Semi-Supervised Learning

Machine Learning II (SS 2008, TU Berlin)

Prof. Dr. Klaus-Robert Müller
Dr. Alex Zien

22. 04. 2008

http://ml.cs.tu-berlin.de/en/

MAX-PLANCK-GESELLSCHAFT

1. Semi-Supervised Learning (SSL)
2. The Semi-Supervised SVM (S^3VM)
   - Training a S^3VM
3. Graph-Based Methods
   - Connections to Low Density Separation
4. Other SSL Approaches
   - Co-Training
   - Transduction
5. Overview of SSL and Summary

- **Unsupervised Learning:**
  given $\{\mathbf{x}_i\}_{i=1,\ldots,N}$, $\mathbf{x}_i \in \mathcal{X}$
  characterize $Pr\left(\mathbf{x}\right)$

- **Supervised Learning:**
  given $\{(\mathbf{x}_i, y_i)\}_{i=1,\ldots,N}$
  estimate $f : \mathcal{X} \to \mathcal{Y}$ such that $f(\mathbf{x}) \approx y$
  in other words, characterize $Pr\left(y \,|\mathbf{x}\,\right)$

- **Semi-Supervised Learning (SSL):**
  goal like for supervised,
  with additional unlabeled data $\{\mathbf{x}_j\}_{j=N+1,\ldots,N+M}$

Why SSL?

Labels are often expensive.

Generative model: $Pr(\mathbf{x}, y)$

$$\begin{aligned} Pr(data\,|\theta) &= \prod_i Pr(\mathbf{x}_i, y_i\,|\theta) \prod_j Pr(\mathbf{x}_j\,|\theta) \\ &= \prod_i Pr(\mathbf{x}_i, y_i\,|\theta) \prod_j \sum_y Pr(\mathbf{x}_j, y\,|\theta) \end{aligned}$$

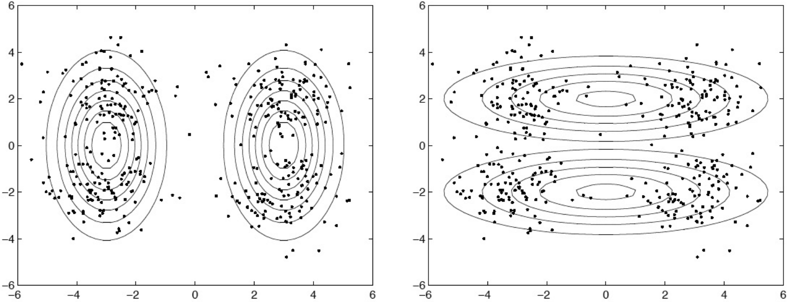
Maximize log likelihood:

$$\log \mathcal{L}(\theta) = \underbrace{\sum_i \log Pr(\mathbf{x}_i, y_i\,|\theta)}_{\textit{typically convex}} + \underbrace{\sum_j \log\left(\sum_y Pr(\mathbf{x}_j, y\,|\theta)\right)}_{\textit{typically non-convex}}$$

Standard tool for optimization (=training):
Expectation-Maximization (EM) algorithm

Unlabeled data can be misleading...

high likelihood wrong | low likelihood correct



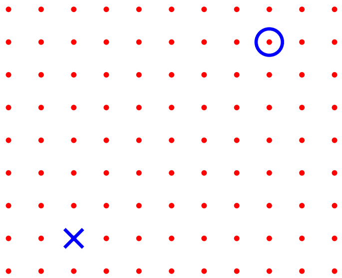
from [Semi-Supervised Learning, ICML 2007 Tutorial; Xiaojin Zhu]

Discriminative model: $Pr\left(y\,|\mathbf{x}\,\right)$

$$\mathcal{L}\left(\theta\right) \;=\; \prod_i Pr\left(y_i\,|\mathbf{x}_i,\theta\,\right)$$

Problem: Density of $\mathbf{x}$ does not help to estimate conditional $Pr\left(y\,|\mathbf{x}\,\right)$!

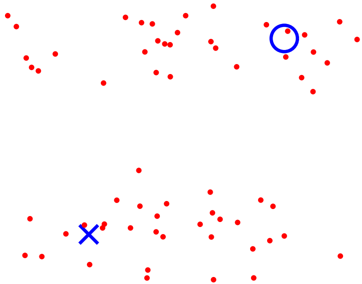
Why would unlabeled data be useful at all?



Uniform data do not help.

Cluster Assumption

1. The data form clusters.
2. Points in the **same cluster** are likely to be of the **same class**.



(Reall the standard **Supervised Learning Assumption**: Similar points tend to have similar labels.)

Cluster Assumption

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

- The cluster assumption seems to hold for many real data sets.
- Most SSL algorithms (implicitly) make use of it.
- No corresponding assumption for regression.

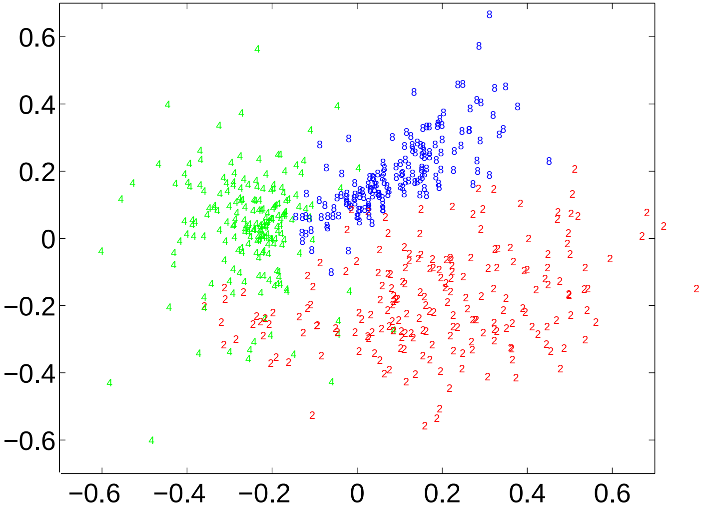
Equivalent assumption:

Low Density Separation Assumption

The decision boundary lies in a low density region.

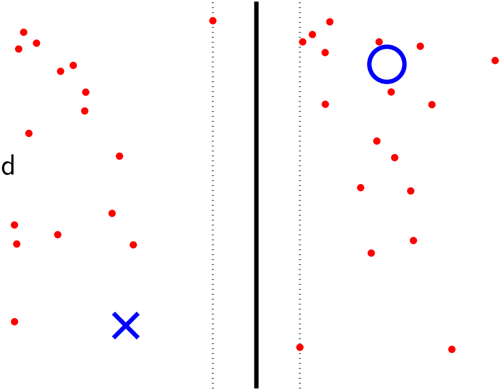
$\Rightarrow$ Algorithmic idea: **Low Density Separation**

Example application: **recognize handwritten digits 2, 4, 8**



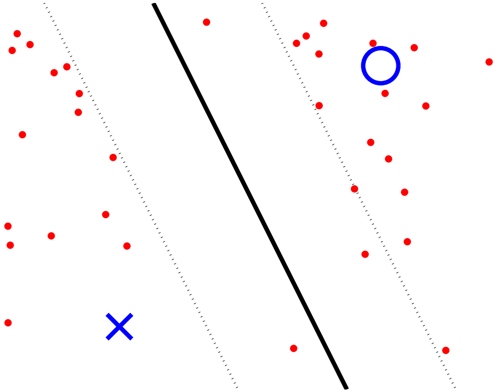
[non-linear 2D-embedding with “Stochastic Neighbor Embedding”]

S^{3}VM:
semi-supervised
SVM



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

soft margin
S^{3}VM



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

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^3VM)

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

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

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

Mixed Integer Programming [Bennett, Demiriz; NIPS 1998]

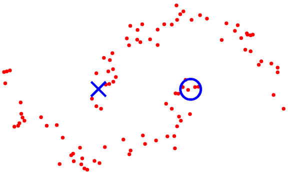
- global optimum found by standard optimization packages (eg CPLEX)
- **NP-hard !** $\Rightarrow$ only works for small sized problems

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!**

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

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

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

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

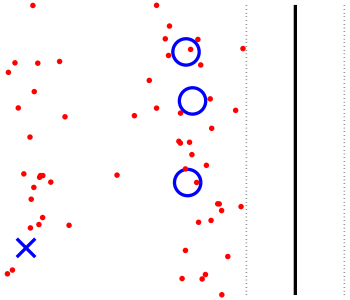
$$\frac{\#\{y_j = +1\}}{\#\text{ unlabeled points}} = \frac{\#\{y_i = +1\}}{\#\text{ labeled points}} \qquad (*)$$

- 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" (*)

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

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

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



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

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

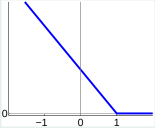
Effective Loss Functions

$$\xi_i = \min\{1 - \color{blue}{y_i}(\langle \mathbf{w}, \color{blue}{\mathbf{x}_i}\rangle + b), 0\}$$

$$\xi_j = \min_{\color{red}{y_j}\in\{+1,-1\}}\{1 - \color{red}{y_j}(\langle \mathbf{w}, \color{red}{\mathbf{x}_j}\rangle + b), 0\}$$

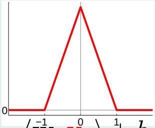
loss functions

ξ_i



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

ξ_j



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

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

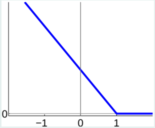
$$s.t. \quad \begin{array}{ll} y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1 - \xi_i & \xi_i \geq 0 \\ y_j(\langle \mathbf{w}, \mathbf{x}_j\rangle + b) \geq 1 - \xi_j & \xi_j \geq 0 \end{array}$$

Resolving the Constraints

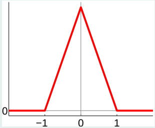
$$\frac{1}{2}\langle \mathbf{w},\mathbf{w}\rangle + C\sum_i \ell_l\left(y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b)\right) + C^*\sum_j \ell_u\left(\langle \mathbf{w}, \mathbf{x}_j\rangle + b\right)$$

loss functions

ℓ_l



ℓ_u

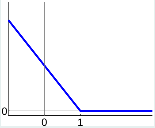


$$\frac{1}{2}\langle \mathbf{w},\mathbf{w}\rangle + C\sum_i \ell_l\left(y_i(\langle \mathbf{w},\mathbf{x}_i\rangle + b)\right) + C^*\sum_j \ell_u\left(\langle \mathbf{w},\mathbf{x}_j\rangle + b\right)$$

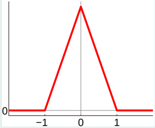
S^{3}VM as Unconstrained Differentiable Optimization Problem

original loss functions

ℓ_l

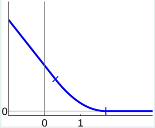


ℓ_u

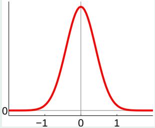


smooth loss functions

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]

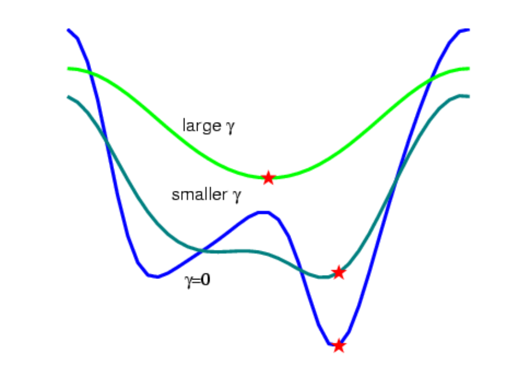
next slide...

The Continuation Method in a Nutshell

Procedure

1. smooth function until convex
2. find minimum
3. track minimum while decreasing amount of smoothing

Illustration

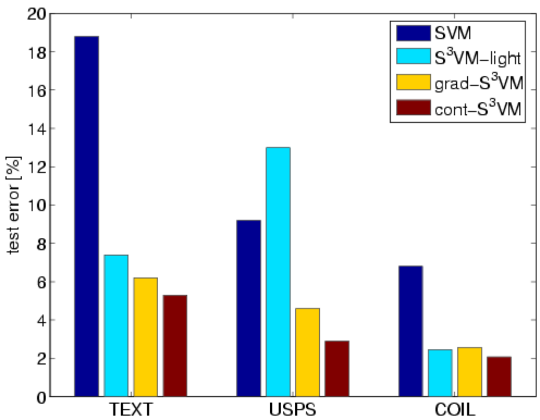


Comparison of S^3VM Optimization Methods

On three tasks (with ∼2000 points each, 100 of which labeled)

- TEXT:
  - do newsgroup texts refert to mac or to windows?
    $\Rightarrow$ binary classification
  - bag of words representation: ∼7500 dimensions, sparse
- USPS
  - recognize handwritten digits
  - 10 classes $\Rightarrow$ 45 one-vs-one binary tasks
  - 16×16 pixel image as input (256 dimensions)
- COIL
  - recognize 20 objects in images: 20 classes
  - 32×32 pixel image as input (1024 dimensions)

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]

Manifold Assumption

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

[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 nodes, ie $f : V \rightarrow \{-1, +1\}$
[recall: for SVMs, $f : \mathcal{X} \rightarrow \{-1, +1\}$, $\mathbf{x} \mapsto sign(\langle \mathbf{w}, \mathbf{x} \rangle + b)$]

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

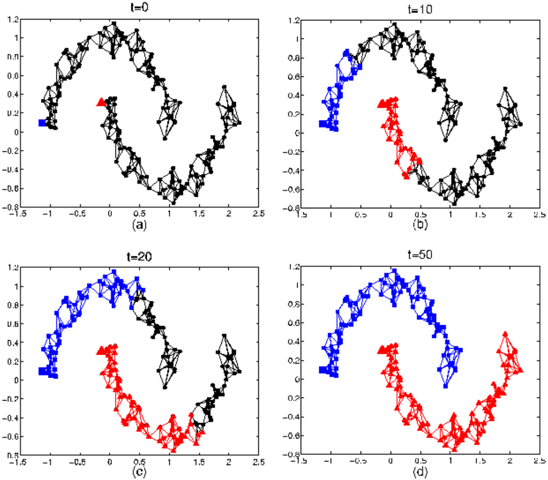
Label Propagation

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 semi-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 $\triangleq$ classes

[images from "Label Propagation Through Linear Neighborhoods", Wang, Zhang, ICML 2006]

“Beyond the Point Cloud” [Sindhwani, Niyogi, Belkin]

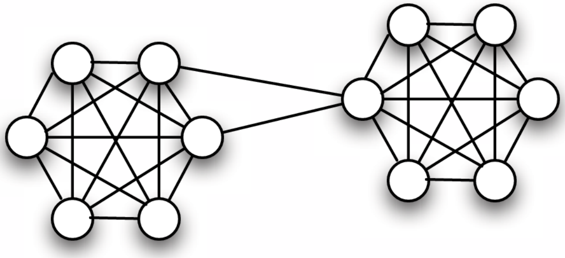
Idea:

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

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**.

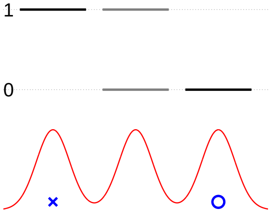
Manifold Assumption

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

Semi-Supervised Smoothness Assumption

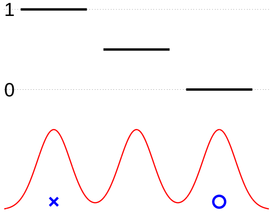
1. The density is non-uniform.
2. If two points are close in a high density region ($\Rightarrow$ connected by a high density path), their outputs are similar.

S^{3}VMs



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

Graph methods

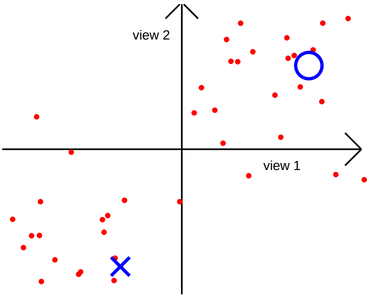


- Semi-Supervised Smoothness
- points within same cluster have **same class probabilities**
- convex

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**

Transduction

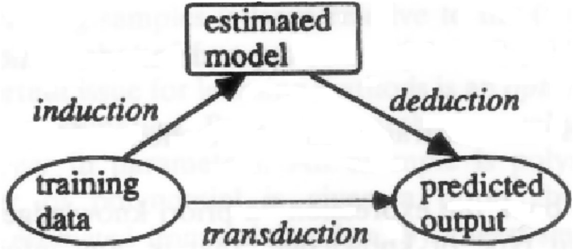


image from [Learning from Data: Concepts, Theory and Methods. V. Cherkassky, F. Mulier. Wiley, 1998.]

- concept introduced by Vladimir Vapnik
- philosophy: solve simpler task
- S^3VM originally called "Transductive SVM" (TSVM)

SSL vs Transduction

- Any SSL algorithm can be run in "transductive setting": use test data as unlabeled data.
- The "Transductive SVM" (S^3VM) is inductive.
- Some graph algorithms are transductive: prediction only available for nodes.

SSL Approaches

Assumption	Approach	Example Algorithm
Cluster Assumption	Low Density Separation	S^{3}VM (and many others)
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 $\sum_j \left(y_j^{(1)} - y_j^{(2)}\right)^2$

SSL Benchmark

average error [%] achieved with 100 labeled, ~ 1400 labeled points

Method	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^3SVM	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	–

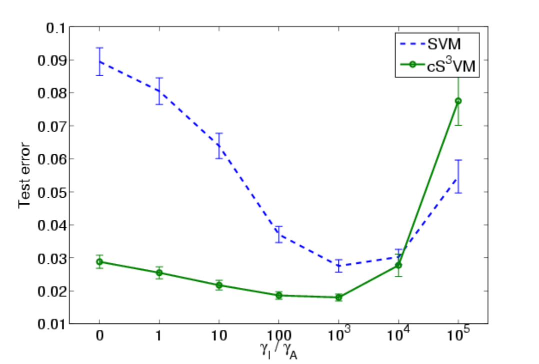
[Semi-Supervised Learning. Chapelle, Schölkopf, Zien. MIT Press, 2006.]

SSL Benchmark

[Semi-Supervised Learning. Chapelle, Schölkopf, Zien. MIT Press, 2006.]

Combining S^3VM with Graph-based Regularizer

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



[A Continuation Method for S^3VM; Chapelle, Chi, Zien; ICML 2006]

Summary

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