Semi-Supervised Learning
in “Advanced Methods in Sequence Analysis”

Alexander Zien

Max Planck Institute for Biological Cybernetics (B. Schölkopf)
Friedrich Mischer Laboratory (G. Rätsch)

Tübingen, 29. Nov. 2006
1 Review of the SVM

2 The Semi-Supervised SVM (S\(^3\)VM)

3 Training a S\(^3\)VM

4 Semi-Supervised Learning (SSL): Assumptions and Methods

5 Overview of SSL and Summary
find a linear classification boundary
Training a S³VM
not robust wrt input noise!
SVM:
maximum margin classifier

\[
\begin{align*}
\langle w, x \rangle + b &= +\rho \\
\langle w, x \rangle + b &= -\rho
\end{align*}
\]

\[
\max_{w,b,\rho} \quad \rho \\
\text{s.t.} \quad y_i (\langle w, x_i \rangle + b) \geq \rho, \quad \|w\| = 1
\]

\text{margin} \quad \text{data fitting} \quad \text{normalization}
Equivalent reformulation of the SVM

\[
\max_{w, b, \rho} \quad \rho \\
\text{s.t.} \quad y_i (\langle w, x_i \rangle + b) \geq \rho, \quad \|w\| = 1
\]

\[\Leftrightarrow \max_{w', b, \rho} \quad \rho^2 \quad \text{s.t.} \quad y_i \left( \frac{\langle w', x_i \rangle}{\|w'\|} + b \right) \geq \rho, \quad \rho \geq 0\]

\[\Leftrightarrow \max_{w', b, \rho} \quad \rho^2 \quad \text{s.t.} \quad y_i \left( \frac{\langle w', x_i \rangle}{\|w'\| \rho} + b \frac{b'}{\rho} \right) \geq 1, \quad \rho \geq 0\]

\[\Leftrightarrow \max_{w'', b''} \quad \frac{1}{\|w''\|^2} \quad \text{s.t.} \quad y_i (\langle w'', x_i \rangle + b'') \geq 1,\]

using \[\|w''\| = \left\| \frac{w'}{\|w'\| \rho} \right\| = \frac{1}{\rho} \cdot \|w'\| = \frac{1}{\rho} \]
SVM:
maximum margin classifier

\[
\langle \mathbf{w}, \mathbf{x} \rangle + b = +1
\]
\[
\langle \mathbf{w}, \mathbf{x} \rangle + b = 0
\]
\[
\langle \mathbf{w}, \mathbf{x} \rangle + b = -1
\]

\[
\min_{\mathbf{w},b} \frac{1}{2} \langle \mathbf{w}, \mathbf{w} \rangle \quad \text{s.t.} \quad y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1
\]

regularizer

data fitting
**S³VM (TSVM):**
semi-supervised (transductive)

SVM

\[
\begin{align*}
\min_{w,b,(y_j)} & \quad \frac{1}{2} \langle w, w \rangle \\
\text{regularizer} & \quad s.t. \quad y_i(\langle w, x_i \rangle + b) \geq 1 \\
& \quad y_j(\langle w, x_j \rangle + b) \geq 1
\end{align*}
\]
Training a $S^3$VM

**Summary**

The soft margin $S^3$VM model minimizes the objective function:

$$
\min_{w, b, (y_j), (\xi_k)} \frac{1}{2} \langle w, w \rangle + C \sum_i \xi_i + C^* \sum_j \xi_j
$$

subject to:

$$
\begin{align*}
\xi_i &\geq 0 & \xi_j &\geq 0 \\
y_i(\langle w, x_i \rangle + b) &\geq 1 - \xi_i \\
y_j(\langle w, x_j \rangle + b) &\geq 1 - \xi_j
\end{align*}
$$

This formulation allows for some data points to be on the wrong side of the margin, introducing a soft margin trade-off.
Supervised Support Vector Machine (SVM)

\[
\min_{w, b, (\xi_k)} \frac{1}{2} \langle w, w \rangle + C \sum_i \xi_i \\
\text{s.t.} \quad y_i(\langle w, x_i \rangle + b) \geq 1 - \xi_i \\
\xi_i \geq 0
\]

- maximize margin on (labeled) points
- convex optimization problem (QP)

Semi-Supervised Support Vector Machine (S\textsuperscript{3}VM)

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

- maximize margin on labeled and unlabeled points
- combinatorial optimization problem (optimize \( y_j \in \{0, 1\} \))
Training a $S^3$VM

$$\min_{w,b,(y_j), (\xi_k)} \frac{1}{2} \langle w, w \rangle + C \sum_i \xi_i + C^* \sum_j \xi_j$$

s.t. $y_i (\langle w, x_i \rangle + b) \geq 1 - \xi_i \quad \xi_i \geq 0$

$y_j (\langle w, x_j \rangle + b) \geq 1 - \xi_j \quad \xi_j \geq 0$

Mixed Integer Programming [Bennett, Demiriz; NIPS 1998]
- global optimum found by standard optimization packages (eg CPLEX)
- **NP-hard** ! $\Rightarrow$ only works for small sized problems

Branch & Bound [Chapelle, Sindhwani, Keerthi; NIPS 2006]
- global optimum found
- problem structure exploited to reduce space to be searched
- again, only works for rather small sized problems
**“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><strong>global min.</strong> {Branch &amp; Bound}</td>
<td>0.0%</td>
<td>7.81</td>
</tr>
<tr>
<td><strong>find local minima</strong></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!**
\[
\begin{align*}
\min_{w,b,(y_j),(\xi)} & \quad \frac{1}{2} \langle w, w \rangle + C \sum_i \xi_i + C^* \sum_j \xi_j \\
\text{s.t.} & \quad y_i(\langle w, x_i \rangle + b) \geq 1 - \xi_i \quad \xi_i \geq 0 \\
& \quad y_j(\langle w, x_j \rangle + b) \geq 1 - \xi_j \quad \xi_j \geq 0
\end{align*}
\]

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

- train SVM on labeled points, predict \(y_j\)'s
- in prediction, always make sure that
  \[
  \frac{\#\{y_j = +1\}}{\# \text{ unlabeled points}} = \frac{\#\{y_i = +1\}}{\# \text{ labeled points}} \quad (*)
  \]
- 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” (*)
Training a S\(^3\)VM

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

Balancing constraint required to avoid degenerate solutions!
min \mathbf{w}, b, (y_j), (\xi_k)
\begin{align*}
\frac{1}{2} \langle \mathbf{w}, \mathbf{w} \rangle + C \sum_i \xi_i + C^* \sum_j \xi_j
\end{align*}
\text{s.t.}
\begin{align*}
y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) &\geq 1 - \xi_i \quad \xi_i \geq 0 \\
y_j(\langle \mathbf{w}, \mathbf{x}_j \rangle + b) &\geq 1 - \xi_j \quad \xi_j \geq 0
\end{align*}

**Effective Loss Functions**

\[ \xi_i = \min \left\{ 1 - y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b), 0 \right\} \]

\[ \xi_j = \min_{y_j \in \{+1, -1\}} \left\{ 1 - y_j(\langle \mathbf{w}, \mathbf{x}_j \rangle + b), 0 \right\} \]
\[
\min_{w, b, (y_j), (\xi_k)} \frac{1}{2} \langle w, w \rangle + C \sum_i \xi_i + C^* \sum_j \xi_j \\
\text{s.t.} \\
y_i (\langle w, x_i \rangle + b) \geq 1 - \xi_i \quad \xi_i \geq 0 \\
y_j (\langle w, x_j \rangle + b) \geq 1 - \xi_j \quad \xi_j \geq 0
\]

Resolving the Constraints

\[
\frac{1}{2} \langle w, w \rangle + C \sum_i \ell_l (y_i (\langle w, x_i \rangle + b)) + C^* \sum_j \ell_u (\langle w, x_j \rangle + b)
\]

loss functions

\[
\ell_l \quad \ell_u
\]
\[
\frac{1}{2} \langle \mathbf{w}, \mathbf{w} \rangle + C \sum_i \ell_i \left( y_i \left( \langle \mathbf{w}, \mathbf{x}_i \rangle + b \right) \right) + C^* \sum_j \ell_u \left( \langle \mathbf{w}, \mathbf{x}_j \rangle + b \right)
\]

**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
\[ \frac{1}{2} \langle \mathbf{w}, \mathbf{w} \rangle + C \sum_i \ell_i \left( y_i (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \right) + C^* \sum_j \ell_u \left( \langle \mathbf{w}, \mathbf{x}_j \rangle + b \right) \]

**S³VM as Unconstrained Differentiable Optimization Problem**

- **Original loss functions**
  - \( \ell_i \)
  - \( \ell_u \)

- **Smooth loss functions**
  - \( \ell_i \)
  - \( \ell_u \)
\[
\frac{1}{2}\langle w, w \rangle + C \sum_i \ell_i \left(y_i(\langle w, x_i \rangle + b)\right) + C^* \sum_j \ell_u \left(\langle w, x_j \rangle + b\right)
\]

\(\nabla S^3VM\) [Chapelle, Zien; AISTATS 2005]
- simply do gradient descent!
- thereby stepwise increase \(C^*\)

\(contS^3VM\) [Chapelle et al.; ICML 2006]
... in more detail on next slides!
\[
\frac{1}{2} \langle \mathbf{w}, \mathbf{w} \rangle + C \sum_i \ell_i (y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b)) + C^* \sum_j \ell_u (\langle \mathbf{w}, \mathbf{x}_j \rangle + b)
\]

**Hard Balancing Constraint**

\[S^3VM^{light}\]

<table>
<thead>
<tr>
<th>Constraint Type</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>(S^3VM^{light}) constraint</td>
<td>(\frac{#{y_j = +1}}{# \text{ unlabeled points}} = \frac{#{y_i = +1}}{# \text{ labeled points}})</td>
</tr>
<tr>
<td>Equivalent constraint</td>
<td>(\frac{1}{m} \sum_{j=1}^{m} \text{sign}(\langle \mathbf{w}, \mathbf{x}<em>j \rangle + b) = \frac{1}{n} \sum</em>{i=1}^{n} y_i)</td>
</tr>
</tbody>
</table>

Average prediction vs. average label
Making the Balancing Constraint Linear

\[
\begin{align*}
\text{hard / non-linear} & \quad \frac{1}{m} \sum_j \text{sign} (\langle \mathbf{w}, \mathbf{x}_j \rangle + b) = \frac{1}{n} \sum_i y_i \\
\text{average prediction} & \quad \text{average label} \\
\text{soft / linear} & \quad \frac{1}{m} \sum_j \langle \mathbf{w}, \mathbf{x}_j \rangle + b = \frac{1}{n} \sum_i y_i \\
\text{mean output on unlabeled points} & \quad \text{average label}
\end{align*}
\]

Implementing the linear soft balancing:
- center the unlabeled data: \( \sum_j x_j = 0 \)
- \( \Rightarrow \) 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**

- Large $\gamma$
- Smaller $\gamma$
- $\gamma=0$
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 $k(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$.

**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:
   $$(A^\top A)^{-1} = K$$
   solve for $A$, eg by KPCA or Choleski decomposition

3. project data $\Phi(x_i)$ on basis $V = (v_j)_j$:
   $$\tilde{x}_i = V^\top \Phi(x_i) = (A)_i$$
Comparison of \( S^3 \text{VM} \) 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.
Cluster Assumption

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

Algorithmic idea: **Low Density Separation**, eg $S^3$VM
Manifold Assumption

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

Algorithmic idea: use **Nearest-Neighbor Graph**

Graph Construction

- nodes: data points $x_k$
- edges: every edge $(x_k, x_l)$ weighted with $a_{kl} \geq 0$
- weights: represent similarity, eg $a_{kl} = \exp(-\gamma \|x_k - x_l\|)$

approximate manifold / achieve sparsity – two choices:
1. $k$ nearest neighbor graph (usually preferred)
2. $\epsilon$ distance graph

Learning on the Graph

estimation of a function on the edges, ie $f : E \rightarrow \{-1, +1\}$
[recall: for SVMs, $f : \mathcal{X} \rightarrow \{-1, +1\}$, $x \mapsto sign(\langle w, x \rangle + b)$ ]
Regularization on a Graph – penalize change along edges

\[
\min_{(y_j)} g(y) \quad \text{with} \quad g(y) := \frac{1}{2} \sum_k \sum_l a_{kl} (y_k - y_l)^2
\]

\[
\begin{align*}
g(y) &= \frac{1}{2} \left( \sum_k \sum_l a_{kl} y_k^2 + \sum_k \sum_l a_{kl} y_l^2 \right) - \sum_k \sum_l a_{kl} y_k y_l \\
&= \sum_k y_k^2 \sum_l a_{kl} - \sum_k \sum_l y_k a_{kl} y_l \\
&= y^\top D y - y^\top A y = y^\top L y
\end{align*}
\]

where \( D \) is the diagonal matrix with \( d_{kl} = \sum_k a_{kl} \)
and \( L := D - A \) is called the graph Laplacian

with constraints \( y_j \in \{-1, +1\} \) essentially yields min-cut problem
Label Propagation

relax: instead of \( y_j \in \{-1, +1\} \), optimize free \( f_j \)

\( \Rightarrow \) fix \( f_l = (f_i) = (y_i) \), solve for \( f_u = (f_j) \), predict \( y_j = \text{sign}(f_j) \)

\( \Rightarrow \) convex QP (\( L \) is positive definite)

\[
0 = \frac{\partial}{\partial f_u} \begin{pmatrix} f_l \\ f_u \end{pmatrix}^\top \begin{pmatrix} L_{ll}L_{ul}^\top \\ L_{ul}L_{uu} \end{pmatrix} \begin{pmatrix} f_l \\ f_u \end{pmatrix} = \frac{\partial}{\partial f_u} \left( f_u^\top L_{ul}f_l + f_l^\top L_{ul}f_u + f_u^\top L_{uu}f_u \right) = 2f_l^\top L_{ul} + 2f_u^\top L_{uu}
\]

\( \Rightarrow \) solve linear system \( L_{uu}f_u = -L_{lu}^\top f_l \) \( (f_u = -L_{uu}^{-1}L_{lu}^\top f_l) \)

- easy to do in \( \mathcal{O}(n^3) \) time; faster for sparse graphs
- solution can be shown to satisfy \( f_j \in [-1, +1] \)
Called **Label Propagation**, as the same solution is achieved by iteratively propagating labels along edges until convergence.

![Diagram](images)

**Note:** here color \( \triangleq \) classes

[images from “Learning with Local and Global Consistency”, Zhou, Bousquet, Lal, Weston, Schölkopf; NIPS 2004]
“Beyond the Point Cloud”

Idea:

- model output \( f_j \) as linear function of the node value \( x_j \)
  \[
  f_k = \mathbf{w}^\top \mathbf{x}_k
  \]  (with kernels: \( f_k = \sum_l \alpha_l k(x_l, x_k) \))

- add graph regularizer to SVM cost function
  \[
  R_g(\mathbf{w}) = \frac{1}{2} \sum_k \sum_l a_{kl} (f_k - f_l)^2 = \mathbf{f}^\top \mathbf{Lf}
  \]

\[
\min_{\mathbf{w}} \quad \sum_i \ell(y_i(\mathbf{w}^\top \mathbf{x}_i)) + \lambda \|\mathbf{w}\|^2 + \gamma R_g(\mathbf{w})
\]

- data fitting
- regularizers

- linear (\( \mathbf{f} = \mathbf{Xw} \)): \( \Rightarrow \lambda \mathbf{w}^\top \mathbf{w} + \gamma \mathbf{w}^\top \mathbf{X}^\top \mathbf{LXw} \)
- w. kernel (\( \mathbf{f} = \mathbf{K}\alpha \)): \( \Rightarrow \lambda \alpha^\top \mathbf{K}\alpha + \gamma \alpha^\top \mathbf{KLK}\alpha \)
Graph Methods

Observation
graphs model **density** on manifold
⇒ graph methods also implement cluster assumption
Cluster Assumption
1. The data form clusters.
2. Points in the same cluster are likely to be of the same class.

Manifold Assumption
1. The data lie on (or close to) a low-dimensional manifold.
2. Its intrinsic distance is relevant for classification.

Semi-Supervised Smoothness Assumption
1. The density is non-uniform.
2. If two points are close in a high density region (⇒ connected by high density path), their outputs are close.
**S\(^3\)VMs**

- Cluster Assumption
- Points within the same cluster are of **same class**
- Non-convex

**Graph methods**

- Semi-Supervised Smoothness
- Points within the same cluster have **same class probabilities**
- Convex
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**
Co-Training with SVM

use multiple views \( v \) on the input data

\[
\begin{align*}
\min_{w^v, (y^j), \xi_k} & \quad \frac{1}{2} \sum_v \|w^v\|^2 + C \sum_i \xi_i + C^* \sum_j \xi_j \\
\text{s.t.} & \quad \forall v: y^i \left( \langle w^v, \Phi^v(x^i) \rangle + b \right) \geq 1 - \xi_i, \quad \xi_i \geq 0 \\
& \quad \forall v: y^j \left( \langle w^v, \Phi^v(x^j) \rangle + b \right) \geq 1 - \xi_j, \quad \xi_j \geq 0
\end{align*}
\]
<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>(S^3\text{VM}; ) Entropy Regularization; (\text{Data-Dependent Regularization}; \ldots)</td>
</tr>
<tr>
<td>Manifold Assumption</td>
<td>Graph-based Methods</td>
<td>- build weighted graph ((w_{kl}))</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- (\min (y_j) \sum_k \sum_l w_{kl} (y_k - y_l)^2)</td>
</tr>
<tr>
<td>Independent Views</td>
<td>Co-Training</td>
<td>- train two predictors (y^{(1)}_j, y^{(2)}_j)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- couple objectives by adding (\sum_j \left(y^{(1)}_j - y^{(2)}_j\right)^2)</td>
</tr>
</tbody>
</table>
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
- successfully used [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 – Benchmark

<table>
<thead>
<tr>
<th>Method</th>
<th>g241c</th>
<th>g241d</th>
<th>Digit1</th>
<th>USPS</th>
<th>COIL</th>
<th>BCI</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-NN</td>
<td>43.93</td>
<td>42.45</td>
<td>3.89</td>
<td>5.81</td>
<td>17.35</td>
<td>48.67</td>
<td>30.11</td>
</tr>
<tr>
<td>SVM</td>
<td>23.11</td>
<td>24.64</td>
<td>5.53</td>
<td>9.75</td>
<td>22.93</td>
<td>34.31</td>
<td>26.45</td>
</tr>
<tr>
<td>MVU + 1-NN</td>
<td>43.01</td>
<td>38.20</td>
<td>2.83</td>
<td>6.50</td>
<td>28.71</td>
<td>47.89</td>
<td>32.83</td>
</tr>
<tr>
<td>LEM + 1-NN</td>
<td>40.28</td>
<td>37.49</td>
<td>6.12</td>
<td>7.64</td>
<td>23.27</td>
<td>44.83</td>
<td>30.77</td>
</tr>
<tr>
<td>Label-Prop.</td>
<td>22.05</td>
<td>28.20</td>
<td>3.15</td>
<td>6.36</td>
<td>10.03</td>
<td>46.22</td>
<td>25.71</td>
</tr>
<tr>
<td>Discrete Reg.</td>
<td>43.65</td>
<td>41.65</td>
<td>2.77</td>
<td>4.68</td>
<td>9.61</td>
<td>47.67</td>
<td>24.00</td>
</tr>
<tr>
<td>$S^3$SVM</td>
<td>18.46</td>
<td>22.42</td>
<td>6.15</td>
<td>9.77</td>
<td>25.80</td>
<td>33.25</td>
<td>24.52</td>
</tr>
<tr>
<td>SGT</td>
<td>17.41</td>
<td>9.11</td>
<td>2.61</td>
<td>6.80</td>
<td></td>
<td>45.03</td>
<td>23.09</td>
</tr>
<tr>
<td>Data-Dep. Reg.</td>
<td>20.31</td>
<td>32.82</td>
<td>2.44</td>
<td>5.10</td>
<td>11.46</td>
<td>47.47</td>
<td></td>
</tr>
<tr>
<td>LDS</td>
<td>18.04</td>
<td>23.74</td>
<td>3.46</td>
<td>4.96</td>
<td>13.72</td>
<td>43.97</td>
<td>23.15</td>
</tr>
<tr>
<td>Graph-Reg.</td>
<td>24.36</td>
<td>26.46</td>
<td>2.92</td>
<td>4.68</td>
<td>11.92</td>
<td>31.36</td>
<td>23.57</td>
</tr>
<tr>
<td>CHM (normed)</td>
<td>24.82</td>
<td>25.67</td>
<td>3.79</td>
<td>7.65</td>
<td></td>
<td>36.03</td>
<td></td>
</tr>
</tbody>
</table>

average error [%] achieved with 100 labeled points
Combining $S^3$VM with Graph-based Regularizer

- apply SVM and $S^3$VM in the “warped space”
- strength of graph regularizer on x-axis
- MNIST digit classification data, “3” vs “5”

“A Continuation Method for $S^3$VM”; Chapelle, Chi, Zien; ICML 2006
Summary

- unlabeled data can improve classification (most useful if few labeled data available)

- verify if assumptions hold!

- two ways to use unlabeled data:
  
  - in the loss function ($S^3$VM, co-training)
    non-convex – optimization method matters!
  
  - in the regularizer (graph methods)
    convex, but graph construction matters

- combination seems to work best

Thank you!