Semi-Supervised Learning for Structured Output Variables

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

Abstract

The problem of learning a mapping between input and structured, interdependent output variables covers sequential, spatial, and relational learning as well as predicting recursive structures. Joint feature representations of the input and output variables have paved the way to leveraging discriminative learners such as SVMs to this class of problems. We address the problem of semi-supervised learning in joint input output spaces. The co-training approach is based on the principle of maximizing the consensus among multiple independent hypotheses; we develop this principle into a semi-supervised support vector learning algorithm for joint input output spaces and arbitrary loss functions. Experiments investigate the benefit of semi-supervised structured models in terms of accuracy and F1 score.

1. Introduction

Learning mappings between arbitrary structured and interdependent input and output spaces is a fundamental problem in machine learning; it covers many natural learning tasks and it challenges the standard model of learning a mapping from independently drawn instances to a small set of labels. Applications of the problem setting of learning with structured output variables include named entity recognition and information extraction (sequential output), natural language parsing (tree-structured output), classification with a class taxonomy – here, the output is a node in a tree –, and collective classification where the output is a set of interdependent class variables.

When the input $\mathbf{x}$ and the desired output $\mathbf{y}$ are structures, it is not generally feasible to model each possible value of $\mathbf{y}$ as an individual class. In addition, not only may there be dependencies between the components of $\mathbf{x}$ (*e.g.,* words of a sentence), but also between the components of $\mathbf{y}$ (for instance, the class labels of hyperlinked web pages), and between the components of $\mathbf{x}$ and $\mathbf{y}$ (the semantic annotation of a word may depend on that word, as well as its neighbors). In order to capture these dependencies it is helpful to represent input and output pairs in a joint feature representation. The learning task is therefore rephrased as finding a function $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ such that

$$\hat{\mathbf{y}} = \operatorname*{argmax}_{\bar{\mathbf{y}} \in \mathcal{Y}} f(\mathbf{x}, \bar{\mathbf{y}})$$

is the desired output for any input $\mathbf{x}$. Thus, f can be a linear discriminator in a joint space $\Phi(\mathbf{x}, \mathbf{y})$ of input and output variables and may depend on arbitrarily defined joint features. Max-margin Markov models (Taskar et al., 2004), support vector machines for structured output spaces (Tsochantaridis et al., 2005) and other discriminative learners exploit this principle.

In many areas, labeled data are rare while unlabeled examples are inexpensive and readily available. For discriminative learning, the co-training principle has proven to be an effective mechanism for utilizing unlabeled data. It is based on the observation that the rate of disagreement between independent hypotheses upper-bounds their individual error rate (Dasgupta et al., 2001). We contribute to the field by leveraging this principle to general joint spaces of input and output variables, and empirically studying it for multi-class classification, sequential learning, and parsing.

The rest of our paper is structured as follows. After reviewing related work in Section 2 and defining the learning setting and feature representation in Section 3, Section 4 presents the co-support vector machine for input output spaces. We report on experimental results in Section 5. Section 6 concludes.

2. Related Work

In the last years, several discriminative algorithms have been studied that utilize joint spaces of input and

Appearing in *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006. Copyright 2006 by the author(s)/owner(s).

output variables; these include max-margin Markov models (Taskar et al., 2004), kernel conditional random fields (Lafferty et al., 2004), hidden Markov support vector machines (Altun et al., 2003), and support vector machines for structured output spaces (Tsochantaridis et al., 2005). These methods utilize kernels to compute the inner product in input output space. Not only does this approach allow to capture long-distance dependencies between inputs and outputs, but it is also sufficiently versatile to cover other structures of outputs, such as (parse) trees, lattices, or graphs. An application-specific learning method is constructed by defining appropriate features, and choosing a decoding procedure that efficiently calculates the argmax, exploiting the dependency structure of the features.

Multi-view methods naturally allow the inclusion of unlabeled examples in discriminative learning. The co-training (Blum & Mitchell, 1998) and co-EM algorithms (Nigam & Ghani, 2000) iteratively increment the consensus of independent hypotheses by exchanging conjectured labels for unlabeled data. Recently, Hardoon et al. (2006) propose a fully supervised variant of a co-support vector machine that minimizes the training error as well as the disagreement between two views.

Dasgupta et al. (2001) give PAC bounds on the error of co-training in terms of the disagreement rate of hypotheses on unlabeled data in two independent views. A corollary of their results that holds under general assumptions is the inequality

$$Pr(f^1 \neq f^2) \geq \max\{Pr(err(f^1)), Pr(err(f^2))\}.$$

That is, the probability that two independent hypotheses disagree upper-bounds the error rate of either hypothesis. Thus, the strategy of semi-supervised multi-view learning can be stated as: Minimize the error for labeled examples and maximize the agreement for unlabeled examples.

Recently, two semi-supervised approaches to label sequence learning have been proposed. Altun et al. (2006) integrate Laplacian priors into structured large margin classifiers for pitch accent prediction. Brefeld et al. (2005) study semi-supervised sequential learning with 0/1 loss.

3. Learning in Input Output Spaces

Input examples $\mathbf{x} \in \mathcal{X}$ and output examples $\mathbf{y} \in \mathcal{Y}$ are represented jointly by a feature map $\Phi(\mathbf{x}, \mathbf{y})$ that allows to capture multiple-way dependencies between inputs and outputs. We apply a generalized linear model $f(\mathbf{x}, \mathbf{y}) = \langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle$ to decode the top scoring output for a given input

$$\hat{\mathbf{y}} = \operatorname*{argmax}_{\bar{\mathbf{y}} \in \mathcal{Y}} f(\mathbf{x}, \bar{\mathbf{y}}). \tag{1}$$

We measure the quality of f by an appropriate, symmetric, nonnegative loss function $\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_0^+$ that details the distance between the true $\mathbf{y}$ and the prediction $\operatorname{argmax}_{\bar{\mathbf{y}}} f(\mathbf{x}, \bar{\mathbf{y}})$; for instance, Δ may be the common 0/1 loss, given by $\Delta(\mathbf{y}, \mathbf{y}') = [[\mathbf{y} \neq \mathbf{y}']]$, where we introduce the indicator function $[[z]] = 1$ if z is *true* and 0 otherwise. Thus, we can restate the optimization problem as finding a function f that minimizes the expected risk

$$R(f) = \int_{\mathcal{X} \times \mathcal{Y}} \Delta\left(\mathbf{y}, \operatorname{argmax}_{\bar{\mathbf{y}}} f(\mathbf{x}_i, \bar{\mathbf{y}})\right) \, dP_{\mathcal{X} \times \mathcal{Y}}(\mathbf{x}, \mathbf{y}),$$

where $P_{\mathcal{X} \times \mathcal{Y}}$ is the (unknown) distribution of inputs and outputs. As in the classical classification setting, we address this problem by searching a minimizer of the empirical risk

$$R_{emp}(f) = \sum_{i=1}^{n} \Delta\left(\mathbf{y}_i, \operatorname{argmax}_{\bar{\mathbf{y}}} f(\mathbf{x}_i, \bar{\mathbf{y}})\right),$$

that is defined on a fixed *iid* sample from $P_{\mathcal{X} \times \mathcal{Y}}$. In the following, we will refer to a sample of n labeled pairs $(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n)$ and m unlabeled examples $\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+m}$, where in general $m \gg n$ holds. The $\mathbf{x}_i \in \mathcal{X}$ denote the i-th input and $\mathbf{y}_i \in \mathcal{Y}$ the corresponding output for labeled examples. Each element in $\mathcal{Y}$ is a composition of elements of the output alphabet Σ.

In the co-learning setting that we discuss here the available attributes $\Phi(\mathbf{x}, \mathbf{y})$ are decomposed into disjoint sets $\Phi^0(\mathbf{x}, \mathbf{y})$ and $\Phi^1(\mathbf{x}, \mathbf{y})$. The spaces spanned by Φ^v, $v = 0, 1$, are called views; *e.g.,* in hypertext classification we have two natural views on a page, either by the contained text or by the anchor text of its inbound links. The representation in each view has to be sufficient for the decoding.

The joint feature map $\Phi(\mathbf{x}, \mathbf{y})$ and the decoding have to be adapted on the application at hand. We present exemplary joint feature mappings and corresponding decoding algorithms in the following sections and report on empirical results in Section 5.

3.1. Multi-Class Classification

We begin with multi-class classification as an introductory example. Multi-class classification can be seen as a special case of learning in joint input output space where the output space equals the output alphabet; *i.e.,* $\mathcal{Y} = \Sigma$ (*e.g.,* compare Crammer & Singer, 2001).

We differentiate between an object (*e.g.,* a text document) $\mathbf{x}$ and its feature vector (*e.g., its tf.idf* vector) $\psi(\mathbf{x})$ and define the class-based feature representation $\phi^\sigma(\mathbf{x}, \mathbf{y})$ by

$$\phi^\sigma(\mathbf{x}, \mathbf{y}) = [[\mathbf{y} = \sigma]]\psi(\mathbf{x}), \tag{2}$$

with $\sigma \in \Sigma$. The joint feature representation is given by stacking up the class-based representations of all classes $\sigma \in \Sigma$

$$\Phi(\mathbf{x}, \mathbf{y}) = (\ldots, \phi^\sigma(\mathbf{x}, \mathbf{y}), \ldots). \tag{3}$$

With this definition, the inner product in input output space reduces to

$$\langle \Phi(\mathbf{x}_i, \mathbf{y}_i), \Phi(\mathbf{x}_j, \mathbf{y}_j) \rangle = [[\mathbf{y}_i = \mathbf{y}_j]]k(\mathbf{x}_i, \mathbf{x}_j),$$

for arbitrary $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \psi(\mathbf{x}_i), \psi(\mathbf{x}_j) \rangle$. Since the number of classes is fixed we do not need an efficient decoding strategy of Equation 1. Instead we compute $f(\mathbf{x}, \bar{\mathbf{y}})$ explicitly for all $\bar{\mathbf{y}} \in \mathcal{Y}$ and return the highest scoring class.

3.2. Label Sequence Learning

In label sequence learning the task is to find a mapping from a sequential input $\mathbf{x}_i = \langle x_{i,1}, \ldots, x_{i,T_i} \rangle$ to a sequential output $\mathbf{y}_i = \langle y_{i,1}, \ldots, y_{i,T_i} \rangle$, where $y_i \in \Sigma$. Each element of $\mathbf{x}$ is annotated with an element of Σ.

We follow Altun et al. (2003) and extract *label-label* interactions $\phi_{\sigma,\tau}(\mathbf{y}|t) = [[y_{t-1} = \sigma \wedge y_t = \tau]]$ and *label-observation* features $\bar{\phi}_{\sigma,j}(\mathbf{x}, \mathbf{y}|t) = [[y_t = \sigma]]\psi_j(x_t)$, with labels $\sigma, \tau \in \Sigma$. Here, $\psi_j(x)$ extracts characteristics of x *e.g.,* $\psi_{123}(x) = 1$ if x starts with a capital letter and 0 otherwise. We will refer to the vector $\psi(x) = (\ldots, \psi_j(x), \ldots)^\mathsf{T}$ and denote the inner product by means of $k(x, \bar{x}) = \langle \psi(x), \psi(\bar{x}) \rangle$.

We define the joint feature representation $\Phi(\mathbf{x}, \mathbf{y})$ of a sequence as the sum of all feature vectors $\Phi(\mathbf{x}, \mathbf{y}|t) = (\ldots, \phi_{\sigma,\tau}(\mathbf{y}|t), \ldots, \bar{\phi}_{\sigma,j}(\mathbf{x}, \mathbf{y}|t), \ldots)^\mathsf{T}$ extracted at position t

$$\Phi(\mathbf{x}, \mathbf{y}) = \sum_{t=1}^{T} \Phi(\mathbf{x}, \mathbf{y}|t).$$

The inner product in input output space decomposes into a *label-label* and a *label-observation* part,

$$\begin{aligned}\langle \Phi(\mathbf{x}_i, \mathbf{y}_i), \Phi(\mathbf{x}_j, \mathbf{y}_j) \rangle = \\ \sum_{s,t} [[y_{i,s-1} = y_{j,t-1} \wedge y_{i,s} = y_{j,t}]] \\ + \sum_{s,t} [[y_{i,s} = y_{j,t}]] k(x_{i,s}, x_{j,t}).\end{aligned}$$

Note that the described feature mapping exhibits a first-order Markov property and as a result, decoding can be performed by a Viterbi algorithm.

3.3. Natural Language Parsing

The goal in natural language parsing is to predict the parse tree $\mathbf{y}$ that generates a given input sentence $\mathbf{x} = \langle x_1, \ldots, x_T \rangle$. Each node in the tree $\mathbf{y}$ is generated by a rule of a weighted context-free grammar Σ that we assume to be in Chomsky normal form. Thus, the output alphabet Σ consists of unary and binary production rules. Binary rules are of the form $A \to BC$, where capital letters indicate non-terminal symbols. An example for a binary rule is $VP \to V, NP$ that substitutes a verb phrase VP by a verb V and a noun phrase NP (compare Figure 1). The head of unary production rules is either substituted by another non-terminal symbol (*e.g.,* $NP \to N$, a noun phrase can be substituted by a noun) or by a terminal symbol (*e.g.,* $N \to cat$).

According to Tsochantaridis et al. (2005) we extract local features from each production rule $\sigma \in \Sigma$

$$\phi^\sigma(\mathbf{x}, \mathbf{y}) = (\ldots, \phi^\sigma_j(\mathbf{x}, \mathbf{y}), \ldots). \tag{4}$$

Thus, the elements of $\phi^\sigma(\mathbf{x}, \mathbf{y})$ count for instance how many times production rule σ occurs in the parse tree $\mathbf{y}$ or how often this rule is at a border.

The joint feature map $\Phi(\mathbf{x}, \mathbf{y}) = (\ldots, \phi^\sigma(\mathbf{x}, \mathbf{y}), \ldots)^\mathsf{T}$ is constructed by stacking up the production rule features $\phi^\sigma(\mathbf{x}, \mathbf{y})$ for each production $\sigma \in \Sigma$. After the learning process the entries of the weight vector may be interpreted as scores that indicate how likely a certain production rule is to be applied given the local context. This feature map gives rise to the following inner product in input output space

$$\langle \Phi(\mathbf{x}_i, \mathbf{y}_i), \Phi(\mathbf{x}_j, \mathbf{y}_j) \rangle = \sum_{\sigma,k} \phi^\sigma_k(\mathbf{x}_i, \mathbf{y}_i) \cdot \phi^\sigma_k(\mathbf{x}_j, \mathbf{y}_j).$$

Note that this feature map allows the use of the CKY parser as efficient decoding procedure (Manning & Schütze, 1999).

4. Co-Support Vector Learning for Structured Output Variables

In this section we present the co-support vector machines for structured output variables and arbitrary loss functions. We briefly review the single-view variant (Tsochantaridis et al., 2005) and extend it to semi-supervised learning.

4.1. Support Vector Learning for Structured Output Variables

The goal in predicting structured output variables is to learn a linear discriminant function $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$

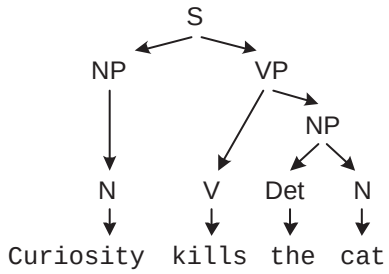

Figure 1. Example of a parse tree.

given by $f(\mathbf{x}, \mathbf{y}) = \langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle$ that correctly decodes any output $\mathbf{y}_i$ of the training sample $(\mathbf{x}_i, \mathbf{y}_i) \in D$; *i.e.,*

$$\mathbf{y}_i = \operatorname*{argmax}_{\bar{\mathbf{y}} \in \mathcal{Y}} f(\mathbf{x}_i, \bar{\mathbf{y}}).$$

This is the case if there exists $\gamma \geq 0$ such that

$$f(\mathbf{x}_i, \mathbf{y}_i) - \max_{\bar{\mathbf{y}} \neq \mathbf{y}_i} f(\mathbf{x}_i, \bar{\mathbf{y}}) \geq \gamma \tag{5}$$

holds for all $i = 1, \ldots, n$. The scalar γ is called the functional margin. Support vector machines enforce confident predictions by maximizing the geometrical margin $\gamma / \|\mathbf{w}\|$; setting $\gamma = 1$ leads us directly to the following hard margin optimization problem.

Optimization Problem 1 *Given n labeled examples; over all $\mathbf{w}$ minimize $\frac{1}{2}\|\mathbf{w}\|^2$ subject to the constraints $\forall_{i=1}^{n}$, $\forall_{\bar{\mathbf{y}} \neq \mathbf{y}_i} \langle \mathbf{w}, \Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \bar{\mathbf{y}}) \rangle \geq 1$.*

In general, we have to allow pointwise relaxations of the hard margin constraints by slack variables. Each slack variable ξ_i is bound to an input example $\mathbf{x}_i$ (Crammer & Singer, 2001), leading to the following soft-margin optimization problem.

Optimization Problem 2 *Given n labeled examples, let $C > 0$ and $r = 1, 2$; over all $\mathbf{w}$ and ξ_i minimize $\frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{r}\sum_{i=1}^{n} \xi_i^r$ subject to the constraints $\forall_{i=1}^{n} \xi_i \geq 0$ and $\forall_{i=1}^{n}, \forall_{\bar{\mathbf{y}} \neq \mathbf{y}_i} \langle \mathbf{w}, \Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \bar{\mathbf{y}}) \rangle \geq 1 - \xi_i$.*

The parameter $r = 1, 2$ denotes a linear or quadratic penalization of the error, respectively, $C > 0$ determines the trade-off between margin maximization and error minimization. The sum $\sum_i \xi_i$ upper-bounds the empirical risk with common 0/1 loss. This, however, might not be the best choice for several applications (Joachims, 2005). Recently, two distinct ways of integrating a loss function Δ into structured optimization problems have been discussed; *i.e.,* a margin rescaling approach (Taskar et al., 2004) and a slack rescaling approach (Tsochantaridis et al., 2005). We follow the latter since rescaling the slack variables still allows the sum $\sum \xi_i$ to be interpreted as an upper bound on the empirical risk which is not the case for rescaling the margin. Note, that our approach is also easily generalizable to the margin rescaling case. The extension of Optimization Problem 2 to arbitrary loss functions leads us to the following optimization problem.

Optimization Problem 3 *Given n labeled examples, loss function $\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_0^+$, tradeoff $C > 0$, and $r = 1, 2$; over all $\mathbf{w}$ and ξ_i minimize $\frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{r}\sum_{i=1}^{n} \xi_i^r$ subject to the constraints $\forall_{i=1}^{n} \xi_i \geq 0$ and $\forall_{i=1}^{n}, \forall_{\bar{\mathbf{y}} \neq \mathbf{y}_i} \langle \mathbf{w}, \Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \bar{\mathbf{y}}) \rangle \geq 1 - \frac{\xi_i}{\sqrt[r]{\Delta(\mathbf{y}_i, \bar{\mathbf{y}})}}$.*

Tsochantaridis et al. (2005) derive corresponding 1- and 2-norm dual optimization problems and propose an iterative optimization algorithm that is proven to converge to the optimal solution in polynomial time. If Δ is the 0/1 loss, Optimization Problems 2 and 3 are equivalent.

4.2. Semi-Supervised Support Vector Learning for Structured Output Variables

In the co-learning setting that we discuss here we have n labeled examples $(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n) \in D^l$ and m unlabeled inputs $\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+m} \in D^u$. The joint decision function is given by the sum $f(\mathbf{x}, \mathbf{y}) = f^0(\mathbf{x}, \mathbf{y}) + f^1(\mathbf{x}, \mathbf{y})$, where each view $f^v(\mathbf{x}, \mathbf{y}) = \langle \mathbf{w}^v, \Phi^v(\mathbf{x}, \mathbf{y}) \rangle$, $v = 0, 1$ has its respective feature map. According to the consensus maximizing principle, the co-support vector machine now has to minimize the number of errors for labeled examples and the disagreement for the unlabeled examples.

For view v this is the case, if the labeled examples fulfill Equation 5 while for the unlabeled examples

$$f^v(\mathbf{x}_i, \hat{\mathbf{y}}_i^{\bar{v}}) - \max_{\bar{\mathbf{y}} \neq \mathbf{y}_i} f^v(\mathbf{x}_i, \bar{\mathbf{y}}) = \gamma_i^v \geq 1 \tag{6}$$

holds for all $i = n+1, \ldots, n+m$. The structure $\hat{\mathbf{y}}_i^{\bar{v}}$ denotes the prediction of the peer view $\bar{v}$ that is treated as correct output for the i-th unlabeled input example. In the following, we omit the superscript v and use the superscript $\bar{v}$ to indicate variables of the peer view. Optimization Problem 3 can be rephrased in the co-learning setting as follows.

Optimization Problem 4 *Given n labeled examples and m unlabeled examples, loss function Δ, let $C, C_u > 0$, $r = 1, 2$, and $v = 0, 1$; over all $\mathbf{w}$ and ξ minimize $\frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{r}\left(\sum_{i=1}^{n} \xi_i^r + C_u \sum_{i=n+1}^{n+m} (\min\{\gamma_i^{\bar{v}}, 1\}) \xi_i^r\right)$ subject to the constraints $\forall_{i=1}^{n+m} \xi_i \geq 0$ and $\forall_{i=1}^{n}, \forall_{\bar{\mathbf{y}} \neq \mathbf{y}_i} \langle \mathbf{w}, \Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \bar{\mathbf{y}}) \rangle \geq 1 - \frac{\xi_i}{\sqrt[r]{\Delta(\mathbf{y}_i, \bar{\mathbf{y}})}}$, $\forall_{i=n+1}^{n+m}, \forall_{\bar{\mathbf{y}} \neq \mathbf{y}^{\bar{v}}} \langle \mathbf{w}, \Phi(\mathbf{x}_i, \mathbf{y}_i^{\bar{v}}) - \Phi(\mathbf{x}_i, \bar{\mathbf{y}}) \rangle \geq 1 - \frac{\xi_i}{\sqrt[r]{\Delta(\mathbf{y}_i^{\bar{v}}, \bar{\mathbf{y}})}}$.*

Algorithm 1 CoSVM Optimization Algorithm

Input: i-th unlabeled example $\mathbf{x}_i$, $S^0_{j\neq i}$, $S^1_{j\neq i}$, C, C_u, norm r, repetitions r_{max}.

Set $S^0_i = S^1_i = \emptyset$, $\alpha^0_{i,\mathbf{y}} = \alpha^1_{i,\mathbf{y}} = 0$ for all $\mathbf{y} \in \mathcal{Y}$
repeat
  for each view $v = 0, 1$ **do**
    $\hat{\mathbf{y}}^v = \text{argmax}_{\mathbf{y}} \langle \mathbf{w}^v, \Phi^v(\mathbf{x}_i, \mathbf{y}) \rangle$
    $\bar{\mathbf{y}}^v = \text{argmax}_{\mathbf{y} \neq \hat{\mathbf{y}}^v} (1 - \langle \mathbf{w}^v, \Phi^v_{i,\hat{\mathbf{y}}^v,\mathbf{y}} \rangle) \sqrt[r]{\Delta(\hat{\mathbf{y}}^v, \mathbf{y})}$
    $\xi^v_i = \max_{\mathbf{y} \in S^v_i} \{(1 - \langle \mathbf{w}, \Phi^v_{i,\hat{\mathbf{y}}^v,\mathbf{y}} \rangle) \sqrt[r]{\Delta(\hat{\mathbf{y}}^v, \mathbf{y})}\}$
    $\gamma^v = f^v(\mathbf{x}_i, \hat{\mathbf{y}}^v) - f^v(\mathbf{x}_i, \bar{\mathbf{y}}^v)$
  end for
  if $[[\hat{\mathbf{y}}^0 \neq \hat{\mathbf{y}}^1]] \vee [[\langle \mathbf{w}^v, \Phi^v_{i,\hat{\mathbf{y}}^v,\bar{\mathbf{y}}^v} \rangle < 1 - \frac{\xi^v_i}{\sqrt[r]{\Delta(\hat{\mathbf{y}}^v, \bar{\mathbf{y}}^v)}}]]$, $v = 0, 1$ **then**
    for each view $v = 0, 1$ **do**
      Substitute former target $\mathbf{y}^v_i = \hat{\mathbf{y}}^{\bar{v}}$
      if $[[\hat{\mathbf{y}}^0 \neq \hat{\mathbf{y}}^1]]$ **then**
        $S^v_i = S^v_i \cup \{\hat{\mathbf{y}}^v\}$
      else
        $S^v_i = S^v_i \cup \{\bar{\mathbf{y}}^v\}$
      end if
      Optimize $\alpha^v_{i,\bar{\mathbf{y}}}$ over S^v_i with $S^v_{j\neq i}$ fixed
      $\forall \bar{\mathbf{y}} \in S^v$ with $\alpha^v_{i,\bar{\mathbf{y}}} = 0$: $S^v_i = S^v_i \backslash \{\bar{\mathbf{y}}\}$
    end for
  end if
until consensus or r_{max} repetitions

Output: Optimized α^0_i and α^1_i, sets S^0_i and S^1_i

The balancing factor C_u regularizes the influence of the unlabeled data. Weights of $\min\{\gamma^{\bar{v}}_i, 1\}$ to the slacks $\xi_{n+1}, \ldots, \xi_{n+m}$ relate errors on unlabeled examples to the confidence (*i.e.,* margin) of the peer view's prediction. Thus, an unlabeled sequence that satisfies the margin constraint has the same influence in the peer view as a labeled example.

The sum of the slack variables now consists of an upper bound on the error for the labeled examples and an upper bound on the disagreement weighted by the confidence of the peer view's prediction. Analogously to the single-view case, the effective influence of the loss function Δ can be adjusted by the trade-off C.

Note that the joint objective function that is given by the sum of the objective functions of both views reduces to that of the transductive SVM (Joachims, 1999) in the case of identical views.

Similarly to the regular support vector machine, the constraints of Optimization Problem 4 can be integrated into the objective function by introducing non-negative Lagrange multipiers $\alpha_{i,\bar{\mathbf{y}}}$ for all $i = 1, \ldots, n+m$ and every $\bar{\mathbf{y}} \in \mathcal{Y}$. Taking the derivative of the Lagrangian with respect to the weight vector $\mathbf{w}$ leads to its dual representation $\mathbf{w} = \sum_{i=1}^{n+m} \sum_{\bar{\mathbf{y}} \neq \mathbf{y}_i} \alpha_{i,\bar{\mathbf{y}}} \Phi_{i,\mathbf{y}^*_i,\bar{\mathbf{y}}}$, where we use $\Phi_{i,\mathbf{y}^*_i,\bar{\mathbf{y}}}$ shorthand for difference vectors

$$\Phi_{i,\mathbf{y}^*_i,\bar{\mathbf{y}}} = \Phi(\mathbf{x}_i, \mathbf{y}^*_i) - \Phi(\mathbf{x}_i, \bar{\mathbf{y}})$$

with output variable $\mathbf{y}^*_i = \mathbf{y}^v_i$ if $1 \leq i \leq n$ and $\mathbf{y}^*_i = \mathbf{y}^{\bar{v}}_i$ for unlabeled examples with $n+1 \leq i \leq n+m$.

Given the derivative with respect to the ξ_i and substituting all derivatives into the Lagrangian removes its dependence on the primal variables and we resolve the corresponding dual optimization problem that has to be maximized with respect to the $\alpha_{i,\bar{\mathbf{y}}}$. For $r = 1$ we derive the following 1-norm co-support vector machine optimization problem.

Optimization Problem 5 *Given n labeled and m unlabeled examples, loss function Δ, $C, C_u > 0$; over all $\alpha_{i,\bar{\mathbf{y}}}$ maximize*

$$\sum_{i=1}^{n+m} \sum_{\bar{\mathbf{y}} \neq \mathbf{y}_i} \alpha_{i,\bar{\mathbf{y}}} - \frac{1}{2} \sum_{i,j=1}^{n+m} \sum_{\substack{\bar{\mathbf{y}} \neq \mathbf{y}_i \\ \bar{\mathbf{y}}' \neq \mathbf{y}_j}} \alpha_{i,\bar{\mathbf{y}}} \alpha_{j,\bar{\mathbf{y}}'} K\left((\mathbf{x}_i, \bar{\mathbf{y}}), (\mathbf{x}_j, \bar{\mathbf{y}}')\right)$$

subject to the constraints $\forall_{i=1}^{n} \sum_{\bar{\mathbf{y}} \neq \mathbf{y}_i} \frac{\alpha_{i,\bar{\mathbf{y}}}}{\Delta(\mathbf{y}_i, \bar{\mathbf{y}})} \leq C$, $\forall_{i=n+1}^{n+m} \sum_{\bar{\mathbf{y}} \neq \mathbf{y}^{\bar{v}}_i} \frac{\alpha_{i,\bar{\mathbf{y}}}}{\Delta(\mathbf{y}^{\bar{v}}_i, \bar{\mathbf{y}})} \leq (\min\{\gamma^{\bar{v}}_i, 1\}) C_u C$, *and* $\forall_{i=1}^{n+m} \forall_{\bar{\mathbf{y}} \neq \mathbf{y}_i} \alpha_{i,\bar{\mathbf{y}}} \geq 0$.

The composite kernel $K\left((\mathbf{x}_i, \bar{\mathbf{y}}), (\mathbf{x}_j, \bar{\mathbf{y}}')\right)$ computes the inner product of two difference vectors in input output space and is given by

$$\begin{aligned} &K\left((\mathbf{x}_i, \bar{\mathbf{y}}), (\mathbf{x}_j, \bar{\mathbf{y}}')\right) \\ &\quad = \langle \Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \bar{\mathbf{y}}) \,,\, \Phi(\mathbf{x}_j, \mathbf{y}_j) - \Phi(\mathbf{x}_j, \bar{\mathbf{y}}') \rangle \\ &\quad = \langle \Phi(\mathbf{x}_i, \mathbf{y}_i), \Phi(\mathbf{x}_j, \mathbf{y}_j) \rangle - \langle \Phi(\mathbf{x}_i, \mathbf{y}_i), \Phi(\mathbf{x}_j, \bar{\mathbf{y}}') \rangle \\ &\qquad - \langle \Phi(\mathbf{x}_i, \bar{\mathbf{y}}), \Phi(\mathbf{x}_j, \mathbf{y}_j) \rangle + \langle \Phi(\mathbf{x}_i, \bar{\mathbf{y}}), \Phi(\mathbf{x}_j, \bar{\mathbf{y}}') \rangle. \end{aligned}$$

The method by Brefeld et al. (2005) can be obtained as a special case of Optimization Problem 5 for 0/1 loss and a sequential feature mapping.

For $r = 2$ we may drop the non-negativity constraints $\xi_i \geq 0$ of Optimization Problem 4 since $\xi_i < 0$ satisfies $\langle \mathbf{w}, \Phi_{i,\bar{\mathbf{y}}} \rangle \geq \sqrt{\Delta(\mathbf{y}_i, \bar{\mathbf{y}})} - \xi_i$ and $\sum_i \xi_i^2$ guarantees the objective to be positive. We derive the dual 2-norm co-support vector machine optimization problem by resubstituting the respective derivatives with respect to $\mathbf{w}$ and ξ_i into the Lagrangian.

Optimization Problem 6 *Given n labeled and m unlabeled examples, loss function Δ, $C, C_u > 0$; over all $\alpha_{i,\bar{\mathbf{y}}}$ maximize*

$$\sum_{i=1}^{n+m} \sum_{\bar{\mathbf{y}} \neq \mathbf{y}_i} \alpha_{i,\bar{\mathbf{y}}} - \frac{1}{2} \sum_{i,j=1}^{n+m} \sum_{\substack{\bar{\mathbf{y}} \neq \mathbf{y}_i \\ \bar{\mathbf{y}}' \neq \mathbf{y}_j}} \alpha_{i,\bar{\mathbf{y}}} \alpha_{j,\bar{\mathbf{y}}'} K'((\mathbf{x}_i, \bar{\mathbf{y}}), (\mathbf{x}_j, \bar{\mathbf{y}}'))$$

subject to the constraints $\forall_{i=1}^{n+m} \forall_{\bar{\mathbf{y}} \neq \mathbf{y}_i} \alpha_{i,\bar{\mathbf{y}}} \geq 0$.

Additional constraints are integrated into the kernel $K'((\mathbf{x}_i, \bar{\mathbf{y}}), (\mathbf{x}_j, \bar{\mathbf{y}}')) = K((\mathbf{x}_i, \bar{\mathbf{y}}), (\mathbf{x}_j, \bar{\mathbf{y}}')) + \delta_{i\bar{\mathbf{y}},j\bar{\mathbf{y}}'}$

Table 1. Error rates for the Cora data set.

	L:200			L:400		
	U:0	U:400	U:800	U:0	U:800	U:2000
SVM	46.74 ± 0.26	-	-	38.39 ± 0.22	-	-
TSVM	46.13 ± 0.41	48.54 ± 0.28	50.84 ± 0.30	37.65 ± 0.25	39.31 ± 0.45	42.72 ± 0.60
coSVM	$\mathbf{41.94 \pm 0.30}$	42.51 ± 0.33	41.52 ± 0.26	$\mathbf{32.80 \pm 0.22}$	32.79 ± 0.21	32.72 ± 0.26

where $\delta_{i\bar{\mathbf{y}},j\bar{\mathbf{y}}'}$ equals $(C\sqrt{\Delta(\mathbf{y}_i,\bar{\mathbf{y}})\Delta(\mathbf{y}_j,\bar{\mathbf{y}}')})^{-1}$ if $1 \leq i = j \leq n$ (*i.e.,* in case of a labeled example), $((\min\{\gamma_j^{\bar{v}},1\})C_u\, C\sqrt{\Delta(\mathbf{y}_i^{\bar{v}},\bar{\mathbf{y}})\Delta(\mathbf{y}_j^{\bar{v}},\bar{\mathbf{y}}')}\,)^{-1}$ if $n+1 \leq i = j \leq n+m$ (*i.e.,* in case of an unlabeled example), and 0 otherwise.

4.3. Optimization Strategy

Since the dual variables $\alpha_{i,\bar{\mathbf{y}}}$ are tied to observations $\mathbf{x}_i$, the dual optimization problem splits into $n+m$ disjoint subspaces spanned by $\alpha_{i,\bar{\mathbf{y}}}$ with fixed values for the $\alpha_{j\neq i,\bar{\mathbf{y}}}$.

In an outer loop, the co-support vector machine iterates over the examples and consecutively optimizes the example's parameters $\alpha_{i,\bar{\mathbf{y}}}$, using distinct working set approaches for labeled (similar to Tsochantaridis et al., 2005) and unlabeled (Algorithm 1) examples, adding a new constraint in each iteration if necessary.

Algorithm 1 computes the top scoring output (line 4) and its best runner-up (line 5) for both views and relates their difference to the current slack (line 6). In case of a disagreement or if a margin violation is detected in one of the views an update is performed (line 10-20). This optimization scheme leads to sparse models, since it suffices to store only those $\alpha_{i,\bar{\mathbf{y}}}$ explicitly whose associated output $\bar{\mathbf{y}}$ is decoded instead of the true $\mathbf{y}_i$. Outputs $\bar{\mathbf{y}}$ with $\alpha_{i,\bar{\mathbf{y}}} = 0$ are removed in order to speed up computation. When the loop reaches an example for the second time, all former outputs $\alpha_{i,\bar{\mathbf{y}}}$ of that example are removed since the errors or disagreements that they used to correct in earlier iterations of the main loop may have been resolved.

Since the cost factors upper-bound the growth of the $\alpha_{i,\bar{\mathbf{y}}}$, consensus might not be established and we therefore integrate a user defined constant r_{max} that bounds the number of iterations.

5. Empirical Results

We investigate our approach by applying the semi-supervised support vector machine to multi-class classification, named entity recognition, and natural language parsing. We explore the benefit of co-learning and investigate its execution time. The baseline SVM is described by Tsochantaridis et al. (2005).

In each setting, the influence of unlabeled examples is determined by a smoothing strategy which exponentially approaches C_u after a fixed number of epochs. We first optimize parameter C_u using resampling; we then fix C_u and present curves that show the average error over distinct randomly drawn training and holdout sets. The baseline methods are trained on concatenated views. We initialize $r_{max} = 10$, $C = 1$.

For all problems and sample sizes we conduct a one-sided t-test at a 1% confidence level. Significant results are indicated by entries in bold face in Tables 1-3.

5.1. Multi-Class Classification

Our multi-class classification experiments are based on the Cora data set that contains 9,947 linked computer science papers. We remove documents without a reference section and obtain 9,555 papers divided into 8 different classes.

We exploit the link structure and generate two natural views of the documents: a term frequency view of the document and an outlink view. In the latter we have one feature for each document pair $\mathbf{x}, \mathbf{x}'$ that equals 1 if document $\mathbf{x}$ cites document $\mathbf{x}'$ and is 0 otherwise.

We use the common 0/1 loss and the respective 2-norm variant of both structured prediction methods and the transductive SVM (Joachims, 1999) as an additional baseline. Table 1 details error rates and standard error in percent for different numbers of labeled (L) and unlabeled (U) training examples and 500 holdout examples. Results are averages over 100 repetitions with distinct training and holdout sets. The performance of the TSVM deteriorates when the number of unlabeled instances is increased. The co-trained SVM significantly outperforms its fully supervised counterpart for all numbers of labeled and unlabeled examples.

5.2. Label Sequence Learning

We study the effectiveness of our approach on two named entity recognition problems. We use the data set provided for task 1A of the Biocreative challenge and the Spanish news wire article corpus of the shared task of CoNLL 2002.

The Biocreative data contains 7500 sentences from biomedical papers; gene and protein names are to be

Table 2. Token error for the Biocreative (BC) and Spanish news wire (SN) data sets.

		L:5		L:10		L:20	
		U:0	U:25	U:0	U:50	U:0	U:100
BC	HMM	17.98 ± 0.69	-	14.32 ± 0.53	-	12.31 ± 0.23	-
	SVM	10.27 ± 0.16	-	9.70 ± 0.07	-	9.47 ± 0.05	-
	coSVM	$\mathbf{9.71 \pm 0.07}$	$\mathbf{9.54 \pm 0.08}$	$\mathbf{9.48 \pm 0.05}$	9.51 ± 0.05	9.4 ± 0.05	9.37 ± 0.06
SN	HMM	23.59 ± 2.00	-	20.04 ± 1.27	-	15.31 ± 0.78	-
	SVM	10.95 ± 0.18	-	9.98 ± 0.09	-	8.97 ± 0.08	-
	coSVM	13.86 ± 0.78	$\mathbf{10.28 \pm 0.14}$	11.26 ± 0.13	$\mathbf{9.60 \pm 0.11}$	11.73 ± 0.43	8.99 ± 0.09

recognized. We discriminate tokens that are parts of gene names against all other tokens. We utilize *label-observation* features like the token itself, letter 2,3 and 4-grams, and surface clues like capitalization, inclusion of Greek symbols, numbers, and others.

The CoNLL2002 data contains 9 label types which distinguish person, organization, location, and other names. We use 3100 sentences of between 10 and 40 tokens. The extracted *label-observation* features cover the token itself and surface clues.

We assure that each label occurs at least once in the labeled training data; otherwise, we discard and draw again. Each holdout set consists of 500 (Biocreative) and 300 (Spanish news wire) sentences, respectively.

Table 2 shows the token error in percent for coSVM, single-view SVM, and also for a supervised hidden Markov model as an additional baseline for both data sets. We utilize a distinct random split of the attributes in two views for each repetition. Both support vector algorithms beat the HMM significantly. The baseline SVM that utilizes only labeled examples is clearly outperformed by the semi-supervised SVM in all but one settings and is never significantly better.

5.3. Natural Language Parsing

For our natural language parsing experiments we learn an unlexicalized, weighted context-free grammar on subsets of the Penn Treebank Wall Street Journal corpus and the Negra corpus. Both are tagged with part-of-speech and completely annotated with syntactic structures.

We use the subsets 2-21 of the Wall Street Journal corpus that contain 39,833 sentences. We extract sentences of length of at most 15 words. From the annotations of the resulting 8,666 sentences we build a context-free grammar of approximately 4,800 distinct production rules.

The Negra corpus contains 20,602 sentences from a German news paper archive. We extract sentences of between 5 and 25 tokens. The resulting 14,137 sentences contain more than 26,700 production rules in Chomsky normal form.

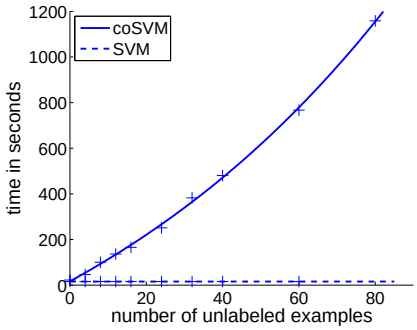

Figure 2. Execution time.

The extracted local feature maps contain the rule itself and binarized border and span width features for both corpora. Each result is averaged over 100 repetitions. In each repetition we use distinct, randomly chosen feature splits and randomly drawn training and holdout sets. The latter is of size 100. We use a modified variant of the CKY implementation by Johnson (1999) for the decoding and apply 2-norm SVMs with the loss $\Delta(\mathbf{y}_i, \bar{\mathbf{y}}) = 1 - F_1(\mathbf{y}_i, \bar{\mathbf{y}})$.

Table 3 details F1 scores for different numbers of labeled (L) and unlabeled (U) training instances for both corpora. Surprisingly, even for no unlabeled data coSVM leads to better F1 scores than regular SVM by simply averaging the predictions of the two views. When we add unlabeled instances, the performance of coSVM increases. Note, that additional unlabeled examples (40+200) further improve F1 score.

5.4. Execution Time

The observed performance benefits of coSVM are at the cost of significantly longer training processes. Figure 2 plots execution time against training set size for 4 labeled and different numbers of unlabeled examples. Empirically, we observe that the execution time of co-trained SVM scales between linearly and quadratically in the number of unlabeled examples.

Table 3. F1 scores for the wall street journal (WSJ) and the Negra (NEG) corpus.

		L:4		L:40		
		U:0	U:80	U:0	U:80	U:200
WSJ	SVM	45.40 ± 0.61	-	71.73 ± 0.29	-	-
	coSVM	$\mathbf{47.92 \pm 0.59}$	$\mathbf{48.23 \pm 0.55}$	$\mathbf{73.85 \pm 0.24}$	$\mathbf{74.07 \pm 0.25}$	$\mathbf{75.01 \pm 0.31}$
NEG	SVM	47.58 ± 0.37	-	63.70 ± 0.29	-	-
	coSVM	$\mathbf{48.81 \pm 0.37}$	$\mathbf{49.46 \pm 0.33}$	$\mathbf{64.94 \pm 0.27}$	$\mathbf{65.13 \pm 0.25}$	$\mathbf{65.70 \pm 0.25}$

5.5. Discussion

We observe that the co-trained support vector machine with no unlabeled examples outperforms the baseline methods in all tasks, significantly. We credit this finding to averaging two independently trained hypotheses. However, the prediction accuracy of coSVM can be further increased by adding unlabeled examples in the NER and parsing experiments.

6. Conclusion

We devised a semi-supervised variant of the support vector machine for structured output variables and arbitrary loss functions (coSVM). It is based on the co-training framework and implements the principle of consensus maximization between hypotheses. We derived 1- and 2-norm optimization problems that allow the use of arbitrary feature mappings and corresponding decoding strategies. We presented exemplary feature maps and decodings for three problem classes.

We used various norms, loss functions, and feature splits in our experiments. Empirical results for multi-class classification, named entity recognition, and natural language parsing tasks showed that coSVM leads to better models in terms of the chosen loss function compared to the fully-supervised SVM. However, this comes at the cost of a longer execution time. The semi-supervised support vector machine benefits from the inclusion of unlabeled examples into the training process. We observed that coSVM outperforms its single-view counterpart significantly in all tasks.

Acknowledgment

This work has been funded by the German Science Foundation DFG under grant SCHE540/10-2. We would like to thank Bernhard Schölkopf, Gabriele Schweikert, Georg Zeller, and Alexander Zien for inspiring discussions.

References

Altun, Y., McAllester, D., & Belkin, M. (2006). Maximum margin semi-supervised learning for structured variables. *Adv. in Neural Information Proc. Systems*.

Altun, Y., Tsochantaridis, I., & Hofmann, T. (2003). Hidden Markov support vector machines. *Proceedings of the International Conference on Machine Learning*.

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

Brefeld, U., Büscher, C., & Scheffer, T. (2005). Multi-view discriminative sequential learning. *Proceedings of the European Conference on Machine Learning*.

Crammer, K., & Singer, Y. (2001). On the algorithmic implementation of multi-class kernel-based vector machines. *Journal of Machine Learning Research*, *2*, 265–292.

Dasgupta, S., Littman, M., & McAllester, D. (2001). PAC generalization bounds for co-training. *Advances in Neural Information Processing Systems*.

Hardoon, D., Farquhar, J. D. R., Meng, H., Shawe-Taylor, J., & Szedmak, S. (2006). Two view learning: SVM-2K, theory and practice. *Advances in Neural Information Processing Systems*.

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

Joachims, T. (2005). A support vector method for multivariate performance measures. *Proceedings of the International Conference on Machine Learning*.

Johnson, M. (1999). PCFG models of linguistic tree representations. *Computational Linguistics*, *24(4)*, 613–632.

Lafferty, J., Zhu, X., & Liu, Y. (2004). Kernel conditional random fields: representation and clique selection. *Proc. of the International Conference on Machine Learning*.

Manning, C. D., & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA.

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

Taskar, B., Guestrin, C., & Koller, D. (2004). Max-margin Markov networks. *Advances in Neural Information Processing Systems*.

Tsochantaridis, I., Joachims, T., Hofmann, T., & Altun, Y. (2005). Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, *6*, 1453–1484.