1}^{K} \mathbb{E}_{\sigma \sim \{\pm 1\}^m} \left[\sum_{i=1}^{m} \sigma_i \max(0, 1 - w_j^\top x_i) \right] \leq K\sqrt{m_2}$$

The bound follows from the Contraction Lemma (Ledoux and Talagrand, 1991), applied to $\max(0, 1 - w^\top x)$, which is 1-Lipschitz function.
Hence, $R(\mathcal{F} \circ S_2) \leq \frac{K}{\sqrt{m_2}}$. ■

Next we provide the uniform generalization bound for the empirical risk of the maximum over hinge losses.

Theorem 8 *Let* $L_D^H(W) = \mathbb{E}_D \left[\ell(W, x, y) \mathbb{1}[y = 1] + 0 \cdot \mathbb{1}[y = -1] \right]$ *be the expected risk, and let* $L_S^H(W) = \frac{1}{m_2} \sum_{i=1}^{m} \ell(W, x_i, 1)$ *be the empirical risk over a positive label training sample of size* m_2. *Then, for any* $\delta \in (0, 1]$ *with probability at least* $1 - \delta$ *over the i.i.d. sample of size* m_2 *it holds simultaneously for all* $\|w_1\|, ..., \|w_k\| \leq 1$ *that whenever* $\|x\| \leq 1$*:*

$$L_D^H(W) \leq L_S^H(W) + \frac{2K}{\sqrt{m_2}} + 8K\sqrt{\frac{2\log(2/\delta)}{m_2}}$$

Proof By noting that $|\ell(W, x, y)| \leq 2K$ and that a maximum over positive numbers is upper bounded by their sum, the result follows immediately, from Bartlett and Mendelson (2003). ■

3. Latent Hinge-Minimax Classifier

The intersection of K positive half-spaces forms a convex set. For non-convex sets, KHHM will produce many false positives (as show in Figure 2 left). To accommodate classes that form non-convex or disjoint sets, we propose a non-convex classifier, which is an ensemble of KHHM models that we call the Latent Hinge-Minimax (LHM) classifier. Specifically, we define the LHM classifier as a union of intersections of positive half-spaces. We assume that each intersection is composed of K hyperplanes: $W^i = [w_1^i, ..., w_K^i]$ and there are C components in the union (see Figure 2 right). Let $W_{LHM} \triangleq (W^1, \dots, W^C)$ denote the LHM model.

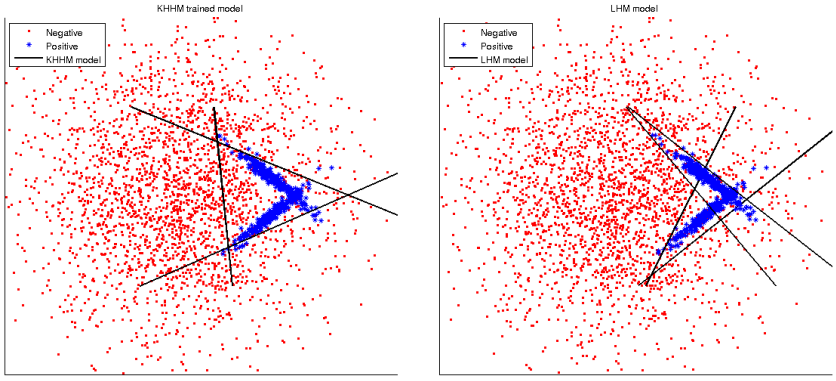

Figure 2: Schematic comparison of KHHM (left) and LHM (right) classifiers on a non-convex positive class. The LHM classifier iteratively discovers a partition of the positive set into convex components and builds KHHM model for each convex component.

3.1. Expected Latent Mixed Risk

Similarly to K-hyperplane model, the latent mixed risk is also composed of the hinge and background parts. However, we extend the risk in Eq. 4 to contain multiple components $Q^i \triangleq \{x \in R^n | {W^i}^T x \geq \vec{0}\}$ and a latent variable $\varphi(x) = i$ ($i \in \{1, \ldots, C\}$) which assigns each positive sample $x \in \mathbb{R}^n$ to one of the C components.

$$L_D(W_{LHM}; \varphi) = L^B_{\mu,\Sigma}(W_{LHM}) + L^H_D(W_{LHM}; \varphi), \tag{17}$$

The background part of the latent mixed risk bounds the probability of the negative class in all components Q^i:

$$L^B_{\mu,\Sigma}(W_{LHM}) = \sum_{i=1}^{C} L^B_{\mu,\Sigma}(W^i) = \sum_{i=1}^{C} \sup_{z \sim Z(\mu,\Sigma)} \Pr(z \in Q^i) \tag{18}$$

The hinge part of the latent mixed risk aggregates the K-hyperplane hinge risk over C components:

$$L^H_D(W_{LHM}; \varphi) = \mathbb{E}_{(x,y)\in D} \left[\sum_{i=1}^{C} \ell(W^i; x, y) \mathbb{1}\left[\varphi(x) = i\right] \right] \tag{19}$$

where

$$\ell(W; x, y) = \max_{j \in \{1..K\}} \{\max\{0, \alpha - y w_j^T x\}\}$$

is the modified K-hyperplane hinge loss (in Eq. 2). We replaced 1 with α to accommodate comparison between different norms of the hyperplanes.

3.2. Empirical Latent Mixed Risk

Recall that S is a training sample of size m, where $S_2 = \{(x, y) \in S : y = 1\}$, and $S_1 = \{(x, y) \in S : y = -1\}$ are the positive and negative training sets correspondingly, m_2 is the size of S_2 and m_1 be the size of S_1. We define the empirical risk over S as follows:

$$L^{HB}_S(W_{LHM}, \varphi) = L^H_{S_2}(W_{LHM}, \varphi) + L^B_{S_1}(W_{LHM}) \tag{20}$$

Both parts of the empirical latent mixed risk are aggregated over C latent components. Specifically, the background part of the risk is defined as a sum of background empirical risks of the model's components:

$$L^B_{S_1}(W_{LHM}) = \sum_{i=1}^{C} L^B_{\hat{\mu},\hat{\Sigma}}(W^i) \tag{21}$$

where $L^B_{\hat{\mu},\hat{\Sigma}}(W^i) = \sup_{Z\in\Omega(\hat{\mu},\hat{\Sigma})} Pr_{z\sim Z}(z \in Q^i)$ and $\hat{\mu}, \hat{\Sigma}$ are the empirical mean and covariance matrix, estimated from the negative training sample S_1.

Let $X^i \triangleq \{x : \varphi(x) = i\}$ define a subset of positive samples. The hinge part of the empirical risk is defined as the follows:

$$L^H_{S_2}(W_{LHM}, \varphi) = \sum_{i=1}^{C} L^H_{S_2}(W^i) \tag{22}$$

where $L^H_{S_2}(W^i) = \sum_{x\in X^i} \ell(W^i, x, 1)$.

Using the above notations, we can define a component empirical risk, as

$$L^{HB}(W^i) = L^H_{S_2}(W^i) + L^B_{\hat{\mu},\hat{\Sigma}}(W^i) \tag{23}$$

Next, we formalize the loss function for a positive sample which we use in the training algorithm in Section 3.3. Each sample with positive label encounters a loss only in a single latent component, specified by its latent variable $\varphi(x)$. The hinge part of this loss is $\ell(W^i, x, 1)$. The background empirical risk of a component $L^M_{\hat{\mu},\hat{\Sigma}}(W^i)$ depends on $\hat{\mu}, \hat{\Sigma}$ and W^i. W^i depends on the latent assignment of the positive samples. Thus the optimal assignment should minimize also the background part of the risk. We implement this by dividing the empirical risk $L^M_{\hat{\mu},\hat{\Sigma}}(W^i)$ equally among the positive samples with $\varphi(x) = i$. Hence the sample loss (for positive samples) is defined as follows:

$$L(W^{\varphi(x)}; x, 1, \varphi(x)) = \ell(W^{\varphi(x)}, x, 1) + \frac{1}{|S^i|} L^M_{\hat{\mu},\hat{\Sigma}}(W^{\varphi(x)}) \tag{24}$$

where $|S^i|$ is the number of samples with $\varphi(x) = i$.

3.3. LHM Training Algorithm

The training aims to minimize the empirical risk in Eq. 20 over the parameters W_{LHM} and the hidden variables φ. We propose an iterative algorithm, which reaches fast convergence and shows good results in practice. The algorithm iterates between two steps: First, given an assignment it produces a model W_{LHM}, second, it updates the latent variables $\varphi(x), \forall(x, y) \in S_2$ to better represent the latent structure of the data.

The **first** step updates the LHM model W^t_{LHM} in iteration t given the latent variables φ from iteration $t-1$. Namely, for each hidden component $i = 1, ..., C$, we find the hyperplanes W^i separating the training samples in S^i from D_{neg} by minimizing the empirical risk in Eq. 23. This risk is minimized by the training algorithm proposed in Algorithm 1.

The **second** step updates the latent variable assignment, given the current W^t_{LHM}. For each positive sample, it finds the best component w.r.t. the risk in Eq 20. Specifically, the hinge risk for x is simply $\ell(W^i, x, 1)$. The background part of the assignment function for $x \notin Q^i$ should consider the probability that this point adds when it is included in the component i (as shown in Figure 3, left). For $x \in Q^i$, the background part should consider the amount of probability released when the component shrinks as a result of change in the assignment of x (as shown in Figure 3, right). The optimal assignment should take both cases into consideration for all components. To define the assignment function we introduce the following notations.

W^{def} is a *deflated model* derived from W^i by parallel translation of the hyperplane closest to x such that $w^T_* x + b_* = 0$. W^{inf} is an *inflated model* derived from W^i by parallel translation of the hyperplanes for

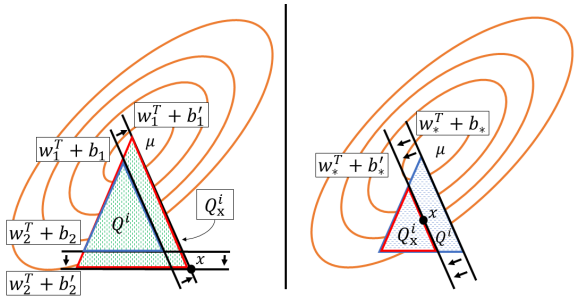

Figure 3: The orange ellipses represent the negative distribution $Z(\hat{\mu}, \hat{\Sigma})$, the red triangles corresponds to Q_x^i and the blue ones to Q^i. **Left**: w_1^i, w_2^i are moved to pass through x, causing the probability Q^i to increase. **Right**:w_*^i is moved to pass through x, causing the probability Q^i to decrease.

which $w_k^T x + b_k < 0$, until they intersect in x, namely, $w_k^T x + b_k' = 0$. W_x^i is a surrogate model, defined as follows,

$$W_x^i \triangleq \begin{cases} W^{def} & \text{if } x \in Q^i \\ W^{inf} & \text{if } x \notin Q^i \end{cases}$$

and $Q_x^i \triangleq \{x : {W_x^i}^T x \geq \vec{0}\}$. Using the above notations, we define the assignment function as follows,

$$\varphi(x) = \underset{i \in \{1..C\}}{\operatorname{argmin}} Pr_{z \sim Z}(z \in Q_x^i) + \lambda \ell(W^i; x, 1) \tag{25}$$

where λ is a balancing parameter. The full training algorithm is summarized in Algorithm 2.

Lemma 9 *Algorithm 2 reduces the empirical risk $L_S^{HB}(W_{LHM}; \varphi)$ in each iteration.*

Proof Since latent mixed risk is a sum of risks over the latent components (Eq. 20), it is minimized by minimizing the empirical risk of each component. In step (5) of the Algorithm 2, we train $W^{i,t}$ model for each latent component $i = 1, ..., C$ using Algorithm 1 (Section 2.2). It is easy to see that $L_S^{HB}(W^i) = L_S^{HB}(W)$ (in Eq. 7), thus step (5) of the Algorithm 2 minimizes the component's risk in Eq. 23.

It is now left to show that the assignment φ^t in iteration t, will cause the reduction in the empirical risk in iteration $t+1$. Since the empirical risk is aggregated over positive samples, it is enough to prove the claim for a single sample. We consider two cases:

1. The assignment of sample x does not change, formally $\varphi^t(x) = \varphi^{t+1}(x)$.
In this case $L(W_{LHM}^{t+1}; \varphi^{t+1}(x))$ will only be affected by the $W^{i,t+1}$ training, thus

$$L\left(W_{LHM}^t; \varphi^t(x)\right) \geq L\left(W_{LHM}^{t+1}; \varphi^{t+1}(x)\right) \tag{26}$$

2. The assignment of sample x is changed. Formally, in iteration t: $\varphi^t(x) = i$ and in iteration $t+1$, exists $j \neq i$, such that

$$\varphi^{t+1}(x) = j = \underset{k \in \{1..C\}}{\operatorname{argmin}} L_{S_1}^B\left(W_x^{k,t}\right) + \lambda L_{S_2}^H\left(W^{k,t}; x\right). \tag{27}$$

Since $x \in Q^i$, reassigning it to a different component will cause the $Pr_{z \sim Z}(z \in Q_x^i)$ to decrease (or stay the same), thus

$$L_{S_1}^B(W_x^{i,t}) - L_{S_2}^B(W^{i,t}) \leq 0. \tag{28}$$

Algorithm 2 LHM Training. T is the threshold on the empirical risk change.

Input: C, K, S_1, S_2,T
Initialization:
$t \leftarrow 1$
$L(W_{LHM}^{t=0}; \varphi^{t=0}) \leftarrow \infty$
$\varphi^t \leftarrow Init(S_2, C)$

Training:
while $L(W_{LHM}^{t}; \varphi^{t}) - L(W_{LHM}^{t-1}; \varphi^{t-1}) \geq T$ **do**
{Model Step}
for i=1 to C do
$W^{i,t}$ =*KHHM-training*(S_1,X^i) {in Algorithm 1}
end for
{Assignment Step}
for$(x, y) \in S_2$
$\varphi^{t+1}(x)$ as defined in Eq. 25
end for
$t \leftarrow t + 1$
end while
Output: W_{LHM}, φ

Hence, the sample loss in component i is larger than the sample loss in the deflated component:

$$L\left(W^{i,t}; x\right) \geq L_{S_1}^{B}\left(W_x^{i,t}\right) + \lambda L_{S_2}^{H}\left(W^{i,t}; x\right). \tag{29}$$

At the same time, j is the optimal assignment, thus

$$L_{S_1}^{B}\left(W_x^{i,t}\right) + \lambda L_{S_2}^{H}\left(W^{i,t}; x\right) \geq L_{S_1}^{B}\left(W_x^{j,t}\right) + \lambda L_{S_2}^{H}\left(W^{j,t}; x\right). \tag{30}$$

Since $W_x^{j,t}$ is a naive inflation of $W^{j,t}$ to include x, the solution $W^{j,t+1}$, provided by KHHM training, would have lower (or same) empirical risk, thus

$$L_{S_1}^{B}\left(W_x^{j,t}\right) \geq L_{S_1}^{B}(W^{j,t+1}). \tag{31}$$

In iteration $t+1$, x is included in X^j for training the j'th latent component, consequently

$$L_{S_2}^{H}\left(W^{j,t}; x\right) \geq L_{S_2}^{H}(W^{j,t+1}; x). \tag{32}$$

(as we assume that $x \in X^j$ leads to $x \in Q^{j,t+1}$). Finally, by combining the inequalities in Eq. 29–32, we obtain:

$$L\left(W^{i,t}; x\right) \geq L\left(W^{j,t+1}; x\right). \tag{33}$$

■

3.4. Generalization Bound for LHM Model with Fixed Assignment

For a fixed $\varphi(x), \forall(x, y) \in S_2$, we can derive a uniform generalization bound for the union of the K-hyperplane models. Similarly to KHHM model (in Section 2.3), we derive the uniform generalization bounds separately for the positive and negative classes. We start with the positive class, for which we use the hinge part of the latent mixed risk.

Theorem 10 *Let φ^* denote a fixed assignment of the positive training samples to components. Let $L_D^H(W_{LHM};\varphi^*) = \mathbb{E}_{(x,y)\in D}\left[\sum_{i=1}^C \ell(W^i;x,y)\mathbb{1}\left[\varphi^*(x)=i\right]\right]$ be the expected risk, and let $L_{S_2}^H(W_{LHM},\varphi^*) = \sum_{i=1}^C L_{S_2}^H(W^i)$ be the empirical risk over a positive label training sample of size m_2 for a fixed assignment φ^* of the positive samples to C components. Then, for any $\delta \in (0,1]$ with probability at least $1-\delta$ over the i.i.d. sample of size m_2 it holds simultaneously for all $\|w_i\| \le 1$ $(i=1,...,C\cdot K)$ that whenever $\|x\| \le 1$:*

$$L_D^H(W_{LHM};\varphi^*) \le L_{S_2}^H(W_{LHM},\varphi^*) + \sqrt{\frac{\log 1/\delta}{2m_2}} L_{S_2}^H(W_{LHM},\varphi^*) + \max_{i\in\{1,...,C\}}\left(\frac{2K}{\sqrt{m^i}} + 8K\sqrt{\frac{2\log(2/\delta)}{m_2}}\right)$$

Proof Let $p_i = E_{(x,1)\sim D}\Big[\mathbb{1}[\varphi^*(x)=i]\Big]$, and let $\frac{m_i}{m^+}$ be its estimated mean. Then,

$$L_D^H(W_{LHM};\varphi^*) - L_{S_2}^H(W_{LHM},\varphi^*) = \sum_{i=1}^C\left(p_i L_D^H(W^i) - \frac{m_i}{m_2}L_{S^+}^H(W^i)\right)$$
$$= \sum_{i=1}^C\left[p_i\left(L_D^H(W^i) - L_{S_2}^H(W^i)\right)\right] + \sum_{i=1}^C\left[(p_i - \frac{m_i}{m^+})L_{S_2}^H(W^i)\right] \quad (34)$$

We can bound the discrepancy between the expected and empirical risks in a component using Theorem 8. Hence, the first term in Eq. 34 is upper bounded by $\max_{i\in\{1,...,C\}}\left(\frac{2K}{\sqrt{m^i}} + 8K\sqrt{\frac{2\log(2/\delta)}{m_2}}\right)$. We can upper bound $(p_i - \frac{m_i}{m_2})$ using the Hoeffding inequality. Rearranging the terms and noting that $\sum_{i=1}^C L_{S_2}^H(W^i) = L_{S_2}^H(W_{LHM},\varphi^*)$ conclude the proof. ■

We formulate the uniform generalization bound for the negative class below.

Theorem 11 *Suppose that D is a distribution over $X\times Y$ such that $Y=\{-1,+1\}$ and $X=\{x:\|x\|\le G\}$. Let $L_{\mu,\Sigma}^B(W_{LHM};\varphi^*)$ be the background risk over the negative labels, where μ,Σ are the mean and covariance of the marginal distribution of x over the negative labels and the positive labeled samples have a fixed assignment φ^*. Consider a training sample S of size m, m_1 of which have negative label and let*

$$L_{\hat\mu,\hat\Sigma}^B(W_{LHM};\varphi^*) = \sum_{i=1}^C L_{\hat\mu,\hat\Sigma}^B(W^i)$$

be the empirical background risk over the negative labels ($\hat\mu,\hat\Sigma$ are the empirical mean and covariance estimation of D_{neg}). With probability at least $1-3\delta$ over m_1 the i.i.d. samples from D_{neg} the following holds uniformly for all W including C components, each with K independent hyperplanes:

$$L_{\mu,\Sigma}^B(W_{LHM};\varphi^*) \le L_{\hat\mu,\hat\Sigma}^B(W_{LHM};\varphi^*) + C\Big(\frac{2G^2}{\alpha} + \frac{6G^4}{\alpha^2}\Big)\sqrt{\frac{32(\log(1/\delta)+1/4)}{m_1}}.$$

The proof is straightforward as the assignment φ^* does not affect negative samples, and thus the bound is a simple summation of bounds for each component which is derived using Theorem 6.

4. Mapping LHM Classifier to a Neural Network

Deep Neural Networks and Convolutional Neural Networks (CNN) in particular have shown impressive results in a variety of domains, including images, speech, text, etc. CNN enables learning very good features for these domains, but requires a lot of labeled training samples. One way to reduce the number of examples

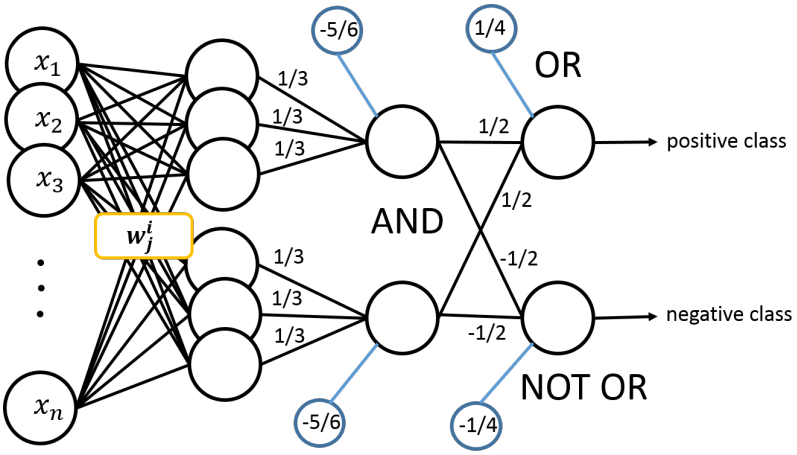

Figure 4: An example of NN equivalent to LHM for two components and three hyperplanes in each.

needed for training of a specific classification task is to pre-train a CNN on a different classification problem in a similar domain and then change the last layer of the CNN to fit the target classification problem and fine-tune the network on a smaller training set associated with the target problem. This approach is referred to as transfer learning.

When the target classification problem is less similar to that used to train CNN features, the classification accuracy of the fine-tuned network could be quite poor. This is because the high-level representation of the original and the target networks are different. One way to approach this problem is by employing a non-linear classifier, such as LHM classifier, on CNN features. In order to fine-tune the feature layers with the LHM classifier, we need to combine them in a single architecture. To this end we propose to map LHM Classifier to a Neural Network and stack it on top of the pre-trained convolutional layers. This enables end-to-end training of feature extraction and classifier. As we show below, mapping of LHM classifier to NN also allows extending it to multi-class problems.

4.1. Binary NN

A union of the intersections of positive half spaces can be implemented by a NN with three hidden layers. The first fully connected hidden layer has $K \times C$ neurons with a sigmoid activation, where K is the number of hyperplanes in an intersection and C is the number of components. The second hidden layer has C nodes with a sigmoid activation, connected only to the neurons associated with hyperplanes forming the corresponding intersection. The weights on these connections and the biases are fixed and mimic **AND** operation, namely, all weights of this layer are equal to $1/K$ and the biases are equal to $-1 + 1/(2K)$. The last hidden layer has two neurons, which are fully connected to the previous layer with the fixed weights and biases, one of which mimics **OR** operation, and the other **NOT OR**. Namely, the neuron, corresponding to OR has weights equal to $1/C$ and the bias of $-1/(2C)$. The neuron corresponding to NOT OR has weights equal to $-1/C$ and the bias of $1/(2C)$. The network has two outputs, one for the positive class (with label 1) and one for the negative class (with label 0). An example of such network for $C = 2$ and $K = 3$ is depicted in Figure 4.

4.2. Multi-Class NN

For a multi-class setting, we suggest to train LHM model for each class using an additional unlabeled data for estimating the statistics of the negative class. We then map these models to a multi-class NN with the following architecture. The first hidden layer is a fully connected layer with $C \times K$ neurons per class,

$C \times K \times G$ neurons in total, where G is the number of classes. These are equivalent to $C \times K \times G$ hyperplanes in the LHM model. For each hidden component, all hyperplanes in the intersection are connected to their corresponding node in the **AND** layer (as detailed in Section 4.1). The **AND** layer comprises $C \times G$ neurons. The next layer is a fully connected layer, comprising G nodes. The weights on the connections to the C components of the corresponding class are initialized with 1's, and the weights on the remaining connections are initialized with very small values from a Gaussian distribution. The network has G outputs and is trained using the cross-entropy loss.

To provide an end-to-end training, one can consider stacking the feature extraction layers of CNN (up to fully connected layers) with one of the above networks.

5. Experiments

We start by evaluating the simple K-hyperplane model (Section 5.1) and then move to a more general LHM model (Section 5.2). Both models are tested on synthetic and real data. Then we show how the hinge-minimax training can be combined with a CNN (Section 5.3) for approaching problems that require more powerful features but do not have large enough data to train a deep model from scratch. We demonstrate this for both binary and multi-class settings.

5.1. K-Hyperplane Hinge-Minimax Classifier

To test the proposed KHHM classifier, we ran experiments in three different scenarios: synthetic 2D data, letter recognition, and large scale scene classification.

During classification, the K-hyperplane classifier incurs only K times the computational complexity of a linear classifier (just K inner products), hence its "natural competitors" are linear classifiers, and we choose linear SVM for the benchmark. We have also compared the hinge-minimax classifiers to kernel SVM and ensemble-based methods, which incur far longer running times (this is especially true for kernel SVM). The classification rates of the hinge-minimax classifier in all our experiments were comparable to ensemble classifiers which required 100-170 basic classifiers in order to reach similar performance. In experiments with high-dimensional data, the KHHM classifiers preformed as well as kernel SVM.

The SVM classifiers were trained using C-SVC in LIBSVM [4]. We used the CVX optimization package [5] to find a single hyperplane in Algorithm 1. The ensemble classifiers were trained using the Matlab Statistic toolbox.

5.1.1. SYNTHETIC DATA EXAMPLE

We construct the KHHM classifier for 2D data to illustrate Algorithm 1. We samples 5000 data points from two highly overlapping Gaussians (see Figure 5) with varying ratio of positive (shown in red) and negative (shown in blue) examples. Each class was equally partitioned into training, validation, and test sets. We estimated the mean and covariance from the training data and tuned the parameters (C and γ) and the bias using the validation set. Table 1 shows the AUC for the different ratios of positive and negative examples using an intersection of 5 hyperplanes. These results demonstrate the robustness of the algorithm to imbalanced sets.

Positive fraction	0.01	0.1	0.2	0.3	0.4	0.5
AUC	94.68	94.91	95.07	94.96	94.89	95.83

Table 1: AUC for different size partitions of positive and negative classes

4. $http://www.csie.ntu.edu.tw/cjlin/libsvm/$
5. $http://cvxr.com/cvx/download/$

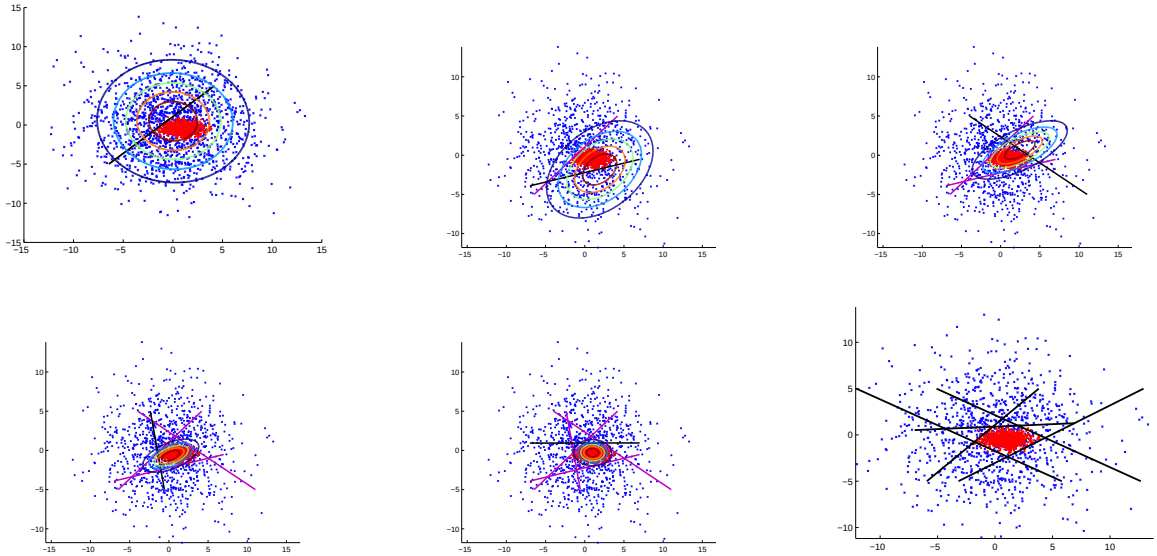

Figure 5: Illustration of KHHM classifier construction on a toy example.The first 5 figures show the greedy initial step. The last figure shows the final classifier after 25 iterations. The contour lines show the covariance matrix of the negative distribution inside the intersection of hyperplane, which is used to find the optimal hyperplane, depicted in black.

The first five plots in Figure 5 show the result of the initial greedy step for the first, second, third, forth, and fifth hyperplanes respectively. The contour lines in Figure 5 illustrate the covariance of the negative distribution inside the intersection, which is used to find the optimal separation hyperplane, depicted in black. The last plot in Figure 5 shows the final classifier after 25 iterations. It illustrates that the approximation algorithm succeeds in separating the positive set from the background, and that the refinement iterations improve the separation boundary.

5.1.2. LETTER RECOGNITION

The following tests were performed on a data set of letters from the UCI Machine Learning Repository (Murphy and Aha (1994)), which includes 16-dimensional feature vectors for the 26 letters in the English alphabet. The letter images were based on 20 different fonts and each letter within these 20 fonts was randomly distorted to produce 20,000 samples. For each letter, we used 100 samples for training, 250 for validation, and the rest for test (about 400 samples per letter). The parameters of all methods have been chosen using the validation set. Since the test set includes 25 times more negatives than positives, which leads to about 96% classification rate by just classifying all inputs as negative, we used EER as a more faithful measure of performance. Table 2 shows the classification rate at EER , averaged over 26 letters, and the average classification times of the tested classifiers.

The KHHM classifiers improve over the linear SVM for all K, and for $K > 1$ outperforms Adaboost with much shorter classification time. For this data set, kernel SVM outperform all methods. However, the KHHM classifier with $K = 4$ comes fairly close to the performance of the kernel SVM, while its classification time is three magnitudes faster.

Method	Classification rate at EER	Classification time
KHHM $K = 1$	89.32	5.6e-07
KHHM $K = 2$	92.98	1.4e-06
KHHM $K = 3$	93.93	1.5e-06
KHHM $K = 4$	94.48	1.7e-06
Linear SVM	84.87	4.6e-07
RBF kernel SVM	96.47	1.7e-03
AdaBoost	92.26	1.0e-03

Table 2: Letter experiments. K corresponds to the number of hyperplanes used in the hinge-minimax classifier. The times are in sec. AdaBoost uses 100 decision trees.

Method	AUC	classification time
KHHM $K = 1$	88.89	9.8e-05
KHHM $K = 2$	90.99	1.34e-04
Linear SVM	88.20	8.6e-05
RBF kernel SVM	90.77	23.97
RUSBoost	90.76	0.08

Table 3: Scene classification with 300 dim. features. The classification time of RBF kernel SVM is very high, since it chooses about 15,000 SVs from 19850 training examples. The RUSBoost uses 100 decision trees.

5.1.3. LARGE SCALE SCENE RECOGNITION

In this test we used 397 scene categories of the SUN data base, which have at least 100 images per category (Xiao et al. (2010)). We represent the images as BOW of dense HOG features with 300 words. We downloaded the features from the SUN web page[6], containing spatial pyramid of BOWs, and used the bottom layer (the details of the feature extraction can be found in Xiao et al. (2010)). The data is divided into 50 training and 50 test images in 10 folds. Training one-against-all classifiers for 397 categories with 50 training samples per category uses very unbalanced training sets. Thus we defined different weights for positive and negative samples in SVM training and we used RUSBoost (Seiffert et al. (2008)) as an ensemble method (it is designed for skewed data and performed significantly better than AdaBoost on this data set). Note that the KHHM classifier naturally handles imbalanced sets. KHHM classifier with more than two hyperplanes didn't improve the performance. Table 5.1.3 shows the average AUC of the tested methods and their running times.

Using a pyramid of BOWs with the histogram intersection kernel improves over the RBF kernel applied to the bottom layer of the pyramid, but then the dimension of the feature vector increases to 6300. The AUC of the KHHM classifier with $K = 2$ is 92.99% and of histogram kernel is 92.85%. Figure 6 shows the ROCs of the first three categories produced by the KHHM classifier and the histogram intersection kernel SVM classifiers.

6. $http://vision.cs.princeton.edu/projects/2010/SUN/$

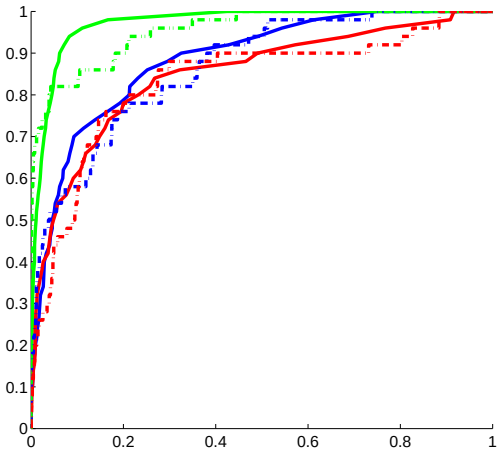

Figure 6: ROCs of the first three categories of the SUN data set, represented by a spatial pyramid of BOWs, obtained from dense HOG. The solid lines correspond to the hinge-minimax classifier, dotted lines correspond to the histogram intersection kernel SVM.

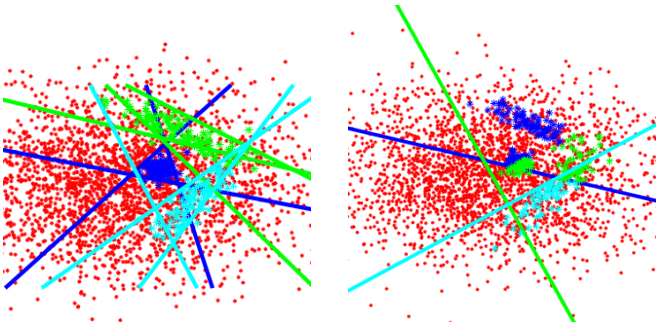

Figure 7: A qualitative comparison of the latent hinge minimax classifier (on the left) to the union of LDA classifiers (on the right).

5.2. Latent Hinge Minimax Classifier

We first show a 2D toy example (Section 5.2.1) to illustrate the ability of the LHM classifier to discover the hidden components in the positive class and to separate each of them from the negative class using a K-hyperplane model.

Then, we compare LHM model to alternative ensembles of hyperplanes (in shallow architectures) on the PASCAL-VOC 2007 dataset (Everingham et al. (2010)) (Section 5.2.2), and show its advantage over those methods and its robustness to the choice of the number of latent components.

5.2.1. SYNTHETIC DATA

A simpler alternative to the LHM model is a two-step algorithm which first finds the structure of the target class by applying some kind of unsupervised learning (e.g, k-means clustering) and then builds a model

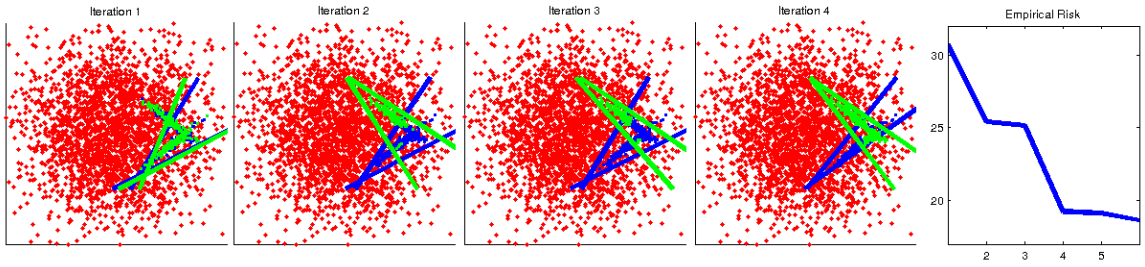

Figure 8: First four iterations of the LHM training on toy example and the corresponding loss convergence.

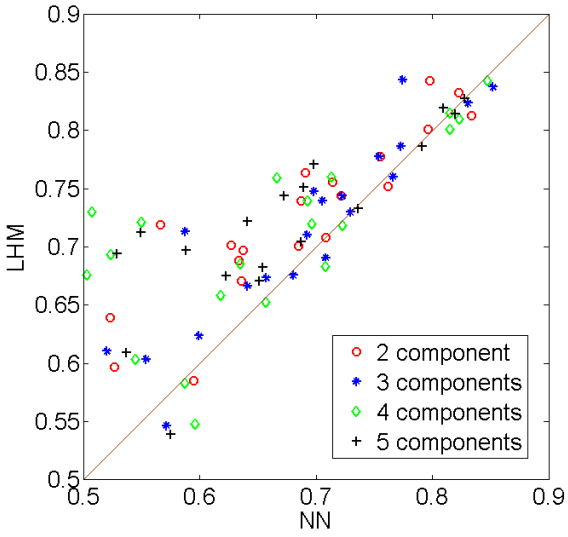

Figure 9: Comparison of the LHM classifier to the equivalent NN for a varying number of hidden components (from 2 to 5) on PASCAL VOC 2007. The points above the diagonal line show the advantage of LHM classifier.

for each component. Such a simple approach was employed in Hariharan et al. (2012) with LDA classifier (Hastie et al. (2001)) trained per cluster. Unless the clusters are very small[7], it relies heavily on the results of the clustering. If an initial clustering is incorrect (as in Figure 7, right), LDA (or any other convex classifier) cannot separate the resulting components from the background without including many false positives. The LHM training finds the underlying structure of the data and the model iteratively, improving both (Figure 7, left). Furthermore, LHM is quite robust to the initial assignment. Figure 8 shows a few iterations and the corresponding loss convergence when the initial assignment of the positive samples to components is chosen at random. Note the LHM training discovers the underlying structure in a 3-4 iterations.

7. as in time consuming exemplar-based approach(Malisiewicz et al. (2011))

LHM	Union of LDAs	NN	KHHM
71.48%	65.17%	67.19%	69.45%

Table 4: The table reports the accuracy at the EER point averaged over 20 classes and different hidden partitions (except for KHHM) on PASCAL VOC-2007 classification task using 80-dimensional HOG features.

5.2.2. ENSEMBLES OF HYPERPLANES

Next, we compared the LHM classifier to alternative ensembles of linear classifiers on PASCAL VOC 2007 dataset (Everingham et al. (2010)). To compare the raw performance of the classifiers we designed the experiment to separate the contribution of the classifier from that of features and the detection system (which usually involves various engineering steps that obscure the actual contribution of the classifier). To this end we used simple features, such as Dalal-Triggs variant of the HOG features (Dalal and Triggs (2005)) with a fixed number of cells (thus keeping the classification problem difficult), and we compared the classification accuracy on the bounding boxes of 20 VOC object categories in test images (instead of running a full detection system).

LDA Union (as a baseline model): We applied k-means clustering on whitened features to find the partition. We then learned an LDA classifier for each cluster in that partition. We varied the number of clusters from 2 to 5.

NN with an architecture equivalent to LHM: We used the model described in Section 4.1 with $K = 2$ and $H = 2, .., 5$, but the weights were initialized at random.

KHHM model This is essentially an LHM model with a single component, thus it is theoretically inferior to LHM. However, we ran this experiment to test the benefits of modeling the hidden structure of the positive class. We varied the number of hyperplanes from 2 to 5.

LHM model: We set the number of hyperplanes in each component to 2 and varied the number of components from 2 to 5. An initial assignment to the components was done using k-means with the Euclidian distance.

All ensembles were trained in one-against-all manner. Similarly to (Hariharan et al. (2012); Osadchy et al. (2012)), we learned the background mean and covariance using bounding boxes from all classes and used them to represent the negative class in LDA union, KHHM, and LHM training. We tested all ensemble classifiers on all bounding boxes from the test set. Table 4 summarizes the accuracy at the EER points of all ensembles averaged over classes and different parameters. It shows that LHM model outperforms all other classifiers. Figure 9 compares LHM to NN on 20 categories (as one-against-all binary classifiers) for varying number of hidden components. The plot shows that LHM outperforms NN independently of the number of components.

5.3. Hinge-Minimax Training in Deep Architecture

In the following experiments, we show that LHM classifier can be combined with CNN via transfer learning. Specifically, we test the LHM classifier on top of the pre-trained CNN feature extraction in imbalanced binary problems and in multi-class tasks with a small number of labeled examples.

We explore the following transfer learning settings. The first setting refers to the **best case** scenario in which the source and the target classification tasks operate on the *same* set of features but differ in the classification problem. The second setting refers to the **worst case** scenario for the transfer learning where the source and the target classification problems *share very little similarity*. The "worst case" scenario is very common in practice, as many classification tasks do not have a large, comprehensive training set (such as ImageNet (Deng et al. (2009)) in object recognition) to be used in transfer learning. No good solution currently exists for such problems.

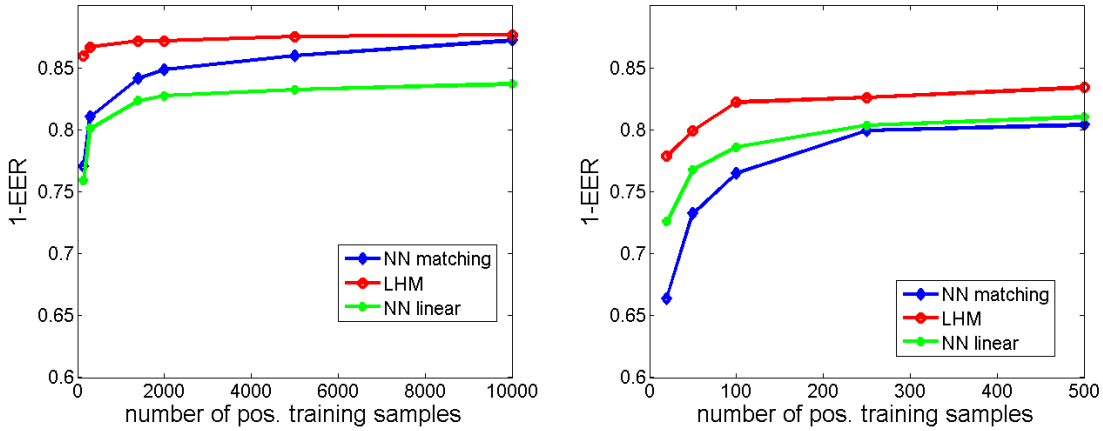

Figure 10: Binary imbalanced classification: left – the "best-case" transfer learning setting, right – the "worst-case" transfer learning setting.

We used the CIFAR-10, composed of 10 categories (airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck) as the source problem. Specifically, we trained the LeNet model implemented in MatConvNet Vedaldi and Lenc (2015) on CIFAR-10. Then we removed the last fully-connected layer and the soft-max and used this trimmed network as a feature extractor which converts images to a 64-dimensional feature vectors.

For the best case transfer learning, we defined a new set of classes by coupling i and $i + 5$ indexes of CIFAR-10 classes. CNN trained on CIFAR-10 maps individual classes to linearly separable sub-spaces, thus using pairs of classes as a target classification problem makes it non-linear. Consequently, we get a new classification problem over the same space of features.

For the worst case transfer learning, we picked a subset of 5 classes (train, bottle, cattle, forest, and sweet peppers) from the CIFAR-100, which do not overlap (in their visual appearance) with the CIFAR-10 categories, to be the target classification task. CIFAR-10 data set is not rich enough to enable learning of features that can be used for an arbitrary category, thus we believe that such setting is especially difficult.

We tested the LHM binary and multi-class classifiers in the best and the worst case transfer learning scenarios and compared their performance to two baselines. One is an NN with a single fully connected layer and the cross-entropy loss (NN linear) and the other is the NN with the architecture matching the LHM model (NN matching). We repeated each experiment 50 times over different random subsets of training samples and random initialization of NN and averaged the results.

5.3.1. BINARY IMBALANCED SETTING

The "Best Case" Transfer Learning: We trained binary classifiers for pairs of classes from CIFAR-10 using imbalanced training sets, in which the negative class included all samples from all other classes (40,000 examples) and the positive class included a varying number of samples (140, 300, 600, 1400, 2000, 5000-all). This resulted in imbalance ratios from 1:256 to 1:4.

LHM model was trained with 2 hidden components and 3 hyperplanes per component. The matching NN mimicked the configuration of LHM model, but the weights were allowed to change in training. Figure 10-left shows the 1-EER (averaged over 5 classification problems) of the LHM classifier and the two NN baselines as a function of the positive training sample size.

The "Worst Case" Transfer Learning for Binary Imbalanced Problems: Since the number of samples per class in CIFAR-100 is significantly smaller, this experiment tests the robustness to imbalanced training data

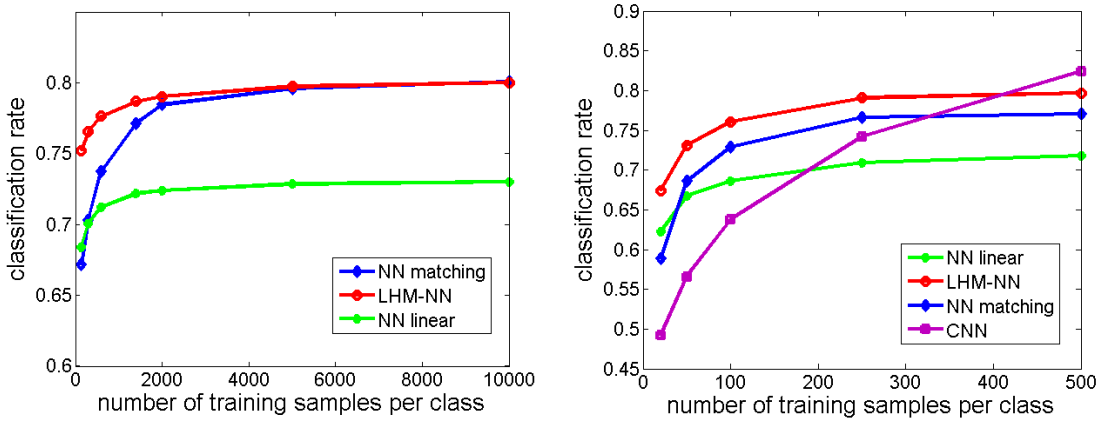

Figure 11: Multi-class classification: left – the "best-case" transfer learning setting, right – "worst-case" transfer learning setting.

and to a small number of examples. We varied the size of the positive training set between 20, 50, 100, 250, 500(all) samples and we used all 2,000 samples of other classes as the negative training set. We compared the LHM model trained with 2 hidden components and 2 hyperplanes per component to NN baselines. Figure 10-right shows the 1-EER of the classifiers averaged over 5 classification problems as a function of the positive training set size.

5.3.2. MULTI-CLASS SETTING

The "Best Case" Transfer Learning: We mapped the LHM binary classifiers trained for 5 pairs of categories to a multi-class NN as described in Section 4.2. We fine-tuned the weights with a very fast training (just a handful of epochs, while training from scratch requires two orders of magnitude more training epochs). Figure 11-left shows the accuracy of the LHM models mapped to a multi-class NN (LHM-NN) with the two baseline NNs as a function of the size of the training set.

The "Worst Case" Transfer Learning for Multi-Class Problems: We mapped the LHM binary classifiers trained for the 5 categories from CIFAR-100 (using CIFAR-10 features) to a multi-class NN and fine-tuned the weights with a small number of epochs.

To test the complexity of the transfer learning problem we also trained a CNN (LeNet model implemented in MatConvNet (Vedaldi and Lenc (2015))) on the target problem. We hoped that due to the small size of the target classification problem, 500 training examples per class would yield relatively good accuracy. Figure 11-right compares the accuracy of LHM-NN, two baseline NNs, and CNN (trained from scratch) as a function of the training sample size. It shows that CNN trained on the target problem is indeed the best as it succeeds to learn features specific for the task, but its accuracy drops very abruptly when the number of training samples becomes smaller. This suggests that when the number of training examples is small, using transfer learning even in a such difficult setting is a better solution than training a CNN from scratch.

The results in Figures 10 and 11 show that the NN models either heavily overfit when the number of training samples is small (NN matching) or they are not expressive enough when the number of training samples increases (NN linear). LHM classifiers are expressive enough to learn from a large set of examples and are more robust to overfitting when the number of examples is small.

6. Training Efficiency

Another advantage of LHM-NN is its training efficiency. A class-specific LHM model converges in 5-10 iterations. Its training time primarily depends on the number of positive samples and the dimension. The negative samples are used to estimate the mean and covariance of the background. The initial estimation (which involves a large number of samples) is done only once and used for all classes. Since the probability of the negative class is evaluated inside the positive region using false positives, the number of which drops very fast, the estimation time of the mean and covariance during the training is negligible. Training of a binary classifier per class is independent of other classes, thus their training can be done in parallel. Finally, the fine-tuning of the multi-class network after mapping is very fast, due to the initialization of all layers (using supervized learning): feature extraction layers with pre-trained CNN and classifier's layers with LHM models.

The LHM-NN is also beneficial for the problems in which classes are dynamically added or removed from the classification task. Adding a class requires training a single binary classifier and fast fine-tuning; removing a class requires only fine-tuning.

7. Conclusions and Future Work

We proposed an efficient method for learning an intersection of finite number of hyperplanes which combines the hinge-risk (for the small number of positive data) with the background risk, based on the "minimax bound" (for a large number of negative data points) and derived a generalization bound for the mixed risk. We showed that the proposed classifier yields results comparable to the popular non-linear classifiers, but at much lower (order of magnitude) computational cost of classification.

We generalized this model to a non-convex classifier (Latent Hinge-Minimax classifier), which discovers the hidden components in the positive class and separates them from the negative class with the intersections of positive half spaces. The main advantage of this classifier is its ability to incorporate unlabeled data in training which improves the robustness to imbalanced sets.

We showed that for multi-class tasks, class-specific LHM models can be mapped to a multi-class NN with matching architecture requiring only a few iterations of fine-tuning. Finally, the showed that LHM architecture can be integrated with CNN features via transfer learning. The entire training procedure is very efficient. Our experiments showed that such classifiers are much more robust to the number of labeled training samples than the equivalent NNs.

This work cay be extended in various directions. A lower Rademacher complexity was shown for the k-fold maxima of hyperplanes in Kontorovich (2018). We plan to extend this result to the maximum over hinge losses and improve the generalization bound for the positive sample. Structured output learning have had an impact on machine vision and can be applied to this framework while improving the multiclass procedure. Another direction is devising a unified probabilistic framework to include both the hinge-loss and background-loss. Also, extensions of Marshall-Olkin theorem to non-convex sets might have a significant impact on robust deep learning methods.

References

Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *The Journal of Machine Learning Research*, 3:463–482, 2003.

R. E. Bellman. *Introduction to Matrix Analysis 2nd ed.* New York: McGraw-Hilla, 1970.

Olivier Bousquet, Stéphane Boucheron, and Gábor Lugosi. Introduction to statistical learning theory. In *Advanced Lectures on Machine Learning*, pages 169–207. Springer, 2004.

Emmanuel J Candes and Yaniv Plan. A probabilistic and ripless theory of compressed sensing. *Information Theory, IEEE Transactions on*, 57(11):7235–7254, 2011.

Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005.

Amit Daniely, Nati Linial, and Shai Shalev-Shwartz. From average case complexity to improper learning complexity. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 441–448, 2014.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition,CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 580–587, 2014.

Bharath Hariharan, Jitendra Malik, and Deva Ramanan. Discriminative decorrelation for clustering and classification. In *Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part IV*, pages 459–472, 2012.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer New York Inc., 2001.

Jean Honorio and Tommi Jaakkola. {Tight Bounds for the Expected Risk of Linear Classifiers and PAC-Bayes Finite-Sample Guarantees}. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, pages 384–392, 2014.

Sham M Kakade, Karthik Sridharan, and Ambuj Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In *Advances in neural information processing systems*, pages 793–800, 2008.

Adam R. Klivans and Alexander A. Sherstov. Cryptographic hardness for learning intersections of halfspaces. *Journal of Computer and System Sciences*, 75(1):2 – 12, 2009.

Aryeh Kontorovich. Rademacher complexity of k-fold maxima of hyperplanes. 2018. URL `https://www.cs.bgu.ac.il/~karyeh/rademacher-max-hyperplane.pdf`.

Ilja Kuzborskij, Francesco Orabona, and Barbara Caputo. From N to N+1: multiclass transfer incremental learning. In *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, pages 3358–3365, 2013.

Gert R.G. Lanckriet, Laurent El Ghaoui, Chiranjib Bhattacharyya, and Michael I. Jordan. A robust minimax approach to classification. *J. Mach. Learn. Res.*, 3:555–582, 2003.

M. Ledoux and M. Talagrand. *Probability in Banach Spaces: Isoperimetry and Processes*. A Series of Modern Surveys in Mathematics Series. Springer, 1991. ISBN 9783540520139.

Tomasz Malisiewicz, Abhinav Gupta, and Alexei A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pages 89–96, 2011.

Albert W. Marshall and Ingram Olkin. Multivariate chebyshev inequalities. *Ann. Math. Statist.*, 31(4):1001–1014, 1960.

P. Murphy and D. Aha. Uci repository of machine learning databases. *Tech. rep., U. California, Dept. of Information and Computer Science*, 1994.

M. Osadchy, D. Keren, and B. Fadida-Specktor. Hybrid classifiers for object classification with a rich background. In *ECCV (5)*, pages 284–297, 2012.

Margarita Osadchy, Daniel Keren, and Dolev Raviv. Recognition using hybrid classifiers. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(4):759–771, 2016.

Chris Seiffert, Taghi M. Khoshgoftaar, Jason Van Hulse, and Amri Napolitano. Rusboost: Improving classification performance when training data is skewed. In *ICPR*, pages 1–4, 2008.

Vladimir Vapnik. *The nature of statistical learning theory*. Springer Science & Business Media, 2000.

Andrea Vedaldi and Karel Lenc. MatConvNet: Convolutional Neural Networks for MATLAB. In *Proc. of the 23rd Annual ACM Conference on Multimedia Conference, Brisbane*, pages 689–692, 2015. doi: 10.1145/2733373.2807412.

J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. *CVPR*, pages 3485–3492, 2010.

Tong Zhang. Covering number bounds of certain regularized linear function classes. *The Journal of Machine Learning Research*, 2:527–550, 2002.