Transductive Support Vector Machines for Structured Variables

Alexander Zien[*†]
Ulf Brefeld[‡]
Tobias Scheffer[‡]

∗ MPI for Biological Cybernetics, Empirical Inference Dept.
† Friedrich Miescher Laboratory, Machine Learning for Bioinformatics
‡ MPI for Informatics, Research Group Machine Learning

June 21, 2007

MAX-PLANCK-GESELLSCHAFT
BIOLOGISCHE KYBERNETIK
mpi max planck institut informatik

Support Vector Machine

original SVM

- binary
- supervised

Semi-Supervised SVM ("Transductive SVM")

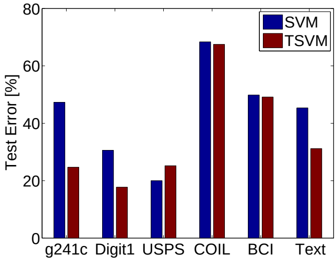
original SVM

- binary
- supervised

⇓

TSVM

- binary
- **semi-supervised**



[*Semi-Supervised Learning*, 2006, MIT Press]

Structured Output SVM

original SVM

- binary
- supervised

⇒

SO-SVM

- **structured output**
- supervised

- True multiclass (not 1-vs-rest or 1-vs-1).
- Accurate label sequence learning [Nguyen, Guo; ICML 2007].
- More complex structures (eg parse trees, RNA secondary structures).

Orthogonal SVM Extensions

original SVM

- binary
- supervised

⇒

SO-SVM

- **structured output**
- supervised

⇓

TSVM

- binary
- **semi-supervised**

Structured Output Semi-Supervised SVM

original SVM
- binary
- supervised

⇒

SO-SVM
- **structured output**
- supervised

⇓

TSVM
- binary
- **semi-supervised**

⇒

SO-TSVM
- **structured output**
- **semi-supervised**

Outline

Structured Output SVM

Use **joint feature map** $\Phi : \mathcal{X} \times \mathcal{Y} \to \mathcal{H}$.

Training: find $\mathbf{w}$ such that

$$\forall_i : \ \forall_{\bar{\mathbf{y}}_i \neq \mathbf{y}_i} : \ \mathbf{w}^\top \Phi(\mathbf{x}_i, \mathbf{y}_i) \ > \ \mathbf{w}^\top \Phi(\mathbf{x}_i, \bar{\mathbf{y}}_i)$$

Prediction:

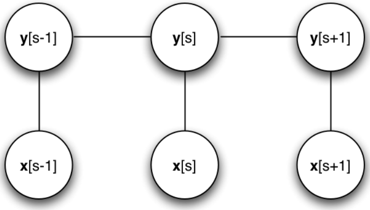
$$\mathbf{x} \mapsto \mathbf{y} := \arg\max_{\mathbf{y}} \mathbf{w}^\top \Phi(\mathbf{x}, \mathbf{y})$$

SO-SVM (aka HM-SVM)

$$\min_{\mathbf{w}, \xi_i} \quad \frac{1}{2} \mathbf{w}^\top \mathbf{w} + C \sum_i \xi_i$$

$$s.t. \quad \forall_{\bar{\mathbf{y}}_i \neq \mathbf{y}_i} : \ \mathbf{w}^\top \Phi(\mathbf{x}_i, \mathbf{y}_i) \ \geq \ \mathbf{w}^\top \Phi(\mathbf{x}_i, \bar{\mathbf{y}}_i) + 1 - \xi_i, \quad \xi_i \geq 0$$

Interlude: Label Sequence Learning



first-order Markov property
⇒ prediction by Viterbi

kernel function decomposes into

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

- label-label part $\sum_{s,t} [[y_{i,s-1} = y_{j,t-1} \wedge y_{i,s} = y_{j,t}]]$
- label-observation part $+ \sum_{s,t} [[y_{i,s} = y_{j,t}]] k(x_{i,s}, x_{j,t})$

Incorporating Unlabeled Data

SO-SVM

$$\min_{\mathbf{w},\xi_i} \quad \frac{1}{2}\mathbf{w}^\top\mathbf{w} + C\sum_i \xi_i$$

$$s.t. \quad \forall_{\bar{\mathbf{y}}_i \neq \mathbf{y}_i} : \mathbf{w}^\top [\Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \bar{\mathbf{y}}_i)] \geq 1 - \xi_i, \quad \xi_i \geq 0$$

How to use unlabeled data $\mathbf{x}_j$?

- For each $\mathbf{x}_j$, $\exists$ true label $\mathbf{y}_j^{true}$.
- Margin shall be maximized on $(\mathbf{x}_i, \mathbf{y}_i)$ and $(\mathbf{x}_j, \mathbf{y}_j^{true})$.
- At optimal solution, $\mathbf{y}_j^{true}$ should score highest, thus estimate

$$\mathbf{y}_j = \arg\max_{\bar{\mathbf{y}}} \mathbf{w}^\top \Phi(\mathbf{x}_j, \bar{\mathbf{y}})$$

Semi-Supervised Structured Output SVM

SO-SVM

$$\min_{\mathbf{w},\xi_i} \quad \frac{1}{2}\mathbf{w}^\top \mathbf{w} + C\sum_i \xi_i$$

$$s.t. \quad \forall_{\bar{\mathbf{y}}_i \neq \mathbf{y}_i} : \mathbf{w}^\top [\Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \bar{\mathbf{y}}_i)] \geq 1 - \xi_i, \quad \xi_i \geq 0$$

SO-TSVM

$$\min_{\mathbf{w},\mathbf{y}_j,\xi_k} \quad \frac{1}{2}\mathbf{w}^\top \mathbf{w} + C\sum_i \xi_i + C^* \sum_j \xi_j$$

$$s.t. \quad \begin{array}{l} \forall_{\bar{\mathbf{y}}_i \neq \mathbf{y}_i} : \mathbf{w}^\top [\Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \bar{\mathbf{y}}_i)] \geq 1 - \xi_i, \quad \xi_i \geq 0 \\ \forall_{\bar{\mathbf{y}}_j \neq \mathbf{y}_j} : \mathbf{w}^\top [\Phi(\mathbf{x}_j, \mathbf{y}_j) - \Phi(\mathbf{x}_j, \bar{\mathbf{y}}_j)] \geq 1 - \xi_j, \quad \xi_j \geq 0 \end{array}$$

Combinatorial SO-TSVM

SO-TSVM

$$\min_{\mathbf{w},\mathbf{y}_j,\xi_k} \quad \frac{1}{2}\mathbf{w}^\top \mathbf{w} + C\sum_i \xi_i + C^* \sum_j \xi_j$$

$$s.t. \quad \begin{array}{l} \forall_{\bar{\mathbf{y}}_i \neq \mathbf{y}_i} : \mathbf{w}^\top [\Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \bar{\mathbf{y}}_i)] \geq 1 - \xi_i, \quad \xi_i \geq 0 \\ \forall_{\bar{\mathbf{y}}_j \neq \mathbf{y}_j} : \mathbf{w}^\top [\Phi(\mathbf{x}_j, \mathbf{y}_j) - \Phi(\mathbf{x}_j, \bar{\mathbf{y}}_j)] \geq 1 - \xi_j, \quad \xi_j \geq 0 \end{array}$$

Problem!

- $\mathbf{y}_j$ **are discrete!**
- **Combinatorial** task.
- **NP-hard!**

For binary TSVM, **continuous** techniques very successfull.

[*Low Density Separation*; 2005; Chapelle, Zien]

Efficient Optimization for SO-TSVM

SO-TSVM

$$\min_{\mathbf{w},\mathbf{y}_j,\xi_k} \quad \frac{1}{2}\mathbf{w}^\top\mathbf{w} + C\sum_i \xi_i + C^*\sum_j \xi_j$$

$$s.t. \quad \forall_{\bar{\mathbf{y}}_i \neq \mathbf{y}_i} : \mathbf{w}^\top [\Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \bar{\mathbf{y}}_i)] \geq 1 - \xi_i, \quad \xi_i \geq 0$$
$$\forall_{\bar{\mathbf{y}}_j \neq \mathbf{y}_j} : \mathbf{w}^\top [\Phi(\mathbf{x}_j, \mathbf{y}_j) - \Phi(\mathbf{x}_j, \bar{\mathbf{y}}_j)] \geq 1 - \xi_j, \quad \xi_j \geq 0$$

Key ideas:

- Plug in effective loss function ⇒ **unconstrained**.
- Make **differentiable**.
- Invoke *Representer Theorem* to use **kernels**.
- Apply efficient **gradient descent** method.

Effective Loss Functions

SO-TSVM

$$\min_{\mathbf{w},\mathbf{y}_j,\xi_k} \quad \frac{1}{2}\mathbf{w}^\top\mathbf{w} + C\sum_i \xi_i + C^*\sum_j \xi_j$$

$$s.t. \quad \forall_{\bar{\mathbf{y}}_i \neq \mathbf{y}_i} : \xi_i \geq 1 - \mathbf{w}^\top\left[\Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \bar{\mathbf{y}}_i)\right], \quad \xi_i \geq 0$$

$$\forall_{\bar{\mathbf{y}}_j \neq \mathbf{y}_j} : \xi_j \geq 1 - \mathbf{w}^\top\left[\Phi(\mathbf{x}_j, \mathbf{y}_j) - \Phi(\mathbf{x}_j, \bar{\mathbf{y}}_j)\right], \quad \xi_j \geq 0$$

At optimum, we have following **effective losses**:

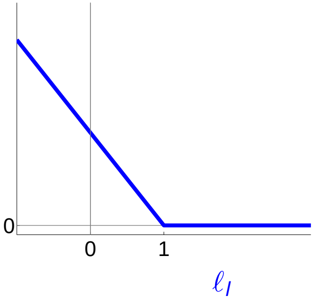
$$\xi_i = \max_{\bar{\mathbf{y}}_i \neq \mathbf{y}_i} \max\left\{ 1 - \mathbf{w}^\top\left[\Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \bar{\mathbf{y}}_i)\right], 0 \right\}$$

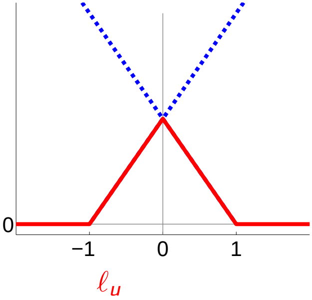
$$\xi_j = \min_{\mathbf{y}_j} \max_{\bar{\mathbf{y}}_j \neq \mathbf{y}_j} \max\left\{ 1 - \mathbf{w}^\top\left[\Phi(\mathbf{x}_j, \mathbf{y}_j) - \Phi(\mathbf{x}_j, \bar{\mathbf{y}}_j)\right], 0 \right\}$$

Original Effective Loss Functions

$$\xi_i = \max_{\bar{\mathbf{y}}_i \neq \mathbf{y}_i} \ell_l \left(\mathbf{w}^\top \Phi(\mathbf{x}_i, \mathbf{y}_i) - \mathbf{w}^\top \Phi(\mathbf{x}_i, \bar{\mathbf{y}}_i) \right)$$

$$\xi_j = \min_{\mathbf{y}_j} \max_{\bar{\mathbf{y}}_j \neq \mathbf{y}_j} \ell_u \left(\mathbf{w}^\top \Phi(\mathbf{x}_j, \mathbf{y}_j) - \mathbf{w}^\top \Phi(\mathbf{x}_j, \bar{\mathbf{y}}_j) \right)$$

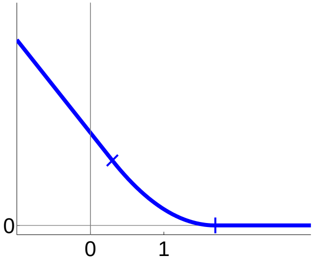




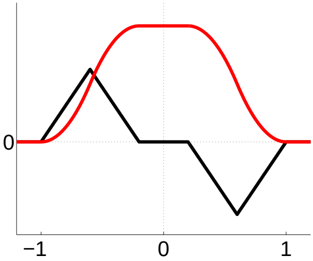
Differentiable Loss Functions

$$\xi_i = \max_{\bar{\mathbf{y}}_i \neq \mathbf{y}_i} \ell_l \left(\mathbf{w}^\top \Phi(\mathbf{x}_i, \mathbf{y}_i) - \mathbf{w}^\top \Phi(\mathbf{x}_i, \bar{\mathbf{y}}_i) \right)$$

$$\xi_j = \min_{\mathbf{y}_j} \max_{\bar{\mathbf{y}}_j \neq \mathbf{y}_j} \ell_u \left(\mathbf{w}^\top \Phi(\mathbf{x}_j, \mathbf{y}_j) - \mathbf{w}^\top \Phi(\mathbf{x}_j, \bar{\mathbf{y}}_j) \right)$$



ℓ_l



ℓ_u, derivative

Differentiable Loss Functions

$$\xi_i = \operatorname*{smax}_{\bar{\mathbf{y}}_i \neq \mathbf{y}_i} \ell_l \left(\mathbf{w}^\top \Phi(\mathbf{x}_i, \mathbf{y}_i) - \mathbf{w}^\top \Phi(\mathbf{x}_i, \bar{\mathbf{y}}_i) \right)$$

$$\xi_j = \min_{\mathbf{y}_j} \operatorname*{smax}_{\bar{\mathbf{y}}_j \neq \mathbf{y}_j} \ell_u \left(\mathbf{w}^\top \Phi(\mathbf{x}_j, \mathbf{y}_j) - \mathbf{w}^\top \Phi(\mathbf{x}_j, \bar{\mathbf{y}}_j) \right)$$

softmax not differentiable $\Rightarrow$ use **softmax**

$$\operatorname*{smax}_{\tilde{\mathbf{y}} \neq \mathbf{y}_k}(s(\tilde{\mathbf{y}})) = \frac{1}{\rho} \log \left(1 + \sum_{\tilde{\mathbf{y}} \neq \mathbf{y}_k} (e^{\rho s(\tilde{\mathbf{y}})} - 1) \right)$$

- approximates max: $\lim_{\rho \to \infty} \operatorname{smax}(s(\tilde{\mathbf{y}})) = \max\{s(\tilde{\mathbf{y}})\}$
- approximates sum: $\lim_{\rho \to 0} \operatorname{smax}(s(\tilde{\mathbf{y}})) = \sum s(\tilde{\mathbf{y}})$

Unconstrained Differentiable Optimization

Unconstrained Differentiable SO-TSVM

$$\min_{\mathbf{w},\xi_k} \quad \frac{1}{2}\mathbf{w}^\top\mathbf{w}$$
$$+C\sum_i \operatorname*{smax}_{\bar{\mathbf{y}}_i \neq \mathbf{y}_i} \ell_l\left(\mathbf{w}^\top\Phi(\mathbf{x}_i,\mathbf{y}_i) - \mathbf{w}^\top\Phi(\mathbf{x}_i,\bar{\mathbf{y}}_i)\right)$$
$$+C^*\sum_j \min_{\mathbf{y}_j} \operatorname*{smax}_{\bar{\mathbf{y}}_j \neq \mathbf{y}_j} \ell_u\left(\mathbf{w}^\top\Phi(\mathbf{x}_j,\mathbf{y}_j) - \mathbf{w}^\top\Phi(\mathbf{x}_j,\bar{\mathbf{y}}_j)\right)$$

- Determine optimial $\mathbf{y}_j$ (Viterbi); repeatedly update.
- Symmetrized loss ℓ_u can account for switching $\mathbf{y}_j \leftrightarrow \bar{\mathbf{y}}_j$.
- $\Rightarrow$ Can optimize $\mathbf{w}$ by gradient descent!

How to Use Kernels?

Representer Theorem

$$\mathbf{w} = \sum_{k=1}^{n+m} \sum_{\mathbf{y} \in \mathcal{Y}(\mathbf{x}_k)} \alpha_{k,\mathbf{y}} \Phi(\mathbf{x}_k, \mathbf{y})$$

- Plug into optimization problem.
  - $\mathbf{w}^\top \Phi(\mathbf{x}_i, \mathbf{y}_i) = \sum_k \sum_{\mathbf{y}} \alpha_{k,\mathbf{y}} \underbrace{\Phi(\mathbf{x}_k, \mathbf{y})^\top \Phi(\mathbf{x}_i, \mathbf{y}_i)}_{k((\mathbf{x}_k,\mathbf{y}),(\mathbf{x}_i,\mathbf{y}_i))}$
  - Similarly for $\mathbf{w}^\top \Phi(\mathbf{x}_j, \mathbf{y}_j)$ and $\mathbf{w}^\top \mathbf{w}$.
- Carry gradients through: $\dfrac{\partial obj}{\partial \alpha_{k,\mathbf{y}}} = \dfrac{\partial obj}{\partial \mathbf{w}} \cdot \dfrac{\partial \mathbf{w}}{\partial \alpha_{k,\mathbf{y}}}$.

Working Set Approach

Problems: Exponential Complexity!

- Exponentially many **variables $\alpha_{k,\mathbf{y}}$ to optimize**.
- Also, exponentially many **arguments $\bar{\mathbf{y}}$'s in (soft)max**.

Observation:

- Only $(\mathbf{x}_i, \bar{\mathbf{y}}_i)$ with positive loss relevant.
- Same for $(\mathbf{x}_j, \bar{\mathbf{y}}_j)$.

Solution: Working Set Approach

- Labeled points: Collect worst margin violators $\bar{\mathbf{y}}_i$ (maximum loss; found by 2-best-decoder).
- Unlabeled points: Both $\mathbf{y}_j$ and $\bar{\mathbf{y}}_j$ found by 2-best-decoder.

Alternating Algorithm

Algorithm

Input: labeled points $\{(\mathbf{x}_i, \mathbf{y}_i)\}$, unlabeled points $\{\mathbf{x}_j\}$.
Output: working set $\mathcal{W}$ and associated $\alpha_{k,\mathbf{y}}$.

Initialize $\mathcal{W} \leftarrow \{(\mathbf{x}_i, \mathbf{y}_i)\}$.

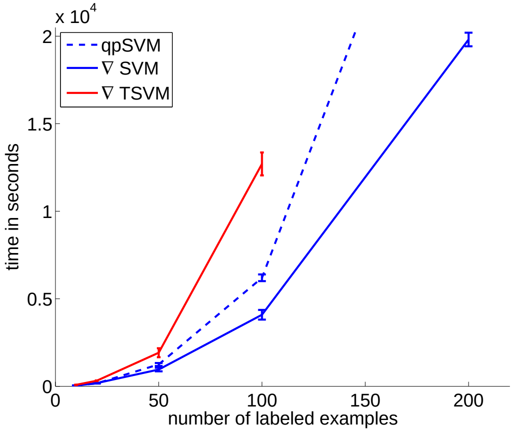
Alternate until convergence:

1. Augment working set $\mathcal{W}$
   - add $\{(\mathbf{x}_i, \bar{\mathbf{y}}_i^*)\}$ to $\mathcal{W}$ (worst margin violators)
   - find $\{\mathbf{y}_j{}^*\}$ (highest scoring labels)
   - add $\{(\mathbf{x}_j, \bar{\mathbf{y}}_j^*)\}$ to $\mathcal{W}$ (2nd highest scoring labels)
2. Optimize α by preconditioned Conjugate Gradient.

Computational Experiments

- Time comparison to QP-based optimization.
- Comparison to supervised learning:
  - Multiclass classification: Text classification.
  - Label sequence learning: Named entity recognition.
- Combination / comparison with Laplacian kernel SO-SVM, another semi-supervised SO learning approach.

Optimization Efficiency

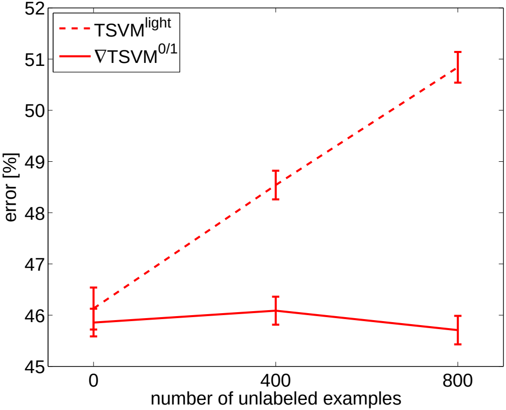


Time Comparison

∇TSVM: on top of labeled points uses 5$\times$ as many unlabeled points

- CG faster than QP-solving...
- ... even when including unlabeled examples.

Cora Dataset [Multiclass]

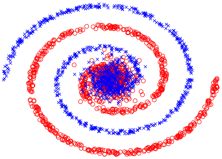


Cora Dataset

- text classification
- multiclass: 8 classes
- 200 labeled examples

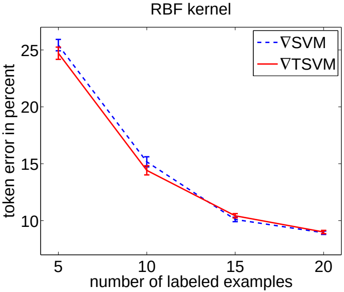
- Combinatorial optimization: error increases.
- Continuous optimization: accuracy essentially unchanged.

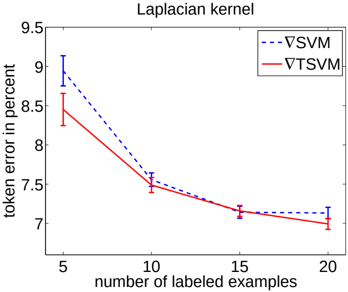
Galaxy Dataset [Laplacian Kernel]



Galaxy Dataset (artificial data)

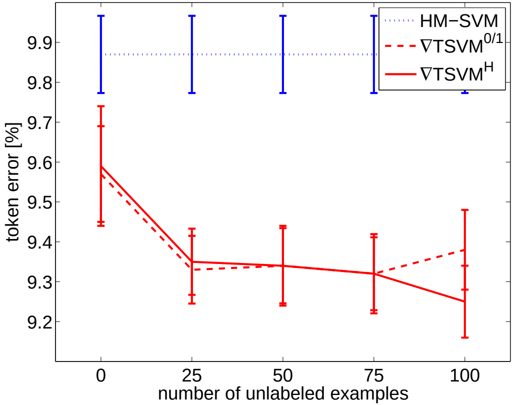
- [Lafferty et al; ICML 2004]
- label sequence learning
- #unlabeled
  $= 100 - \#\textit{labeled}$





- **Here, ∇SO-TSVM only slightly better than ∇SO-SVM.**

Spanish News Wire Dataset



Spanish News Wire Dataset

- named entity recognition
- label sequence learning
- 9 types of labels

- **Here, ∇SO-TSVM clearly outperforms HM-SVM.**

Conclusions

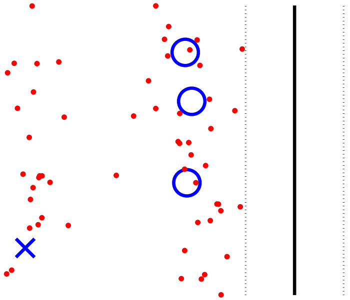
Summary

- TSVM for structured outputs:
  - Use information from unlabeled (test) examples.
  - Unconstrained, differentiable optimization criterion.
  - Efficient conjugate gradient optimization.
- SVM criterion is convex; TSVM criterium has many local minima.
- Empirically:
  - Often, no improvement – but also no deterioration.
  - Sometimes, unlabeled data increase accuracy significantly.

Thank you!

Class Balancing

binary classification: **balancing** of class sizes to avoid degenerate solutions.



Balancing for Structured Outputs

- soft constraints on label frequencies can be implemented
- however, empirically not necessary