Semi-Supervised Support Vector Machines and Application to Spam Filtering

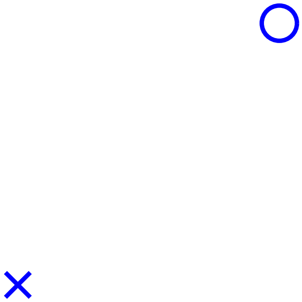
Alexander Zien

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

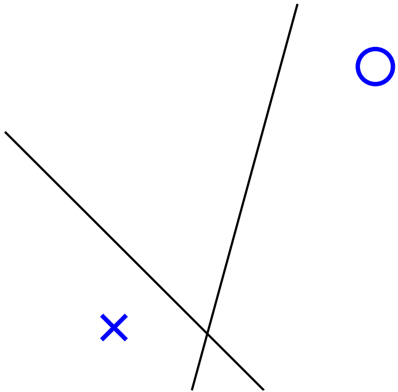
ECML 2006 – Discovery Challenge

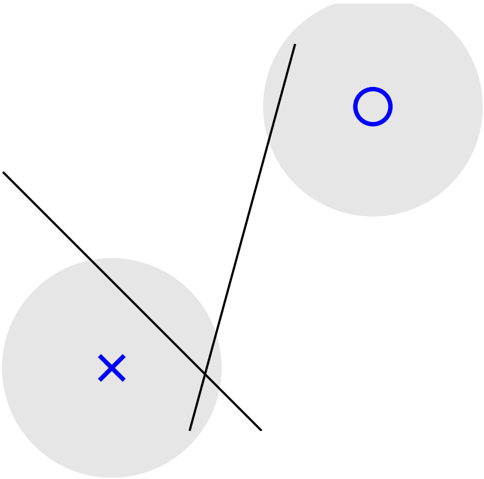
MAX-PLANCK-GESELLSCHAFT

BIOLOGISCHE KYBERNETIK



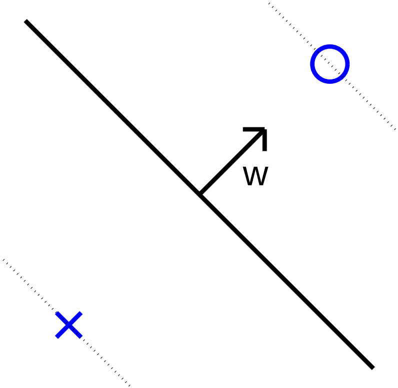
find a linear classification boundary





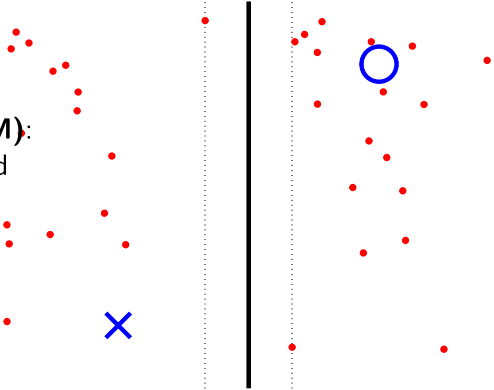
not robust wrt input noise!

SVM:
maximum margin classifier



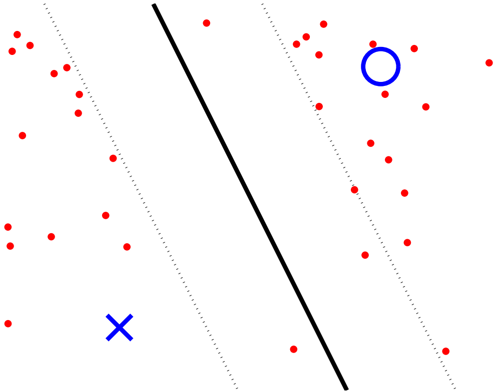
$$\min_{\mathbf{w},b} \quad \underbrace{\frac{1}{2}\mathbf{w}^\top\mathbf{w}}_{\text{regularizer}} \qquad s.t. \quad y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1$$

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



$$\min_{\mathbf{w},b,(y_j)} \quad \underbrace{\frac{1}{2}\mathbf{w}^\top\mathbf{w}}_{\text{regularizer}} \quad s.t. \quad \begin{array}{l} y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1 \\ y_j(\mathbf{w}^\top\mathbf{x}_j + b) \geq 1 \end{array}$$

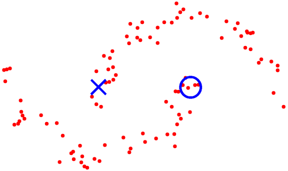
soft margin
S^{3}VM



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

“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!**

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

$$s.t. \quad \begin{array}{ll} y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1 - \xi_i & \xi_i \geq 0 \\ y_j(\mathbf{w}^\top\mathbf{x}_j + b) \geq 1 - \xi_j & \xi_j \geq 0 \end{array}$$

Mixed Integer Programming [Bennett, Demiriz; NIPS 1998]

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

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

$$s.t. \quad \begin{array}{ll} y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i & \xi_i \geq 0 \\ y_j(\mathbf{w}^\top \mathbf{x}_j + b) \geq 1 - \xi_j & \xi_j \geq 0 \end{array}$$

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

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

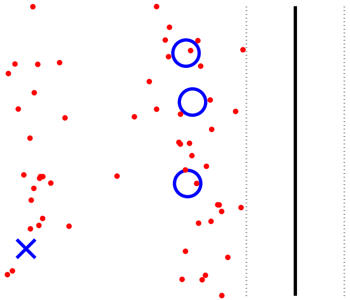
$$\frac{\#\{y_j = +1\}}{\#\text{ unlabeled points}} = \frac{\#\{y_i = +1\}}{\#\text{ labeled points}} \tag{1}$$

- 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" (*)

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

$$s.t. \quad \begin{array}{ll} y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1 - \xi_i & \xi_i \geq 0 \\ y_j(\mathbf{w}^\top\mathbf{x}_j + b) \geq 1 - \xi_j & \xi_j \geq 0 \end{array}$$

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



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

$$s.t. \quad \begin{array}{ll} {\color{blue}y_i}(\mathbf{w}^\top {\color{blue}\mathbf{x}_i} + b) \geq 1 - \xi_i & \xi_i \geq 0 \\ {\color{red}y_j}(\mathbf{w}^\top {\color{red}\mathbf{x}_j} + b) \geq 1 - \xi_j & \xi_j \geq 0 \end{array}$$

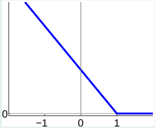
Effective Loss Functions

$$\xi_i = \min\left\{1 - {\color{blue}y_i}(\mathbf{w}^\top {\color{blue}\mathbf{x}_i} + b), 0\right\}$$

$$\xi_j = \min_{{\color{red}y_j}\in\{+1,-1\}}\left\{1 - {\color{red}y_j}(\mathbf{w}^\top {\color{red}\mathbf{x}_j} + b), 0\right\}$$

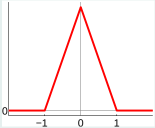
loss functions

ξ_i



${\color{blue}y_i}(\mathbf{w}^\top {\color{blue}\mathbf{x}_i} + b)$

ξ_j



$\mathbf{w}^\top {\color{red}\mathbf{x}_j} + b$

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

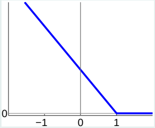
$$s.t. \quad \begin{array}{ll} y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1 - \xi_i & \xi_i \geq 0 \\ y_j(\mathbf{w}^\top\mathbf{x}_j + b) \geq 1 - \xi_j & \xi_j \geq 0 \end{array}$$

Resolving the Constraints

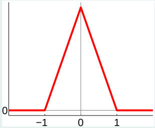
$$\frac{1}{2}\mathbf{w}^\top\mathbf{w} + C\sum_i \ell_l\left(y_i(\mathbf{w}^\top\mathbf{x}_i + b)\right) + C^*\sum_j \ell_u\left(\mathbf{w}^\top\mathbf{x}_j + b\right)$$

loss functions

ℓ_l



ℓ_u



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

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]

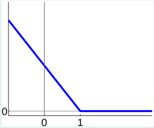
- similar approach
- restricted to linear S^{3}VMs

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

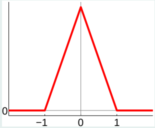
S^3VM as Unconstrained Differentiable Optimization Problem

original loss functions

ℓ_l

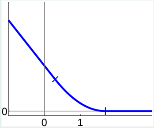


ℓ_u

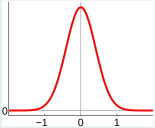


smooth loss functions

ℓ_l



ℓ_u



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

∇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!

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

Hard Balancing Constraint

S^3VM^{light} constraint

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

equivalent constraint

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

Making the Balancing Constraint Linear

hard / non-linear

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

soft / linear

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

Implementing the linear soft balancing:

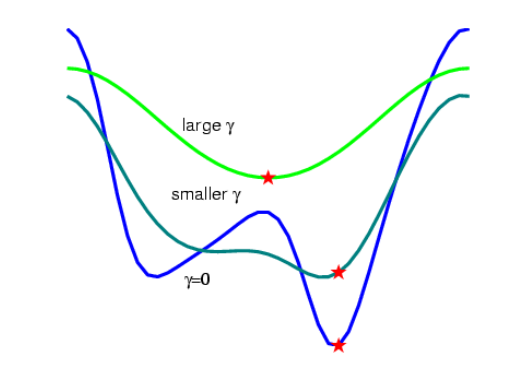
- center the unlabeled data: $\sum_j \mathbf{x}_j = \mathbf{0}$
- $\Rightarrow$ just fix b; unconstrained optimization over $\mathbf{w}$!

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

- 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 $k(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^\top \Phi(\mathbf{x}_j)$.

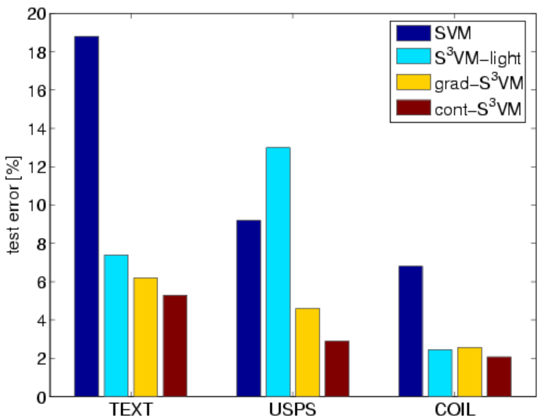
Representer Theorem: S^{3}VM solution is in span E of data points

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

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
3. project data $\Phi(\mathbf{x}_i)$ on basis $V = (\mathbf{v}_j)_j$: $\tilde{\mathbf{x}}_i = V^\top \Phi(\mathbf{x}_i) = (A)_i$

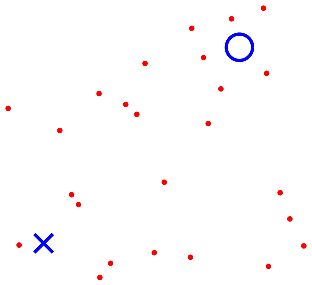
Comparison of S^{3}VM Optimization Methods



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

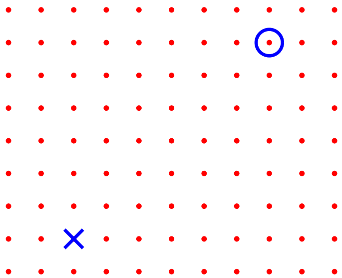
[Chapelle, Chi, Zien; ICML 2006]

Why would unlabeled data be useful at all?



Uniform data do not help.

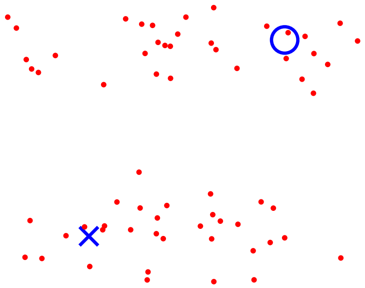
Why would unlabeled data be useful at all?



Uniform data do not help.

Cluster Assumption

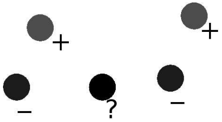
Points in the **same cluster** are likely to be of the **same class**.

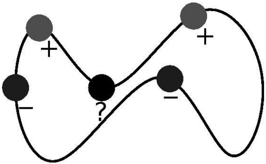


Algorithmic idea: **Low Density Separation**

Manifold Assumption

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





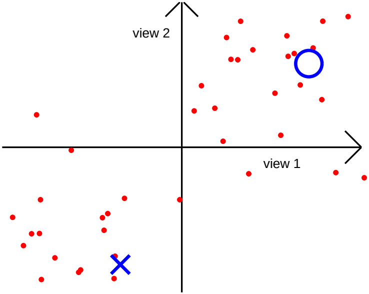
[images from "The Geometric Basis of Semi-Supervised Learning", Sindhwani, Belkin, Niyogi in "Semi-Supervised Learning" Chapelle, Schölkopf, Zien]

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

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**

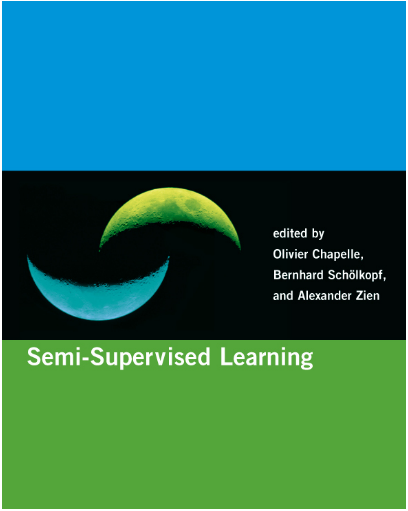
Assumption	Approach	Example Algorithm
Cluster Assumption	Low Density Separation	S^{3}VM; Entropy Regularization; Data-Dependent Regularization; ...
Manifold Assumption	Graph-based Methods	• build weighted graph (w_{kl}) • $\min_{(y_j)} \sum_k \sum_l w_{kl}(y_k - y_l)^2$ • relax y_j to be real $\Rightarrow$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

Generative Learning (Sampling Paradigm)

- **model** $p(\mathbf{x}|y)$
- predict via Bayes: $p(y|\mathbf{x}) = \dfrac{p(y)p(\mathbf{x}|y)}{\sum_{y'} p(y')p(\mathbf{x}|y')}$
- $\Rightarrow$ missing data problem
- EM algorithm (expectation-maximization) is a natural tool
- 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

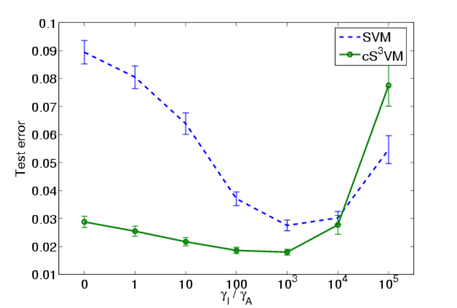
http://www.kyb.tuebingen.mpg.de/ssl-book/

SSL Book – Text Benchmark

	error [%]		AUC [%]	
	l=10	l=100	l=10	l=100
1-NN	38.12	30.11	–	–
SVM	45.37	26.45	67.97	84.26
MVU + 1-NN	45.32	32.83	–	–
LEM + 1-NN	39.44	30.77	–	–
QC + CMN	40.79	25.71	70.71	84.62
Discrete Reg.	40.37	24.00	53.79	71.53
TSVM	31.21	24.52	73.42	80.96
SGT	29.02	23.09	80.09	85.22
Cluster-Kernel	42.72	24.38	73.09	85.90
LDS	27.15	23.15	80.68	84.77
Laplacian RLS	33.68	23.57	76.55	85.05

Combining S^3VM with Graph-based Regularizer

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



[1] "Beyond the Point Clound"; Sindhwani, Niyogi, Belkin; ICML 2005
[2] "A Continuation Method for S^3VM"; Chapelle, Chi, Zien; ICML 2006

Combining S^3VM with Co-Training

"SSL for Structured Output Variables"; Brefeld, Scheffer; ICML 2006

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

$$s.t. \quad \begin{array}{ll} y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1 - \xi_i & \xi_i \geq 0 \\ y_j(\mathbf{w}^\top\mathbf{x}_j + b) \geq 1 - \xi_j & \xi_j \geq 0 \end{array}$$

How to set C ?

- data fitting, $y_i\mathbf{w}^\top\mathbf{x}_i \geq 1$, and regularization, $\min ||\mathbf{w}||^2$:

$$|\mathbf{w}^\top\mathbf{x}_i| = \mathcal{O}(1) \quad \Rightarrow \quad ||\mathbf{w}||^2 \approx Var[\mathbf{x}]^{-1}$$

- balance influence: $||\mathbf{w}||^2 \approx C\xi_i \quad \Rightarrow \quad C \approx Var[\mathbf{x}]^{-1}$

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

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

	task A	**task B**
S^{3}VM^{light}	94.53% (4) [6]	92.34% (2) [4]
∇S^{3}VM	96.72% (1) [3]	93.74% (2) [4]
contS^{3}VM	96.01% (1) [3]	93.56% (2) [4]

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^3VM

	task A	**task B**
$C = C^* = 1$ (guessed model)	96.01%	93.56%
model selection	89.31%	90.09%

- **significant drop in accuracy!**
- CV relys 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!