Transductive Support Vector Machines for Structured Variables

Alexander Zien ALEXANDER.ZIEN@TUEBINGEN.MPG.DE
Max Planck Institute for Biological Cybernetics

Ulf Brefeld BREFELD@MPI-INF.MPG.DE
Tobias Scheffer SCHEFFER@MPI-INF.MPG.DE
Max Planck Institute for Computer Science

Abstract

We study the problem of learning kernel machines transductively for structured output variables. Transductive learning can be reduced to combinatorial optimization problems over all possible labelings of the unlabeled data. In order to scale transductive learning to structured variables, we transform the corresponding non-convex, combinatorial, constrained optimization problems into continuous, unconstrained optimization problems. The discrete optimization parameters are eliminated and the resulting differentiable problems can be optimized efficiently. We study the effectiveness of the generalized TSVM on multiclass classification and label-sequence learning problems empirically.

1. Introduction

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

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. It is then helpful to represent input and output pairs in a joint feature representation, and to rephrase the learning task as finding $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ such that

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

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 arbitrary joint features. Max-margin Markov models (Taskar et al., 2003), kernel conditional random fields (Lafferty et al., 2004), and support vector machines for structured output spaces (Tsochantaridis et al., 2004) use kernels to compute the inner product in input output space. 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. The decoder can be a Viterbi algorithm when joint features are constrained to depend only on adjacent outputs, or a chart parser for tree-structured outputs.

Several semi-supervised techniques in joint input output spaces have been studied. One of the most promising approaches is the integration of unlabeled instances by Laplacian priors into structured large margin classifiers (Lafferty et al., 2004; Altun et al., 2005). Brefeld and Scheffer (2006) include unlabeled examples into structural support vector learning by modeling distinct views of the data and applying the consensus maximization princple between peer hypotheses. Lee et al. (2007) study semi-supervised CRFs and include unlabeled data via an entropy criterion such that their objective acts as a probabilistic analogon to the transductive setting we discuss here. Xu et al. (2006) derive unsupervised M^{3}Networks by employing SDP relaxation techniques. Their optimization problem is similar to the transductive criterion derived in this paper.

Traditional binary TSVM implementations (Joachims,

Appearing in *Proceedings of the 24th International Conference on Machine Learning*, Corvallis, OR, 2007.

1999) solve a combinatorial optimization problem over pseudo-labels $\hat{y}_j$ for the unlabeled $\mathbf{x}_j$. These additional combinatorial optimization parameters can be removed altogether when the constraints $\hat{y}_j\langle \mathbf{w}, \mathbf{x}_i\rangle \geq 1-\xi_i$ are expressed using absolute values: $\xi_j = \max\{1-|\langle \mathbf{w}, \mathbf{x}_j\rangle|, 0\}$ (Chapelle, 2007). The resulting problem remains non-convex, but is now continuous and has fewer parameters. It can therefore be optimized faster, and the retrieved local minima are substantially better on average (Chapelle & Zien, 2005).

The structure of this paper and its main contributions are as follows. Section 2 recalls the basics of learning in structured output spaces. We then leverage the technique of continuous optimization of the primal to structured input output spaces, addressing both the supervised (Section 3) and the transductive case (Section 4). Our treatment covers general loss functions and linear discrimination as well as general kernels. We study the benefit of the generalized transductive SVM empirically in Section 5. Section 6 provides a discussion of the experimental results and Section 7 concludes.

2. Learning in Input Output Spaces

When dealing with discriminative structured prediction models, input variables $\mathbf{x}_i \in \mathcal{X}$ and outputs variables $\mathbf{y}_i \in \mathcal{Y}$ are represented jointly by a feature map $\Phi(\mathbf{x}_i, \mathbf{y}_i)$ that allows to capture multiple-way dependencies. 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 $\hat{\mathbf{y}} = \operatorname{argmax}_{\mathbf{y}\in\mathcal{Y}} f(\mathbf{x}, \mathbf{y})$ for input $\mathbf{x}$.

We measure the quality of f by an appropriate, symmetric, non-negative loss function $\Delta : \mathcal{Y}\times\mathcal{Y} \to \mathbb{R}_0^+$ that details the distance between the true $\mathbf{y}$ and the prediction; for instance, Δ may be the common 0/1 loss, given by $\Delta(\mathbf{y}, \hat{\mathbf{y}}) = 1_{[[\mathbf{y}\neq\hat{\mathbf{y}}]]}$. Thus, the expected risk of f is given as

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

where $P_{\mathcal{X}\times\mathcal{Y}}$ is the (unknown) distribution of inputs and outputs. We address this problem by searching for a minimizer of the empirical risk on a fixed *iid* sample of pairs $(\mathbf{x}_i, \mathbf{y}_i)$, $1 \leq i \leq n$, drawn *iid* from $P_{\mathcal{X}\times\mathcal{Y}}$, regularized with the inverse margin $||\mathbf{w}||^2$. The feature map $\Phi(\mathbf{x}, \mathbf{y})$ and the decoder have to be adapted to the application at hand. We briefly skim the feature spaces and decoders used in our experiments.

Multi-class classification is a special case of a joint input output space with the output space equaling the finite output alphabet; *i.e.,* $\mathcal{Y} = \Sigma$. Let $\psi(\mathbf{x})$ be the feature vector (*e.g.,* a tf.idf vector) of $\mathbf{x}$. Then, the class-based feature representation $\phi^\sigma(\mathbf{x}, y)$ is given by $\phi^\sigma(\mathbf{x}, y) = 1_{[[y=\sigma]]}\psi(\mathbf{x})$, with $\sigma \in \Sigma$. The joint feature representation is given by "stacking up" the class-based representations of all classes $\sigma \in \Sigma$; thus, $\Phi(\mathbf{x}, y) = (\ldots, \phi^\sigma(\mathbf{x}, y), \ldots)$. With this definition, the inner product in input output space reduces to $\langle \Phi(\mathbf{x}_i, y_i), \Phi(\mathbf{x}_j, y_j)\rangle = 1_{[[y_i=y_j]]}k(\mathbf{x}_i, \mathbf{x}_j)$, 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 limited we do not need a special decoding strategy: the argmax can efficiently be determined by enumerating all y and returning the highest-scoring class.

In *label sequence learning*, the task is to find a mapping from a sequential input $\mathbf{x}_i = \langle x_{i,1}, \ldots, x_{i,|\mathbf{x}_i|}\rangle$ to a sequential output $\mathbf{y}_i = \langle y_{i,1}, \ldots, y_{i,|\mathbf{x}_i|}\rangle$ of the same length $|\mathbf{y}_i| = |\mathbf{x}_i|$. Each element of $\mathbf{x}$ is annotated with an element $y_{i,t} \in \Sigma$. We follow Altun et al. (2003) and extract *label-label* interactions $\phi_{\sigma,\tau}(\mathbf{y}|t) = 1_{[[y_{t-1}=\sigma\wedge y_t=\tau]]}$ and *label-observation* features $\bar{\phi}_{\sigma,l}(\mathbf{x}, \mathbf{y}|t) = 1_{[[y_t=\sigma]]}\psi_l(x_t)$, with labels $\sigma, \tau \in \Sigma$. Here, $\psi_l(x)$ extracts characteristics of x; *e.g.,* $\psi_{123}(x) = 1$ if x starts with a capital letter and 0 otherwise. We refer to the vector $\psi(x) = (\ldots, \psi_l(x), \ldots)^\mathsf{T}$ and denote the inner product by means of $k(x, \bar{x}) = \langle \psi(x), \psi(\bar{x})\rangle$. The joint feature representation $\Phi(\mathbf{x}, \mathbf{y})$ of a sequence is the sum of all feature vectors $\Phi(\mathbf{x}, \mathbf{y}|t) = (\ldots, \phi_{\sigma,\tau}(\mathbf{y}|t), \ldots, \bar{\phi}_{\sigma,l}(\mathbf{x}, \mathbf{y}|t), \ldots)^\mathsf{T}$ 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,

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

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

Once an appropriate feature mapping Φ and the corresponding decoding are found for a problem at hand, both can be plugged into the learning algorithms presented in the following Sections. In the remainder we will use the shorthand

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

for difference vectors in joint input output space.

3. Unconstrained Optimization for Structured Output Spaces

Optimization Problem 1 is the known SVM learning problem in input output spaces with cost-based margin rescaling, which includes the fixed size margin with 0/1-loss as special case. All presented results can also be derived for slack rescaling approaches; however, the corresponding constraint generation becomes more difficult. The norm of $\mathbf{w}$ plus the sum of the slack terms ξ_i be minimized, subject to the constraint that, for all examples $(\mathbf{x}_i, \mathbf{y}_i)$, the correct label $\mathbf{y}_i$ receives the highest score by a margin.

OP 1 (SVM) *Given n labeled training pairs, $C > 0$,*

$$\begin{array}{lll} \min\limits_{\mathbf{w},\boldsymbol{\xi}} & ||\mathbf{w}||^2 + C\sum\limits_{i=1}^{n}\xi_i & \\ s.t. & \forall_{i=1}^{n}\,\forall_{\bar{\mathbf{y}}\neq\mathbf{y}_i}\ \langle \mathbf{w}, \Phi_{i\mathbf{y}_i\bar{\mathbf{y}}}\rangle & \geq \quad \Delta(\mathbf{y}_i,\bar{\mathbf{y}}) - \xi_i, \\ & \forall_{i=1}^{n}\ \xi_i & \geq \quad 0. \end{array}$$

In general, unconstrained optimization is easier to implement than constrained optimization. For the SVM, it is possible to resolve the slack terms:

$$\begin{aligned} \xi_i &= \max\left\{\max_{\bar{\mathbf{y}}\neq\mathbf{y}_i}\left\{\Delta(\mathbf{y}_i,\bar{\mathbf{y}}) - \langle \mathbf{w}, \Phi_{i\mathbf{y}_i\bar{\mathbf{y}}}\rangle\right\}, 0\right\} \\ &= \max_{\bar{\mathbf{y}}\neq\mathbf{y}_i}\left\{\ell_{\Delta(\mathbf{y}_i,\bar{\mathbf{y}})}\left(\langle \mathbf{w}, \Phi_{i\mathbf{y}_i\bar{\mathbf{y}}}\rangle\right)\right\}, \end{aligned} \tag{1}$$

where $\ell_\Delta(t) = \max\{\Delta - t, 0\}$ is the hinge loss with margin rescaling. We can now pose Optimization Problem 2 for structured outputs, a simple quadratic optimization function *without constraints.*

OP 2 (∇SVM) *Given n labeled pairs and $C > 0$,*

$$\min_{\mathbf{w}} \quad ||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i \tag{2}$$

where $\xi_i = \max_{\bar{\mathbf{y}}\neq\mathbf{y}_i}\left\{\ell_{\Delta(\mathbf{y}_i,\bar{\mathbf{y}})}\left(\langle \mathbf{w}, \Phi_{i\mathbf{y}_i\bar{\mathbf{y}}}\rangle\right)\right\}$.

The ξ_i remain in Optimization Problem 2 for better comprehensibility; when they are expanded, the criterion is a closed expression. When the maximum is approximated by the softmax, and a smooth approximation of the hinge loss is used, Equation 1 is also differentiable. The softmax and its derivative are displayed in the following Equations,

$$\begin{aligned} \operatorname*{smax}_{\tilde{\mathbf{y}}\neq\mathbf{y}_k}(s(\tilde{\mathbf{y}})) &= \frac{1}{\rho}\log\left(1 + \sum\nolimits_{\tilde{\mathbf{y}}\neq\mathbf{y}_k}(e^{\rho s(\tilde{\mathbf{y}})} - 1)\right) \\ \frac{\partial}{\partial s(\bar{\mathbf{y}})}\operatorname*{smax}_{\tilde{\mathbf{y}}\neq\mathbf{y}_k}(s(\tilde{\mathbf{y}})) &= \frac{e^{\rho s(\bar{\mathbf{y}})}}{1 + \sum_{\tilde{\mathbf{y}}\neq\mathbf{y}_k}(e^{\rho s(\tilde{\mathbf{y}})} - 1)}\ , \end{aligned}$$

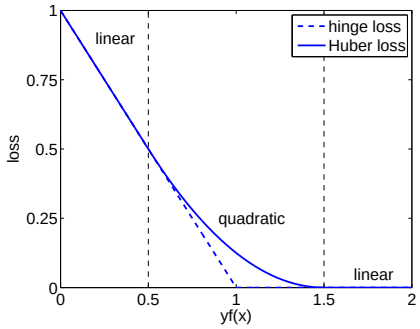

Figure 1. The differentiable Huber loss $\ell_{\Delta=1,\epsilon=0.5}$.

where we will use $s(\tilde{\mathbf{y}}) = \ell_{\Delta(\mathbf{y}_i,\tilde{\mathbf{y}})}\left(\langle \mathbf{w}, \Phi_{i\mathbf{y}_i\tilde{\mathbf{y}}}\rangle\right)$. The Huber loss $\ell_{\Delta,\epsilon}$, displayed in Figure 1, is given by

$$\begin{aligned} \ell_{\Delta,\epsilon}(t) &= \begin{cases} \Delta - t & : \ t \leq \Delta - \epsilon \\ \frac{(\Delta+\epsilon-t)^2}{4\epsilon} & : \ \Delta - \epsilon \leq t \leq \Delta + \epsilon \\ 0 & : \ \text{otherwise} \end{cases} \\ \ell'_{\Delta,\epsilon}(t) &= \begin{cases} -1 & : \ t \leq \Delta - \epsilon \\ -\frac{1}{2}\left(\frac{\Delta-t}{\epsilon} + 1\right) & : \ \Delta - \epsilon \leq t \leq \Delta + \epsilon \\ 0 & : \ \text{otherwise}\ . \end{cases} \end{aligned}$$

An application of the representer theorem shows that $\mathbf{w}$ can be expanded as

$$\mathbf{w} = \sum\nolimits_k \sum\nolimits_{\bar{\mathbf{y}}\neq\mathbf{y}_k} \alpha_{k\mathbf{y}_k\bar{\mathbf{y}}}\, \Phi_{k\mathbf{y}_k\bar{\mathbf{y}}}. \tag{3}$$

The gradient is a vector over the coefficients $\alpha_{k\mathbf{y}_k\bar{\mathbf{y}}}$ for each example $\mathbf{x}_k$ with true label $\mathbf{y}_k$ and each possible incorrect labeling $\bar{\mathbf{y}}$. Computationally, only nonzero coefficients have to be represented. The gradient of Equation 2 with respect to $\mathbf{w}$ is given by

$$\nabla_{OP2} = 2\mathbf{w}\nabla_{\mathbf{w}} + C\sum_{i=1}^{n}\nabla_{\xi_i}.$$

Thus, applying Equation 3 gives us the first derivative in terms of the $\alpha_{k\mathbf{y}_k\bar{\mathbf{y}}}$

$$\frac{\partial OP2}{\partial \alpha_{k\mathbf{y}_k\bar{\mathbf{y}}}} = 2\mathbf{w}\frac{\partial \mathbf{w}}{\partial \alpha_{k\mathbf{y}_k\bar{\mathbf{y}}}} + C\sum_{i=1}^{n}\frac{\partial \xi_i}{\partial \alpha_{k\mathbf{y}_k\bar{\mathbf{y}}}}.$$

The partial derivative $\frac{\partial \mathbf{w}}{\partial \alpha_{k\mathbf{y}_k\bar{\mathbf{y}}}}$ resolves to $\Phi_{k\mathbf{y}_k\bar{\mathbf{y}}}$; that of ξ_i can be decomposed by the chain rule into

$$\frac{\partial \xi_i}{\partial \alpha_{k\mathbf{y}_k\bar{\mathbf{y}}}} = \frac{\partial \xi_i}{\partial \mathbf{w}}\frac{\partial \mathbf{w}}{\partial \alpha_{k\mathbf{y}_k\bar{\mathbf{y}}}} = \frac{\partial \xi_i}{\partial \mathbf{w}}\Phi_{k\mathbf{y}_k\bar{\mathbf{y}}}\ ,$$

$$\frac{\partial \xi_i}{\partial \mathbf{w}} = \sum_{\bar{\mathbf{y}}\neq\mathbf{y}_i}\frac{\partial \operatorname*{smax}_{\tilde{\mathbf{y}}\neq\mathbf{y}_i} s(\tilde{\mathbf{y}})}{\partial s(\bar{\mathbf{y}})}\cdot \ell'_{\Delta(\mathbf{y}_i,\bar{\mathbf{y}})}\left(\langle \mathbf{w}, \Phi_{i\mathbf{y}_i\bar{\mathbf{y}}}\rangle\right)\cdot \Phi_{i\mathbf{y}_i\bar{\mathbf{y}}}$$

This solution generalizes Chapelle (2007) for general input output spaces. The global minimum of Optimization Problem 2 can now easily be found with a

standard gradient algorithm, such as conjugate gradient descent. By rephrasing the problem as an unconstrained optimization problem, its intrinsic complexity has not changed. We will observe the benefit of this approach in the following Sections.

4. Unconstrained Transductive SVMs

In semi-supervised learning, unlabeled $\mathbf{x}_j^*$ for $n+1 \leq j \leq n+m$ are given in addition to the labeled pairs $(\mathbf{x}_i, \mathbf{y}_i)$ for $1 \leq i \leq n$, where usually $n \ll m$. Optimization Problem 3 requires the unlabeled data to be classified by a large margin, but the actual label is unconstrained; this favors a low-density separation.

OP 3 (TSVM) *Given n labeled and m unlabeled training pairs, let $C_l, C_u > 0$,*

$$\min_{\mathbf{w},\boldsymbol{\xi},\boldsymbol{\xi}^*} ||\mathbf{w}||^2 + C_l \sum_{i=1}^{n} \xi_i + C_u \sum_{j=n+1}^{n+m} \xi_j^*$$

subject to the constraints

$$\begin{aligned} \forall_{i=1}^{n} \forall_{\bar{\mathbf{y}} \neq \mathbf{y}_i} \langle \mathbf{w}, \Phi_{i\mathbf{y}_i\bar{\mathbf{y}}} \rangle &\geq \Delta(\mathbf{y}_i, \bar{\mathbf{y}}) - \xi_i \\ \forall_{j=n+1}^{n+m} \exists_{\mathbf{y}_j^*} \forall_{\bar{\mathbf{y}} \neq \mathbf{y}_j^*} \langle \mathbf{w}, \Phi_{j\mathbf{y}_j^*\bar{\mathbf{y}}} \rangle &\geq \Delta(\mathbf{y}_j^*, \bar{\mathbf{y}}) - \xi_j^* \\ \forall_{i=1}^{n}\ \xi_i \geq 0; \quad \forall_{j=n+1}^{n+m}\ \xi_j^* &\geq 0. \end{aligned}$$

Optimization problem 3 requires that there be a $\mathbf{y}_j^*$ such that all other labels $\bar{\mathbf{y}}$ violate the margin by no more than ξ_j^*. Hence, the value of slack variable ξ_j^* is determined by the label $\bar{\mathbf{y}}$ that incurs the strongest margin violation. Alternatively, the sum of margin violations over all $\bar{\mathbf{y}} \neq \mathbf{y}_j^*$ may be upper bounded by ξ_j^*. In fact we can interpolate between max and sum by varying the softmax parameter ρ. Note that the optimum expansion $\boldsymbol{\alpha}$ is sparse, as only margin violating labels $\bar{\mathbf{y}}$ contribute to the aggregation. As we will see later, these $\bar{\mathbf{y}}$ can be efficiently determined.

The constraints on ξ_j^* involve a *disjunction* over all possible labelings $\mathbf{y}_j^*$ of the unlabeled $\mathbf{x}_j^*$ which causes non-convexity and renders QP-solvers not directly applicable. The TSVM implementation in SVM^{light} (Joachims, 1999) treats the pseudo-labels $\mathbf{y}_j^*$ as additional combinatorial parameters. The existential quantifier is thus removed, but the criterion has to be minimized over all possible values of $(\mathbf{y}_{n+1}^*, \ldots, \mathbf{y}_{n+m}^*)$ and, in a nested step of convex optimization, over the $\mathbf{w}$. Analogously to the ξ_i (Equation 1), we replace the constraints on ξ_j^*:

$$\begin{aligned} \xi_j^* &= \min_{\mathbf{y}_j^*} \max \left\{ \max_{\bar{\mathbf{y}} \neq \mathbf{y}_j^*} \left\{ \Delta(\mathbf{y}_j^*, \bar{\mathbf{y}}) - \langle \mathbf{w}, \Phi_{j\mathbf{y}_j^*\bar{\mathbf{y}}} \rangle \right\}, 0 \right\} \\ &= \min_{\mathbf{y}_j^*} \max_{\bar{\mathbf{y}} \neq \mathbf{y}_j^*} \left\{ u_{\Delta(\mathbf{y}_j^*, \bar{\mathbf{y}})}(\langle \mathbf{w}, \Phi_{j\mathbf{y}_j^*\bar{\mathbf{y}}} \rangle) \right\}. \end{aligned} \quad (4)$$

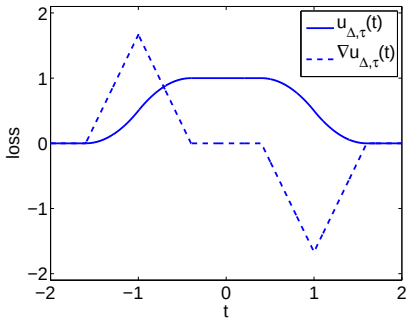

Figure 2. Loss $u_{\Delta=1,\tau=0.6}(t)$ and first derivative.

We quantify the loss induced by unlabeled instances $u_{\Delta,\tau}$ by a function slightly different from Huber loss $\ell_{\Delta,\epsilon}$. Diverging from ℓ, we engineer u to be symmetric, and to have a vanishing derivative at (and around) the point of symmetry. At this point, two labels score equally good (and better than all others), and the corresponding margin violation can be mitigated by moving $\mathbf{w}$ in two symmetric ways.

$$u_{\Delta,\tau}(t) = \begin{cases} 1 & : |t| \leq \Delta - \tau \\ 1 - \frac{(|t| - \Delta + \tau)^2}{2\tau^2} & : \Delta - \tau \leq |t| \leq \Delta \\ \frac{(|t| - \Delta - \tau)^2}{2\tau^2} & : \Delta \leq |t| \leq \Delta + \tau \\ 0 & : \text{otherwise} \end{cases}$$

$$u'_{\Delta,\tau}(t) = \begin{cases} 0 & : |t| \leq \Delta - \tau \\ -\frac{\operatorname{sgn}(t)}{\tau^2}(|t| - \Delta + \tau) & : \Delta - \tau \leq |t| \leq \Delta \\ +\frac{\operatorname{sgn}(t)}{\tau^2}(|t| - \Delta - \tau) & : \Delta \leq |t| \leq \Delta + \tau \\ 0 & : \text{otherwise.} \end{cases}$$

Having rephrased the constraints on ξ_j^* as an equation, we can pose the unconstrained transductive SVM optimization problem for structured outputs.

OP 4 (∇TSVM) *Given n labeled and m unlabeled training pairs, let $C_l, C_u > 0$,*

$$\min_{\mathbf{w}} \quad ||\mathbf{w}||^2 + C_l \sum_{i=1}^{n} \xi_i + C_u \sum_{j=n+1}^{n+m} \xi_j^* \quad (5)$$

with placeholders $\xi_i = \max_{\bar{\mathbf{y}} \neq \mathbf{y}_i} \left\{ \ell_{\Delta(\mathbf{y}_i, \bar{\mathbf{y}})} (\langle \mathbf{w}, \Phi_{i\mathbf{y}_i\bar{\mathbf{y}}} \rangle) \right\}$ and $\xi_j^* = \min_{\mathbf{y}_j^*} \max_{\bar{\mathbf{y}} \neq \mathbf{y}_j^*} \left\{ u_{\Delta(\mathbf{y}_j^*, \bar{\mathbf{y}})} \left(\langle \mathbf{w}, \Phi_{j\mathbf{y}_j^*\bar{\mathbf{y}}} \rangle \right) \right\}$.

Variables ξ_i and ξ_j^* remain in Optimization Problem 4 for notational harmony, they can be expanded to yield a closed, unconstrained optimization criterion.

Again we invoke the representer theorem (3) and optimize along the gradient $\frac{\partial OP4}{\partial \boldsymbol{\alpha}}$. In addition to the derivatives calculated in the Section 3, we need the partial derivatives of the ξ_j^*. They are analogous to

Table 1. THE ∇TSVM ALGORITHM

Input: Labeled data $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$, unlabeled data $\{\mathbf{x}_j^*\}_{j=n+1}^{n+m}$; parameters $C_l, C_u, \epsilon_\alpha > 0$.

1: **repeat**
2: **for** each labeled example $(\mathbf{x}_i, \mathbf{y}_i)$ **do**
3: $\bar{\mathbf{y}} \leftarrow \operatorname{argmax}_{\mathbf{y} \neq \mathbf{y}_i} \{\Delta(\mathbf{y}_i, \mathbf{y}) - \langle \mathbf{w}, \Phi(\mathbf{x}_i, \mathbf{y})\rangle\}$ // compute worst margin violater
4: **if** $\ell_{\Delta(\mathbf{y}_i,\bar{\mathbf{y}}),\epsilon}(\langle \mathbf{w}, \Phi(\mathbf{x}_i, \mathbf{y}_i)\rangle - \langle \mathbf{w}, \Phi(\mathbf{x}_i, \bar{\mathbf{y}})\rangle) > 0$ **then**
5: $W \leftarrow W \cup \{(i, \mathbf{y}_i, \bar{\mathbf{y}})\}$ // add difference vector to working set
6: **end if**
7: **end for**
8: **for** each unlabeled example $\mathbf{x}_j^*$ **do**
9: $\hat{\mathbf{y}}_j^* \leftarrow \operatorname{argmax}_{\mathbf{y}} \left\{\langle \mathbf{w}, \Phi(\mathbf{x}_j^*, \mathbf{y})\rangle\right\}$ // compute top scoring output
10: $\bar{\mathbf{y}} \leftarrow \operatorname{argmax}_{\mathbf{y} \neq \mathbf{y}_j^*} \left\{\Delta(\mathbf{y}_j^*, \mathbf{y}) - \langle \mathbf{w}, \Phi(\mathbf{x}_j^*, \mathbf{y})\rangle\right\}$ // compute runner-up
11: **if** $\exists \mathbf{y}_j^* \in W \wedge \mathbf{y}_j^* \neq \hat{\mathbf{y}}_j^*$ **then**
12: $\forall \bar{\mathbf{y}} : W \leftarrow W \backslash \{(j, \mathbf{y}_j^*, \bar{\mathbf{y}})\}$ // delete old constraints
13: **end if**
14: **if** $u_{\Delta(\mathbf{y}_j^*,\bar{\mathbf{y}}),\tau}(\langle \mathbf{w}, \Phi(\mathbf{x}_j^*, \mathbf{y}_j^*)\rangle - \langle \mathbf{w}, \Phi(\mathbf{x}_j^*, \bar{\mathbf{y}})\rangle) > 0$ **then**
15: $\mathbf{y}_j^* \leftarrow \hat{\mathbf{y}}_j^*$
16: $W \leftarrow W \cup \{(j, \mathbf{y}_j^*, \bar{\mathbf{y}})\}$ // add difference vector to working set
17: **end if**
18: **end for**
19: $\boldsymbol{\alpha} \leftarrow \operatorname{argmin}_{\boldsymbol{\alpha}'} \text{TSVM}(\boldsymbol{\alpha}', W)$ // minimize Eq. 5 by conjugate gradient descent
20: $\forall \alpha_{k\mathbf{y}\bar{\mathbf{y}}} < \epsilon_\alpha : W \leftarrow W \backslash \{(k, \mathbf{y}_k, \bar{\mathbf{y}})\}$ // delete unnecessary constraints
21: **until** convergence

Output: Optimized $\boldsymbol{\alpha}$, working set W.

those of ξ_i; let $\bar{s}(\tilde{\mathbf{y}}) = u_{\Delta(\mathbf{y}_j^*,\tilde{\mathbf{y}})}(\langle \mathbf{w}, \Phi_{j\mathbf{y}_j^*\tilde{\mathbf{y}}}\rangle)$, we have

$$\frac{\partial \xi_j^*}{\partial \mathbf{w}} = \sum_{\bar{\mathbf{y}} \neq \mathbf{y}_j^*} \frac{\partial \operatorname{smax}_{\tilde{\mathbf{y}} \neq \mathbf{y}_j^*} \bar{s}(\tilde{\mathbf{y}})}{\partial \bar{s}(\bar{\mathbf{y}})} u'_{\Delta(\mathbf{y}_j^*,\bar{\mathbf{y}})} \left(\langle \mathbf{w}, \Phi_{j\mathbf{y}_j^*\bar{\mathbf{y}}}\rangle\right) \Phi_{j\mathbf{y}_j^*\bar{\mathbf{y}}}.$$

Every expansion coefficient $\alpha^*_{j\mathbf{y}_j^*\bar{\mathbf{y}}}$ influences how strongly f favors label $\mathbf{y}_j^*$ over $\bar{\mathbf{y}}$ for the unlabeled example j. This solution generalizes Chapelle and Zien (2005) for general input output spaces.

Algorithmically, continuous optimization over all parameters $\alpha_{k\mathbf{y}_k\bar{\mathbf{y}}}$ is impossible due to exponentially many $\bar{\mathbf{y}}$'s. However, our loss functions cause the solution to be sparse. In order to narrow the search to the non-zero variables, generalized ∇TSVM training interleaves two steps. In the decoding step, the algorithm iterates over all training instances and uses a 2-best decoder to produce the highest-scoring output $\hat{\mathbf{y}}$ and the worst margin violator $\bar{\mathbf{y}} \neq \hat{\mathbf{y}}$. For labeled examples $(\mathbf{x}_i, \mathbf{y}_i)$, output $\hat{\mathbf{y}}$ has to be equal to the desired $\mathbf{y}_i$, and $\bar{\mathbf{y}}$ must not violate the margin. Otherwise, the difference vector $\Phi_{i\mathbf{y}_i\bar{\mathbf{y}}}$ is added to the (initially empty) working set of the i-th example. For unlabeled data, the highest-scoring output of the joint classifier $\hat{\mathbf{y}}_j^*$ serves as desired labeling and the runner-up as margin violator $\bar{\mathbf{y}}_j$. Again, in case of a margin violation $\Phi_{j\mathbf{y}^*\bar{\mathbf{y}}}$ is added to the working set for $\mathbf{x}_j$.

In the optimization step, conjugate gradient descent (CG) is executed over the parameters $\alpha_{k\mathbf{y}_k\bar{\mathbf{y}}}$, given by all examples $\mathbf{x}_k$, desired outputs $\mathbf{y}_k$, and all associated pseudo-labels $\bar{\mathbf{y}}$ currently in the working set. As proposed in (Chapelle, 2007) we use the kernel matrix as preconditioner, which speeds up the convergence of the CG considerably. The inner loop of the ∇TSVM algorithm is depicted in Table 1.

In an outer loop, ∇TSVM first increases C_l in a barrier fashion to avoid numerical instabilities, and eventually increases the the influence of the unlabeled examples C_u. The algorithm terminates when the working set remains unchanged over two consecutive iterations and C_l and C_u have reached the desired maximum value. Notice that ∇TSVM reduces to ∇SVM when no unlabeled examples are included into the training process; *i.e.,* for ∇SVM, lines 8-18 are removed from Table 1.

For binary TSVMs it has proven useful to add a *balancing constraint* to the optimization problem that ensures that the relative class sizes of the predictions are similar to those of the labeled points (Joachims, 1999). For structured outputs, the relative frequencies of the output symbols $\sigma \in \Sigma$ may be constrained:

$$\frac{\sum_{j=n+1}^{n+m} \sum_{t=1}^{|\mathbf{x}_j|} 1_{[[y_{j,t}=\sigma]]}}{\sum_{j=n+1}^{n+m} |\mathbf{x}_j|} = \frac{\sum_{i=1}^{n} \sum_{s=1}^{|\mathbf{x}_i|} 1_{[[y_{i,s}=\sigma]]}}{\sum_{i=1}^{n} |\mathbf{x}_i|}.$$

Analoguously to binary TSVMs (Chapelle & Zien, 2005), this can be relaxed to "soft" linear constraints:

$$\sum_{j=n+1}^{n+m} \sum_{t=1}^{|\mathbf{x}_j|} \left(\mathbf{w}^\top \Phi(x_{j,t}, \sigma) + b_\sigma - \mathbf{w}^\top \bar{\Phi}(x_{j,t}) - \bar{b}\right) = \hat{p}_\sigma$$

where $\Phi(x_{j,t}, \sigma)$ are the feature maps corresponding to predicting σ for position t of $\mathbf{x}_j$, $\bar{\Phi}(x_{j,t}) = \sum_{\omega \in \Sigma} \Phi(x_{j,t}, \omega)/|\Sigma|$ is their average, the b_σ are newly introduced label biases with average $\bar{b} = \sum_\sigma b_\sigma/|\Sigma|$, and $\hat{p}_\sigma = (\sum_j |\mathbf{x}_j|)(\sum_i \sum_s 1_{[[y_{is}=\sigma]]}/(\sum_i |\mathbf{x}_i|) - 1/|\Sigma|)$ are centered predicted class sizes. By appropriately centering the unlabeled data these constraints can be equivalently transformed into fixing the b_σ to constants. However, here we do not implement any balancing, as we empirically observe the fractions of predicted symbols to roughly agree to the corresponding fractions on the known labels.

5. Experiments

We investigate unconstrained optimization of structured output support vector machines by comparing differentiable ∇SVM and ∇TSVM to SVMs solved by quadratic programming (QP) approaches.

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 optimize C_u using resampling and then fix C_u and present curves that show the average error over 100 randomly drawn training and holdout sets; errorbars indicate standard error. In all experiments we set $C_l = 1$, $\epsilon = 0.3$, and $\tau = 0.4$.

5.1. Execution Time

Figure 3 compares the execution times of CG-based ∇SVM and ∇TSVM to a QP-based SVM where we used the same convergence criteria for all optimizers. ∇TVSM is trained with the respective number of labeled examples and a 5 times larger set of unlabeled instances. Besides being faster than a solution based on solving QPs, the continuous optimization is remarkably efficient at utilizing the unlabeled data. For instance, ∇TSVM with 50 labeled and 250 unlabeled examples converges considerably faster than ∇SVM and qpSVM with only 200 labeled instances.

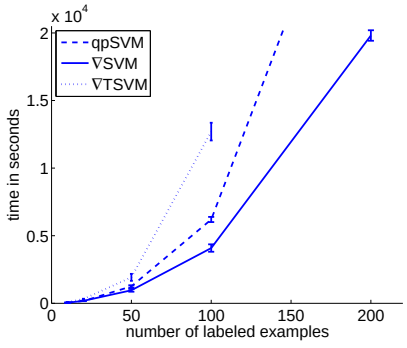

Figure 3. Execution time.

5.2. Multi-Class Classification

For the multi-class classification experiments, we use a cleaned variant of the Cora data set that contains 9,555 linked computer science papers with a reference section. The data set is divided into 8 different classes. We extract term frequencies of the document and of the anchor text of the inbound links. The latter are drawn from three sentences, respectively, centered at the occurrence of the reference. We compare the performances of ∇TSVM with 0/1 loss to the performance of TSVM^{light}, trained with a one-vs-rest strategy. Figure 4 details the error-rates for 200 labeled examples and varying numbers of unlabeled instances. For no unlabeled data both transductive methods reduce to their fully-supervised, inductive counterparts. Both SVMs perform equally well for the labeled instances. However, when unlabeled examples are included into the training process, the performance of TSVM^{light} deteriorates. The error-rates of ∇TSVM show a slight improvement with 800 unlabeled instances.

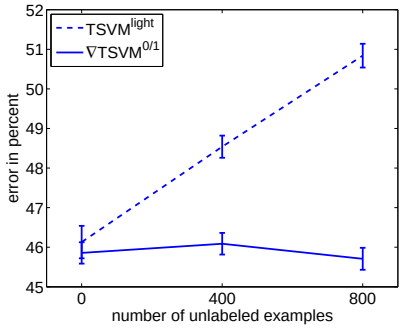

Figure 4. Error-rates for the Cora data set.

We also apply our method to the 6-class dataset COIL as used in (Chapelle et al., 2006), and compare to the reported one-vs-rest TSVM results. For $n = 10$ labeled points, we achieve 68.87% error, while the one-vs-rest TSVM achieves 67.50%. For $n = 100$ points, the results are 25.42% as compared to 25.80%.

5.3. Artificial Sequential Data

The artificial galaxy data set (Lafferty et al., 2004) consists of 100 sequences of length 20, generated by a two state hidden Markov model. The initial state is chosen uniformly and there is a 10% chance of switching the state. Each state emits instances uniformly from one of the two classes, see Figure 5 (left).

We run ∇SVM and ∇TSVM using Hamming loss with two different kernels, a Gaussian RBF kernel with bandwidth $\sigma = 0.35$ and a semi-supervised graph kernel. The graph kernel is constructed from a 10-nearest neighbor graph and given by $K = 10\,(L + \mathbb{1}\rho)^{-1}$, with

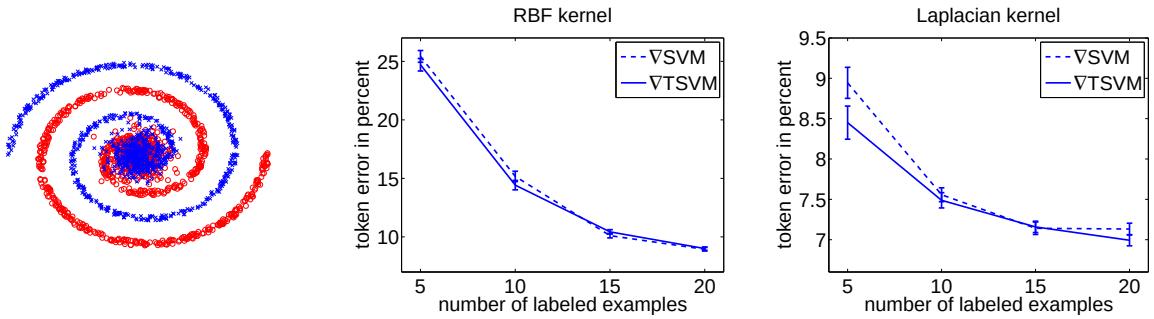

Figure 5. The galaxy data set (left) and error-rates for ∇SVM and ∇TSVM using RBF (center) and graph kernels (right).

graph Laplacian L and $\rho = 10^{-6}$ as proposed by Lafferty et al. (2004).

In each experiment we draw a certain number of labeled sequences at random and use the rest either as unlabeled examples or as holdout set. We report on averages over 20 runs. Figure 5 (center and right) details the results for semi-supervised vs. supervised algorithm and semi-supervised vs. standard kernel. Since the approaches are orthogonal, we apply all 4 combinations. For increasing numbers of labeled examples, the error rates of the tested models decrease. The continuous TSVM performs just slightly better than the supervised SVM; the differences are significant only in few cases. This problem is extremely well tailored for the Laplacian kernel. The error rates achieved with the semi-supervised kernel are between 20% to 3% lower than the corresponding results for the RBF kernel.

5.4. Named Entity Recognition

The CoNLL2002 data consists of sentences from a Spanish news wire archive and contains 9 label types which distinguish person, organization, location, and other names. We use $3,100$ sentences of between 10 and 40 tokens, leading to $\approx 24,000$ distinct tokens in the dictionary. Moreover, we extract surface clue features, like capitalization features and others. We use a window of size 3, centered around each token.

In each experiment we draw a specified number of labeled and unlabeled training and holdout data without replacement at random in each iteration. We assure that each label occurs at least once in the labeled training data; otherwise, we discard and draw again. We compare ∇TSVM with 0/1 loss and Hamming loss to the HM-SVM (Altun et al., 2003), trained by incrementally solving quadratic programs over subspaces associated with individual input examples. Figure 6 details the results for 10 labeled sequences.

∇SVM converges to better local optima than HM-SVM due to global conjugate gradient based optimization compared to solving local quadratic programs. When unlabeled examples are included in the training process the error of the ∇TSVM decreases significantly. ∇TSVM^H with Hamming loss performs slightly better than $\nabla\text{TSVM}^{0/1}$ using 0/1 loss.

6. Discussion

The TSVM criterion is non-convex and the maximization can be difficult even for binary class variables. In order to scale the TSVM to structured outputs, we employ a technique that eliminates the discrete parameters and allows for a conjugate gradient descent in the space of expansion coefficients $\boldsymbol{\alpha}$. Empirical comparisons of execution time show that the continuous approaches are more efficient than standard approaches based on quadratic programming.

For the Cora text classification problem, transductive learning does not achieve a substantial benefit over supervised learning. Worse yet, the combinatorial TSVM increases the error substantially, whereas ∇TSVM has negligible effect. In order to draw an unbiased picture, we present this finding with as much emphasis as any positive result. For the Spanish news named entity recognition problem, we consistently observe small but significant improvements over purely supervised learning.

One might intuitively expect transductive learning to outperform supervised learning, because more information is available. However these test instances introduce non-convexity, and the local minimum retrieved by the optimizer may be worse than the global minimum of the convex supervised problem. Our experiments indicate that this might occationally occur.

For the galaxy problem, the benefit of ∇TSVM over ∇SVM is marginal, and observable only for very few labeled examples. By its design this problem is very well suited for graph kernels, which reduce the error rate by 50%. In the graph Laplacian approach (Sindhwani et al., 2005), an SVM is trained on the labeled

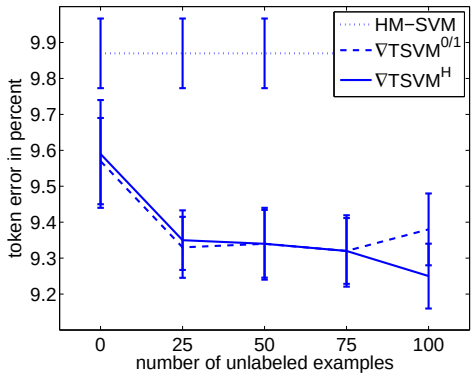

Figure 6. Token error for the Spanish news wire data set with 10 labeled instances.

data, but in addition to the standard kernel, the graph Laplacian derived from labeled and unlabeled points serves as regularizer. For binary classification, combining TSVM and graph Laplacian yields the greatest benefit (Chapelle & Zien, 2005). For structured variables, we observe a similar effect, though much weaker.

The presented ∇TSVM rests on a cluster assumption for entire structures, while graph-based methods (Lafferty et al., 2004; Altun et al., 2005) exploit the distribution of parts of structures. Both approaches improve over supervised learning on some datasets and fail to do so on others. This raises the question how to determine which kind of assumptions are appropriate for a given task at hand.

7. Conclusion

We devised a transductive support vector machine for structured variables (∇TSVM). We transformed the original combinatorial and constrained optimization problem into a differentiable and unconstrained one. The resulting optimization problem is still non-convex but can be optimized efficiently, for instance via a conjugate gradient descent. A differentiable variant of the SVM for structured variables (∇SVM) is obtained for the special case of a fully labeled training set.

We applied both methods with various loss functions to multi-class classification and sequence labeling problems. Due to our empirical findings, we can rule out the hypothesis that ∇TSVM generally improves learning with structured output variables over purely supervised learning, as well as the hypothesis that ∇TSVM never improves accuracy.

We conjecture that transductive structured output learning could benefit from more research on (i) improved non-convex optimization techniques and on (ii) appropriate (cluster) assumptions.

Acknowledgment

We thank John Lafferty, Yan Liu, and Xiaojin Zhu for providing their data. We also thank the anonymous reviewers for detailed comments and suggestions. This work has been partially funded by the German Science Foundation DFG under grant SCHE540/10-2, and it has in part been supported by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778.

References

Altun, Y., McAllester, D., & Belkin, M. (2005). Maximum margin semi-supervised learning for structured variables. *Advances in Neural Information Processing Systems*.

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

Brefeld, U., & Scheffer, T. (2006). Semi-supervised learning for structured output variables. *Proceedings of the International Conference on Machine Learning*.

Chapelle, O. (2007). Training a support vector machine in the primal. *Neural Computation*, *19*, 1155–1178.

Chapelle, O., Schölkopf, B., & Zien, A. (Eds.). (2006). *Semi-supervised learning*. Cambridge, MA: MIT Press.

Chapelle, O., & Zien, A. (2005). Semi-supervised classification by low density separation. *Proceedings of the International Workshop on AI and Statistics*.

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

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

Lee, C., Wang, S., Jiao, F., Greiner, R., & Schuurmans, D. (2007). Learning to model spatial dependency: Semi-supervised discriminative random fields. *Advances in Neural Information Processing Systems*.

Sindhwani, V., Niyogi, P., & Belkin, M. (2005). Beyond the point cloud: From transductive to semisupervised learning. *Proceedings of the International Conference on Machine Learning*.

Taskar, B., Guestrin, C., & Koller, D. (2003). Max-margin Markov networks. *Advances in Neural Information Processing Systems*.

Tsochantaridis, I., Hofmann, T., Joachims, T., & Altun, Y. (2004). Support vector machine learning for interdependent and structured output spaces. *Proceedings of the International Conference on Machine Learning*.

Xu, L., Wilkinson, D., Southey, F., & Schuurmans, D. (2006). Discriminative unsupervised learning of structured predictors. *Proceedings of the International Conference on Machine Learning*.