Co-EM Support Vector Learning

Ulf Brefeld BREFELD@INFORMATIK.HU-BERLIN.DE
Tobias Scheffer SCHEFFER@INFORMATIK.HU-BERLIN.DE
Humboldt-Universität zu Berlin, Department of Computer Science, Unter den Linden 6, 10099 Berlin, Germany

Abstract

Multi-view algorithms, such as co-training and co-EM, utilize unlabeled data when the available attributes can be split into independent and compatible subsets. Co-EM outperforms co-training for many problems, but it requires the underlying learner to estimate class probabilities, and to learn from probabilistically labeled data. Therefore, co-EM has so far only been studied with naive Bayesian learners. We cast linear classifiers into a probabilistic framework and develop a co-EM version of the Support Vector Machine. We conduct experiments on text classification problems and compare the family of semi-supervised support vector algorithms under different conditions, including violations of the assumptions underlying multi-view learning. For some problems, such as course web page classification, we observe the most accurate results reported so far.

1. Introduction

Semi-supervised learning algorithms utilize unlabeled data to improve classification performance. The EM approach in which a classifier labels unlabeled data, and then learns from that data, is complemented by the multi-view framework. Multi-view algorithms – such as co-training (Blum & Mitchell, 1998) – split the attributes into two independent subsets, each of which has to be sufficient for learning. An example of a domain that is suitable for multi-view learning is web page classification: a page can be classified based on its content as well as based on the anchor texts of its inbound hyperlinks.

Multi-view algorithms learn two independent classifiers based on independent attribute subsets. These classifiers then provide each other with labels for the unlabeled data. The co-EM algorithm (Nigam & Ghani, 2000) combines multi-view learning with the probabilistic EM approach. This, however, requires the learning algorithm to process probabilistically labeled training data and the classifier to output class probabilities. Hence, the co-EM algorithm has so far only been studied with naive Bayes as underlying learner – even though the Support Vector Machine is known to better fit the characteristics of many interesting problems, such as text classification. We close this gap by developing and studying a co-EM version of the Support Vector Machine.

The rest of our paper is organized as follows. We discuss related work in Section 2. We formulate the problem setting and review known multi-view and semi-supervised learning algorithms which are relevant for our empirical studies in Section 3. In Section 4, we develop the co-EM Support Vector algorithm and report on our experimental results in Section 5. Section 6 concludes.

2. Related Work

Semi-supervised learning (Cooper & Freeman, 1970; for an overview, see Seeger, 2001) has a long tradition in statistics and machine learning; the Expectation Maximization (EM) algorithm (Dempster et al., 1977) is probably the most prominent approach to learning from labeled and unlabeled data (McCallum & Nigam, 1998; Nigam & Ghani, 2000). The EM algorithm is wrapped around learning algorithms that fit model parameters to probabilistically labeled data.

Linear separators, such as Support Vector Machines (SVMs), cannot immediately be trained from probabilistically labeled examples. The transductive SVM – TSVM – (Vapnik, 1998; Bennett, 1999; Joachims, 1999b) still utilizes unlabeled data by EM-like self-labeling and a modification of the optimization criterion (see Section 3). The TSVM is motivated by

Appearing in *Proceedings of the 21st International Conference on Machine Learning*, Banff, Canada, 2004. Copyright 2004 by the authors.

the idea that the test instances which are to be classified are often available (without class labels) during training. Besides the transductive SVM, a transductive version of the k-NN algorithm (the spectral graph partitioning algorithm; Joachims, 2003) has been studied.

The co-training algorithm (Blum & Mitchell, 1998) learns two decision functions on independent attribute subsets but does not operate with class probabilities – which makes it easily applicable for support vector learning. The co-EM algorithm (Nigam & Ghani, 2000; Ghani, 2002) combines multi-view learning with EM. Co-EM (with naive Bayes as underlying classifier) has been found to outperform co-training in some cases (Nigam & Ghani, 2000); in particular, when the compatibility and independence assumptions (see Section 3) are not violated (Muslea et al., 2002a).

Applications of co-training that have been studied include classification of web pages (Blum & Mitchell, 1998), named entity recognition (Collins & Singer, 1999), text classification (*e.g.,* Denis et al., 2003), wrapper induction (Muslea et al., 2002b), classification of emails (Kiritchenko & Matwin, 2002; Kockelkorn et al., 2003), and word form normalization (Mladenic, 2002). For text classification, experiments have clearly shown that the co-trained Support Vector Machine (in fact, even the "vanilla" Support Vector Machine) substantially outperforms co-trained naive Bayes (Kiritchenko & Matwin, 2002; Kockelkorn et al., 2003). Together with the observation that co-EM outperforms co-training for problems with compatible and independent views, this raises the question whether there is a co-EM version of the Support Vector Machine, and whether this is possibly the most effective classifier for text classification problems with compatible views.

However, it should be noted that semi-supervised learning does not *necessarily* lead to better results than supervised learning. When the target distribution is not in the assumed model class, then the best approximation of the unlabeled data can sometimes lie further away from the optimal classifier than the best approximation of (even few) labeled data (Cozman et al., 2003). While additional unlabeled data have often been observed to improve classifier performance (Baluja, 1998; Collins & Singer, 1999; Nigam et al., 2000; Mladenic, 2002), there are some cases in which they have been found to deteriorate performance – often, but not always, when the labeled sample is large (Shahshahani & Landgrebe, 1994; Baluja, 1998; Nigam et al., 2000; Kockelkorn et al., 2003).

3. Semi-Supervised and Multi-View Learning

We focus on the *semi-supervised learning* setting in which *labeled data* $D_l = \langle(x_1, y_1), \ldots, (x_{m_l}, y_{m_l})\rangle$, $y_i \in \{+1, -1\}$ and *unlabeled data* $D_u = \langle x_1^*, \ldots, x_{m_u}^*\rangle$ are available. Our goal is to learn a *decision function* $f(x)$ which assigns high values to positive and low values to negative examples. The ability of a decision function to discriminate positives against negatives is naturally characterized by the *receiver operating characteristic (ROC)* analysis (Bradley, 1997; Provost et al., 1998). The ROC curve displays the number of true positives against the number of false positives for the range of decision function values. The area under the ROC curve, called the AUC performance, is equal to the probability that, when we draw one positive and one negative example at random, the decision function assigns a higher value to the positive than to the negative example. Depending on the application at hand, the decision function may itself be the learning result, or it may be thresholded to yield a *classifier* $h(x) = sign(f(x) - \theta)$, where θ is adjusted to minimize the application-specific cost function.

In the multi-view setting that we discuss, the available attributes V are split into disjoint sets V_1 and V_2. A labeled instance (x, y) is decomposed and viewed as (x_1, x_2, y), where x_1 and x_2 are vectors over the attributes V_1 and V_2, respectively. These views have to satisfy the *independence* and *compatibility* assumptions.

Definition 1 *Views* V_1 *and* V_2 *are* independent *when* $\forall x_1 \in V_1, x_2 \in V_2 : p(x_1, x_2|y) = p(x_1|y)p(x_2|y)$.

Definition 2 *Views* V_1 *and* V_2 *are* compatible *with target concept* $t : x \mapsto y$ *when there are hypotheses* $h_1 : V_1 \rightarrow \{-1, +1\}$ *and* $h_2 : V_2 \rightarrow \{-1, +1\}$ *such that, for all* $x = (x_1, x_2)$, $f_1(x_1) = f_2(x_2) = t(x)$.

Table 1 shows the semi-supervised learning algorithms in our discourse area, EM, TSVM, co-training, and co-EM. Let us briefly review these methods. In the *EM algorithm*, one single hypothesis iteratively labels the unlabeled data probabilistically, and then adapts its parameters to the data.

The *TSVM algorithm* has a similar structure but does not operate with class probabilities. Instead, the labeling of the unlabeled data is changed when, by switching a pair of labels, the optimization function of the current separator f_i is improved. The TSVM optimization problem is defined as follows (Joachims, 1999b).

Table 1. Semi-supervised learning algorithms.

```
Input: Labeled data $D_l$, unlabeled data $D_u$, parameters.

Semi-supervised learning with EM.
  1. Train $f_0$ on labeled data $D_l$.
  2. For $i = 1 \dots T$:
     (a) Estimate class probabilities $\hat{p}(y|x_j^*)$ for $x_j^* \in D_u$, based on hypothesis $f_{i-1}$.
     (b) Train $f_i$ using $D_l$, $D_u$, and the $\hat{p}(y|x_j^*)$.

Transductive Support Vector Learning.
  1. Train $f_0$ on labeled data $D_l$.
  2. Let $C_S$ be some small number.
  3. Use $f_0$ to label unlabeled data $D_u$, restore fixed ratio of positives to negatives.
  4. For $i = 1 \dots T$
     (a) Find $f_i$ by minimizing Equation 5 subject to constraints 2, 3, and 4. +-
     (b) While the margin can be improved by switching labels of a pair of examples in $D_u$, switch labels, retrain $f_i$.
     (c) Let $C_S = \min\{2 \times C_S, C^*\}$.

Co-training.
  1. Train $f_0^1$ and $f_0^2$ on $D_l$ using attribute sets $V_1$ and $V_2$, respectively.
  2. For $i = 1 \dots T$ until $D_u = \emptyset$:
     (a) For $v = 1 \dots 2$: Remove $n_p$ elements with greatest $f_{i-1}^v(x_j^*)$, from $D_u$ and add $(x_j^*, +1)$ to $D_l$.
     (b) For $v = 1 \dots 2$: Remove $n_n$ elements with smallest $f_{i-1}^v(x_j^*)$, from $D_u$, add $(x_j^*, -1)$ to $D_l$.
     (c) Train $f_i^1$ and $f_i^2$ on $D_l$ using attribute sets $V_1$ and $V_2$, respectively.

Co-EM.
  1. Train $f_0^2$ on labeled data $D_l$ with attributes $V_2$.
  2. For $i = 1 \dots T$; For $v = 1 \dots 2$:
     (a) Estimate class probabilities $\hat{p}(y|x_j^*)$ based on peer hypothesis $f_{i-1}^{\bar{v}}$ with complementary view $V_{\bar{v}}$.
     (b) Train $f_i^v$ using attribute set $V_v$, labeled data $D_l$ and probabilistically labeled data $D_u$ with label probabilities $\hat{p}(y|x_j^*)$.

Return: single-view: $f_T$; multi-view: $\frac{1}{2}(f_T^1 + f_T^2)$.
```

Definition 3 *Given labeled data D_l, unlabeled data D_u, and parameters C and C^*, the* TSVM optimization problem *is to minimize Equation 1 over all possible values of w, b, $y_1^*, \dots, y_{m_u}^*$, $\xi_1, \dots, \xi_{m_l}$, and $\xi_1^*, \dots, \xi_{m_u}^*$ subject to the constraints 2, 3, and 4.*

$$\min_{w,b,\xi,\xi^*,y^*} \frac{1}{2}|w|^2 + C\sum_{j=1}^{m_l}\xi_j + C^*\sum_{j=1}^{m_u}\xi_j^* \quad (1)$$

$$\forall_{j=1}^{m_l} y_j(wx_j + b) \geq 1 - \xi_j \quad (2)$$

$$\forall_{j=1}^{m_u} y_j^*(wx_j + b) \geq 1 - \xi_j^* \quad (3)$$

$$\forall_{j=1}^{m_l}\xi_j > 0,\ \forall_{j=1}^{m_u}\xi_j^* > 0, \quad (4)$$

In order to avoid local optima, the TSVM algorithm minimizes a smooth approximation of Equation 1; the contribution of the unlabeled data is weighted with a smoothing factor C_S that is doubled in each iteration until it reaches one. In order to obtain a desired ratio $\hat{p}(y)$ of positive and negative labels for the unlabeled data, the contributions of positive and negative slack values are weighted accordingly (Equation 5).

$$\min_{w,b,\xi,\xi^*,y^*} \frac{1}{2}|w|^2 + C\sum_{j=1}^{m_l}\xi_j + C_S\hat{p}(y{=}-1)\sum_{j:y_j^*=+1}\xi_j^*$$
$$+C_S\hat{p}(y{=}1)\sum_{j:y_j^*=-1}\xi_j^* \quad (5)$$

In each iteration of the *co-training algorithm* (Table 1, top right), each of the two decision functions commits to class labels for (at least) one positive and one negative example – the ones that are most confidently rated positive and negative. In contrast to EM and TSVM, co-training never revises conjectured labels for unlabeled data. The co-training algorithm has a favorable theoretical property: because of their independence, the two decision functions can provide each other with labels for the unlabeled data in a way that is essentially equivalent to drawing (slightly noisy) labeled examples at random (Blum & Mitchell, 1998). A co-training step improves the classifier performance when one classifier errs for an unlabeled instance, whereas the peer classifier is very confident and adds the correct class label to the labeled data. The independence of the views reduces the chance of both hypotheses agreeing on an erroneous label of an unlabeled instance.

The *co-EM algorithm* (Table 1, bottom right) combines the two paradigms. Unlike co-training, the co-EM algorithm does not commit to the generated class labels but rather re-estimates class probabilities after each iteration. The key difference to the self-training strategy of EM is that each decision function produces labels that are used to train the independent peer hypothesis.

4. Co-EM Support Vector Learning

In this section, we present the co-EM Support Vector Machine. We have to address two principal difficulties: The co-EM algorithm requires each classifier to yield class probability estimates for the unlabeled data. Additionally, we have to construct a learning algorithm that utilizes data which have been labeled with class probabilities for training.

Let us first address the problem of estimating class probabilities. A linear classifier f gives us an uncalibrated decision function $f(x) = wx$, but we need to have an estimate of the class posterior $p(y|x)$. We assume a parametric model: the decision function values for a class, $p(f(x)|y)$, are assumed to be governed by a normal distribution $N[\mu, \sigma^2]$. We estimate the parameters μ and σ^2 during training; given labeled and unlabeled training data and a decision function f, we proceed as follows.

Firstly, we estimate the prior probabilities $\hat{p}(y)$ from the *labeled* data. We split the unlabeled data into positives D_u^+ and negatives D_u^- according to the fixed ratio $\hat{p}(y)$; the unlabeled instances x_j^* with highest $f(x_j^*)$ are selected into D_u^+. Secondly, we estimate the mean decision function values μ_+ and μ_- (Equation 6) and corresponding variances σ_+^2 and σ_-^2 (Equation 7).

$$\mu_y = \frac{1}{|D_l^y| + |D_u^y|}\left(\sum_{(x,y)\in D_l} f(x) + \sum_{x\in D_u^y} f(x)\right) \quad (6)$$

$$\sigma_y^2 = \frac{1}{|D_l^y| + |D_u^y|}\left(\sum_{(x,y)\in D_l; x\in D_u^y} (f(x) - \mu_y)^2\right) \quad (7)$$

From the priors $\hat{p}(y)$ and Gaussian likelihoods with parameters μ_+, μ_-, σ_+^2, and σ_-^2, we can now infer the desired class probabilities $\hat{p}(y|x_j^*)$ (Equation 8).

$$\hat{p}(y|x_j^*) = \frac{N[\mu_y, \sigma_y^2](f(x_j^*))\hat{p}(y)}{N[\mu_y, \sigma_y^2](f(x_j^*))\hat{p}(y) + N[\mu_{\bar{y}}, \sigma_{\bar{y}^2}^2](f(x_j^*))\hat{p}(\bar{y})} \quad (8)$$

Now we address the second problem: Given labeled data D_l, and unlabeled data D_u with class probability estimates $\hat{p}(y|x_j^*)$, how can we train a support vector classifier? Intuitively, if $\hat{p}(y|x_j^*) = 1$ for some instance x, then that instance is essentially a labeled example and should contribute to the optimization criterion accordingly. On the other hand, $\hat{p}(y|x_j^*) = 1/2$ indicates a lack of information about the class label of x_j^*; the optimization criterion should not be influenced by the class label it assigns to such an x_j^*.

We introduce an individual weight for each example into the optimization criterion analogously to Brefeld et al. (2003); we define the weight such that we achieve a smooth transition from full contribution for $\hat{p}(y|x_j^*) = 1$ to no contribution for $\hat{p}(y|x_j^*) = 1/2$. We label an unlabeled instance x_j^* with $y = \text{argmax}_y \hat{p}(y|x_j^*)$ and define its weight to be $c_{x_j^*} = \max_{y'} \hat{p}(y'|x_j^*) - \min_{y'} \hat{p}(y'|x_j^*)$.

Table 2. The co-EM SVM algorithm.

```
Co-EM SVM. Input: Labeled data $D_l$, unlabeled data $D_u$, slack parameter $C$, number of iterations $T$.

1. Initialize smoothing factor $C_S = \frac{1}{2^T}$
2. Train initial support vector machine $f_0^2$ on labeled data $D_l$ using the attributes in $V_2$.
3. Estimate $\hat{p}(y)$ using the labeled data $D_l$.
4. For $i = 1 \dots T$: For $v = 1 \dots 2$:
    (a) Let $D_u^+$ be the $\hat{p}(y{=}1)|D_u|$ many unlabeled examples with highest decision function values $f_{i-1}^{\bar{v}}(x_j^*)$ (use decision function with complementary view $\bar{v}$); Let $D_u^- = D_u \setminus D_u^+$.
    (b) Estimate $\mu_+$, $\mu_-$, $\sigma_+^2$, and $\sigma_-^2$ from $D_l$ and $D_u$ according to Equations 6 and 7.
    (c) For all unlabeled data $x_j^*$, estimate $\hat{p}(y|x_j^*)$ according to Equation 8, based on $f_{i-1}^{\bar{v}}$.
    (d) Train Support Vector Machine $f_i^v$ by solving the optimization problem of Definition 4 with smoothing factor $C_S$; that is, minimize Equation 13 subject to the constraints 10, 11, and 12, using the attributes in $V_v$.
    (e) End For $v$; Let $C_S = 2C_S$; End For $i$.
5. Return the combined function $\frac{1}{2}(f_T^1 + f_T^2)$.
```

Definition 4 *Given labeled data D_l and unlabeled data $D_u = \langle x_1^*, \dots, x_m^* \rangle$ with label probabilities $\hat{p}(y|x_j^*)$, the* probabilistic SVM optimization problem *is to minimize Equation 9 over all possible values of w, b, $\xi_1, \dots, \xi_{m_l}$, and $\xi_1^*, \dots, \xi_{m_u}^*$, subject to the constraints 10, 11, and 12, where $c_{x_j*} = (\max_y \hat{p}(y|x_j^*) - \min_y \hat{p}(y|x_j^*))$.*

$$\min_{w,b,\xi,\xi^*} \frac{1}{2}|w|^2 + C\left(\sum_{j=1}^{m_l} \xi_j + \sum_{j=1}^{m_u} c_{x_j^*} \xi_j^*\right) \quad (9)$$

$$\forall_{j=1}^{m_l} y_j(wx_j + b) \geq 1 - \xi_j \quad (10)$$

$$\forall_{j=1}^{m_u} (\text{argmax}_y \hat{p}(y|x_j^*))(wx_j^* + b) \geq 1 - \xi_j^* \quad (11)$$

$$\forall_{j=1}^{m_l} \xi_j > 0, \quad \forall_{j=1}^{m_u} \xi_j^* > 0 \quad (12)$$

In order to reduce the risk of finding local minima, we copy the smoothing strategy of the TSVM and multiply the contributions of the unlabeled data to Equation 9 by an initially small number C_S which is doubled in each iteration until it reaches one (Equation 13). The resulting algorithm is shown in Table 2.

$$\min_{w,b,\xi,\xi^*} \frac{1}{2}|w|^2 + C\left(\sum_{j=1}^{m_l} \xi_j + C_S \sum_{j=1}^{m_u} c_{x_j^*} \xi_j^*\right) \quad (13)$$

We can trivially extend the co-EM SVM to non-linear functions by moving from the primal to the dual representation of the optimization criterion and replacing the inner products by kernel functions. As a by-product, we obtain another semi-supervised single-view algorithm: the EM SVM algorithm is a self-training strategy that is just the co-EM SVM algorithm with $V_1 = V_2$.

How does co-EM improve the performance of a Support Vector Machine? Intuitively, when x is a large margin example for f^1, then f^1 has a small error probability for x. When V_1 and V_2 are truly independent, then the projection of x into V_2 is a randomly drawn instance in V_2; x may be a support vector in V_2 even though it is a large-margin example in V_1. The co-EM SVM labels each unlabeled example in V_2 with the class label assigned by f^1. Co-EM assigns a weight to the example that is derived from the probability that this class label is in fact correct. This only holds for independent views; in the other extreme of equal views, co-EM training becomes EM self-training.

5. Empirical Studies

Our experiments are based on the course data set (Blum & Mitchell, 1998; Nigam & Ghani, 2000), and the well-known Reuters-21578 and 20-newsgroups data sets. In the course data set, the task is to decide whether a web page is a course home page, based on its content (V_1) as well as on the anchor texts of inbound links (V_2); the split of attributes into V_1 and V_2 is explicit for the data set.

All curves that we present in this section are averages of 20 runs of the focused algorithm, with distinct, randomly drawn samples. Our implementation of the co-EM algorithm is built into SVM^{light} (Joachims, 1999a). We use the default parameters of SVM^{light} and linear kernels for all experiments. We want to shed light on the following list of questions.

How fast does co-EM SVM converge? The curves for the co-EM SVM in Figure 1 (for the course data set), third column, show a sharp increase in the second iteration, and another increase (in few cases, a decrease) towards the end of the training process. The increase after the first iteration is caused by the unlabeled data which are first perceived in the second round. The change towards the end of the training process is caused by the smoothing factor which exponentially approaches one in the last round. The flat region in between indicates that the chosen 30 iterations are more than sufficient. In many cases, the maximal AUC value is reached for a smoothing weight C_S of less than one. This implies that we could improve the performance of the co-EM SVM by adjusting the maximal C_S as a learning parameter. However, we refrain from adjusting any parameters and report on results for a maximal smoothing parameter of 1.

How does the relative benefit of semi-supervised support vector algorithms depend on the number of available labeled data? We vary the number of labeled examples and observe ROC curves over the co-training and EM iterations. Figure 1, top row, compares the curves for co-training, co-EM SVM, and EM SVM. The right-most curve in the top row summarizes these results and compares them to the performance of the "vanilla" SVM and the TSVM. For all labeled sample sizes, the co-EM SVM outperforms all other variants.

How does the relative benefit of semi-supervised support vector algorithms depend on the number of available unlabeled data? The second row of Figure 1 shows the results for 2 positive and 8 negative and various unlabeled sample sizes, the third row for 4 positive and 16 negative labeled examples and various unlabeled sample sizes. The right-most diagrams summarize the results and present the baselines SVM and TSVM.

The performance of all variants scales down linearly as we reduce the amount of unlabeled data. Except for 2 positive and 8 negative examples using 12.5% of the unlabeled data, co-EM SVM is most effective. The former case is dominated by EM SVM that is least affected by the amount of unlabeled data. Here, co-training behaves brittly and the performance decreases over the iterations. This decrease becomes stronger as we reduce the amount of unlabeled data.

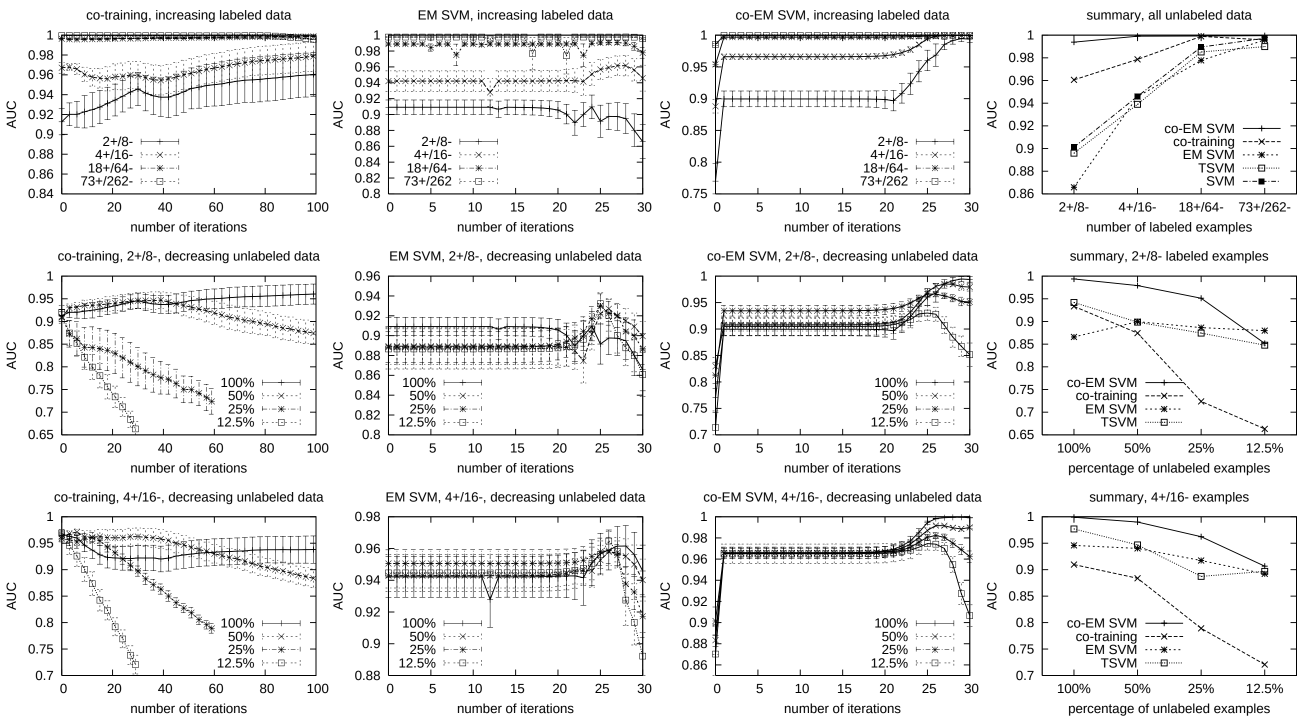

Figure 1. Semi-supervised support vector learning for the course data set.

How does the relative benefit of semi-supervised support vector algorithms depend on the compatibility of the views? In order to add controlled amounts of incompatibility and dependence into the experiment, we adapt an experimental setting of Nigam and Ghani (2000) and Muslea et al. (2002a). We use four of the 20 newsgroups: rec.autos, comp.graphics, sci.space, and talk.politics.misc.

After building tfidf vectors, we generate positive examples by concatenating vectors x_1 from rec.auto with randomly drawn vectors x_2 from sci.space to construct multi-view examples (x_1, x_2). We generate negative examples by concatenating vectors from comp.graphics with vectors from talk.politics.misc. This procedure generates views which are perfectly independent (peers are selected randomly) and compatible (either group can be discriminated from the other).

In each run we choose 5 positive and 5 negative labeled examples and add noise and dependencies, respectively, at random. Figure 2, top row, shows the results for increasingly large incompatibility (percentage of labels flipped). With up to 20% noise, both co-EM and co-training learn extremely accurate separators (both achieve AUC values of 1). As we add increasingly much noise, the performance of co-training deteriorates faster than the performance of co-EM SVM.

How does the relative benefit of semi-supervised support vector algorithms depend on the independence of the views? In order to add dependencies into the data set we proceed as follows. Each vector is a concatenation of attributes $x_1, \ldots, x_k$ (view V_1), and $x_{k+1}, \ldots, x_{2k}$ (view V_2). For each vector, each attribute $k+i$ assumes the value of attribute i (as opposed to its original value) with probability p_{dep}. For $p_{dep} = 0$, the views V_1 and V_2 are perfectly independent. For $p_{dep} = 1$, the projections of each instance into either view are equal; the views are totally dependent. This procedure allows to add much stronger dependencies than the related procedure proposed by Muslea et al. (2002a)

Figure 2, bottom row, shows the curves for varying levels of dependency. The performance of the co-EM SVM deteriorates faster than the performance of co-training as we add strong dependencies. As expected, the SVM shows only marginal deteriorations and outperforms all other variants for stronger dependencies.

So, how does the co-EM SVM algorithm compare to results of co-training and co-EM with naive Bayes? We focus on the course data set for which several results are published that are based on

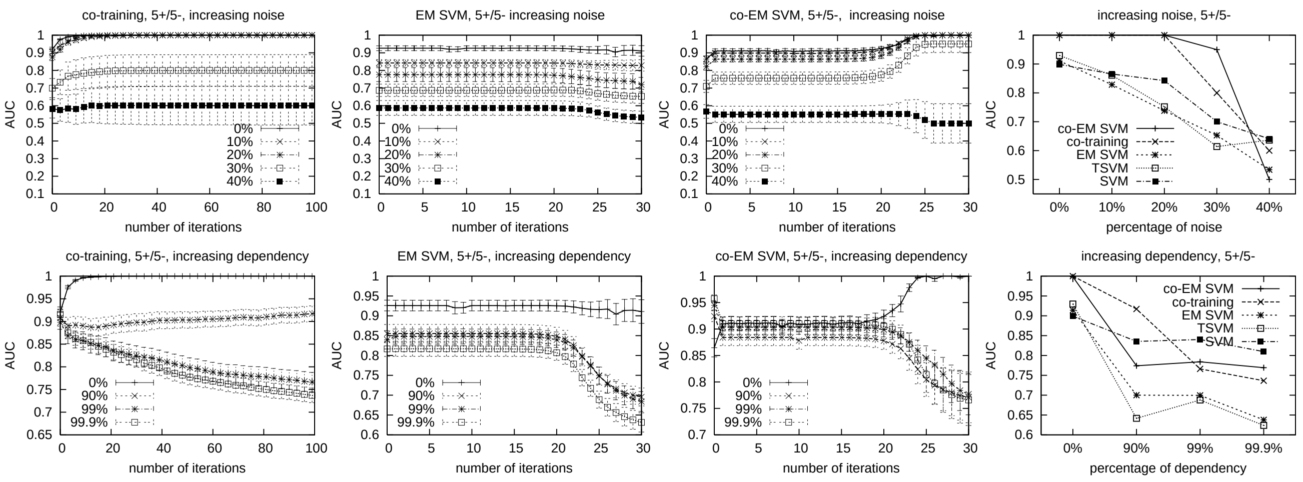

Figure 2. Semi-supervised support vector learning for the semi-artificial 20 newsgroups data set.

Table 3. Results for the course problem.

Method	Error rate
naive Bayes	13.0%
co-trained NB	5.0%
co-EM NB (65 labeled ex.)	$5.08 \pm 0.7\%$
SVM	$10.39\% \pm 0.7\%$
TSVM	$8.35\% \pm 0.7\%$
EM SVM	$8.02\% \pm 1.0\%$
co-trained SVM	$4.45\% \pm 0.9\%$
co-EM SVM	**$0.99\% \pm 1.3\%$**

naive Bayes (Blum & Mitchell, 1998; Nigam & Ghani, 2000; Muslea et al., 2002a).

Table 3 summarizes the results. After 100 rounds, the co-trained SVM achieves an error of 4.45% while the co-EM SVM outperforms all other support vector algorithms significantly with an error rate of 0.99%. Since 3 positive and 9 negative examples do not reflect the true prior distribution we used the natural ratio of 2 positive and 8 negative examples for shifting the decision hyperplane.

Do the obtained results hold for larger data sets? We conduct another set of experiments in which we discriminate each of the seven most frequent classes of the Reuters-21587 data set from all other classes. In each of the seven binary classification problems we draw 190 labeled examples (1% of the data) at random – the positive/negative ratio varies due to different class sizes – and obtain 18853 unlabeled examples that we use as hold out set as well. In each trial we randomly split the available attributes into two subsets; we average over 20 distinct samples and attribute splits. The results are shown in Figure 3.

Analogously to the course data set experiment, the multi-view algorithms outperform all other variants of supervised and semi-supervised support vector algorithms that we studied. Here, co-training beats the baseline SVM significantly in four out of six cases followed by the co-EM SVM with three out of five significant improvements.

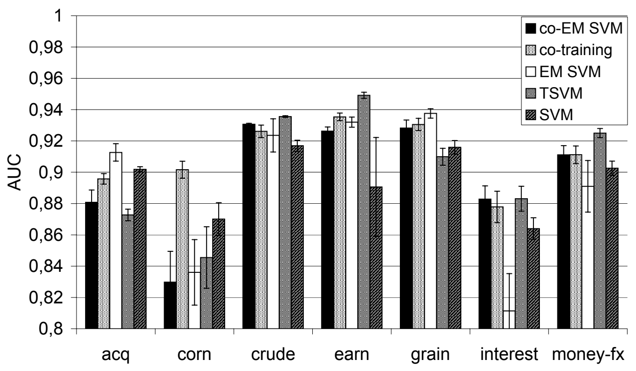

Figure 3. Results for the Reuters-21587 data set.

6. Conclusion

We developed a co-EM version of the Support Vector Machine. The co-EM SVM algorithm utilizes unlabeled data when the available attributes can be split into two independent subsets each of which has to be sufficient for learning. We observed that the co-EM SVM outperforms all other variations of semi-supervised SVM algorithms for the course problem, in most trials with the 20 newsgroups data set, and performs second-best for the Reuters data set. When we reduce the amount of unlabeled data, the performance of the co-EM SVM deteriorates less severely than the performance of co-training. The single-view counterpart of the co-EM SVM behaves similar to the transductive SVM. Furthermore, we found that multi-view learning improves the performance on the Reuters data set even though the views are generated by splitting the attributes at random.

Acknowledgment

This work has been supported by the German Science Foundation DFG under grant SCHE540/10-1.

References

Baluja, S. (1998). Probabilistic modeling for face orientation discrimination: Learning from labeled and unlabeled data. *Advances in Neural Information Processing Systems.*

Bennett, K. (1999). Combining support vector and mathematical programming methods for classification. *Advances in Kernel Methods - Support Vector Learning.* MIT Press.

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. *Proceedings of the Conference on Computational Learning Theory* (pp. 92–100).

Bradley, A. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, *30*, 1145–1159.

Brefeld, U., Geibel, P., & Wysotzki, F. (2003). Support vector machines with example dependent costs. *Proceedings of the European Conference on Machine Learning.*

Collins, M., & Singer, Y. (1999). Unsupervised models for named entity classification. *Proceedings of the Conference on Empirical Methods in Natural Language Processing.*

Cooper, D., & Freeman, J. (1970). On the asymptotic improvement in the outcome of supervised learning provided by additional nonsupervised learning. *IEEE Transactions on Computers*, *C-19*, 1055–1063.

Cozman, F., Cohen, I., & Cirelo, M. (2003). Semi-supervised learning of mixture models. *Proceedings of the International Conference on Machine Learning* (pp. 99–106).

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, *39*.

Denis, F., Laurent, A., Gilleron, R., & Tommasi, M. (2003). Text classification and co-training from positive and unlabeled examples. *ICML Workshop on the Continuum from Labeled to Unlabeled Data.*

Ghani, R. (2002). Combining labeled and unlabeled data for multiclass text categorization. *Proceedings of the International Conference on Machine Learning.*

Joachims, T. (1999a). Making large-scale SVM learning practical. *Advances in Kernel Methods - Support Vector Learning.* MIT Press.

Joachims, T. (1999b). Transductive inference for text classification using support vector machines. *Proceedings of the International Conference on Machine Learning.*

Joachims, T. (2003). Transductive learning via spectral graph partitioning. *Proceedings of the International Conference on Machine Learning.*

Kiritchenko, S., & Matwin, S. (2002). *Email classification with co-training* (Technical Report). University of Ottawa.

Kockelkorn, M., Lüneburg, A., & Scheffer, T. (2003). Using transduction and multi-view learning to answer emails. *Proceedings of the European Conference on Principle and Practice of Knowledge Discovery in Databases.*

McCallum, A., & Nigam, K. (1998). Employing EM in pool-based active learning for text classification. *Proceedings of the International Conference on Machine Learning.*

Mladenic, D. (2002). Learning word normalization using word suffix and context from unlabeled data. *Proceedings of the International Conference on Machine Learning* (pp. 427–434).

Muslea, I., Kloblock, C., & Minton, S. (2002a). Active + semi-supervised learning = robust multi-view learning. *Proceedings of the International Conference on Machine Learning* (pp. 435–442).

Muslea, I., Kloblock, C., & Minton, S. (2002b). Adaptive view validation: A first step towards automatic view detection. *Proceedings of the International Conference on Machine Learning* (pp. 443–450).

Nigam, K., & Ghani, R. (2000). Analyzing the effectiveness and applicability of co-training. *Proceedings of Information and Knowledge Management.*

Nigam, K., McCallum, A. K., Thrun, S., & Mitchell, T. M. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning*, *39*.

Provost, F., Fawcett, T., & Kohavi, R. (1998). The case against accuracy estimation for comparing inductive algorithms. *Proceedings of the International Conference on Machine Learning* (pp. 445–453).

Seeger, M. (2001). *Learning with labeled and unlabeled data.* (Technical Report). University of Edinburgh.

Shahshahani, B., & Landgrebe, D. (1994). The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon. *IEEE Transactions on Geoscience and Remote Sensing*, *32*, 1087–1095.

Vapnik, V. (1998). *Statistical learning theory.* Wiley.