Proceedings of Machine Learning Research vol 65:1–17, 2017

Efficient Co-Training of Linear Separators under Weak Dependence

Avrim Blum AVRIM@CS.CMU.EDU
Carnegie Mellon University, Computer Science Department

Yishay Mansour MANSOUR@TAU.AC.IL
Tel Aviv University, Blavatnik School of Computer Science

Abstract

We develop the first polynomial-time algorithm for co-training of homogeneous linear separators under *weak dependence*, a relaxation of the condition of independence given the label. Our algorithm learns from purely unlabeled data, except for a single labeled example to break symmetry of the two classes, and works for any data distribution having an inverse-polynomial margin and with center of mass at the origin.

Keywords: co-training, unsupervised learning, linear classifier.

1. Introduction

Co-training, also known as multi-view learning, is an approach to learning from primarily unlabeled data when examples x can be partitioned into two "views" x_1, x_2 that each separately are sufficient to determine the example's classification. The high-level idea is that rather than directly trying to maximize the agreement of one's hypothesis h with the target function f, one instead aims to learn *two* hypothesis h_1 and h_2, one over each view, and then to maximize agreement of the hypotheses with each other. The key point is that this can be done over *unlabeled* data. Under appropriate conditions (Blum and Mitchell, 1998; Abney, 2002; Balcan and Blum, 2010), and with a small amount of labeled data to "break symmetry" and provide a small amount of signal, maximizing agreement between views will also maximize agreement with the target function. That is, any pair of hypotheses that closely agree with each other, and satisfy other simple conditions such as being roughly balanced in their predictions and agreeing with a small labeled sample, will also closely agree with the target function. There has been substantial work, both theoretical and applied, on algorithms and analysis of multi-view learning for a wide range of settings (e.g., Blum and Mitchell (1998); Dasgupta et al. (2001); Abney (2002); Levin et al. (2003); Wan (2009); Chaudhuri et al. (2009); Balcan and Blum (2010); Kumar and Daumé (2011); Kiritchenko and Matwin (2011); Liu et al. (2014); Do Thi et al. (2016)).

One challenge with this approach to semi-supervised learning, however, is that the optimization problem is often computationally intractable even when the analogous optimization problem for the fully supervised case would be easy. For example and most notably, while finding a linear separator consistent with a set of labeled examples is easy when one exists, finding *two* linear separators, one over each view, that agree with each other over unlabeled examples and agree with the target over a small labeled sample is NP-hard. On the other hand, polynomial-time algorithms for learning linear separators in the multi-view setting have been developed under the assumption that data satisfies the condition of *independence given the label* (Blum and Mitchell, 1998; Balcan and Blum, 2010). This means that each example can be thought of as consisting of two independently drawn positive

sub-examples (x_1 and x_2) or two independently drawn negative sub-examples, from their respective domains X_1 and X_2. Essentially, this assumption allows one to convert a weak-predictor over one view into a labeler corrupted by *random classification noise* for the other view. However, it has been a challenge to relax this condition while maintaining computational efficiency.

In this work, we develop the first polynomial-time algorithm for co-training of linear separators under a relaxation of the independence condition known as *weak dependence* (Abney, 2002). Under weak dependence, an example (x_1, x_2) is produced in the following manner. First, x_1 is drawn at random according to some fixed but unknown distribution over its domain X_1. Then, with some probability λ, x_2 is drawn independently from x_1 given the label; however, with probability $1 - \lambda$, x_2 is drawn from X_2 in an *arbitrarily-correlated* way, though still subject to having the same label as x_1. Thus, the selection of x_2 can be viewed as partly random and partly adversarial. Information-theoretically, it has long been known that this setting is solvable in the sense that maximizing agreement between views subject to satisfying a few simple conditions will indeed lead to a low error hypothesis (Abney, 2002). However, the efficient algorithms used to solve the case of independence given the label break down, because the induced noise no longer looks like random classification noise but instead like malicious misclassification noise (Sloan, 1995), for which no efficient algorithms are known. In this work, we give the first polynomial-time procedure for learning linear separators from primarily unlabeled data under data with weak dependence. More specifically, as in the algorithm of Balcan and Blum (2010) for the setting of independence given the label, our algorithm requires polynomially many unlabeled examples and a single labeled example. However, our algorithm does not produce a linear separator as its hypothesis.

One caveat: we make two assumptions that when combined are not without loss of generality. They are: (1) the target linear separators f_i for each view $i \in \{1, 2\}$ go through the origin, and (2) the underlying distributions for each view have center of mass at the origin. While we are able to relax these somewhat, we do not know how to remove them completely. We additionally assume that there is a $1/poly$ margin.

The paper is organized as follows. The model is detailed in Section 2. Section 3 has an overview of our algorithm, while the details are in Section 4. The analysis of the algorithm is in Section 5.

2. Model

We consider the following setting. There is a domain $X = X_1 \times X_2$ where $X_1 = \mathbb{R}^{d_1}$ and $X_2 = \mathbb{R}^{d_2}$. That is, each example is of the form $x = (x_1, x_2)$ where $x_1 \in X_1$ and $x_2 \in X_2$. There is also an unknown target f that classifies points in X as either positive or negative; i.e., $f : X \to \{+1, -1\}$. We make the co-training assumption that each component, or "view", of an example is sufficient for classification. Formally, there exist two functions f_1 over X_1 and f_2 over X_2 and an (unknown) distribution D over X such that for any (x_1, x_2) in the support of D we have $f_1(x_1) = f(x_1, x_2) = f_2(x_2)$. The classification $y = f(x_1, x_2)$ is never observed by the learner. (At the end we allow for a single labeled example to break symmetry.)

For each classification $y \in \{+1, -1\}$, let D_y denote the distribution D conditioned on the example having label y, and let α_y be the probability that a random example is of label $y \in \{+1, -1\}$. So, $D = \alpha_{+1}D_{+1} + \alpha_{-1}D_{-1}$, where $\alpha_{-1} + \alpha_{+1} = 1$. For simplicity we will assume that both α_y are bounded away from 0 by constants, e.g., $\alpha_{+1}, \alpha_{-1} \in [0.1, 0.9]$. Let $D_{y,i}$ be the marginal probability over x_i of distribution D_y where the label is $y \in \{+1, -1\}$ and the view is $i \in \{1, 2\}$.

We assume access to unlabeled data only, and our goal will be to learn a hypothesis h such that either $h \approx f$ or $h \approx -f$ (at that point, we would just need one random labeled example to determine whether to output h or $-h$). We will denote our (unlabeled) training examples as $(x_1^{(i)}, x_2^{(i)})$.

2.1. Assumptions about the target function

We will assume that in both views, the target function belongs to the class of homogeneous linear separators. Namely, there exist w_1, w_2 such that $f_1(x_1) = sign(x_1^\top w_1)$ and $f_2(x_2) = sign(x_2^\top w_2)$, where $sign(z) = +1$ if $z \geq 0$ and $sign(z) = -1$ if $z < 0$. We assume that $\|w_1\| \leq 1, \|w_2\| \leq 1$ and that $\|x_1\| \leq 1, \|x_2\| \leq 1$ for all (x_1, x_2) in the support of D.

We say that a classifier f has a γ-*margin* under distribution D if for any $(x_1, x_2) \in support(D_y)$ we have $y(x_1^\top w_1) \geq \gamma$ and $y(x_2^\top w_2) \geq \gamma$. Namely, for (x_1, x_2) such that $f(x_1, x_2) = +1$ we have $x_1^\top w_1 \geq \gamma$ and $x_2^\top w_2 \geq \gamma$, and for (x_1, x_2) such that $f(x_1, x_2) = -1$ we have $x_1^\top w_1 \leq -\gamma$ and $x_2^\top w_2 \leq -\gamma$. We assume there exists a γ-margin for some $\gamma \geq 1/poly(d_1 d_2)$. Putting these together we have:

Assumption 1 *We assume that the target function f is a homogeneous linear separator in each view and there is a $\gamma = 1/poly(d_1 d_2)$ margin.*

2.2. Assumptions about the distribution

We say that the distribution D satisfies λ-*weak dependence* (Abney, 2002), if for each label $y \in \{+1, -1\}$, for every $(x_1, x_2) \in support(D_y)$ we have that

$$D_y(x_1, x_2) \geq \lambda D_{y,1}(x_1) D_{y,2}(x_2) \ .$$

Namely, both D_{+1} and D_{-1} can be viewed as with probability λ, sampling independently from the marginal distributions, and with probability $1 - \lambda$, sampling in some arbitrarily dependent manner.

We in fact can relax this to allow the weak dependence assumption to be limited to only a subset of non-negligible probability of the domain X_1. Given a set $X_1' \subseteq X_1$ such that $\Pr_{x_1 \sim D_{y,1}}[x_1 \in X_1'] \geq \eta$ for $y \in \{+1, -1\}$, we say that the distribution D satisfies (λ, X_1')-*weak dependence* if for each $y \in \{+1, -1\}$, for every $(x_1, x_2) \in support(D_y) \cap (X_1' \times X_2)$ we have that

$$D_y(x_1, x_2) \geq \lambda D_{y,1}(x_1) D_{y,2}(x_2) \ .$$

We say that a distribution is *homogeneous* if $E_D[x] = \mathbf{0}$, namely, the mean of the distribution is the origin. We can relax it by only requiring that the origin can be described as a mixture of the positive and negative distribution. Formally,

Definition 2 *A distribution D is (ρ, X_1')-semi homogeneous, for $X_1' \subseteq X_1$ and $\rho = (\rho_{+1}, \rho_{-1}) \in (0,1)^2$ s.t. $\rho_{+1} + \rho_{-1} = 1$, if $\rho_{+1} E_{x_1 \sim D_{+1,1}}[x_1 | x_1 \in X_1'] + \rho_{-1} E_{x_1 \sim D_{-1,1}}[x_1 | x_1 \in X_1'] = \mathbf{0}$.*

We make the following assumption about the distribution D.

Assumption 3 *We assume that there exists a $X_1' \subseteq X_1$ such that $\Pr_{x_1 \sim D_{y,1}}[x_1 \in X_1'] \geq \eta$ for $y \in \{+1, -1\}$ and there exist $\rho = (\rho_{+1}, \rho_{-1}) \in (0,1)^2$ with $\rho_{+1} + \rho_{-1} = 1$ such that the distribution D is (ρ, X_1')-semi homogeneous and satisfies (λ, X_1')-weak dependence.*

2.3. Notation

We will use the notation $VC(d, \epsilon, \delta)$ to specify the number of examples required to guarantee that a hypothesis class H of VC dimension d would have, with probability $1 - \delta$, all hypotheses of zero empirical error having true error at most ϵ. For the most part we will ignore the dependency on confidence δ and use $VC(d, \epsilon)$, since dependency on δ is only a lower order term.

3. Algorithm: overview

We start by giving a high level intuition for the algorithm, and the challenges that the analysis is faced with. We assume that we are in the realizable setting. Namely, that each function f_1 and f_2 is a homogeneous hyperplane and gives a perfect labeling. This implies that for any pair $(x_1^{(i)}, x_2^{(i)})$ in the support of D we have that $f_1(x_1^{(i)})f_2(x_2^{(i)}) = 1$, since f_1 and f_2 agree on the label $y \in \{-1, +1\}$. Since f_1 and f_2 are homogeneous hyperplanes, we have that

$$sign(w_1^\top x_1^{(i)})sign(w_2^\top x_2^{(i)}) = +1.$$

Recall that we also assume that each hyperplane w_i classifies all points in the support of the distribution D correctly by margin γ. Therefore,

$$(w_1^\top x_1^{(i)})^\top (w_2^\top x_2^{(i)}) = (x_1^{(i)})^\top (w_1 w_2^\top) x_2^{(i)} = (x_1^{(i)})^\top W x_2^{(i)} \geq \gamma^2$$

where $W = (w_1 w_2^\top)$ is a rank 1 symmetric matrix and we use the γ-margin assumption.

Our algorithm will try to reconstruct an approximation to W from a sample of m_1 points. We have the following feasibility convex program, where M is a $d_1 \times d_2$ real-valued matrix.

$$\begin{aligned} &\text{Find a matrix } M \text{such that} \\ \forall i \in [1, m_1]: \quad &(x_1^{(i)})^\top M x_2^{(i)} \geq \gamma^2 \\ &\|M\|_F^2 \leq 1 \ , \end{aligned}$$

where $\|M\|_F^2 = \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} M_{i,j}^2$ is the squared Forbenius norm of matrices. By our γ-margin assumption we are guaranteed that the convex program is feasible.

Our first step in the algorithm will be to take a large sample from D and find a matrix M that solves the above convex program. The main challenge in the algorithm is how to use the matrix M in order to predict labels. When given a test example (x_1, x_2), we know that both x_1 and x_2 have the same label, the real issue is to decide which pairs will get the positive label (i.e., $+1$) and which will get the opposite negative label (i.e., -1).

Ideally, we would like to claim that if we take two pairs (x_1, x_2) and (x_1', x_2') and have $x_1^\top M x_2' < 0$ then the two pairs have different labels. However, in our training we never observe such pairs! The main part of the algorithm and the analysis is aimed at overcoming this hurdle.

To gain intuition, consider the case that D satisfies full independence given the label, and is uniform in each view. In this case when we consider two examples with the same label, say $(x_1^{(i)}, x_2^{(i)})$ and $(x_1^{(j)}, x_2^{(j)})$, their probability is equal to that of $(x_1^{(i)}, x_2^{(j)})$ and $(x_1^{(j)}, x_2^{(i)})$. This implies that the matrix M we learn will have, with high probability, $(x_1^{(i)})^\top M x_2^{(j)} > 0$ and similarly $(x_1^{(j)})^\top M x_2^{(i)} > 0$, since those examples are equally probable under the uniform distribution.

Now consider two examples of different labels, $(x_1^{(i)}, x_2^{(i)})$ and $(x_1^{(j)}, x_2^{(j)})$. Since the hyperplanes are homogeneous we know that $(x_1^{(i)}, x_2^{(i)})$ and $(-x_1^{(j)}, -x_2^{(j)})$ have the same label. Also, $(x_1^{(j)}, x_2^{(j)})$ and $(-x_1^{(j)}, -x_2^{(j)})$ are equally likely under the uniform distribution. These together imply that with high probability $(x_1^{(i)})^\top M(-x_2^{(j)}) > 0$ or equivalently, $(x_1^{(i)})^\top M(x_2^{(j)}) < 0$.

Our goal is to recover a predictor that with high probability will have a low error rate. Assume we have one positive example $(x_1^{(i)}, x_2^{(i)})$ and one negative $(x_1^{(j)}, x_2^{(j)})$ (by sampling two examples, this will happen with a constant probability, i.e., $2\alpha_{+1}\alpha_{-1}$). We can take an additional sample of size m_2 and use M and $x_1^{(i)}$ to classify the points. Namely, for an example $(x_1^{(k)}, x_2^{(k)})$ if $(x_1^{(i)})^\top M(x_2^{(k)}) > 0$ label $x_1^{(k)}$ positive, and otherwise label it negative. Assume that we have learned M to some error $\epsilon < 1/m_2$, then we will have a constant probability that *all* our predicted labels are correct. We can now learn a separating hyperplane h_1. In a similar way we can learn a separating hyperplane h_2. With constant probability we learn a good predictor, we just need to verify now that what we have is a good predictor.

In order to verify that a pair of hypotheses (h_1, h_2) are a good predictor, we can do the following. We first test that the probability that $h_1 \neq h_2$ is small. Then we test that the probability that $h_1(x) = +1$ is about α_{+1}. The first test guarantees that the two hypotheses will be consistent (with high probability). However, two hypothesis can be consistent by simply always predicting $+1$; for this, the second test verifies that they give the right proportion of $+1$ and -1. (Note that if $\alpha_{+1} = \alpha_{-1}$ we cannot really distinguish positive and negatives. Also, we do not really need to know α_{+1}: it is enough to know that α_{+1} is in some bounded range, say $[0.1, 0.9]$.)

The above outline assumed that the distribution D is uniform in each view and satisfies independence given the label. When we move to our more general distributions, a challenge is that when you flip an example through the origin, the result may not even be in the support of D, so one cannot directly conclude anything about how M behaves on it. Additionally, we wish to allow for much more dependence between the views, namely (λ, X_1')-weak dependence. Both of these make our setting more challenging.

4. Algorithm

We start by describing our algorithm. The algorithm is composed of two main parts. The first part is a function `GetMatrix` that computes the matrix M. The second part is a function `GetHypo` for computing a pair of hypotheses (h_1, h_2). Formally,

$M \leftarrow$ `GetMatrix`(m_1).
$(h_1, h_2) \leftarrow$ `GetHypo`$(M, m_2, m_3, limit_2)$
`RETURN` (h_1, h_2)

We will specify the setting of the parameters m_1, m_2, m_3 and $limit_2$ with the appropriate functions.

The function `GetMatrix` draws m_1 examples, where $m_1 = VC(d_1 d_2, \epsilon_1 \alpha_{+1}\alpha_{-1})$ and $\epsilon_1 = (\gamma^2 \lambda \eta)/(6m_2)$ (the parameter m_2 will be specified later). Given the sample of the m_1 examples, it writes a convex program to recover a matrix M with margin γ^2 with respect to all the examples. Formally,

Function `GetMatrix`(m_1).
Sample m_1 examples $(x_1^{(i)}, x_2^{(i)})$.

Solve the quadratic program:
Find $d_1 \times d_2$ matrix M such that
$(x_1^{(i)})^\top M x_2^{(i)} \geq \gamma^2$
$\sum_{i=1}^{d_1} \sum_{j=1}^{d_2} M_{i,j}^2 \leq 1$.
RETURN M.

We will later show that with high probability the matrix M is such that points drawn from X_1' and X_2, independently, but with the same label, are likely to have a margin of at least $\gamma^2/2$.

Given the matrix M, the function GetHypo needs to find a pair (h_1, h_2) which will (hopefully) have an error of at most ϵ. The function runs in iterations until it finds a suitable pair (h_1, h_2). The function GetHypo1 tries to find an accurate hypothesis h_1. In the analysis we show that it has a non-negligible success probability. Given a candidate h_1, the function GetHypo2 finds a suitable hypothesis h_2. If some hypothesis h_2 was found (i.e., $h_2 \neq \perp$) the function verify tests the pair (h_1, h_2). Once verify succeeds, the program terminates and returns the pair (h_1, h_2). (We defer the parameters specification to later.) Formally,

Function GetHypo$(M, m_2, m_3, limit_2, \epsilon_3, \epsilon_4)$
Pass$\leftarrow$ FALSE
REPEAT
$h_1 \leftarrow$ GetHypo1$(M, m_2, limit_2)$
$h_2 \leftarrow$ GetHypo2(h_1, m_3)
IF $h_2 \neq \perp$ THEN
Pass $\leftarrow$ verify$(h_1, h_2, m_4, \epsilon_3, \epsilon_4)$
UNTIL Pass = TRUE.
RETURN (h_1, h_2).

The function GetHypo1 needs to produce a candidate h_1 hypothesis. It uses as a subroutine large_sample which returns a subset of examples of size at least $limit_2$ out of m_2 examples it samples, where $m_2 = \Theta(limit_2/\gamma^2)$, $limit_2 = VC(d_1, \epsilon_2)/\lambda$ and $\epsilon_2 = \frac{1}{m_3}$ (m_3 will be specified later). We rather arbitrarily assume that the first call returns a set of only positive examples and the second call a set of only negative examples. In the analysis we show that this occurs with non-negligible probability. When this happens, we can find a separating hyperplane h_1 and return it. Formally,

Function GetHypo1$(M, m_2, limit_2)$
REPEAT
$S_{+1} \leftarrow$ large_sample$(M, m_2, limit_2)$.
$S_{-1} \leftarrow$ large_sample$(M, m_2, limit_2)$.
UNTIL Finding h_1 separating S_{+1} from S_{-1}.
RETURN h_1

The function large_sample samples a point (x_1, x_2) and a sample of size m_2 of $(x_1^{(i)}, x_2^{(i)})$. It considers examples $(x_1^{(i)}, x_2^{(i)})$ such that $x_1^\top M x_2^{(i)} < 0$, which are candidates to have the opposite label of x_1, and collects their first view $x_1^{(i)}$. It continues the process until it finds a large enough such subset, namely of size at least $limit_2$, and returns it. Formally,

Function `large_sample`$(m_2, limit_2)$
`REPEAT`
  sample one example (x_1, x_2)
  sample m_2 examples $(x_1^{(i)}, x_2^{(i)})$.
  Let $S' = \{x_1^{(i)} : x_1^\top M x_2^{(i)} < 0\}$.
`UNTIL` $|S'| \geq limit_2$
`RETURN` S'

The function `GetHypo2` needs to produce a candidate h_2 given h_1. It samples m_3 examples, where $m_3 = VC(d_2, \epsilon_3)$ and $\epsilon_3 = \epsilon\eta\lambda/60$. It uses the hypothesis h_1 to label the sample, and then searches for a hypothesis h_2 that agrees with the labeling. The basic goal is that if h_1 is highly accurate, then we should be able to recover a hypothesis h_2. Formally,

Function `GetHypo2`(h_1, m_3)
  Sample m_3 examples $(x_1^{(i)}, x_2^{(i)})$.
  Let $T_{+1} = \{x_2^{(i)} : h_1(x_1^{(i)}) = +1\}$ and $T_{-1} = \{x_2^{(i)} : h_1(x_1^{(i)}) = -1\}$.
  Find h_2 separating T_{+1} from T_{-1}.
  If no such h_2 set $h_2 \leftarrow \perp$
`RETURN` h_2.

The function `verify` receives a pair (h_1, h_2) and tests whether it is an adequate pair. This is done by sampling $m_4 = \Theta(\frac{1}{\epsilon_4^2}\log(2/\delta))$ and performing two tests. The first is that the two hypothesis almost always agree, i.e., disagree on at most an $\epsilon_3 + \epsilon_4$ fraction, where $\epsilon_4 = \epsilon_3$. The second is that for each label h_2 gives approximately the correct probability. Formally,

Function `verify`$(h_1, h_2, m_4, \epsilon_3, \epsilon_4)$
Sample m_4 examples $(x_1^{(i)}, x_2^{(i)})$.
Let $S = \{(x_1^{(i)}, x_2^{(i)}) : h_1(x_1^{(i)}) \neq h_2(x_2^{(i)})\}$.
Let $S_{2,y} = \{(x_1^{(i)}, x_2^{(i)}) : h_2(x_2^{(i)}) = y\}$
`IF` $(|S|/m_4 \leq \epsilon_3 + \epsilon_4)$ and $(\alpha_y - 3\epsilon_4 \leq |S_{2,y}|/m_4$ for $y \in \{+1, -1\})$
`THEN test`$\leftarrow$`TRUE ELSE test` $\leftarrow$`FALSE`
`RETURN test`

The pseudo-code of all the program appears in Appendix A.

5. Analysis

5.1. Analysis of `GetMatrix`

We start by analyzing the function `GetMatrix` which returns the matrix M. Clearly, the running time of `GetMatrix` is polynomial, since it solves a feasible convex program. We need to analyze the properties of the matrix returned by `GetMatrix`.

We claim that on unseen examples we are likely to have similar margin. (Note that on W we are guarantee to have margin at least γ^2, but for M it will follow from the generalization bounds of linear classifiers.)

Lemma 4 *A sample of size* $m_1 = VC(d_1 d_2, \epsilon_1 \alpha_{+1} \alpha_{-1}, \delta)$ *guarantees that with probability at least* $1 - \delta$ *we have:*

$$\Pr_{(x_1,x_2)\sim D_y}[x_1^\top M x_2 < \gamma^2/2] < \epsilon_1$$

where $y \in \{+1, -1\}$.

Proof Since $D(x) = \alpha_{+1} D_{+1}(x) + \alpha_{-1} D_{-1}(x)$, we have that

$$\Pr_{(x_1,x_2)\sim D_{+1}}[x_1^\top M x_2 < \gamma^2/2] \leq \frac{1}{\alpha_{+1}} \Pr_{(x_1,x_2)\sim D}[x_1^\top M x_2 < \gamma^2/2];,$$

and similarly for D_{-1},

$$\Pr_{(x_1,x_2)\sim D_{-1}}[x_1^\top M x_2 < \gamma^2/2] \leq \frac{1}{\alpha_{-1}} \Pr_{(x_1,x_2)\sim D}[x_1^\top M x_2 < \gamma^2/2] \ .$$

The value of $x_1^\top M x_2$ can be viewed as a dot-product in $d_1 d_2$ dimensions between a weight vector with entries M_{ij} and an example vector with entries $x_{1i} x_{2j}$. Using the VC dimension generalization bound we have that

$$\Pr_{(x_1,x_2)\sim D}[x_1^\top M x_2 < \gamma^2/2] \leq \epsilon_1 \alpha_{+1} \alpha_{-1} \ ,$$

which implies the lemma. ■

We remark that an alternative bound can be derived for $m_1 = \Theta(\frac{1}{\epsilon_1 \gamma^4 \alpha_{+1} \alpha_{-1}})$ based on a generalization bound using the margin γ.

We now claim that since the distribution D satisfies (λ, X_1')-weak dependence, even if we sample from the marginal distributions we will get a similar bound.

Lemma 5 *Assume that* D *satisfies* (λ, X_1')*-weak dependence as in Assumption 3. Let* M *be the matrix output by* `GetMatrix`$(m_1, \epsilon_1, \delta)$*. Then, with probability* $1 - \delta$*, for* $y \in \{+1, -1\}$.

$$\Pr_{x_1\sim D_{y,1}, x_2\sim D_{y,2}}[x_1^\top M x_2 < \gamma^2/2 | x_1 \in X_1'] < \frac{\epsilon_1}{\lambda \eta}$$

Proof Since D satisfies (λ, X_1')-weak dependence,

$$\lambda \Pr_{x_1\sim D_{y,1}, x_2\sim D_{y,2}}[x_1^\top M x_2 < \gamma^2/2 | x_1 \in X_1'] \leq \Pr_{(x_1,x_2)\sim D_y}[x_1^\top M x_2 < \gamma^2/2 | x_1 \in X_1']$$

Clearly we have that,

$$\Pr_{(x_1,x_2)\sim D_y}[x_1 \in X_1'] \Pr_{(x_1,x_2)\sim D_y}[x_1^\top M x_2 < \gamma^2/2 | x_1 \in X_1'] \leq \Pr_{(x_1,x_2)\sim D_y}[x_1^\top M x_2 < \gamma^2/2].$$

From Lemma 4, given our setting of m_1, with probability $1 - \delta$, for each $y \in \{-1, +1\}$, we have

$$\Pr_{(x_1,x_2)\sim D_y}[x_1^\top M x_2 < \gamma^2/2] < \epsilon_1$$

Since distribution D satisfies Assumption 3, it implies that $\Pr_{(x_1,x_2)\sim D_y}[x_1 \in X_1'] \geq \eta$ and the lemma follows. ■

It will be convenient to call matrices that satisfies Lemma 5, *good* matrices, and for the most part we will assume that our matrix M is a good matrix.

Definition 6 *We call a matrix* M good *if for* $y \in \{+1, -1\}$.

$$\Pr_{x_1\sim D_{y,1}, x_2\sim D_{y,2}}[x_1^\top M x_2 < \gamma^2/2 | x_1 \in X_1'] < \frac{\epsilon_1}{\lambda \eta} \ .$$

5.2. Analysis `large_sample`

We start with defining the inputs we wish to target. Those are inputs for which the first view, x_1, has the property that there is a non-negligible probability that a random second view of the opposite label, x_2', will cause the bilinear form with M to be negative, i.e., $x_1^\top M x_2' < 0$.

Definition 7 *We say $x_1 \in support(D_{-y,1})$ is $(\xi, M, +y)$-*heavy *if $\Pr_{x_2' \sim D_{+y,2}}[x_1^\top M x_2' < 0] \geq \xi$.*

The next lemma shows that if the distribution is semi-homogeneous, then with high probability, the expectation of $x_1^\top M x_2'$ is noticeably negative over $x_1 \in X_1'$ and x_2' of opposite label from x_1.

Lemma 8 *Assuming that D is (ρ, X_1')-semi homogeneous and has margin γ, then for a good matrix M, we have:*

$$E_{x_1 \sim D_{-y,1}, x_2' \sim D_{y,2}}[x_1^\top M x_2' | x_1 \in X_1'] \leq \frac{-\gamma^2}{4} \frac{\rho_{-y}}{\rho_y}.$$

Proof Since the matrix M is good, for $\epsilon_1 \leq \gamma^2 \lambda \eta / 6$ and $\gamma \leq 1$, we have,

$$E_{x_1 \sim D_{+y,1}, x_2' \sim D_{+y,2}}[x_1^\top M x_2' | x_1 \in X_1'] \geq \left(1 - \frac{\epsilon_1}{\lambda \eta}\right) \frac{\gamma^2}{2} - \frac{\epsilon_1}{\lambda \eta} \geq \frac{\gamma^2}{4},$$

since $|x_1^\top M x_2'| \leq 1$ for all $x_1 \in X_1, x_2' \in X_2$. Note also that since D is (ρ, X_1')-semi homogeneous we have that $\rho_y E_{x_1 \sim D_{y,1}}[x_1 | x_1 \in X_1'] + \rho_{-y} E_{x_1 \sim D_{-y,1}}[x_1 | x_1 \in X_1'] = 0$, which implies that $E_{x_1 \sim D_{-y,1}}[x_1 | x_1 \in X_1'] = \frac{-\rho_y}{\rho_{-y}} E_{x_1 \sim D_{+y,1}}[x_1 | x_\in X_1']$. Therefore,

$$\begin{aligned}
\frac{\gamma^2}{4} &\leq E_{x_1 \sim D_{+y,1}, x_2' \sim D_{+y,2}}[x_1^\top M x_2' | x_1 \in X_1'] \\
&= E_{x_1 \sim D_{+y,1}}[x_1^\top | x_1 \in X_1']\ M\ E_{x_2' \sim D_{+y,2}}[x_2'] \\
&= \frac{-\rho_y}{\rho_{-y}} E_{x_1 \sim D_{-y,1}}[x_1^\top | x_1 \in X_1']\ M\ E_{x_2' \sim D_{+y,2}}[x_2'] \\
&= \frac{-\rho_y}{\rho_{-y}} E_{x_1 \sim D_{-y,1}, x_2' \sim D_{+y,2}}[x_1^\top M x_2' | x_1 \in X_1']
\end{aligned}$$

This implies that

$$E_{x_1 \sim D_{-y,1}, x_2' \sim D_{+y,2}}[x_1^\top M x_2' | x_1 \in X_1'] \leq -\frac{\gamma^2 \rho_{-y}}{4 \rho_y}$$

which completes the proof. ■

It will be convenient to have a notation $\beta_y = \frac{\rho_{-y}}{8\rho_y}$. Since $x_1^\top M x_2' \geq -1$, from Lemma 8 we have that $\gamma^2 \beta_y \leq 1/2$. We also have that $\beta_y \geq \gamma^2/32$.[1] It would be convenient to assume that β_y is a constant, e.g., $\beta_{+1}, \beta_{-1} \in [0.01, 0.99]$

Lemma 9 *For $y \in \{-1, +1\}$, a good matrix M has the property that $x_1 \sim D_{-y,1}$, conditioned on $x_1 \in X_1'$, is $(\gamma^2 \beta_y, M, y)$-heavy with probability at least $\gamma^2 \beta_y$.*

1. Since $\frac{\gamma^2}{4} \leq \frac{-\rho_y}{\rho_{-y}} E_{x_1 \sim D_{-y,1}, x_2' \sim D_{+y,2}}[x_1^\top M x_2' | x_1 \in X_1'] \leq \frac{\rho_y}{\rho_{-y}} = 8\beta_y$

Proof Since the matrix M is good, by Lemma 8 we have that,

$$E_{x_1 \sim D_{-y,1}, x_2' \sim D_{+y,2}}[x_1^\top M x_2' | x_1 \in X_1'] \leq -\frac{\gamma^2}{4} \cdot \frac{\rho_{-y}}{\rho_y} = -2\gamma^2 \beta_y \ .$$

The claim now follows from Markov's inequality, using the fact that $|x_1^\top M x_2'| \leq 1$ for all $x_1 \in X_1, x_2' \in X_2$. In particular, if the probability, conditioned on $x_1 \in X_1'$, that x_1 is $(\gamma^2\beta_y, M, +y)$-heavy is less than $\gamma^2\beta_y$, then this would imply that $\Pr_{x_1 \sim D_{-y,1}, x_2' \sim D_{+y,2}}[x_1^\top M x_2' \geq 0 | x_1 \in X_1'] \geq (1-\gamma^2\beta_y)^2 > 1 - 2\gamma^2\beta_y$ and so the expected value of $x_1^\top M x_2'$ would have to be greater than $-2\gamma^2\beta_y$, a contradiction to Lemma 8. ■

Lemma 10 *Fix $y \in \{+1, -1\}$. Assuming the matrix M is good, in each iteration of function* `large_sample`*, with probability $\eta\alpha_{-y}\gamma^2\beta_y/2 = \Theta(\eta\gamma^2)$ it selects x_1 from $D_{-y,1}$, $x_1 \in X_1'$, and the size of S' is at least $limit_2 \leq \gamma^2\beta_y m_2$. Furthermore, with an additional probability of $1/e$, for $\epsilon_1 < \gamma\eta/m_2$ we have that all the examples in S' are of label y.*

Proof Assuming the matrix M is good, by Lemma 9, there is at least a $\gamma^2\beta_y$ probability that a random x_1 from $D_{-y,1}$, conditioned on $x_1 \in X_1'$, is $(\gamma^2\beta_y, M, +y)$-heavy. This implies that a random x_1 from D has this property with probability at least $\eta\alpha_{-y}\gamma^2\beta_y = \Theta(\eta\gamma^2)$.

Assume this is the case, i.e., x_1 has label $-y$, $x_1 \in X_1'$ and is $(\gamma^2\beta_y, M, +y)$-heavy. This implies that the expected size of S' is at least $\gamma^2\beta_y m_2 = \Theta(\gamma^2 m_2) = limit_2$. Therefore, with probability at least half, S' has at least $limit_2$ examples of label $+y$.

By Lemma 5, for $\epsilon_1 < \lambda\eta/m_2$, with an additional probability of at least $1/e$, S' is likely to have *no* examples labeled $-y$. ■

5.3. Analysis `GetHypo1`

Lemma 11 *Assuming the matrix M is good, in function* `GetHypo1` *each iteration, with probability at least*

$$\frac{\eta^2\gamma^4\beta_{+1}\beta_{-1}\alpha_{+1}\alpha_{-1}}{16e^2} = \frac{\eta^2\gamma^4\alpha_{+1}\alpha_{-1}}{1024e^2} = \Theta(\eta^2\gamma^4),$$

terminates and outputs a hypothesis h_1 that has error at most ϵ_2.

Proof By Lemma 10 we have that with probability $\eta\gamma^2\alpha_{-1}\beta_{+1}/(2e)$ the set S_{+1} are all positive examples and with probability $\eta\gamma^2\alpha_{+1}\beta_{-1}/(2e)$ the set S_{-1} are all negative examples and each of the sets is of size at least $limit_2$. Given that this happens, the sets S_{+1} and S_{-1} are linearly separable, and therefore, in such a case, function `GetHypo1` finds a separating h_1 and terminates.

With probability at least $1/4$, there are subsets $I_{+1} \subseteq S_{+1}$ and $I_{-1} \subseteq S_{-1}$ of size at least $\lambda\ limit_2$ which contain a random independent samples from the marginal distributions of D_{+1} and D_{-1}, respectively. (This is due to the (λ, X_1')-weak dependence of D.) Since $\lambda\ limit_2 = VC(d_1, \epsilon_2)$ we can apply the VC dimension generalization bounds to bound the error of h_1 by ϵ_2. ■

5.4. Analysis `GetHypo2`

Lemma 12 *Assuming the matrix M is good, with probability $\Theta(\eta^2\gamma^4)$ the function* `GetHypo2` *outputs a hypothesis h_2 that: (1) has error at most ϵ_3 and (2) has probability of disagreeing with h_1 at most ϵ_3.*

Proof By Lemma 11 we have, with probability at least $\Theta(\eta^2\gamma^4)$ that $\Pr[h_1 \neq f_1] \leq \epsilon_2 = 1/m_3$. Assume that holds. This implies that with probability at least $1/e$ we have that all the m_3 examples sampled in `GetHypo2` are classified correctly by h_1. In such a case we will find a separating hyperplane h_2. The size of $m_3 = VC(d_2, \epsilon_3)$ guarantees that, with constant probability, the error of h_2 is at most ϵ_3 and the probability that h_1 and h_2 disagree is also ϵ_3. ■

5.5. Analysis `verify`

We first show that any hypothesis $h = (h_1, h_2)$ with the property that (1) h_1 and h_2 agree with high probability over examples (x_1, x_2) from D, and (2) h_1 and h_2 label a reasonable fraction of examples positive and a reasonable fraction negative, must either be close to the target f or close to its complement $-f$. This extends Theorem 15 from Balcan and Blum (2010) to the case that D satisfies (λ, X_1')-weak dependence, and the proof is essentially the same.

Lemma 13 *Assume $h = (h_1, h_2)$ such that: (1) $\Pr_{(x_1,x_2)\sim D}[h_1(x_1) \neq h_2(x_2)] \leq \mu$, and (2) $\Pr_{(x_1,x_2)\sim D}[h(x_2) = y] > \frac{2\mu}{\lambda\eta}$, for $y \in \{+1, -1\}$. Then either: (1) $\Pr_{(x_1,x_2)\sim D}[h_2(x_1) \neq f(x_1, x_2)] \leq \frac{2\mu}{\eta\lambda}$ or (2) $\Pr_{(x_1,x_2)\sim D}[h_2(x_1) \neq -f(x_1, x_2)] \leq \frac{2\mu}{\eta\lambda}$.*

Proof By definition of D we have,

$$\Pr_{(x_1,x_2)\sim D}[h_1(x_1) \neq h_2(x_2)] = \alpha_{+1} \Pr_{(x_1,x_2)\sim D_{+1}}[h_1(x_1) \neq h_2(x_2)] + \alpha_{-1} \Pr_{(x_1,x_2)\sim D_{-1}}[h_1(x_1) \neq h_2(x_2)]$$

Fix $y \in \{+1, -1\}$. Since distribution D satisfies (λ, X_1')-weak dependence we have

$$\begin{aligned}
\Pr_{(x_1,x_2)\sim D_y}[h_1(x_1) \neq h_2(x_2)] \geq & \eta \Pr_{(x_1,x_2)\sim D_y}[h_1(x_1) \neq h_2(x_2) | x_1 \in X_1'] \\
\geq & \eta\lambda \Pr_{x_1\sim D_{y,1}, x_2\sim D_{y,2}}[h_1(x_1) \neq h_2(x_2) | x_1 \in X_1'] \\
= & \eta\lambda \Big(\Pr_{x_1\sim D_{y,1}}[h_1(x_1) = y | x_1 \in X_1'] \Pr_{x_2\sim D_{y,2}}[h_2(x_2) \neq y] \\
& + \Pr_{x_1\sim D_{y,1}}[h_1(x_1) \neq y | x_1 \in X_1'] \Pr_{x_2\sim D_{y,2}}[h_2(x_2) = y] \Big) \\
= & \eta\lambda \big((1 - \chi_y) \Pr_{x_2\sim D_{y,2}}[h_2(x_2) \neq y] + \chi_y \Pr_{x_2\sim D_{y,2}}[h_2(x_2) = y] \big),
\end{aligned}$$

where $\chi_y = \Pr_{x_1\sim D_{y,1}}[h_1(x_1) \neq y | x_1 \in X_1']$. If $\chi_y \leq 1/2$ we have that

$$\frac{2}{\eta\lambda} \Pr_{(x_1,x_2)\sim D_y}[h_1(x_1) \neq h_2(x_2)] \geq \Pr_{x_2\sim D_{y,2}}[h_2(x_2) \neq y]$$

and if $\chi_y \geq 1/2$ we have that

$$\frac{2}{\eta\lambda} \Pr_{(x_1,x_2)\sim D_y}[h_1(x_1) \neq h_2(x_2)] \geq \Pr_{x_2\sim D_{y,2}}[h_2(x_2) = y].$$

We now consider all the possibilities depending on y and χ_y. First assume that for some $y \in \{+1, -1\}$, we have,

$$\frac{2}{\eta\lambda} \Pr_{(x_1,x_2)\sim D_{+y}}[h_1(x_1) \neq h_2(x_2)] \geq \Pr_{x_2\sim D_{+y,2}}[h_2(x_2) \neq +y] = \Pr_{x_2\sim D_{+y,2}}[h_2(x_2) = -y]$$

and

$$\frac{2}{\eta\lambda} \Pr_{(x_1,x_2)\sim D_{-y}}[h_1(x_1) \neq h_2(x_2)] \geq \Pr_{x_2\sim D_{-y,2}}[h_2(x_2) = -y].$$

This implies that

$$\frac{2\mu}{\eta\lambda} \geq \frac{2}{\eta\lambda} \Pr_{(x_1,x_2)\sim D}[h_1(x_1) \neq h_2(x_2)] \geq \Pr_{x_2\sim D}[h_2(x_2) = -y],$$

which is a contradiction to the assumption that $\Pr_{x_2\sim D}[h_2(x_2) = -y] > \frac{2\epsilon}{\eta\lambda}$ for both $y \in \{-1, +1\}$.

Therefore we have two remaining possibilities. The first possibility,

$$\frac{2}{\eta\lambda} \Pr_{(x_1,x_2)\sim D_{+1}}[h_1(x_1) \neq h_2(x_2)] \geq \Pr_{x_2\sim D_{+1,2}}[h_2(x_2) \neq +1] = \Pr_{x_2\sim D_{+1,2}}[h_2(x_2) \neq f_2(x_2)]$$

and

$$\frac{2}{\eta\lambda} \Pr_{(x_1,x_2)\sim D_{-1}}[h_1(x_1) \neq h_2(x_2)] \geq \Pr_{x_2\sim D_{-1,2}}[h_2(x_2) = +1] = \Pr_{x_2\sim D_{-1,2}}[h_2(x_2) \neq f(x_2)],$$

implies that

$$\frac{2\mu}{\eta\lambda} \geq \frac{2}{\eta\lambda} \Pr_{(x_1,x_2)\sim D}[h_1(x_1) \neq h_2(x_2)] \geq \Pr_{x_2\sim D}[h_2(x_2) \neq f(x)],$$

which is conclusion (1) of the lemma. The second possibility,

$$\frac{2}{\eta\lambda} \Pr_{(x_1,x_2)\sim D_{+1}}[h_1(x_1) \neq h_2(x_2)] \geq \Pr_{x_2\sim D_{+1,2}}[h_2(x_2) \neq -1] = \Pr_{x_2\sim D_{+1,2}}[h_2(x_2) \neq -f_2(x_2)]$$

and

$$\frac{2}{\eta\lambda} \Pr_{(x_1,x_2)\sim D_{-1}}[h_1(x_1) \neq h_2(x_2)] \geq \Pr_{x_2\sim D_{-1,2}}[h_2(x_2) = -1] = \Pr_{x_2\sim D_{-1,2}}[h_2(x_2) \neq -f_2(x_2)],$$

implies that

$$\frac{2\mu}{\eta\lambda} \geq \frac{2}{\eta\lambda} \Pr_{(x_1,x_2)\sim D}[h_1(x_1) \neq h_2(x_2)] \geq \Pr_{x_2\sim D}[h_2(x_2) \neq -f(x)],$$

which is conclusion (2) of the lemma. ■

Lemma 14 *Assume that $h = (h_1, h_2)$ such that: (1) $\Pr_{(x_1,x_2)\sim D}[h_1(x_1) \neq h_2(x_2)] \leq \epsilon_3$, and (2) $\Pr_{(x_1,x_2)\sim D}[h_2(x_2) = y] > \alpha_y - 2\epsilon_4 > \frac{2\epsilon_3}{\lambda\eta}$, for $y \in \{+1, -1\}$. Then, for $m_4 = \Theta(\frac{1}{\epsilon_4^2}\log(2/\delta))$, with probability at least $1 - \delta$ the function* `verify` *will return* `TRUE`.

Proof With probability $1-\delta$ we have two events. The first is that the empirical estimate of $h_1 \neq h_2$, i.e., $|S|/m_4$, is within ϵ_4 from its true probability. That is,

$$\left| \Pr_D[h_1(x_1) \neq h_2(x_2)] - \frac{|S|}{m_4} \right| \leq \epsilon_4$$

which implies that $|S|/m_4 \leq \epsilon_3 + \epsilon_4$, which is the first test of `verify`.

The second is that the empirical estimate of $h_2 = y$, i.e., $|S_{2,y}|/m_4$, is within ϵ_4 from its true probability, i.e.,

$$\left| \Pr_D[h_2(x_2) = y] - \frac{|S_{2,y}|}{m_4} \right| \leq \epsilon_4$$

which implies that $\alpha_y - 3\epsilon_4 \leq |S_{2,y}|/m_4$ for $y \in \{+1, -1\}$, which is the second test of `verify`. ■

Lemma 15 *Assume that $h = (h_1, h_2)$ such that: (1) $\Pr_{(x_1,x_2)\sim D}[h_1(x_1) \neq h_2(x_2)] > \epsilon_3 + 2\epsilon_4$, or (2) $\Pr_{(x_1,x_2)\sim D}[h_2(x_2) = y] < \alpha_y - 4\epsilon_4$, for $y \in \{+1, -1\}$. Then, for $m_4 = \Theta(\frac{1}{\epsilon_4^2}\log(2/\delta))$, with probability at least $1-\delta$ the function* `verify` *will return* `FALSE`.

Proof With probability $1-\delta$ we have two events. The first is that the empirical estimate of $h_1 \neq h_2$, i.e., $|S|/m_4$, is within ϵ_4 from its true probability. That is,

$$\left| \Pr_D[h_1(x_1) \neq h_2(x_2)] - \frac{|S|}{m_4} \right| \leq \epsilon_4$$

The second is that the empirical estimate of $h_2 = y$, i.e., $|S_{2,y}|/m_4$, is within ϵ_4 from its true probability, i.e.,

$$\left| \Pr_D[h_2(x_2) = 1] - \frac{|S_{2,y}|}{m_4} \right| \leq \epsilon_4$$

Assume both events hold. If we have $\Pr_{(x_1,x_2)\sim D}[h_1(x_1) \neq h_2(x_2)] > \epsilon_3 + 2\epsilon_4$, then we have $|S|/m_4 > \epsilon_3 + \epsilon_4$, and `verify` returns `FALSE`.

If for some $y \in \{+1, -1\}$ we have $\Pr_{(x_1,x_2)\sim D}[h(x_2) = y] < \alpha_y - 4\epsilon_4$, this implies that $\alpha_y - 3\epsilon_4 > |S_{2,y}|/m_4$, and `verify` returns `FALSE`. ■

Note that the above lemmas do not apply for the case that both: (1) $\epsilon_3 < \Pr_{(x_1,x_2)\sim D}[h_1(x_1) \neq h_2(x_2)] \leq \epsilon_3 + 2\epsilon_4$, and (2) $\alpha_y - 4\epsilon_4 < \Pr_{(x_1,x_2)\sim D}[h_2(x_2) = y] \leq \alpha_y - 2\epsilon_4$, for $y \in \{+1, -1\}$. However, in this case we are fine with `verify` returning either `TRUE`, since the pair (h_1, h_2) is good enough, or `FALSE`, which implies that we keep searching for a new pair (h_1, h_2).

5.6. Analysis `GetHypo`

Lemma 16 *Assuming that the matrix M is good, w.h.p. the following holds: (1) in function* `GetHypo` *each iteration terminates with probability at least $\Theta(\eta^2\gamma^4)$, and (2) When it terminates the output hypotheses (h_1, h_2) have error at most ϵ.*

Proof By Lemma 11 we have that with probability $\Theta(\eta^2\gamma^4)$ the procedure `GetHypo1` generates a hypothesis h_1 which has error at most ϵ_2. By Lemma 12 we have that procedure `GetHypo2` generates a hypothesis h_2 for which $\Pr_D[h_1 \neq h_2] \leq \epsilon_3$, and $\Pr_D[f \neq h_2] \leq \epsilon_3$. (Note that success probability of both depends mostly on the same events so the joint success probability is $\Theta(\eta^2\gamma^4)$.) If this holds, then by Lemma 14, with high probability, function `verify` will return `TRUE` on (h_1, h_2) and in such a case, by Lemma 13 they have error at most $2\epsilon_3/(\eta\lambda) < \epsilon$. The probability of all the events to hold is at least $\Theta(\eta^2\gamma^4)$.

By Lemma 15 if either: (1) $\Pr[h_1 \neq h_2] > \epsilon_3 + 2\epsilon_4$, or (2) $\Pr[h_2 = y] < \alpha_y - 4\epsilon_4$, then w.h.p. `verify` will return `FALSE`. Therefore, if `verify` returns `TRUE`, then w.h.p. we have both (1) $\Pr[h_1 \neq h_2] \leq \epsilon_3 + 2\epsilon_4 = \mu$ and (2) $\Pr[h_2 = y] \geq \alpha_y - 4\epsilon_4 > 2\mu/(\lambda\eta)$ for $y \in \{+1, -1\}$. By Lemma 13, this implies that (h_1, h_2) has error at most $2\mu/(\lambda\eta) < \epsilon$. ■

5.7. Correctness and complexity

Theorem 17 *With probability $1-\delta$ we have that the output (h_1, h_2) has error at most ϵ. In addition the expected running time is polynomial in $\eta^{-1}, \gamma^{-1}, \epsilon^{-1}, \log(1/\delta), d_1, d_2$.*

Proof For the correctness: By Lemma 5 the matrix M output by `GenMatrix` is good. By Lemma 16 we have that when `GetHypo` terminates it outputs (h_1, h_2) has error at most ϵ. ■

6. Open problems

Our learning algorithm requires the target linear separators to pass through the origin and for the center of mass of the distribution on x_1 to be the origin as well (or at least for there to exist X_1', η, ρ_{+1}, and ρ_{-1} such that the distribution is (ρ, X_1')-semi-homogeneous as in Assumption 3). One natural open problem is whether this assumption on the distribution can be removed. Another open problem is whether one can remove the assumption that the data distribution has an inverse-polynomial margin.

Acknowledgments

AB was supported in part by the National Science Foundation under grants CCF-1525971, CCF-1535967, and CCF-1331175.
YM was supported in part by a grant from the Israel Science Foundation, a grant from the United States-Israel Binational Science Foundation (BSF), and the Israeli Centers of Research Excellence (I-CORE) program (Center No. 4/11).

References

S. Abney. Bootstrapping. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 360–367, 2002.

M.-F. Balcan and A. Blum. A discriminative model for semi-supervised learning. *JACM*, 57(3), 2010. Article 19.

A. Blum and T. M. Mitchell. Combining labeled and unlabeled data with co-training. In *Proc. 11th Annual Conf. Computational Learning Theory*, pages 92–100, 1998.

Kamalika Chaudhuri, Sham M. Kakade, Karen Livescu, and Karthik Sridharan. Multi-view clustering via canonical correlation analysis. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 129–136, 2009. ISBN 978-1-60558-516-1.

Sanjoy Dasgupta, Michael L Littman, and David McAllester. Pac generalization bounds for co-training. In *NIPS*, volume 1, pages 375–382, 2001.

Ngoc Quynh Do Thi, Steven Bethard, and Marie-Francine Moens. Facing the most difficult case of semantic role labeling: A collaboration of word embeddings and co-training. In *Proceedings of the 26th International Conference on Computational Linguistics*. ACL, 2016.

Svetlana Kiritchenko and Stan Matwin. Email classification with co-training. In *Proceedings of the 2011 Conference of the Center for Advanced Studies on Collaborative Research*, CASCON '11, pages 301–312, 2011.

Abhishek Kumar and Hal Daumé. A co-training approach for multi-view spectral clustering. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 393–400, 2011.

Anat Levin, Paul A Viola, and Yoav Freund. Unsupervised improvement of visual detectors using co-training. In *ICCV*, pages 626–633, 2003.

Weifeng Liu, Yang Li, Xu Lin, Dacheng Tao, and Yanjiang Wang. Hessian-regularized co-training for social activity recognition. *PloS one*, 9(9):e108474, 2014.

Robert H. Sloan. Four types of noise in data for PAC learning. *Information Processing Letters*, 54 (3):157–162, 1995.

Xiaojun Wan. Co-training for cross-lingual sentiment classification. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-volume 1*, pages 235–243. Association for Computational Linguistics, 2009.

Appendix A. Pseudo Code of the algorithm

$\epsilon_1 = (\gamma^2\lambda\eta)/(6m_2)$;$\epsilon_2 = \frac{1}{m_3}$; $\epsilon_3 = \epsilon\eta\lambda/60$; $\epsilon_4 = \epsilon_3$
$m_1 = VC(d_1 d_2, \epsilon_1)/(\alpha_{+1}\alpha_{-1})$; $m_2 = \Theta(limit_2/\gamma^2)$;
$limit_2 = VC(d_1, \epsilon_2)/\lambda$; $m_3 = VC(d_2, \epsilon_3)$; $m_4 = \Theta(\frac{1}{\epsilon_4^2}\log(2/\delta))$;

$M \leftarrow$ `GetMatrix`(m_1).
$(h_1, h_2) \leftarrow$ `GetHypo`$(M, m_2, m_3, limit_2)$
`RETURN` (h_1, h_2).

Function `GetMatrix`(m_1).
Sample m_1 examples $(x_1^{(i)}, x_2^{(i)})$.
Solve the quadratic program:
Find $d_1 \times d_2$ matrix M such that
$(x_1^{(i)})^\top M x_2^{(i)} \geq \gamma^2$
$\sum_{i=1}^{d_1}\sum_{j=1}^{d_2} M_{i,j}^2 \leq 1$.
`RETURN` M.

Function `GetHypo`$(M, m_2, m_3, limit_2, \epsilon_3, \epsilon_4)$
Pass$\leftarrow$ FALSE
`REPEAT`
$h_1 \leftarrow$ `GetHypo1`$(M, m_2, limit_2)$
$h_2 \leftarrow$ `GetHypo2`(h_1, m_3)
`IF` $h_2 \neq \perp$ `THEN`
`Pass` $\leftarrow$ `verify`$(h_1, h_2, m_4, \epsilon_3, \epsilon_4)$
`UNTIL` Pass =TRUE.
`RETURN` (h_1, h_2).

Function `GetHypo1`$(M, m_2, limit_2)$
`REPEAT`
$S_{+1} \leftarrow$ `large_sample`$(M, m_2, limit_2)$.
$S_{-1} \leftarrow$ `large_sample`$(M, m_2, limit_2)$.
`UNTIL` Finding h_1 separating S_{+1} from S_{-1}.
`RETURN` h_1

Function `GetHypo2`(h_1, m_3)
Sample m_3 examples $(x_1^{(i)}, x_2^{(i)})$.
Let $T_{+1} = \{x_2^{(i)} : h_1(x_1^{(i)}) = +1\}$ and $T_{-1} = \{x_2^{(i)} : h_1(x_1^{(i)}) = -1\}$.
Find h_2 separating T_{+1} from T_{-1}.
If no such h_2 set $h_2 \leftarrow \perp$
`RETURN` h_2.

Function `large_sample`$(m_2, limit_2)$

REPEAT
  sample (x_1, x_2)
  sample m_2 examples $(x_1^{(i)}, x_2^{(i)})$.
  Let $S' = \{x_1^{(i)} : x_1^\top M x_2^{(i)} < 0\}$.
UNTIL $|S'| \geq limit_2$
RETURN S'

Function verify$(h_1, h_2, m_4, \epsilon_3, \epsilon_4)$
Sample m_4 examples $(x_1^{(i)}, x_2^{(i)})$.
Let $S = \{(x_1^{(i)}, x_2^{(i)}) : h_1(x_1^{(i)}) \neq h_2(x_2^{(i)})\}$.
Let $S_{2,y} = \{(x_1^{(i)}, x_2^{(i)}) : h_2(x_2^{(i)}) = y\}$
IF $(|S|/m_4 \leq \epsilon_3 + \epsilon_4)$ and $(\alpha_y - 3\epsilon_4 \leq |S_{2,y}|/m_4$ for $y \in \{+1, -1\})$
THEN test←TRUE ELSE test ←FALSE
RETURN test