Semi-Supervised Support Vector Machines and Application to Spam Filtering

Alexander Zien

Empirical Inference Department, Bernhard Schölkopf
Max Planck Institute for Biological Cybernetics

ECML 2006 – Discovery Challenge
1. Introduction

2. Training a $S^3$VM
   - Why It Matters
   - Some $S^3$VM Training Methods
   - Gradient-based Optimization
   - The Continuation $S^3$VM

3. Overview of SSL
   - Assumptions of SSL
   - A Crude Overview of SSL
   - Combining Methods

4. Application to Spam Filtering
   - Naive Application
   - Proper Model Selection

5. Conclusions
find a linear classification boundary
not robust wrt input noise!
SVM:
maximum margin
classifier

\[
\min_{w,b} \quad \frac{1}{2} w^\top w \\
\text{s.t.} \quad y_i (w^\top x_i + b) \geq 1
\]
**S³VM (TSVM):**
semi-supervised (transductive) SVM

\[
\begin{align*}
\min_{w,b,(y_j)} & \quad \frac{1}{2} w^\top w \\
\text{regularizer} & \quad s.t. \\
& \quad y_i(w^\top x_i + b) \geq 1 \\
& \quad y_j(w^\top x_j + b) \geq 1
\end{align*}
\]
min \begin{align*}
    & \frac{1}{2} w^\top w \\
    & + C \sum_i \xi_i \\
    & + C^* \sum_j \xi_j
\end{align*}
\quad s.t. \quad \begin{align*}
\xi_i & \geq 0 \\
\xi_j & \geq 0 \\
y_i(w^\top x_i + b) & \geq 1 - \xi_i \\
y_j(w^\top x_j + b) & \geq 1 - \xi_j
\end{align*}

soft margin S\textsuperscript{3}VM
“Two Moons” toy data

- easy for human (0% error)
- hard for $S^3$VMs!

<table>
<thead>
<tr>
<th>$S^3$VM optimization method</th>
<th>test error</th>
<th>objective value</th>
</tr>
</thead>
<tbody>
<tr>
<td>global min. {Branch &amp; Bound}</td>
<td>0.0%</td>
<td>7.81</td>
</tr>
<tr>
<td>find local minima</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCCP</td>
<td>64.0%</td>
<td>39.55</td>
</tr>
<tr>
<td>$S^3$VM$_{light}$</td>
<td>66.2%</td>
<td>20.94</td>
</tr>
<tr>
<td>$\nabla S^3$VM</td>
<td>59.3%</td>
<td>13.64</td>
</tr>
<tr>
<td>c$S^3$VM</td>
<td>45.7%</td>
<td>13.25</td>
</tr>
</tbody>
</table>

- objective function is good for SSL
- $\Rightarrow$ try to find better local minima!
\[
\begin{align*}
\text{min} & \quad \frac{1}{2} \mathbf{w}^\top \mathbf{w} + C \sum_i \xi_i + C^* \sum_j \xi_j \\
\text{s.t.} & \quad y_i (\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i \quad \xi_i \geq 0 \\
& \quad y_j (\mathbf{w}^\top \mathbf{x}_j + b) \geq 1 - \xi_j \quad \xi_j \geq 0
\end{align*}
\]

**Mixed Integer Programming** [Bennett, Demiriz; NIPS 1998]
- global optimum found by standard optimization packages (e.g., CPLEX)
- **combinatorial & NP-hard**!
- only works for small sized problems
\[
\min_{w,b,(y_j),(\xi_k)} \frac{1}{2} w^\top w + C \sum_i \xi_i + C^* \sum_j \xi_j \\
\text{s.t.} \quad y_i(w^\top x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \\
y_j(w^\top x_j + b) \geq 1 - \xi_j, \quad \xi_j \geq 0
\]

**\(S^3VM_{\text{light}} [T. Joachims; ICML 1999] \)**

- train SVM on labeled points, predict \(y_j\)'s
- in prediction, always make sure that
  \[
  \frac{\#\{y_j = +1\}}{\# \text{unlabeled points}} = \frac{\#\{y_i = +1\}}{\# \text{labeled points}} \tag{1}
  \]
- with stepwise increasing \(C^*\) do
  1. train SVM on all points, using labels \((y_i), (y_j)\)
  2. predict new \(y_j\)'s s.t. “balancing constraint” (*)
\[
\min_{w, b, (y_j), (\xi_k)} \quad \frac{1}{2} w^\top w + C \sum_i \xi_i + C^* \sum_j \xi_j \\
\text{s.t.} \\
y_i (w^\top x_i + b) \geq 1 - \xi_i \quad \xi_i \geq 0 \\
y_j (w^\top x_j + b) \geq 1 - \xi_j \quad \xi_j \geq 0
\]
Introduction

Training a S³VM

Overview of SSL

Application to Spam Filtering

Conclusions

\[
\min_{w,b,(y_j),(\xi_k)} \frac{1}{2} w^\top w + C \sum_i \xi_i + C^* \sum_j \xi_j \\
\text{s.t.} \\
y_i(w^\top x_i + b) \geq 1 - \xi_i \quad \xi_i \geq 0 \\
y_j(w^\top x_j + b) \geq 1 - \xi_j \quad \xi_j \geq 0 
\]

Effective Loss Functions

\[
\xi_i = \min \left\{ 1 - y_i(w^\top x_i + b), 0 \right\} \\
\xi_j = \min_{y_j \in \{+1,-1\}} \left\{ 1 - y_j(w^\top x_j + b), 0 \right\}
\]

loss functions

\[
y_i(w^\top x_i + b) \\
w^\top x_j + b
\]
min \( w, b, (y_j), (\xi_k) \)

\[
\begin{align*}
\frac{1}{2} w^\top w + C \sum_i \xi_i + C^* \sum_j \xi_j \\

s.t. \\
y_i (w^\top x_i + b) \geq 1 - \xi_i \quad \xi_i \geq 0 \\
y_j (w^\top x_j + b) \geq 1 - \xi_j \quad \xi_j \geq 0
\end{align*}
\]

Resolving the Constraints

\[
\frac{1}{2} w^\top w + C \sum_i \ell_l \left( y_i (w^\top x_i + b) \right) + C^* \sum_j \ell_u \left( w^\top x_j + b \right)
\]

loss functions

\( \ell_l \)  \hspace{2cm}  \( \ell_u \)
\[
\frac{1}{2}w^Tw + C \sum_i \ell_i \left( y_i (w^Tx_i + b) \right) + C^* \sum_j \ell_u \left( w^Tx_j + b \right)
\]

**CCCP-S\(^3\)VM [R. Collobert et al.; ICML 2006]**

- **CCCP**: “Concave Convex Procedure”
- **objective** = convex function + concave function
- starting from SVM solution, iterate:
  1. approximate concave part by linear function at given point
  2. solve resulting convex problem

**[Fung, Mangasarian; 1999]**

- similar approach
- restricted to linear S\(^3\)VMs
\[
\frac{1}{2}w^Tw + C\sum_i \ell_l \left( y_i(w^Tx_i + b) \right) + C^*\sum_j \ell_u \left( w^Tx_j + b \right)
\]
\[
\frac{1}{2}w^T w + C \sum_i \ell_i \left( y_i (w^T x_i + b) \right) + C^* \sum_j \ell_u \left( w^T x_j + b \right)
\]

∇S³VM [Chapelle, Zien; AISTATS 2005]
- simply do gradient descent!
- thereby stepwise increase \( C^* \)

contS³VM [Chapelle et al.; ICML 2006]
... in more detail on next slides!
\[
\frac{1}{2} \mathbf{w}^\top \mathbf{w} + C \sum_i \ell_l \left( y_i (\mathbf{w}^\top \mathbf{x}_i + b) \right) + C^* \sum_j \ell_u \left( \mathbf{w}^\top \mathbf{x}_j + b \right)
\]

**Hard Balancing Constraint**

\[
\text{S}^3\text{VM}^{light} \\
\text{constraint} \\
\frac{\# \{ y_j = +1 \}}{\# \text{ unlabeled points}} = \frac{\# \{ y_i = +1 \}}{\# \text{ labeled points}}
\]

\[
\frac{1}{m} \sum_j \text{sign} \left( \mathbf{w}^\top \mathbf{x}_j + b \right) = \frac{1}{n} \sum_i y_i
\]

**Hard Balancing Constraint**

\[
\text{S}^3\text{VM}^{light} \\
\text{constraint} \\
\frac{\# \{ y_j = +1 \}}{\# \text{ unlabeled points}} = \frac{\# \{ y_i = +1 \}}{\# \text{ labeled points}}
\]

\[
\frac{1}{m} \sum_j \text{sign} \left( \mathbf{w}^\top \mathbf{x}_j + b \right) = \frac{1}{n} \sum_i y_i
\]
Making the Balancing Constraint Linear

\[
\frac{1}{m} \sum_j \text{sign} \left( \mathbf{w}^\top \mathbf{x}_j + b \right) = \frac{1}{n} \sum_i y_i
\]

average prediction

average label

\[
\frac{1}{m} \sum_j \mathbf{w}^\top \mathbf{x}_j + b = \frac{1}{n} \sum_i y_i
\]

mean output on unlabeled points

average label

Implementing the linear soft balancing:

- center the unlabeled data: \( \sum_j \mathbf{x}_j = 0 \)
- \( \Rightarrow \) just fix \( b \); unconstrained optimization over \( \mathbf{w} \)
The Continuation Method in a Nutshell

**Procedure**

1. Smooth function until convex
2. Find minimum
3. Track minimum while decreasing amount of smoothing

**Illustration**
Smoothing the S$^3$VM Objective $f(\cdot)$

Convolution of $f(\cdot)$ with Gaussian of width $\sqrt{\gamma}/2$:

$$f_\gamma(w) = (\pi \gamma)^{-d/2} \int f(w - t) \exp(-||t||^2/\gamma) dt$$

Closed form solution!

Smoothing Sequence

- choose $\gamma_0 > \gamma_1 > \ldots \gamma_{p-1} > \gamma_p = 0$
  - choose $\gamma_0$ such that $f_{\gamma_0}(\cdot)$ is convex
  - choose $\gamma_{p-1}$ such that $f_{\gamma_{p-1}}(\cdot) \approx f_{\gamma_p}(\cdot) = f(\cdot)$
  - $p = 10$ steps (equidistant on log scale) sufficient
Handling Non-Linearity

Consider non-linear map $\Phi(x)$, kernel $k(x_i, x_j) = \Phi(x_i)^\top \Phi(x_j)$.

**Representer Theorem:** $S^3$VM solution is in span $E$ of data points

$$E := \text{span}\{\Phi(x_i)\} \overset{\wedge}{=} \mathbb{R}^{n+m}$$

**Implementation**

1. expand basis vectors $v_i$ of $E$:

   $$v_i = \sum_k A_{ik} \Phi(x_k)$$

2. orthonormality gives:

   solve for $A$, eg by KPCA or Choleski

   $$(A^\top A)^{-1} = K$$

3. project data $\Phi(x_i)$ on basis $V = (v_j)_j$:

   $$\tilde{x}_i = V^\top \Phi(x_i) = (A)_i$$
Comparison of $S^3$VM Optimization Methods

- averaged over splits (and pairs of classes)
- fixed hyperparams (close to hard margin)
- similar results for other hyperparameter settings

[Chapelle, Chi, Zien; ICML 2006]
Why would unlabeled data be useful at all?

Uniform data do not help.
Why would unlabeled data be useful at all?

Uniform data do not help.
Cluster Assumption

Points in the same cluster are likely to be of the same class.

Algorithmic idea: Low Density Separation
**Manifold Assumption**

The data lie on (close to) a low-dimensional manifold.


Algorithmic idea: use **Nearest-Neighbor Graph**
Assumption: Independent Views Exist

There exist **subsets of features, called views**, each of which

- is **independent** of the others given the class;
- is **sufficient** for classification.

Algorithmic idea: **Co-Training**
<table>
<thead>
<tr>
<th>Assumption</th>
<th>Approach</th>
<th>Example Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster Assumption</td>
<td>Low Density Separation</td>
<td>( \text{S}^3\text{VM}; \text{ Entropy Regularization}; \text{ Data-Dependent Regularization}; \ldots )</td>
</tr>
</tbody>
</table>
| Manifold Assumption              | Graph-based Methods            | - build weighted graph \( (w_{kl}) \)  
- \( \min_{(y_j)} \sum_k \sum_l w_{kl} (y_k - y_l)^2 \)  
- relax \( y_j \) to be real \( \Rightarrow \text{QP} \) |
| Independent Views                | Co-Training                    | - train two predictors \( y_j^{(1)}, y_j^{(2)} \)  
- couple objectives by adding  
  \( \sum_j \left( y_j^{(1)} - y_j^{(2)} \right)^2 \) |
### Discriminative Learning (Diagnostic Paradigm)

- **model** $p(y|x)$ (or just boundary: $\{x \mid p(y|x) = \frac{1}{2}\}$)
- examples: $S^3$VM, graph-based methods

### Generative Learning (Sampling Paradigm)

- **model** $p(x|y)$
- predict via Bayes: $p(y|x) = \frac{p(y)p(x|y)}{\sum_{y'} p(y')p(x|y')}$
- $\Rightarrow$ missing data problem
- EM algorithm (expectation-maximization) is a natural tool
- successful for text data [Nigam et al.; Machine Learning, 2000]
SSL Book

- MIT Press, Sept. 2006
- edited by B. Schölkopf, O. Chapelle, A. Zien
- contains many state-of-art algorithms by top researchers
- extensive SSL benchmark
- online material:
  - sample chapters
  - benchmark data
  - more information

http://www.kyb.tuebingen.mpg.de/ssl-book/
## SSL Book – Text Benchmark

<table>
<thead>
<tr>
<th></th>
<th>error [%]</th>
<th>AUC [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>l=10</td>
<td>l=100</td>
</tr>
<tr>
<td></td>
<td>l=10</td>
<td>l=100</td>
</tr>
<tr>
<td>1-NN</td>
<td>38.12</td>
<td>30.11</td>
</tr>
<tr>
<td>SVM</td>
<td>45.37</td>
<td>26.45</td>
</tr>
<tr>
<td>MVU + 1-NN</td>
<td>45.32</td>
<td>32.83</td>
</tr>
<tr>
<td>LEM + 1-NN</td>
<td>39.44</td>
<td>30.77</td>
</tr>
<tr>
<td>QC + CMN</td>
<td>40.79</td>
<td>25.71</td>
</tr>
<tr>
<td>Discrete Reg.</td>
<td>40.37</td>
<td>24.00</td>
</tr>
<tr>
<td>TSVM</td>
<td>31.21</td>
<td>24.52</td>
</tr>
<tr>
<td>SGT</td>
<td>29.02</td>
<td>23.09</td>
</tr>
<tr>
<td>Cluster-Kernel</td>
<td>42.72</td>
<td>24.38</td>
</tr>
<tr>
<td>LDS</td>
<td>27.15</td>
<td>23.15</td>
</tr>
<tr>
<td>Laplacian RLS</td>
<td>33.68</td>
<td>23.57</td>
</tr>
</tbody>
</table>
Combining $S^3$VM with Graph-based Regularizer

- LapSVM [1]: modify kernel using graph, then train SVM
- combination with $S^3$VM even better [2]
- MNIST, “3” vs “5”

[1] “Beyond the Point Cloud”; Sindhwani, Niyogi, Belkin; ICML 2005

Combining $S^3$VM with Co-Training

“SSL for Structured Output Variables”; Brefeld, Scheffer; ICML 2006
\[ \min_{w,b,(y_j),(\xi_k)} \frac{1}{2} w^\top w + C \sum \xi_i + C^* \sum \xi_j \]

s.t.

\[ y_i (w^\top x_i + b) \geq 1 - \xi_i \quad \xi_i \geq 0 \]

\[ y_j (w^\top x_j + b) \geq 1 - \xi_j \quad \xi_j \geq 0 \]

**How to set \( C \)?**

- data fitting, \( y_i w^\top x_i \geq 1 \), and regularization, \( \min ||w||^2 \):

\[ ||w^\top x_i|| = O(1) \ \Rightarrow \ ||w||^2 \approx \text{Var}[x]^{-1} \]

- balance influence: \( ||w||^2 \approx C\xi_i \ \Rightarrow \ C \approx \text{Var}[x]^{-1} \)

**How to set \( C^* \)?**

- \( C^* = C \)

- \( C^* = \lambda \frac{\# \text{unlabeled points}}{\# \text{labeled points}} C \)
Naive Application:

- **Transductive setting** on each user/inbox:
  - use inbox of given user as unlabeled data
  - test data = unlabeled data

- **Guess the model:**
  - $\text{Var}[x] \approx 1$, so set $C = 1$
  - $C^* = C$
  - linear kernel

### Results: AUC (rank) [rank in unofficial list]

<table>
<thead>
<tr>
<th></th>
<th>task A</th>
<th>task B</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S^3\text{VM}_{\text{light}}$</td>
<td>94.53% (4) [6]</td>
<td>92.34% (2) [4]</td>
</tr>
<tr>
<td>$\nabla S^3\text{VM}$</td>
<td>96.72% (1) [3]</td>
<td>93.74% (2) [4]</td>
</tr>
<tr>
<td>cont$S^3\text{VM}$</td>
<td>96.01% (1) [3]</td>
<td>93.56% (2) [4]</td>
</tr>
</tbody>
</table>
Model selection:
- \( C \in \{10^{-2}, 10^{-1}, 10^0, 10^1, 10^2\} \)
- \( C^* \in \{10^{-2}, 10^{-1}, 10^0, 10^1, 10^2\} \cdot C \)
- cross-validation (3-fold for task A; 5-fold for task B)

Results: AUC for contS\(^3\)VM

<table>
<thead>
<tr>
<th>Model</th>
<th>task A</th>
<th>task B</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C = C^* = 1 ) (guessed model)</td>
<td>96.01%</td>
<td>93.56%</td>
</tr>
<tr>
<td>model selection</td>
<td>89.31%</td>
<td>90.09%</td>
</tr>
</tbody>
</table>

- significant drop in accuracy!
- CV relies on iid assumption: that the data are independent identically distributed
Take Home Messages

- $S^3$VM implements "low density separation" (margin maximization)
- optimization technique matters (non-convex objective)
- works well for text classification (texts form clusters)
- $S^3$VM-based hybrids may be even better
- for spam filtering, further methods needed to cope with non-iid situation (mail inboxes)!

Thank you!