Semi-Supervised Learning

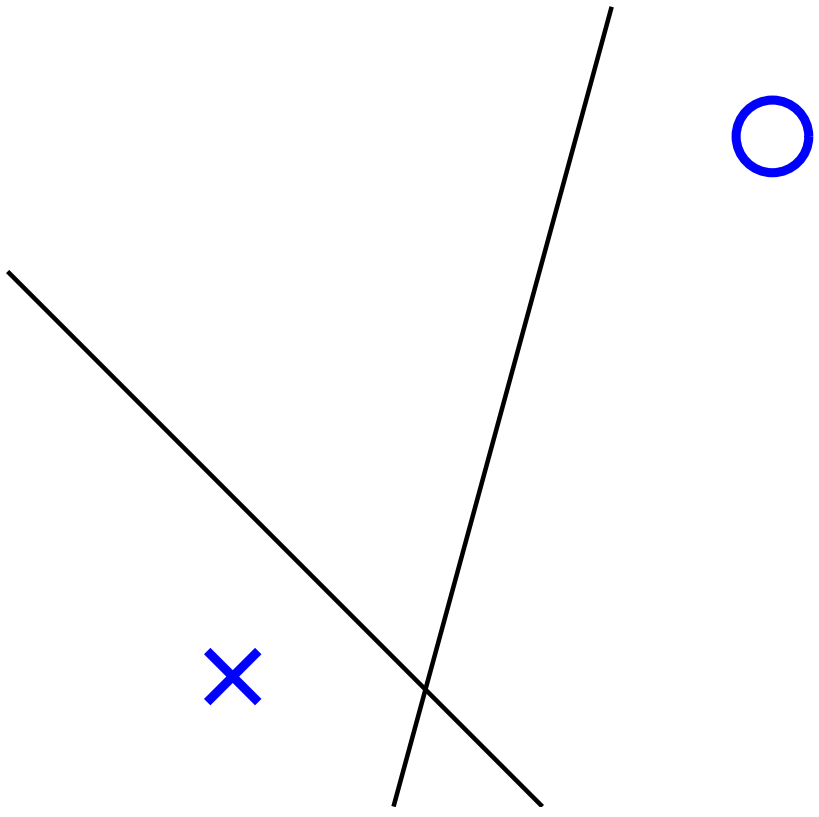
in "Advanced Methods in Sequence Analysis"

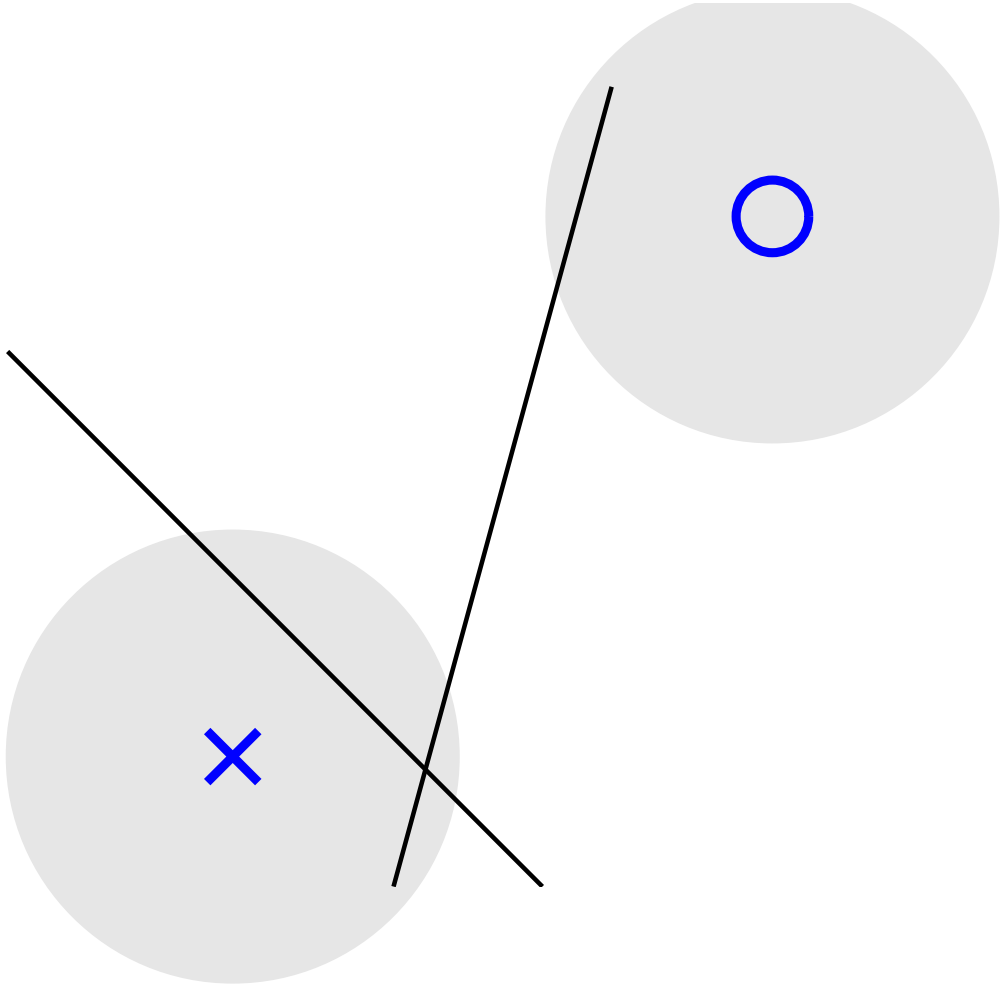
Alexander Zien

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

Tübingen, 29. Nov. 2006

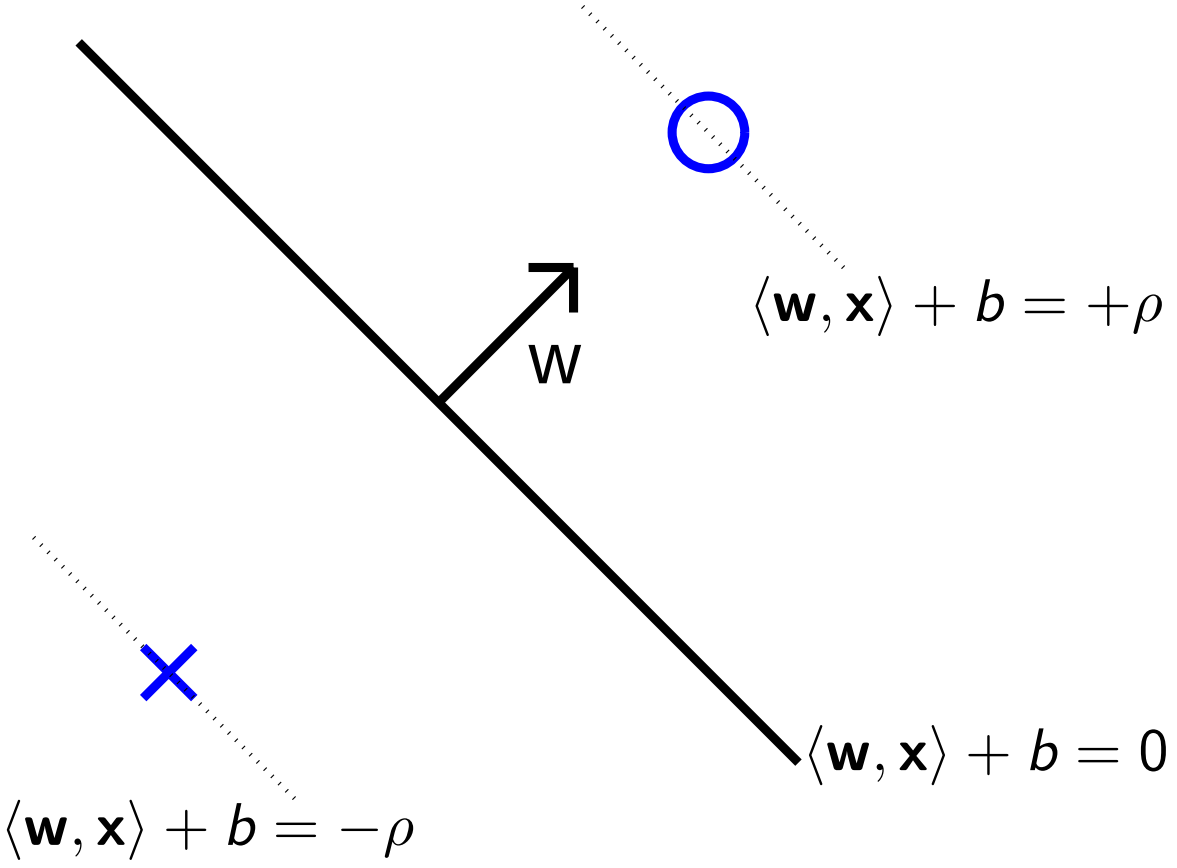
find a linear classification boundary





not robust wrt input noise!

SVM:
maximum margin classifier



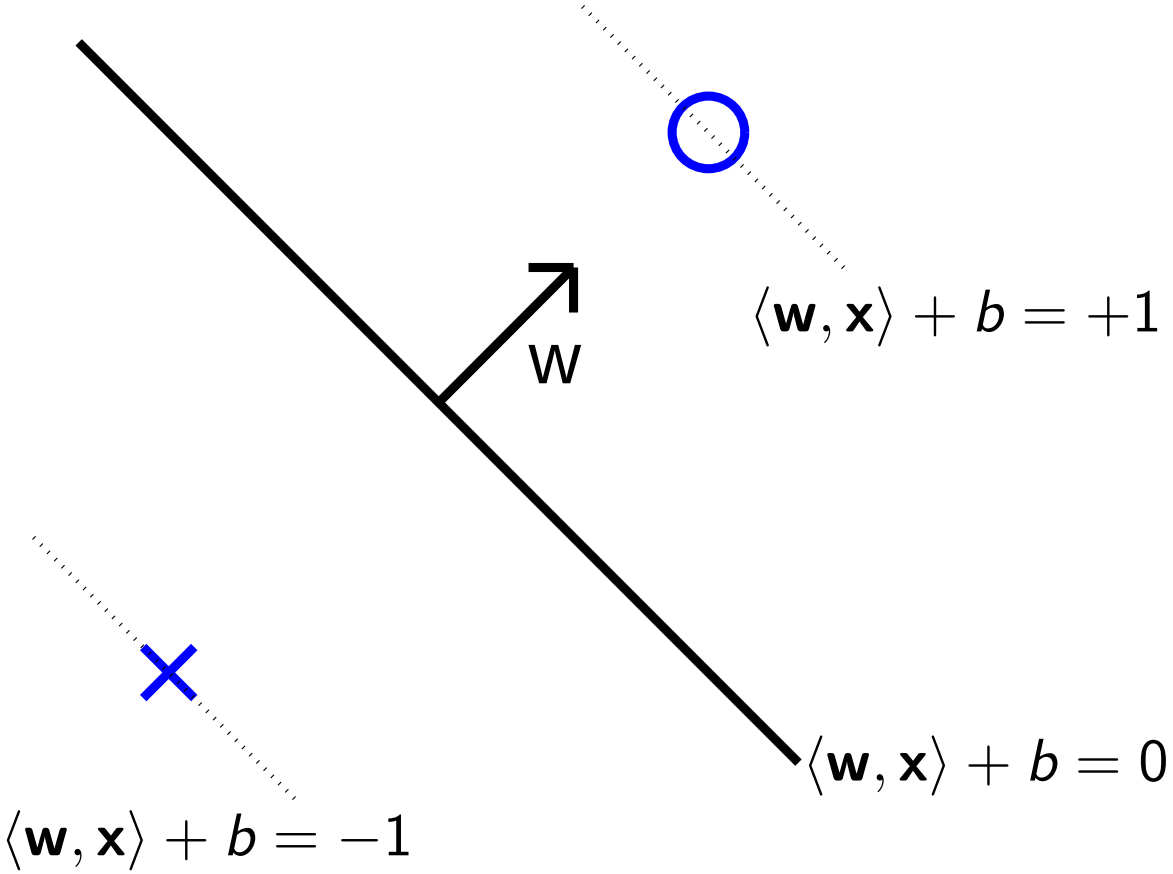
$$\max_{\mathbf{w},b,\rho} \underbrace{\rho}_{\text{margin}} \quad s.t. \quad \underbrace{y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq \rho}_{\text{data fitting}}, \quad \underbrace{\|\mathbf{w}\| = 1}_{\text{normalization}}$$

Equivalent reformulation of the SVM

$$
\begin{aligned}
& \max_{\mathbf{w},b,\rho} \quad \rho && s.t. \quad y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq \rho, \quad \|\mathbf{w}\| = 1 \\
\Leftrightarrow \quad & \max_{\mathbf{w}',b,\rho} \quad \rho^2 && s.t. \quad y_i \left(\left\langle \frac{\mathbf{w}'}{\|\mathbf{w}'\|}, \mathbf{x}_i \right\rangle + b \right) \geq \rho, \quad \rho \geq 0 \\
\Leftrightarrow \quad & \max_{\mathbf{w}',b,\rho} \quad \rho^2 && s.t. \quad y_i \left(\left\langle \underbrace{\frac{\mathbf{w}'}{\|\mathbf{w}'\| \rho}}_{\mathbf{w}''}, \mathbf{x}_i \right\rangle + \underbrace{\frac{b}{\rho}}_{b''} \right) \geq 1, \quad \rho \geq 0 \\
\Leftrightarrow \quad & \max_{\mathbf{w}'',b''} \quad \frac{1}{\|\mathbf{w}''\|^2} && s.t. \quad y_i \left(\langle \mathbf{w}'', \mathbf{x}_i \rangle + b'' \right) \geq 1,
\end{aligned}
$$

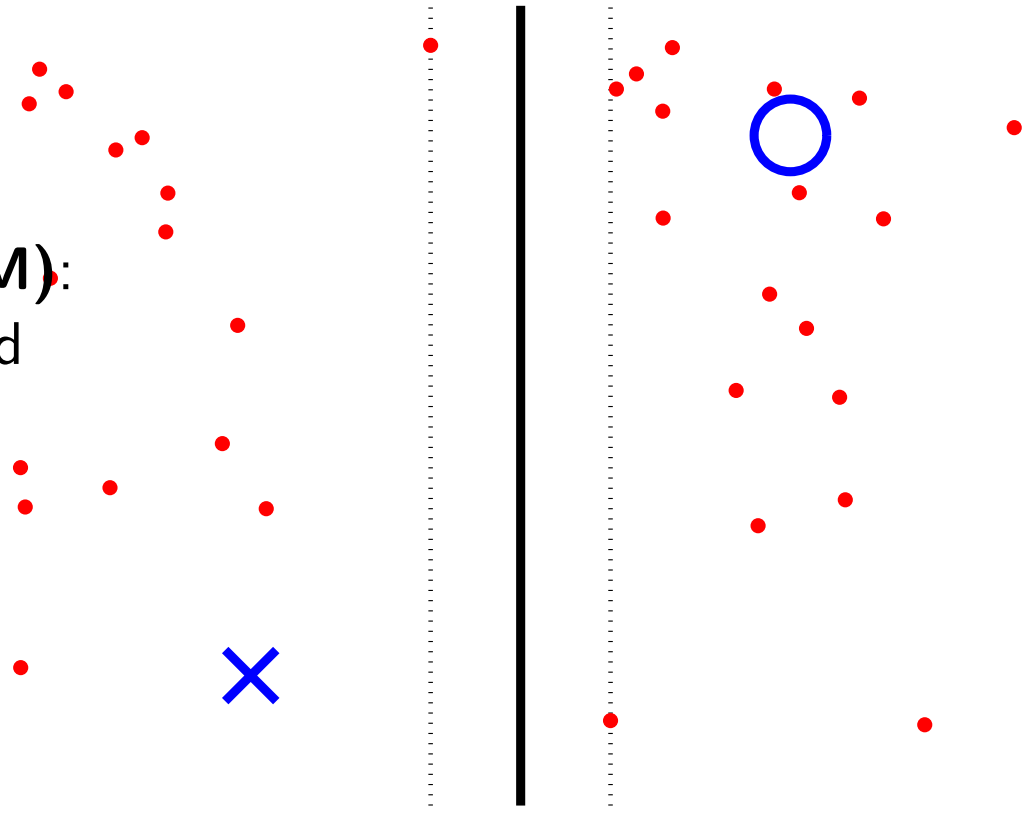
using $\|\mathbf{w}''\| = \left\| \frac{\mathbf{w}'}{\|\mathbf{w}'\| \rho} \right\| = \left| \frac{1}{\rho} \right| \cdot \left\| \frac{\mathbf{w}'}{\|\mathbf{w}'\|} \right\| = \frac{1}{\rho}$

SVM:
maximum margin classifier



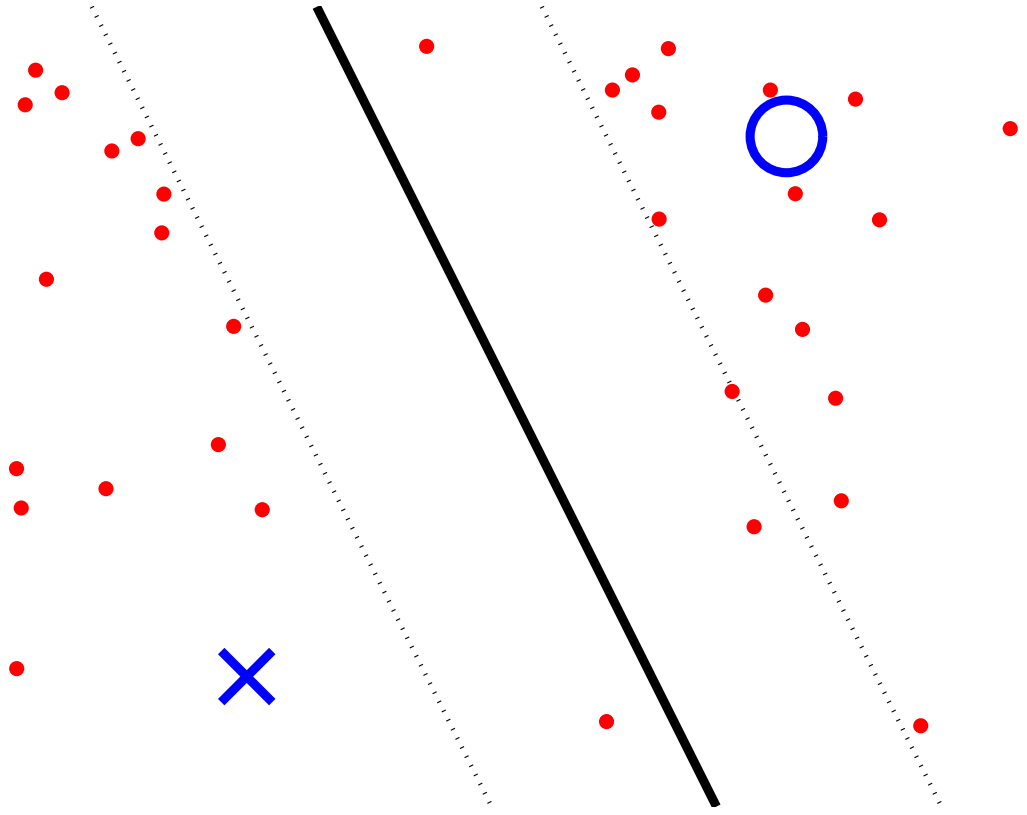
$$\min_{\mathbf{w},b} \quad \underbrace{\frac{1}{2} \langle \mathbf{w}, \mathbf{w} \rangle}_{\text{regularizer}} \quad s.t. \quad \underbrace{y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1}_{\text{data fitting}}$$

S^{3}VM (TSVM):
semi-supervised
(transductive)
SVM



$$\min_{\mathbf{w},b,(y_j)} \underbrace{\frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle}_{\text{regularizer}} \quad s.t. \quad \begin{aligned} y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) &\geq 1 \\ y_j(\langle \mathbf{w}, \mathbf{x}_j \rangle + b) &\geq 1 \end{aligned}$$

soft margin S^{3}VM



$$\min_{\mathbf{w},b,(y_j),(\xi_k)} \quad \begin{array}{l} \frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle \\ +C \sum_i \xi_i \\ +C^* \sum_j \xi_j \end{array} \quad s.t. \quad \begin{array}{c} \xi_i \geq 0 \quad \xi_j \geq 0 \\ y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i \\ y_j(\langle \mathbf{w}, \mathbf{x}_j \rangle + b) \geq 1 - \xi_j \end{array}$$

Supervised Support Vector Machine (SVM)

$$\min_{\mathbf{w},b,(\xi_k)} \quad \begin{array}{c} \frac{1}{2}\langle \mathbf{w},\mathbf{w}\rangle \\ +C\sum_i \xi_i \end{array} \quad s.t. \quad \begin{array}{c} \xi_i \geq 0 \\ y_i(\langle \mathbf{w},\mathbf{x}_i\rangle + b) \geq 1-\xi_i \end{array}$$

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

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

$$\min_{\mathbf{w},b,(y_j),(\xi_k)} \quad \begin{array}{c} \frac{1}{2}\langle \mathbf{w},\mathbf{w}\rangle \\ +C\sum_i \xi_i \\ +C^*\sum_j \xi_j \end{array} \quad s.t. \quad \begin{array}{c} \xi_i \geq 0 \quad \xi_j \geq 0 \\ y_i(\langle \mathbf{w},\mathbf{x}_i\rangle + b) \geq 1-\xi_i \\ y_j(\langle \mathbf{w},\mathbf{x}_j\rangle + b) \geq 1-\xi_j \end{array}$$

- maximize margin on labeled and unlabeled points
- combinatorial optimization problem (optimize $y_j \in \{0,1\}$)

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

$$s.t. \quad y_i(\langle \mathbf{w},\mathbf{x}_i\rangle + b) \geq 1-\xi_i \quad \xi_i \geq 0$$
$$y_j(\langle \mathbf{w},\mathbf{x}_j\rangle + b) \geq 1-\xi_j \quad \xi_j \geq 0$$

Mixed Integer Programming [Bennett, Demiriz; NIPS 1998]

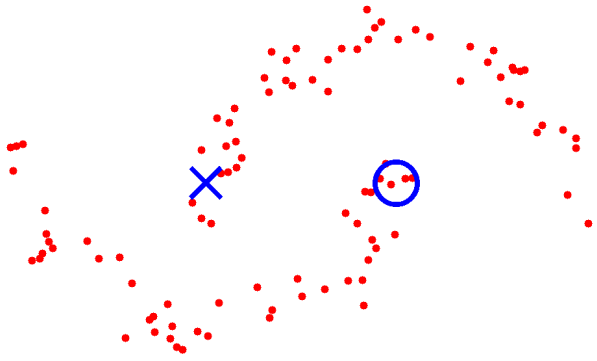
- global optimum found by standard optimization packages (eg CPLEX)
- **NP-hard !** $\Rightarrow$ only works for small sized problems

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!



S^{3}VM optimization method		test error	objective value
global min.	{Branch & Bound	0.0%	7.81
find local minima	CCCP	64.0%	39.55
	S^{3}VM^{light}	66.2%	20.94
	∇S^{3}VM	59.3%	13.64
	cS^{3}VM	45.7%	13.25

- objective function is good for SSL
- $\Rightarrow$ **try to find better local minima!**

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

$$s.t. \quad \begin{aligned} y_i(\langle \mathbf{w},\mathbf{x}_i\rangle + b) &\geq 1-\xi_i \quad \xi_i \geq 0 \\ y_j(\langle \mathbf{w},\mathbf{x}_j\rangle + b) &\geq 1-\xi_j \quad \xi_j \geq 0 \end{aligned}$$

S^3VM^{light} [T. Joachims; ICML 1999]

- train SVM on labeled points, predict y_j's
- in prediction, always make sure that

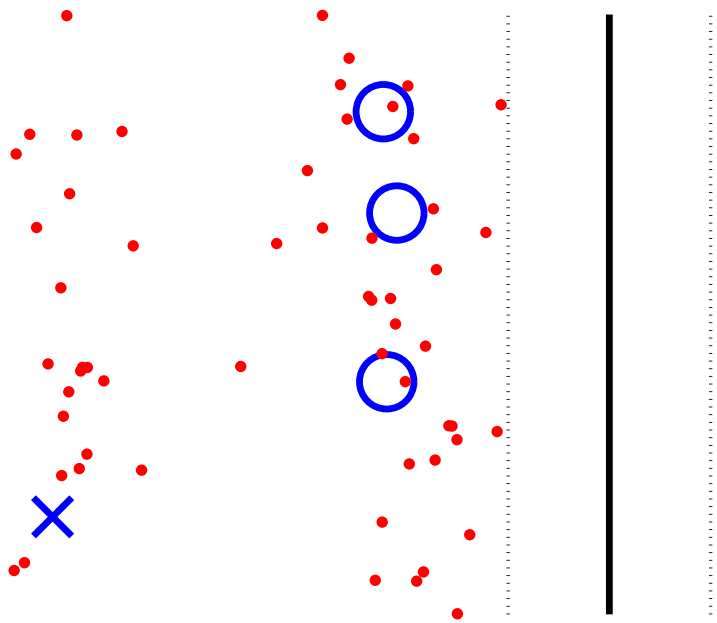
$$\frac{\#\{y_j = +1\}}{\#\text{ unlabeled points}} = \frac{\#\{y_i = +1\}}{\#\text{ labeled points}} \qquad (*)$$

- 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" (*)

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

$$s.t. \quad y_i(\langle \mathbf{w},\mathbf{x}_i\rangle + b) \geq 1-\xi_i \quad \xi_i \geq 0$$
$$y_j(\langle \mathbf{w},\mathbf{x}_j\rangle + b) \geq 1-\xi_j \quad \xi_j \geq 0$$

Balancing constraint required to avoid **degenerate solutions**!



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

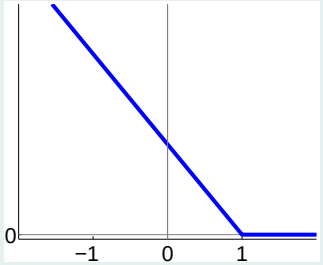
$$s.t. \quad y_i(\langle \mathbf{w},\mathbf{x}_i\rangle + b) \geq 1-\xi_i \quad \xi_i \geq 0$$
$$y_j(\langle \mathbf{w},\mathbf{x}_j\rangle + b) \geq 1-\xi_j \quad \xi_j \geq 0$$

Effective Loss Functions

$$\xi_i = \min\{1 - y_i(\langle \mathbf{w},\mathbf{x}_i\rangle + b), 0\}$$
$$\xi_j = \min_{y_j\in\{+1,-1\}} \{1 - y_j(\langle \mathbf{w},\mathbf{x}_j\rangle + b), 0\}$$

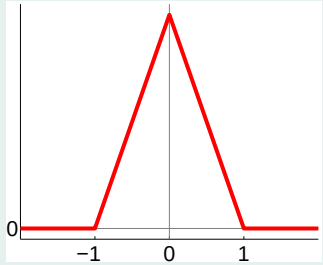
loss functions

ξ_i



$y_i(\langle \mathbf{w},\mathbf{x}_i\rangle + b)$

ξ_j



$\langle \mathbf{w},\mathbf{x}_j\rangle + b$

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

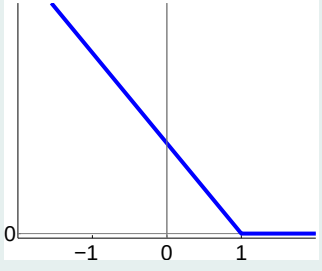
$$s.t. \quad y_i(\langle \mathbf{w},\mathbf{x}_i\rangle + b) \geq 1-\xi_i \quad \xi_i \geq 0$$
$$y_j(\langle \mathbf{w},\mathbf{x}_j\rangle + b) \geq 1-\xi_j \quad \xi_j \geq 0$$

Resolving the Constraints

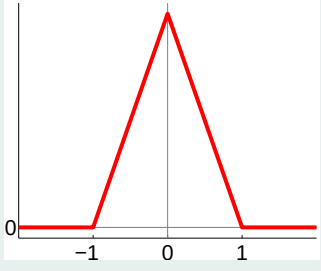
$$\frac{1}{2}\langle \mathbf{w},\mathbf{w}\rangle + C\sum_i \ell_l\left(y_i(\langle \mathbf{w},\mathbf{x}_i\rangle + b)\right) + C^*\sum_j \ell_u\left(\langle \mathbf{w},\mathbf{x}_j\rangle + b\right)$$

loss functions

ℓ_l



ℓ_u



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

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]

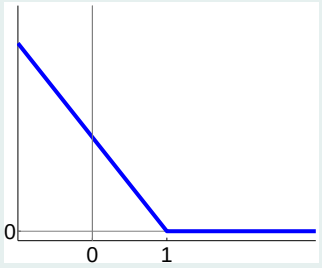
- similar approach
- restricted to linear S^{3}VMs

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

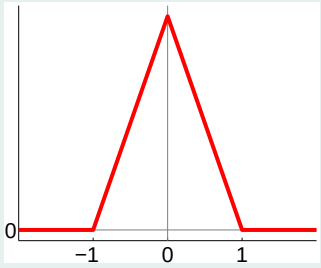
S^{3}VM as Unconstrained Differentiable Optimization Problem

original loss functions

ℓ_l

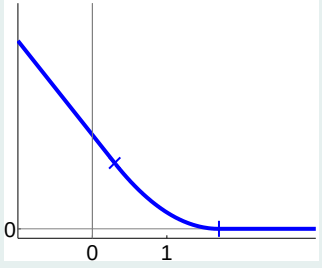


ℓ_u

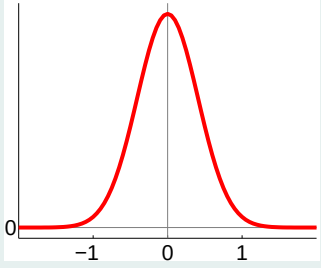


smooth loss functions

ℓ_l



ℓ_u



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

∇S^{3}VM [Chapelle, Zien; AISTATS 2005]

- simply do gradient descent!
- thereby stepwise increase C^*

contS^{3}VM [Chapelle et al.; ICML 2006]

... in more detail on next slides!

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

Hard Balancing Constraint

S^{3}VM^{light} constraint

$$\frac{\#\{y_j = +1\}}{\#\text{ unlabeled points}} = \frac{\#\{y_i = +1\}}{\#\text{ labeled points}}$$

equivalent constraint

$$\underbrace{\frac{1}{m}\sum_{j=1}^{m} sign\left(\langle \mathbf{w}, \mathbf{x}_j \rangle + b\right)}_{\text{average prediction}} = \underbrace{\frac{1}{n}\sum_{i=1}^{n} y_i}_{\text{average label}}$$

Making the Balancing Constraint Linear

hard / non-linear

$$\underbrace{\frac{1}{m}\sum_j sign\left(\langle \mathbf{w}, \mathbf{x}_j \rangle + b\right)}_{\text{average prediction}} = \underbrace{\frac{1}{n}\sum_i y_i}_{\text{average label}}$$

soft / linear

$$\underbrace{\frac{1}{m}\sum_j \langle \mathbf{w}, \mathbf{x}_j \rangle + b}_{\text{mean output on unlabeled points}} = \underbrace{\frac{1}{n}\sum_i y_i}_{\text{average label}}$$

Implementing the linear soft balancing:

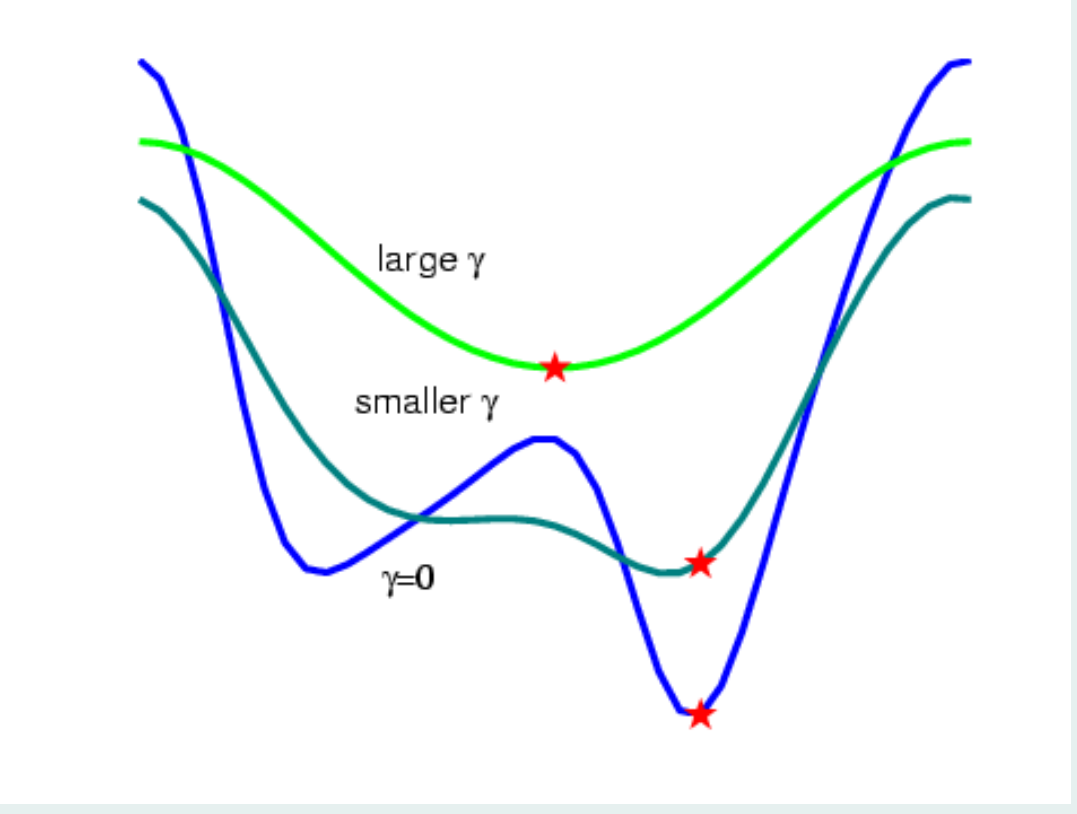
- center the unlabeled data: $\sum_j \mathbf{x}_j = \mathbf{0}$
- $\Rightarrow$ just fix b; unconstrained optimization over $\mathbf{w}$!

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 $\sqrt{\gamma/2}$:

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

Closed form solution!

Smoothing Sequence

choose $\gamma_0 > \gamma_1 > \ldots \gamma_{p-1} > \gamma_p = 0$

- choose γ_0 such that $f_{\gamma_0}(\cdot)$ is convex
- choose γ_{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 $\Phi(\mathbf{x})$, kernel $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$.

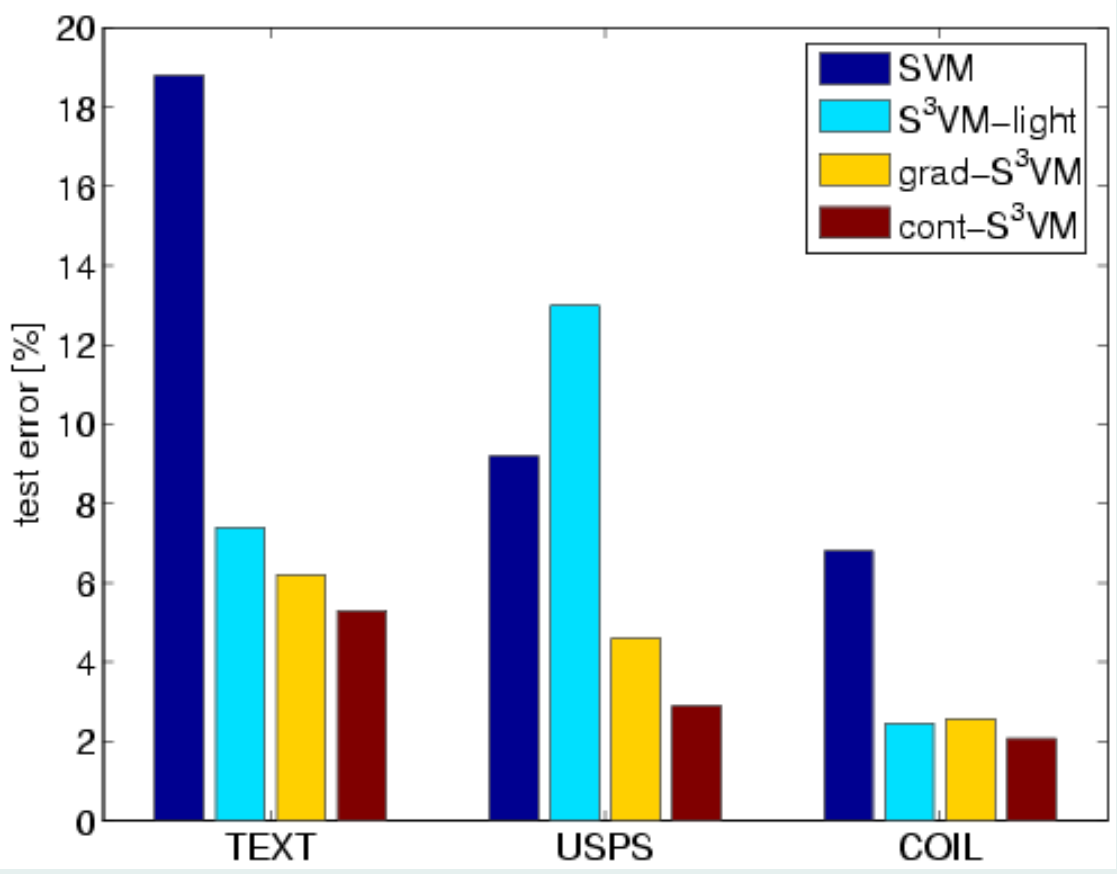
Representer Theorem: S^{3}VM solution is in span E of data points

$$E := span\{\Phi(\mathbf{x}_i)\} \stackrel{\triangle}{=} \mathbb{R}^{n+m}$$

Implementation

1. expand basis vectors $\mathbf{v}_i$ of E: $\mathbf{v}_i = \sum_k A_{ik} \Phi(\mathbf{x}_k)$
2. orthonormality gives: $(A^\top A)^{-1} = K$
   solve for A, eg by KPCA or Choleski decomposition
3. project data $\Phi(\mathbf{x}_i)$ on basis $V = (\mathbf{v}_j)_j$: $\tilde{\mathbf{x}}_i = V^\top \Phi(\mathbf{x}_i) = (A)_i$

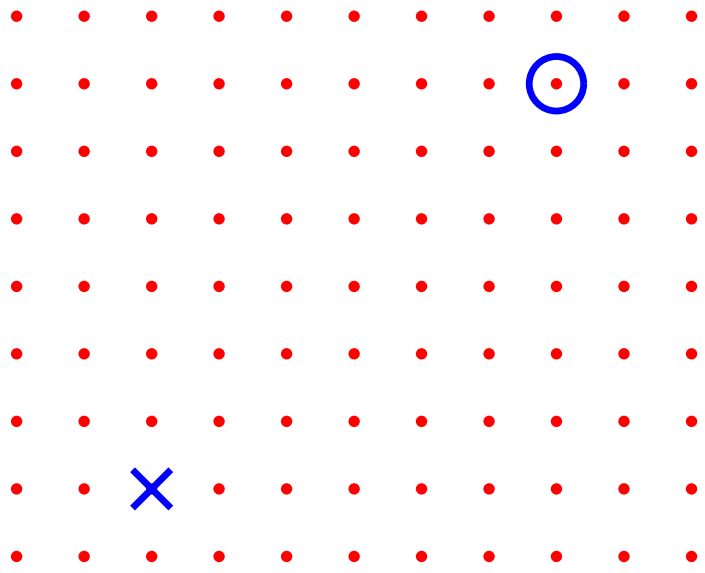
Comparison of S^3VM Optimization Methods



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

[Chapelle, Chi, Zien; ICML 2006]

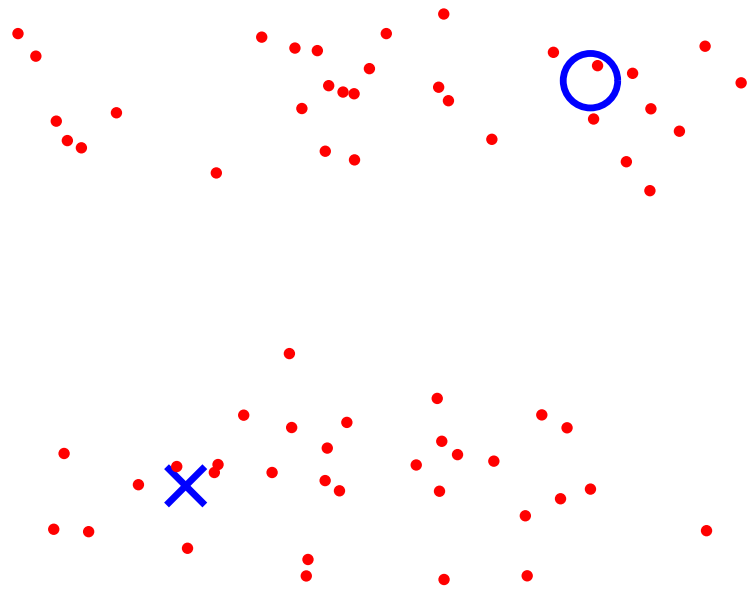
Why would unlabeled data be useful at all?



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^3VM**

Manifold Assumption

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

[images from "The Geometric Basis of Semi-Supervised Learning", Sindhwani, Belkin, Niyogi in "Semi-Supervised Learning" Chapelle, Schölkopf, Zien]

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

Graph Construction

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

approximate manifold / achieve sparsity – two choices:

1. k nearest neighbor graph (usually prefered)
2. ϵ distance graph

Learning on the Graph

estimation of a function on the edges, ie $f : E \to \{-1, +1\}$
[recall: for SVMs, $f : \mathcal{X} \to \{-1, +1\}$, $\mathbf{x} \mapsto sign(\langle \mathbf{w}, \mathbf{x} \rangle + b)$]

Regularization on a Graph – penalize change along edges

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

$$\begin{aligned} g(\mathbf{y}) &= \frac{1}{2} \left(\sum_k \sum_l a_{kl} y_k^2 + \sum_k \sum_l a_{kl} y_l^2 \right) - \sum_k \sum_l a_{kl} y_k y_l \\ &= \sum_k y_k^2 \sum_l a_{kl} - \sum_k \sum_l y_k a_{kl} y_l \\ &= \mathbf{y}^\top \mathbf{D} \mathbf{y} - \mathbf{y}^\top \mathbf{A} \mathbf{y} \quad = \quad \mathbf{y}^\top \mathbf{L} \mathbf{y} \end{aligned}$$

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

with constraints $y_j \in \{-1, +1\}$ essentially yields min-cut problem

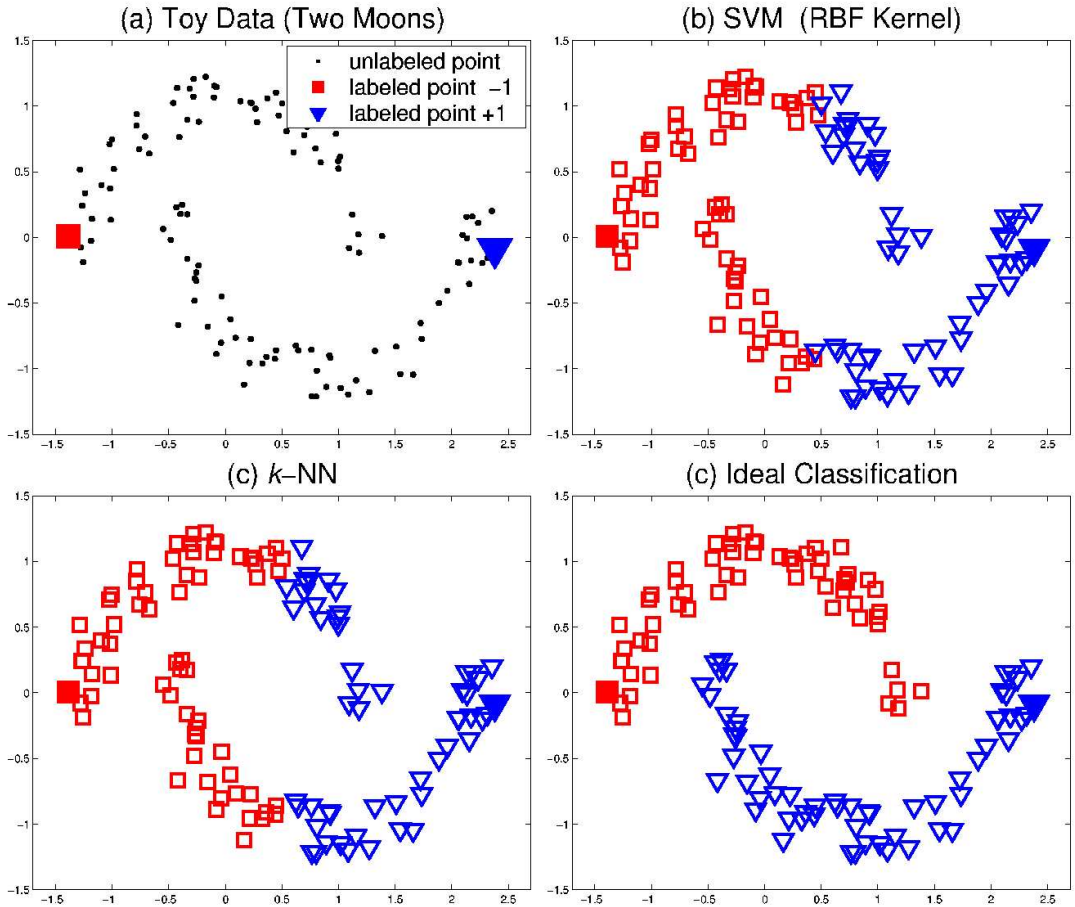
Label Propagation

relax: instead of $y_j \in \{-1, +1\}$, optimize free f_j
$\Rightarrow$ fix $\mathbf{f}_l = (f_i) = (y_i)$, solve for $\mathbf{f}_u = (f_j)$, predict $y_j = sign(f_j)$
$\Rightarrow$ convex QP ($\mathbf{L}$ is positive definite)

$$\begin{aligned} 0 &= \frac{\partial}{\partial \mathbf{f}_u} \begin{pmatrix} \mathbf{f}_l \\ \mathbf{f}_u \end{pmatrix}^\top \begin{pmatrix} \mathbf{L}_{ll} \mathbf{L}_{ul}^\top \\ \mathbf{L}_{ul} \mathbf{L}_{uu} \end{pmatrix} \begin{pmatrix} \mathbf{f}_l \\ \mathbf{f}_u \end{pmatrix} \\ &= \frac{\partial}{\partial \mathbf{f}_u} \left(\mathbf{f}_u{}^\top \mathbf{L}_{ul} \mathbf{f}_l + \mathbf{f}_l{}^\top \mathbf{L}_{ul}^\top \mathbf{f}_u + \mathbf{f}_u{}^\top \mathbf{L}_{uu}^\top \mathbf{f}_u \right) \\ &= 2\mathbf{f}_l{}^\top \mathbf{L}_{ul}^\top + 2\mathbf{f}_u{}^\top \mathbf{L}_{uu}^\top \end{aligned}$$

- $\Rightarrow$ solve linear system $\quad \mathbf{L}_{uu}\mathbf{f}_u = -\mathbf{L}_{lu}^\top \mathbf{f}_l \quad (\mathbf{f}_u = -\mathbf{L}_{uu}^{-1}\mathbf{L}_{lu}^\top \mathbf{f}_l)$
- 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



Note: here color $\hat{=}$ classes

[images from "Learning with Local and Global Consistency", Zhou, Bousquet, Lal, Weston, Schölkopf; NIPS 2004]

"Beyond the Point Cloud"

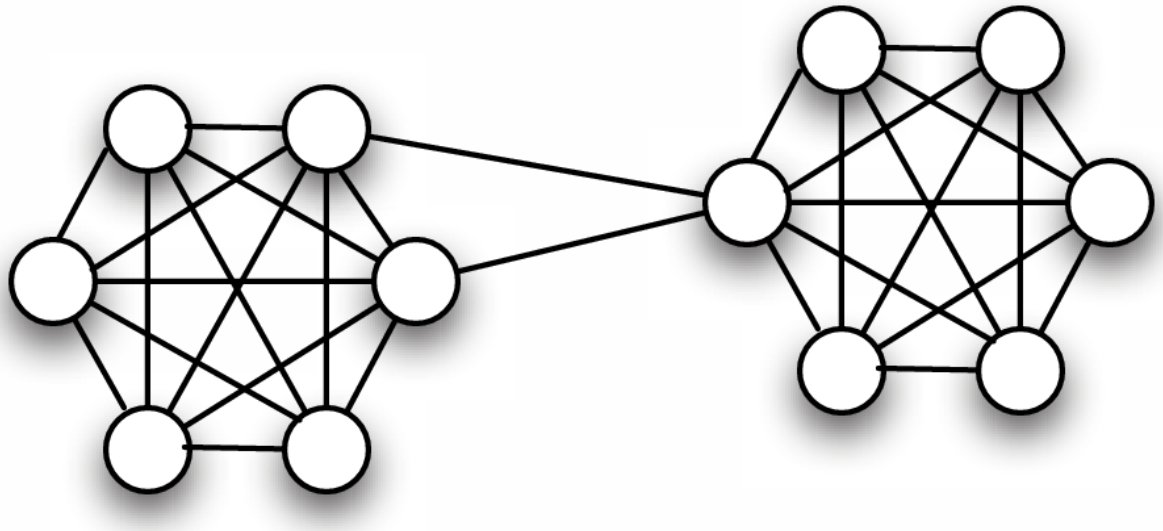
Idea:

- model output f_j as linear function of the node value $\mathbf{x}_j$
  $f_k = \mathbf{w}^\top \mathbf{x}_k$ (with kernels: $f_k = \sum_l \alpha_l k(\mathbf{x}_l, \mathbf{x}_k)$)
- add graph regularizer to SVM cost function
  $R_g(\mathbf{w}) = \frac{1}{2} \sum_k \sum_l a_{kl}(f_k - f_l)^2 = \mathbf{f}^\top \mathbf{L}\mathbf{f}$

$$\min_{\mathbf{w}} \quad \underbrace{\sum_i \ell(y_i(\mathbf{w}^\top \mathbf{x}_i))}_{\text{data fitting}} + \underbrace{\lambda \|\mathbf{w}\|^2 + \gamma R_g(\mathbf{w})}_{\text{regularizers}}$$

- linear ($\mathbf{f} = \mathbf{X}\mathbf{w}$): $\Rightarrow \quad \lambda \mathbf{w}^\top \mathbf{w} + \gamma \mathbf{w}^\top \mathbf{X}^\top \mathbf{L}\mathbf{X}\mathbf{w}$
- w. kernel ($\mathbf{f} = \mathbf{K}\alpha$): $\Rightarrow \quad \lambda \alpha^\top \mathbf{K}\alpha + \gamma \alpha^\top \mathbf{K}\mathbf{L}\mathbf{K}\alpha$

Graph Methods



Observation

graphs model **density** on manifold

$\Rightarrow$ 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**.

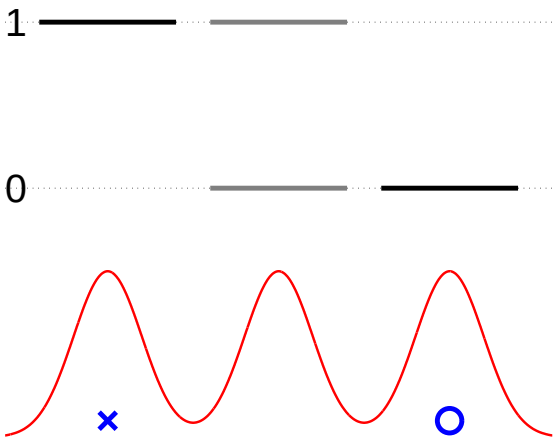
Manifold Assumption

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

Semi-Supervised Smoothness Assumption

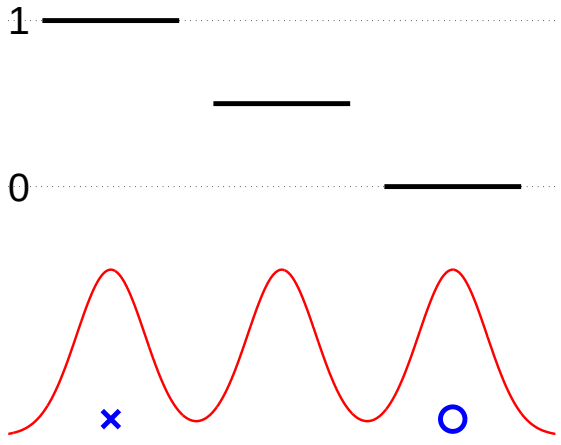
1. The density is non-uniform.
2. If two points are close in a high density region ($\Rightarrow$ connected by high density path), their outputs are close.

S^3VMs



- Cluster Assumption
- points within same cluster are of **same class**
- non-convex

Graph methods

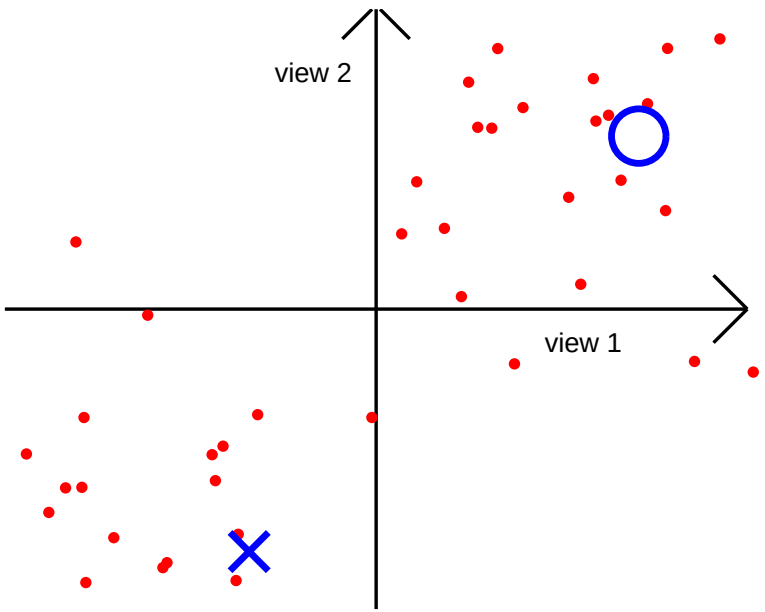


- Semi-Supervised Smoothness
- points within same cluster have **same class probabilities**
- convex

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

$$\min_{\mathbf{w}^v,(y_j),\xi_k} \quad \frac{1}{2}\sum_v \|\mathbf{w}_v\|^2 + C\sum_i \xi_i + C^*\sum_j \xi_j$$

$$s.t. \quad \begin{aligned} \forall_v : y_i\left(\langle \mathbf{w}_v, \Phi_v(\mathbf{x}_i)\rangle + b\right) &\geq 1-\xi_i, \quad \xi_i \geq 0 \\ \forall_v : y_j\left(\langle \mathbf{w}_v, \Phi_v(\mathbf{x}_j)\rangle + b\right) &\geq 1-\xi_j, \quad \xi_j \geq 0 \end{aligned}$$

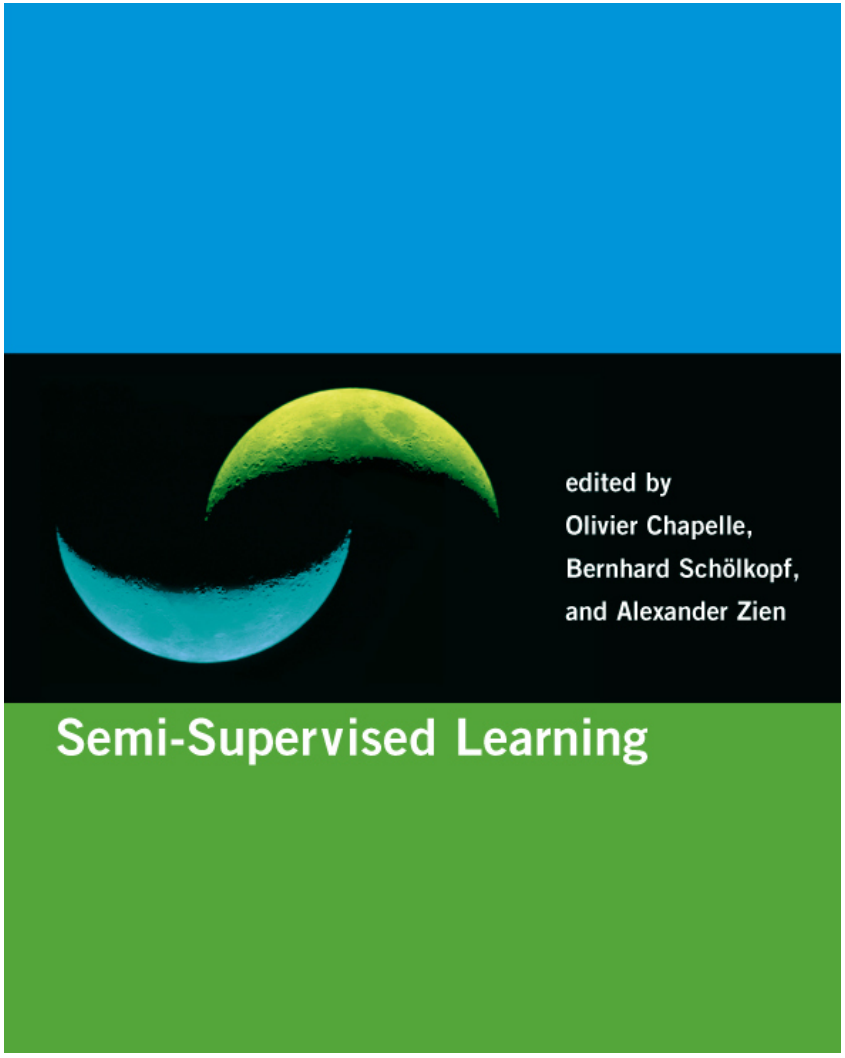
Assumption	**Approach**	**Example Algorithm**
Cluster Assumption	Low Density Separation	S^{3}VM; Entropy Regularization; Data-Dependent Regularization; ...
Manifold Assumption	Graph-based Methods	• build weighted graph (w_{kl}) • $\min_{(y_j)} \sum_k \sum_l w_{kl}(y_k - y_l)^2$
Independent Views	Co-Training	• 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** $p(y|\mathbf{x})$ (or just boundary: $\left\{\mathbf{x} \,\middle|\, p(y|\mathbf{x}) = \frac{1}{2}\right\}$)
- examples: S^3VM, graph-based methods

Generative Learning (Sampling Paradigm)

- **model** $p(\mathbf{x}|y)$
- predict via Bayes: $p(y|\mathbf{x}) = \frac{p(y)p(\mathbf{x}|y)}{\sum_{y'} p(y')p(\mathbf{x}|y')}$
- $\Rightarrow$ 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/`

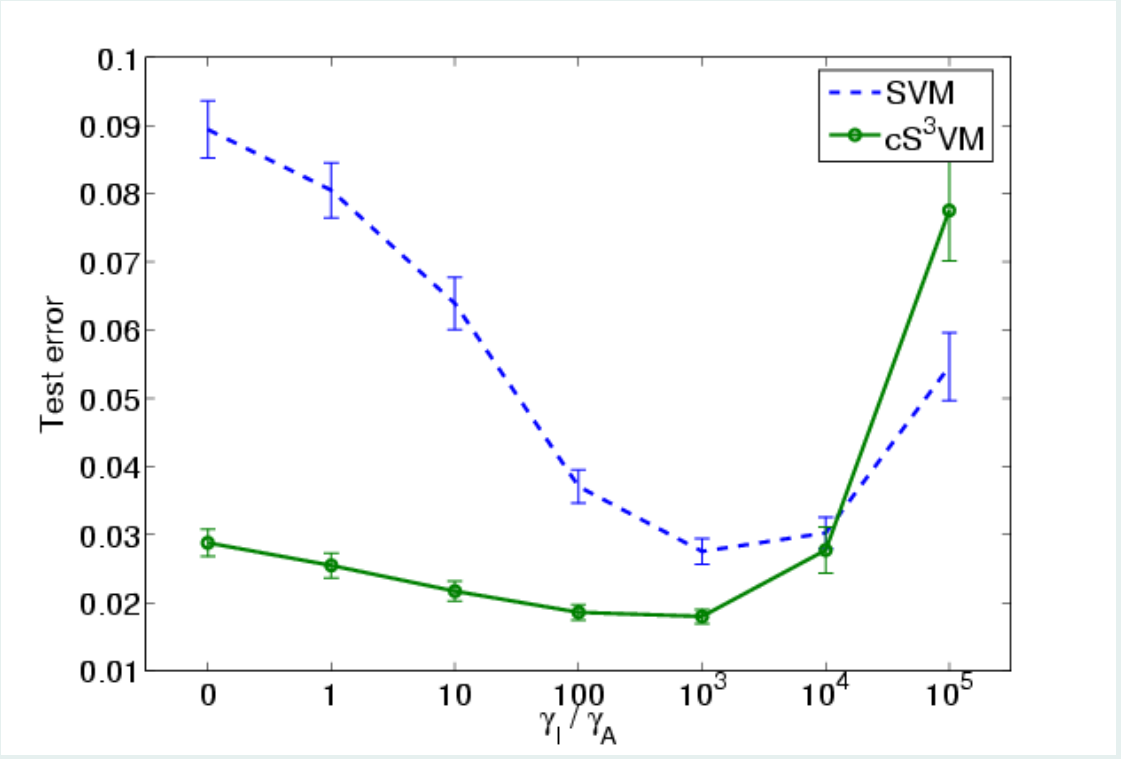
SSL Book – Benchmark

	g241c	g241d	Digit1	USPS	COIL	BCI	Text
1-NN	43.93	42.45	3.89	5.81	17.35	48.67	30.11
SVM	23.11	24.64	5.53	9.75	22.93	34.31	26.45
MVU + 1-NN	43.01	38.20	2.83	6.50	28.71	47.89	32.83
LEM + 1-NN	40.28	37.49	6.12	7.64	23.27	44.83	30.77
Label-Prop.	22.05	28.20	3.15	6.36	10.03	46.22	25.71
Discrete Reg.	43.65	41.65	2.77	4.68	9.61	47.67	24.00
S^{3}SVM	18.46	22.42	6.15	9.77	25.80	33.25	24.52
SGT	17.41	9.11	2.61	6.80	–	45.03	23.09
Cluster-Kernel	13.49	4.95	3.79	9.68	21.99	35.17	24.38
Data-Dep. Reg.	20.31	32.82	2.44	5.10	11.46	47.47	–
LDS	18.04	23.74	3.46	4.96	13.72	43.97	23.15
Graph-Reg.	24.36	26.46	2.92	4.68	*11.92*	31.36	23.57
CHM (normed)	24.82	25.67	3.79	7.65	–	36.03	–

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^3VM, co-training)
    non-convex – optimization method matters!
  - in the regularizer (graph methods)
    convex, but graph construction matters
- combination seems to work best

Thank you!