Semi-supervised Methods

Irwin King
Joint work with Zenglin Xu

Department of Computer Science & Engineering
The Chinese University of Hong Kong

Academia Sinica 2009

Outline

Outline

Semi-supervised learning

Semi-supervised learning

semi-supervised learning is a class of machine learning techniques that make use of both labeled and unlabeled data for training - typically a small amount of labeled data with a large amount of unlabeled data.

Advantages

- save manual labor in labeling data
- improve accuracy in labeling data

Example

labeled

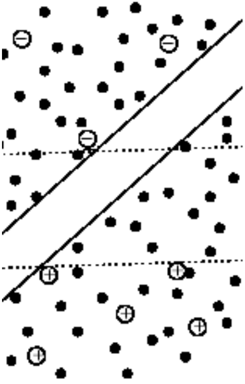

unlabeled


semi-supervised learning

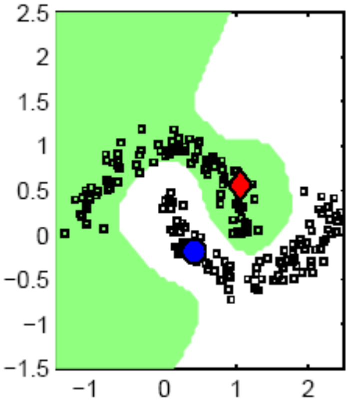
- drawn from the same distribution
- share the same label
- manifold assumption
- low density assumption
- surveys: [Zhu, 2005], [Chapelle et al., 2006]

Assumptions in semi-supervised learning

- low density assumption
- manifold assumption

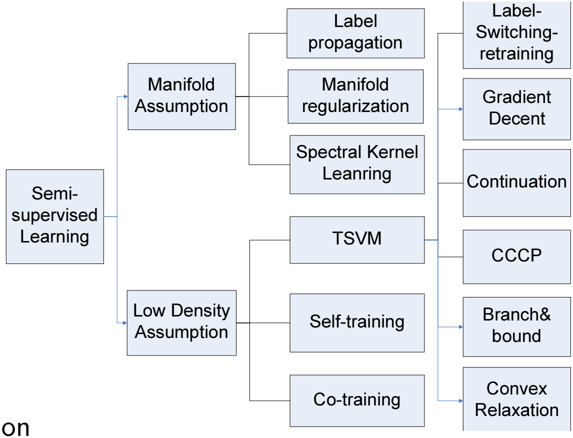


(a) Low density assumption



(b) Manifold assumption

Methods in semi-supervised learning



We will focus on

- Transductive Support Vector Machine (TSVM)

TSVM

Optimization in TSVM

- combinatorial optimization
- exponential complexity

Approximation methods for TSVM

- e.g., gradient descent optimization, label-switching-retraining, continuation method, Convex Concave Procedure, brunch and bound
- either local optima, or
- high complexity

Some challenges in semi-supervised learning

Challenges

perspectives for improving accuracy and efficiency:

- optimization technique for TSVM
- kernel for semi-supervised learning
- sparse models

Topics of this talk

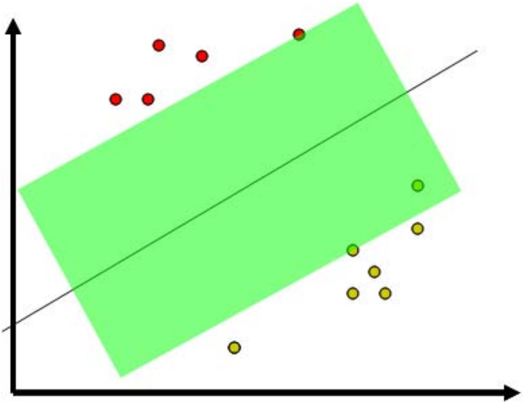
Topics

- How to learn an efficient Convex relaxation for TSVM?
- How to efficiently learn a kernel for semi-supervised learning?
- How to select features for semi-supervised learning?

Outline

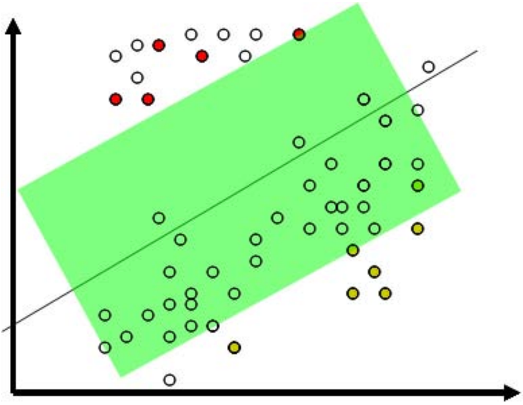
Transductive SVM

- SVM



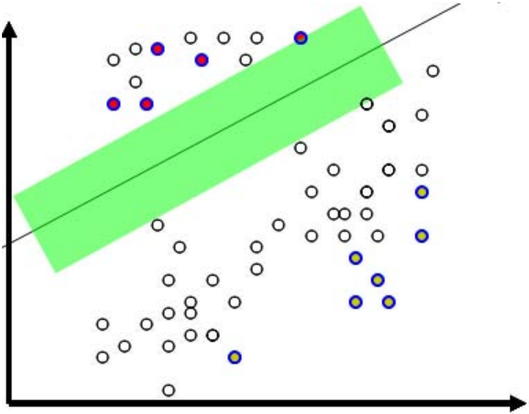
Transductive SVM

- SVM
- SVM with unlabeled data



Transductive SVM

- SVM
- SVM with unlabeled data
- Transductive SVM



Notations

- training data: $\{\mathbf{x}_i\}_{i=1}^{n}$
- kernel matrix: $\mathbf{K}$
- number of labeled examples: l
- labels of training data: $\mathbf{y}^{\ell}$
- decision function: $f = \mathbf{w}^{\top}\mathbf{x} - b$

Transductive SVM

TSVM: label **y** as a free variable

TSVM

$$\min_{\mathbf{w},b,\mathbf{y}\in\{-1,+1\}^n,\xi} \quad \|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n}\xi_i \tag{1}$$

$$\text{s. t.} \quad y_i(\mathbf{w}^\top \mathbf{x}_i - b) \geq 1 - \xi_i,$$

$$\xi_i \geq 0,\ i = 1, 2, \ldots, n$$

$$y_i = y_i^\ell,\ i = 1, 2, \ldots, l,$$

- margin error: ξ
- tradeoff parameter: C

Transductive SVM

- $\circ$: element-wise product;
- $\mathbf{e}$: vector of all ones

Dual form

$$\begin{aligned} \max_{\alpha, \mathbf{y} \in \{-1,+1\}^n} \quad & \alpha^\top \mathbf{e} - \frac{1}{2}(\alpha \circ \mathbf{y})^\top \mathbf{K}(\alpha \circ \mathbf{y}) \\ \text{s. t.} \quad & 0 \leq \alpha \leq C, \\ & y_i = y_i^\ell, \ i = 1, 2, \ldots, l, \end{aligned} \tag{2}$$

problems:

- non-convex problem
- difficult to solve

Transductive SVM

Calculate the Lagrangian of (2)

$$L(\alpha, \nu, \delta, \lambda) = \alpha^\top \mathbf{e} - \frac{1}{2}(\alpha \circ \mathbf{y})^\top \mathbf{K}(\alpha \circ \mathbf{y}) + \nu^\top \alpha + \lambda \mathbf{y}^\top \alpha + \delta(C\mathbf{e} - \alpha)$$

- $\nu \in \mathbb{R}^n$: $\alpha \geq 0$
- $\delta \in \mathbb{R}^n$: $\alpha \leq C$
- λ: $\alpha^\top \mathbf{y} = 0$

Set $\frac{\partial L}{\partial \alpha} = 0$,

Dual form

$$\begin{aligned} \min_{\nu, \mathbf{y}, \lambda, \delta} \quad & \frac{1}{2}(\mathbf{e} + \nu - \delta + \lambda \mathbf{y})^\top \mathrm{D}(\mathbf{y}) \mathbf{K}^{-1} \mathrm{D}(\mathbf{y})(\mathbf{e} + \nu - \delta + \lambda \mathbf{y}) \qquad (3) \\ \text{s. t.} \quad & \nu \geq 0, \\ & y_i = y_i^\ell, \ i = 1, 2, \ldots, l, \\ & y_i^2 = 1, \ i = l+1, l+2, \ldots, n. \end{aligned}$$

Primal form of TSVM

introduce $\frac{1}{2}(\mathbf{e}+\nu-\delta+\lambda\mathbf{y})^\top \mathrm{D}(\mathbf{y})\mathbf{K}^{-1}\mathrm{D}(\mathbf{y})(\mathbf{e}+\nu-\delta+\lambda\mathbf{y}) \leq t$
According to Schur complement, (3) is equivalent to

Semi-definite programming for TSVM [Lanckriet et al., 2004]

$$\min_{\mathbf{y}\in\{-1,+1\}^n, t, \nu, \delta, \lambda} \quad t \tag{4}$$

$$\text{s. t.} \quad \begin{pmatrix} \mathbf{y}\mathbf{y}^\top \circ \mathbf{K} & \mathbf{e}+\nu-\delta+\lambda\mathbf{y} \\ (\mathbf{e}+\nu-\delta+\lambda\mathbf{y})^\top & t-2C\delta^\top\mathbf{e} \end{pmatrix} \succeq 0$$

$$\nu \geq 0,\ \delta \geq 0,\ y_i = y_i^\ell,\ i = 1, 2, \ldots, l,$$

problem:

- still non-convex for $\mathbf{y}$

Convex Relaxation of TSVM

replace $\mathbf{y}\mathbf{y}^\top$ with matrix $\mathbf{M}$ [Xu & Schuurmans, 2004]:

Convex Relaxation of TSVM

$$\begin{aligned} \min_{\mathbf{M},t,\nu,\delta,\lambda} \quad & t \\ \text{s. t.} \quad & \begin{pmatrix} \mathbf{M} \circ \mathbf{K} & \mathbf{e} + \nu - \delta \\ (\mathbf{e} + \nu - \delta)^\top & t - 2C\delta^\top \mathbf{e} \end{pmatrix} \succeq 0 \\ & \nu \geq 0, \ \delta \geq 0, \\ & \mathbf{M} \succeq 0, \ M_{i,i} = 1, \ i = 1, 2, \ldots, n, \\ & M_{ij} = y_i^\ell y_j^\ell, \ 1 \leq i,j \leq l \end{aligned} \tag{5}$$

- polynomial solution

Problems of the relaxation

1. $\mathcal{O}(n^2)$ parameters in the SDP cone
   - high worst-case computational complexity: $\mathcal{O}(n^{6.5})$
   - high storage complexity
2. drop the rank constraint of the matrix $\mathbf{y}\mathbf{y}^\top$
   - not tight approximation

Our solution

Start from the hard margin case ($\delta = 0$) of optimization problem (3),

$$\begin{aligned}
\min_{\nu,\mathbf{y},\lambda} \quad & \frac{1}{2}(\mathbf{e}+\nu+\lambda\mathbf{y})^\top \mathrm{D}(\mathbf{y})\mathbf{K}^{-1}\mathrm{D}(\mathbf{y})(\mathbf{e}+\nu+\lambda\mathbf{y}) \\
\text{s. t.} \quad & \nu \geq 0, \\
& y_i = y_i^\ell,\ i = 1, 2, \ldots, l, \\
& y_i^2 = 1,\ i = l+1, l+2, \ldots, n.
\end{aligned}$$

Our solution

- introduce $\mathbf{z} = \mathrm{D}(\mathbf{y})(\mathbf{e}+\nu) = \mathbf{y} \circ (\mathbf{e}+\nu)$
- $\mathbf{z}$ can be used as the prediction function

Hard margin TSVM

$$\begin{aligned} \min_{\mathbf{z},\lambda} \quad & \frac{1}{2}(\mathbf{z}+\lambda\mathbf{e})^\top \mathbf{K}^{-1}(\mathbf{z}+\lambda\mathbf{e}) \\ \text{s. t.} \quad & y_i^\ell z_i \geq 1,\ i = 1, 2, \ldots, l, \\ & z_i^2 \geq 1,\ i = l+1, l+2, \ldots, n. \end{aligned} \tag{6}$$

Our solution

reformulation:

- $\mathbf{w} = (\mathbf{z}, \lambda) \in \mathbb{R}^{n+1}$
- $\mathbf{P} = (\mathbf{I}_n, \mathbf{e}) \in \mathbb{R}^{n\times(n+1)}$

balance constraint (to avoid putting unlabeled examples to one side):

- $-\epsilon \leq \frac{1}{l}\sum_{i=1}^{l} w_i - \frac{1}{n-l}\sum_{i=l+1}^{n} w_i \leq \epsilon$

$$
\begin{aligned}
\min_{\mathbf{w}} \quad & \mathbf{w}^\top \mathbf{P}^\top \mathbf{K}^{-1} \mathbf{P} \mathbf{w} \\
\text{s. t.} \quad & y_i^\ell w_i \geq 1,\ i = 1, 2, \ldots, l, \\
& w_i^2 \geq 1,\ i = l+1, l+2, \ldots, n, \\
& -\epsilon \leq \frac{1}{l}\sum_{i=1}^{l} w_i - \frac{1}{n-l}\sum_{i=l+1}^{n} w_i \leq \epsilon.
\end{aligned}
\tag{7}
$$

Our solution

Lagragian of (7):

$$L = \mathbf{w}^\top \mathbf{P}^\top \mathbf{K}^{-1} \mathbf{P} \mathbf{w} + \sum_{i=1}^{l} \gamma_i (1 - y_i^\ell w_i) + \sum_{i=l+1}^{n} \gamma_i (1 - w_i^2) + \alpha(\mathbf{c}^\top \mathbf{w} - \epsilon) + \beta(-\mathbf{c}^\top \mathbf{w} - \epsilon) \quad (8)$$

Solution

$$\mathbf{w} = \frac{1}{2} \left[\mathbf{A} - \mathrm{D}(\gamma \circ \mathbf{b})\right]^{-1} (\gamma \circ \mathbf{a} - (\alpha - \beta)\mathbf{c}),$$

- $\mathbf{a} = (\mathbf{y}^l, \mathbf{0}^{n-l}, 0) \in \mathbb{R}^{n+1}$
- $\mathbf{b} = (\mathbf{0}^l, \mathbf{1}^{n-l}, 0) \in \mathbb{R}^{n+1}$
- $\mathbf{c} = (\frac{1}{l}\mathbf{1}^l, -\frac{1}{u}\mathbf{1}^{n-l}, 0) \in \mathbb{R}^{n+1}$
- $\mathbf{A} = \mathbf{P}^\top \mathbf{K}^{-1} \mathbf{P}$

Our solution

TSVM in dual

$$\max_{\gamma,t} \quad -\frac{1}{4}t + \sum_{i=1}^{n} \gamma_i - \epsilon(\alpha + \beta) \tag{9}$$

$$s.\ t. \quad \begin{pmatrix} \mathbf{A} - \mathrm{D}(\gamma \circ \mathbf{b}) & \gamma \circ \mathbf{a} - (\alpha - \beta)\mathbf{c}, \\ (\gamma \circ \mathbf{a} - (\alpha - \beta)\mathbf{c})^\top & t \end{pmatrix} \succeq 0$$

$$\alpha \geq 0,\ \beta \geq 0,\ \gamma_i \geq 0,\ i = 1, 2, \ldots, n.$$

Properties of the proposed convex relaxation model

- Lower worst-case computational complexity of $\mathcal{O}(n^{4.5})$: $\mathcal{O}(n)$ parameters and $\mathcal{O}(n)$ linear equality constraints
- Our prediction function f^* provides a tighter approximation [Hiriart et al., 1993].
- Related to the solution of [Zhu et al., 2003] :

$$\mathbf{z} = \left(\mathbf{I}_n - \sum_{i=l+1}^{n} \gamma_i \mathbf{K}\mathbf{I}_n^i\right)^{-1} \left(\sum_{i=1}^{l} \gamma_i y_i^\ell \mathbf{K}(\mathbf{x}_i, \cdot)\right) \tag{10}$$

Data sets

Table: Data sets used in the experiments, where d represents the data dimensionality, l means the number of labeled data points, and n denotes the total number of examples.

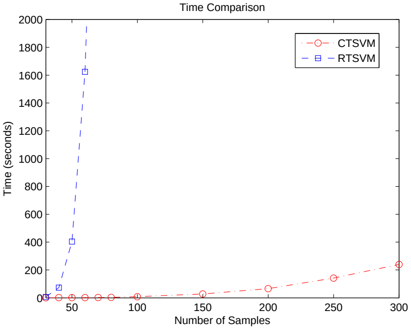
Data set	d	l	n	Data set	d	l	n
Iono	34	20	351	WinMac-m	7511	20	300
Sonar	60	20	208	IBM-m	11960	20	300
Banana	4	20	400	Course-m	1800	20	300
Breast	9	20	300	WinMac-l	7511	50	1000
IBM-s	11960	10	60	IBM-l	11960	50	1000
Course-s	1800	10	60	Course-l	1800	50	1000

Comparison algorithms

- SVM: baseline
- label-switching-retraining in SVM-light [Joachims,1999]
- Convex concave procedure [Collobert et al., 2006]
- Gradient decent (∇TSVM), [Chapelle et al., 2005]

Computation time comparison

- CTSVM: convex relaxation TSVM proposed [Xu et al., 2007]
- RTSVM: previous semi-definite relaxation TSVM [Xu & Schuurmans, 2004]



- Course, labeled 20

Accuracy comparison

Table: The classification performance of Transductive SVMs on benchmark data sets. (Note: ∇TSVM = Gradient Decent TSVM, CCCP = Concave Convex Procedure)

Data Set	SVM	SVM-light	∇TSVM	CCCP	CTSVM
IBM-s	52.75±15.01	67.60±9.29	65.80±6.56	65.62±14.83	**75.25**±7.49
Course-s	63.52±5.82	76.82±4.78	75.80±12.87	74.20±11.50	**79.75**±8.45
Iono	78.55±4.83	78.25±0.36	81.72±4.50	**82.11**±3.83	80.09±2.63
Sonar	51.76±5.05	55.26±5.88	**69.36**±4.69	56.01±6.70	67.39±6.26
Banana	58.45±7.15	-	71.54±7.28	79.33±4.22	**79.51**±3.02
Breast	96.46±1.18	95.68±1.82	97.17±0.35	96.89±0.67	**97.79**±0.23
WinMac-m	57.64±9.58	79.42±4.60	81.03±8.23	84.28±8.84	**84.82**±2.12
IBM-m	53.00±6.83	67.55±6.74	64.65±13.38	69.62±11.03	**73.17**±0.89
Course-m	80.18±1.27	**93.89**±1.49	90.35±3.59	88.78±2.87	92.92±2.28
WinMac-l	60.86±10.10	89.81±2.10	90.19±2.65	91.00±2.42	**91.25**±2.67
IBM-l	61.82±7.26	**75.40**±2.26	73.11±1.99	74.80±1.87	73.42±3.23
Course-l	83.56±3.10	92.35±3.02	93.58±2.68	91.32±4.08	**94.62**±0.97

Discussion

- More efficient than that in [Xu & Schuurmans, 2004]
- Effective prediction accuracy compared with other semi-supervised SVM algorithms
- All algorithms sensitive to data sets
- Consistent to the results in [Chapelle et al., 2008]

Outline

Semi-supervised kernels

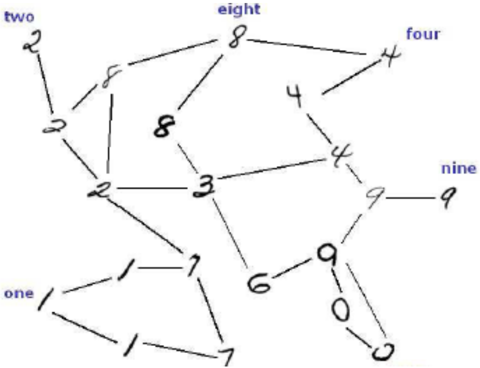
- spectral kernel learning
  - the eigenvalues are readjusted according to some principle
  - Gaussian field kernel [Zhu et. al, 2002]
  - cluster kernel [Chapelle et. al, 2003]
  - spectral kernel [Zhang & Ando, 2005]
- multiple kernel learning
  - a linear combination of a batch of base kernels [Lancriet et. al, 2004]
- graph embedding
  - embed a graph structure into the supervised kernel [Sindhwani et. al, 2005]

Graph embedding

Graph construction

- $\mathcal{G} =< \mathcal{V}, \mathcal{E} >$, where $\mathcal{V} = \{\mathbf{x}_i\}_{i=1}^{N}$
- build adjacency graph
  - ϵ-NN. $\epsilon \in \mathbb{R}^+$. Nodes $\mathbf{x}_i$ and $\mathbf{x}_j$ are connected if $\mathrm{dist}(\mathbf{x}_i, \mathbf{x}_j) \geq \epsilon$
  - k-NN. $k \in \mathbb{N}^+$. Nodes $\mathbf{x}_i$ and $\mathbf{x}_j$ are connected if $\mathbf{x}_i$ is among the k nearest neighbors of $\mathbf{x}_j$.
- graph weighting
  - Heat kernel. If $\mathbf{x}_i$ and $\mathbf{x}_j$ are connected, the weight $W_{ij} = \exp^{-\frac{dist(\mathbf{x}_i, \mathbf{x}_j)}{t}}$, where $t \in \mathbb{R}^+$.
  - Simple-minded. $W_{ij} = 1$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ are connected.

Graph embedding



- $\mathcal{G} =< \mathcal{V}, \mathcal{E} >$
- W_{ij}: weights on edge $(\mathbf{x}_i, \mathbf{x}_j)$
- $D_{ii} = \sum_{j=1}^{n} W_{ij}$
- graph Laplacian: $\mathcal{L} = \mathbf{D} - \mathbf{W}$
- weighted graph Laplacian: $\mathcal{L} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{D} - \mathbf{W})\mathbf{D}^{-\frac{1}{2}}$

Graph embedding

Question?

Can we define a kernel that is adapted to the geometry of the data distribution?

$\Downarrow$

Solution

Define a new RKHS to incorporate the data geometry, such that

$$\langle f, g\rangle_{\tilde{\mathcal{H}}} = \langle f, g\rangle_{\mathcal{H}} + \langle Sf, Sg\rangle_{\mathcal{V}} \tag{11}$$

- $\kappa(\mathbf{x}, \cdot)$: functional in the **Reproducing Kernel Hilbert Space** (RKHS) $\mathcal{H}$
- $\tilde{\kappa}(\mathbf{x}, \cdot)$: functional in the new RKHS $\tilde{\mathcal{H}}$
- $f(\mathbf{x}) = \langle f, \kappa(\mathbf{x}, \cdot)\rangle, \quad \kappa(\mathbf{x}, \mathbf{z}) = \langle \kappa(\mathbf{x}, \cdot), \kappa(\mathbf{x}, \cdot)\rangle_{\mathcal{H}}$
- $S : \mathcal{H} \rightarrow \mathcal{V}$: bounded linear operator

Graph embedding

Define

- $S(f) = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n))$.
- $\|Sf\|_{\mathcal{V}}^2 = \mathbf{f}^\top \mathcal{L} \mathbf{f}$

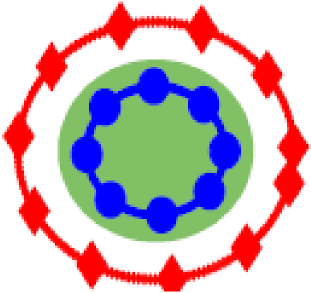
Graph embedding

According to [Sindhwani et. al, 2005], Given a kernel function $\kappa(\cdot,\cdot)$, the new kernel $\tilde{\kappa}(\cdot,\cdot)$ embedded with the graph structure is defined as

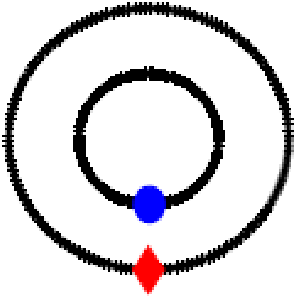
$$\tilde{\kappa}(\mathbf{x},\mathbf{z}) = \kappa(\mathbf{x},\mathbf{z}) - \mathbf{k}_{\mathbf{x}}(\mathbf{I} + \mathcal{L}\mathbf{K})^{-1}\mathcal{L}\mathbf{k}_{\mathbf{z}}. \tag{12}$$

- $\mathbf{k}_{\mathbf{x}} = (\kappa(\mathbf{x}_1,\mathbf{x}), \ldots, \kappa(\mathbf{x}_n,\mathbf{x}))^\top$

Illustration



(a) *data*

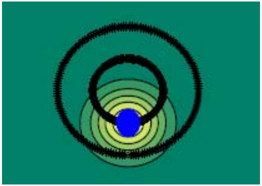


(b) *labeled data*

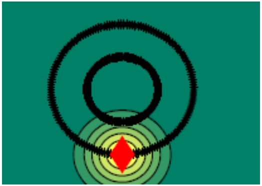
Figure: scatter plot of data

Illustration

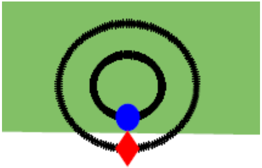
RBF kernel



(a) *Gaussian kernel centered on labeled point 1*



(b) *Gaussian kernel centered on labeled point 2*

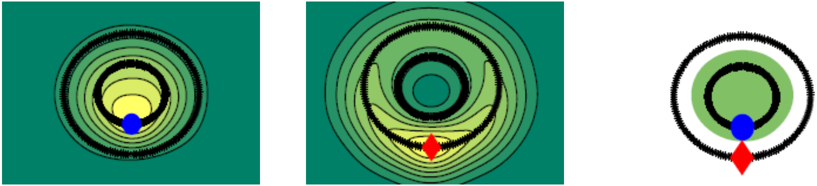


(c) *classifier learnt in the RKHS*

Figure: Gaussian kernel

Illustration

kernel embedded with the graph structure



(a) *embedded kernel centered on labeled point 1*

(b) *embedded kernel centered on labeled point 2*

(c) *classifier learnt in the deformed RKHS*

Figure: Kernel embedded with the graph structure

Challenges of graph embedding

Challenges

- the kernel function $\kappa(\cdot, \cdot)$ for embedding, and
- the graph structure that is used to calculate the graph Laplacian $\mathcal{L}$.

Solutions

- employ multiple kernel learning to select the kernel function $\kappa(\cdot, \cdot)$ and the graph structure

Multiple kernel learning (MKL)

Multiple kernel learning

Given a list of base kernel functions/matrices $\mathbf{K}_i$, $i = 1, \ldots, m$, MKL searches for a linear combination of the base kernel functions that maximizes a generalized performance measure.

Linear combination of kernels

$$\mathbf{K} = \sum_{i=1}^{m} p_i \mathbf{K}_i,\ i = 1, \ldots, m$$

where $\mathbf{p} = (p_1, \ldots, p_m)$ are combination weights in domain $\mathcal{P}$

$$\mathcal{P} = \{\mathbf{p} \in \mathbb{R}^m : \mathbf{p}^\top \mathbf{e} = 1,\ 0 \leq \mathbf{p} \leq 1\}$$

Candidate graphs for semi-supervised learning

Parameter sets when constructing the graph

- distance function: $\mathcal{D} = \{d_1, \ldots, d_r\}$
  - e.g. Euclidean distance, tangent distance
- neighborhood number: $\mathcal{K} = \{k_1, \ldots, k_s\}$
  - e.g. 2, 10, 100, ...
- heat kernel width: $\mathrm{T} = \{t_1, \ldots, t_q\}$
  - e.g. $1e^{-2}$, $1e^{-1}$, 1, 10, ...

Candidate graphs

- $u = r \times s \times q$ graphs
- $\mathcal{L}_i = \mathbf{D}_i - \mathbf{W}_i$ for $i = 1, \ldots, u$

Candidate embedded kernels for semi-supervised learning

For

- i-th ($i = 1, \ldots, u$) candidate graph
- j-th ($j = 1, \ldots, v$) base kernel

embedded kernels

$$\tilde{\kappa}_{ij}(\mathbf{x}, \mathbf{z}) = \kappa_i(\mathbf{x}, \mathbf{z}) - \mathbf{k}_\mathbf{x}(\mathbf{I} + \mathcal{L}_j \mathbf{K}_i)^{-1} \mathcal{L}_j \mathbf{k}_\mathbf{z}. \qquad (13)$$

Multiple kernel learning

- number of base kernels $m = u \times v$

Multiple kernel learning in semi-supervised setting

$$\min_{\mathbf{p}\in\mathcal{P}} \max_{\alpha\in\mathcal{Q}} f(\mathbf{p},\alpha) = \alpha^\top \mathbf{e} - \frac{1}{2}(\alpha \circ \mathbf{y})^\top \left(\sum_{i=1}^{m} p_i \tilde{\mathbf{K}}_i\right)(\alpha \circ \mathbf{y}),$$

Properties

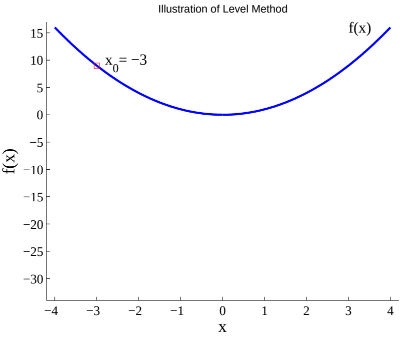
- convex-concave problem (convex in $\mathbf{p}$ and concave in α)
- saddle point $(\mathbf{p}^*, \alpha^*)$ exists and corresponds to the optimal solution

$$f(\mathbf{p},\alpha^*) \leq f(\mathbf{p}^*,\alpha^*) \leq f(\mathbf{p}^*,\alpha), \forall \mathbf{p} \in \mathcal{P}, \alpha \in \mathcal{Q}$$

Level Method

$$\min_x \{f(x) = [x]^2 : x \in \mathcal{X}, \mathcal{X} = [-4, 4]\}$$

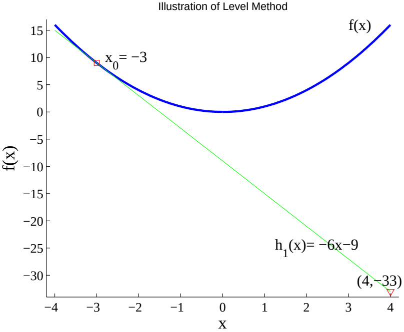
- Initialization: $x_0 = -3$, $\lambda = 0.9$



Level Method

$$\min_{x}\{f(x) = [x]^2 : x \in \mathcal{X}, \mathcal{X} = [-4, 4]\}$$

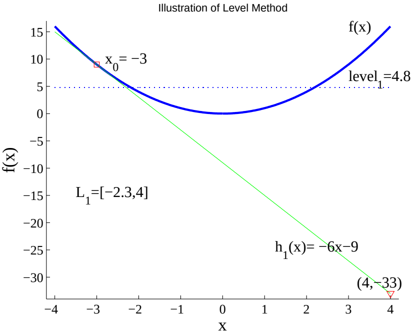
- Initialization: $x_0 = -3$, $\lambda = 0.9$
- Construct a cutting plane model $g_1(x)$



Level Method

$$\min_x \{f(x) = [x]^2 : x \in \mathcal{X}, \mathcal{X} = [-4, 4]\}$$

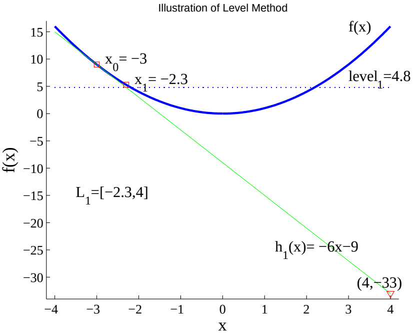
- Initialization: $x_0 = -3$, $\lambda = 0.9$
- Construct a cutting plane model $g_1(x)$
- Construct a level set $\mathcal{L}_1$
  $\text{level}_1 = \lambda \times f(x_0) + (1-\lambda) \times (-33)$
  $\mathcal{L}_1 = \{x \in \mathcal{X} : g_1(x) \leq \text{level}_1\}$



Level Method

$$\min_x \{f(x) = [x]^2 : x \in \mathcal{X}, \mathcal{X} = [-4, 4]\}$$

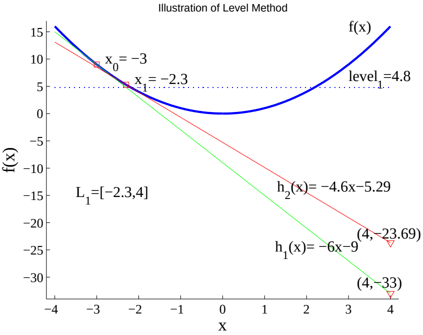
- Initialization: $x_0 = -3$, $\lambda = 0.9$
- Construct a cutting plane model $g_1(x)$
- Construct a level set $\mathcal{L}_1$
  $\text{level}_1 = \lambda \times f(x_0) + (1 - \lambda) \times (-33)$
  $\mathcal{L}_1 = \{x \in \mathcal{X} : g_1(x) \leq \text{level}_1\}$
- Project x_0 to level set $\mathcal{L}_1$, i.e.,
  $x_1 = \arg\min_x \left\{\|x - x_0\|_2^2 : x \in \mathcal{L}_1\right\}$



Level method

$$\min_x \{f(x) = [x]^2 : x \in [-4, 4]\}$$

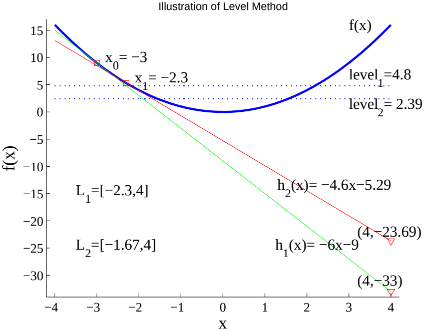
- Construct a new cutting plane model $g_2(x) = \min_x h_i(x)$



Level method

$$\min_x \{f(x) = [x]^2 : x \in [-4, 4]\}$$

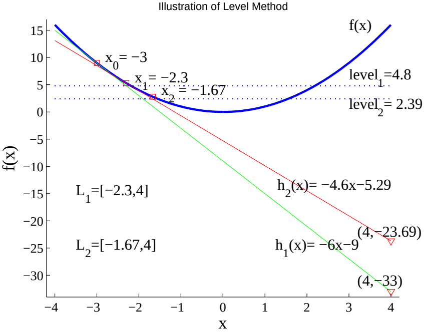
- Construct a new cutting plane model $g_2(x) = \min_x h_i(x)$
- Construct a new level set $\mathcal{L}_2$



Level method

$$\min_x \{f(x) = [x]^2 : x \in [-4, 4]\}$$

- Construct a new cutting plane model $g_2(x) = \min_x h_i(x)$
- Construct a new level set $\mathcal{L}_2$
- Project x_1 to $\mathcal{L}_2$



Cutting plane models

Cutting plane models

$$g^i(\mathbf{p}) = \max_{1 \le j \le i} f(\mathbf{p}^j, \alpha^j) + (\mathbf{p} - \mathbf{p}^j)^\top \nabla_{\mathbf{p}} f(\mathbf{p}^j, \alpha^j)$$

Properties

For any $\mathbf{p} \in \mathcal{P}$, we have

- $g^{i+1}(\mathbf{p}) \ge g^i(\mathbf{p})$, and
- $g^i(\mathbf{p}) \le \max_{\alpha \in \mathcal{Q}} f(\mathbf{p}, \alpha)$

Lower and upper bounds

Lower and upper bounds

$$\underline{f}^i = \min_{\mathbf{p}\in\mathcal{P}} g^i(\mathbf{p}), \quad \overline{f}^i = \min_{1\le j\le i} f(\mathbf{p}^j, \alpha^j)$$

Properties

$$\underline{f}^i \le f(\mathbf{p}^*, \alpha^*) \le \overline{f}^i,$$
$$\overline{f}^1 \ge \overline{f}^2 \ge \ldots \ge \overline{f}^i,$$
$$\underline{f}^1 \le \underline{f}^2 \le \ldots \le \underline{f}^i.$$

where $\mathbf{p}^*$ and α^* are the optimal solution.

Projection to level set

Level set

$$\mathcal{L}^i = \{\mathbf{p} \in \mathcal{P} : g^i(\mathbf{p}) \leq \ell^i = \lambda \overline{f}^i + (1-\lambda)\underline{f}^i\},$$

where $\lambda \in (0,1)$ is a predefined constant.

- larger $\lambda \rightarrow$ more regularization
- $\lambda = 0$: the level method becomes the cutting plane method

Projection to level set

$$\mathbf{p}^{i+1} = \arg\min_{\mathbf{p}\in\mathcal{P}} \left\{ \|\mathbf{p} - \mathbf{p}^i\|_2^2 : \mathbf{p} \in \mathcal{L}^i \right\}$$

Stopping Criterion

Define the gap Δ^i as

$$\Delta^i = \overline{f}^i - \underline{f}^i.$$

Corollary

1. $\Delta^j \geq 0, j = 1, \ldots, i$
2. $\Delta^1 \geq \Delta^2 \geq \ldots \geq \Delta^i$
3. $|f(\mathbf{p}^j, \alpha^j) - f(\mathbf{p}^*, \alpha^*)| \leq \Delta^i$

- Δ^i measures how close the current solution is from the optimal one, serving as the stopping criterion.

The level method for multiple kernel learning

Given: λ (level set) and ε (desired accuracy)

1. Initialize: $\mathbf{p}^0 = \mathbf{e}/m$, and $i = 0$

The level method for multiple kernel learning

Given: λ (level set) and ε (desired accuracy)

1. Initialize: $\mathbf{p}^0 = \mathbf{e}/m$, and $i = 0$
2. REPEAT

The level method for multiple kernel learning

Given: λ (level set) and ε (desired accuracy)

1. Initialize: $\mathbf{p}^0 = \mathbf{e}/m$, and $i = 0$
2. REPEAT
3. Solve dual SVM with $\tilde{\mathbf{K}} = \sum_{j=1}^{m} p_j^i \tilde{\mathbf{K}}_j$ for α^i

The level method for multiple kernel learning

Given: λ (level set) and ε (desired accuracy)

1. Initialize: $\mathbf{p}^0 = \mathbf{e}/m$, and $i = 0$
2. REPEAT
3. Solve dual SVM with $\tilde{\mathbf{K}} = \sum_{j=1}^{m} p_j^i \tilde{\mathbf{K}}_j$ for α^i
4. Construct the cutting plane model $g^i(\mathbf{p})$

The level method for multiple kernel learning

Given: λ (level set) and ε (desired accuracy)

1. Initialize: $\mathbf{p}^0 = \mathbf{e}/m$, and $i = 0$
2. REPEAT
3. Solve dual SVM with $\tilde{\mathbf{K}} = \sum_{j=1}^{m} p_j^i \tilde{\mathbf{K}}_j$ for α^i
4. Construct the cutting plane model $g^i(\mathbf{p})$
5. Compute the lower & upper bounds $\underline{f}^i$ and $\overline{f}^i$, and gap Δ^i

The level method for multiple kernel learning

Given: λ (level set) and ε (desired accuracy)

1. Initialize: $\mathbf{p}^0 = \mathbf{e}/m$, and $i = 0$
2. REPEAT
3. Solve dual SVM with $\tilde{\mathbf{K}} = \sum_{j=1}^{m} p_j^i \tilde{\mathbf{K}}_j$ for α^i
4. Construct the cutting plane model $g^i(\mathbf{p})$
5. Compute the lower & upper bounds $\underline{f}^i$ and $\overline{f}^i$, and gap Δ^i
6. $\mathbf{p}^{i+1} \leftarrow$ projection of $\mathbf{p}^i$ to the level set $\mathcal{L}^i$

The level method for multiple kernel learning

Given: λ (level set) and ε (desired accuracy)

1. Initialize: $\mathbf{p}^0 = \mathbf{e}/m$, and $i = 0$
2. REPEAT
3. Solve dual SVM with $\tilde{\mathbf{K}} = \sum_{j=1}^{m} p_j^i \tilde{\mathbf{K}}_j$ for α^i
4. Construct the cutting plane model $g^i(\mathbf{p})$
5. Compute the lower & upper bounds $\underline{f}^i$ and $\overline{f}^i$, and gap Δ^i
6. $\mathbf{p}^{i+1} \leftarrow$ projection of $\mathbf{p}^i$ to the level set $\mathcal{L}^i$
7. Update $i = i + 1$

The level method for multiple kernel learning

Given: λ (level set) and ε (desired accuracy)

1. Initialize: $\mathbf{p}^0 = \mathbf{e}/m$, and $i = 0$
2. REPEAT
3. Solve dual SVM with $\tilde{\mathbf{K}} = \sum_{j=1}^{m} p_j^i \tilde{\mathbf{K}}_j$ for α^i
4. Construct the cutting plane model $g^i(\mathbf{p})$
5. Compute the lower & upper bounds $\underline{f}^i$ and $\overline{f}^i$, and gap Δ^i
6. $\mathbf{p}^{i+1} \leftarrow$ projection of $\mathbf{p}^i$ to the level set $\mathcal{L}^i$
7. Update $i = i + 1$
8. UNTIL$\Delta^i \leq \varepsilon$

Experimental setup: semi-supervised setting

- Base kernel matrices for embedding
  - Gaussian kernels with 10 different widths ($\{2^{-3}, 2^{-2}, \ldots, 2^{6}\}$) on all features,
  - Polynomial kernels of degree 1 to 3 on all features,
- Graphs: 10 NN, cosine similarity
- heat kernel width: $\{0.5, 1, 2, 4, 8\}$
- Other settings similar to the supervised setting

Experimental setup: semi-supervised setting

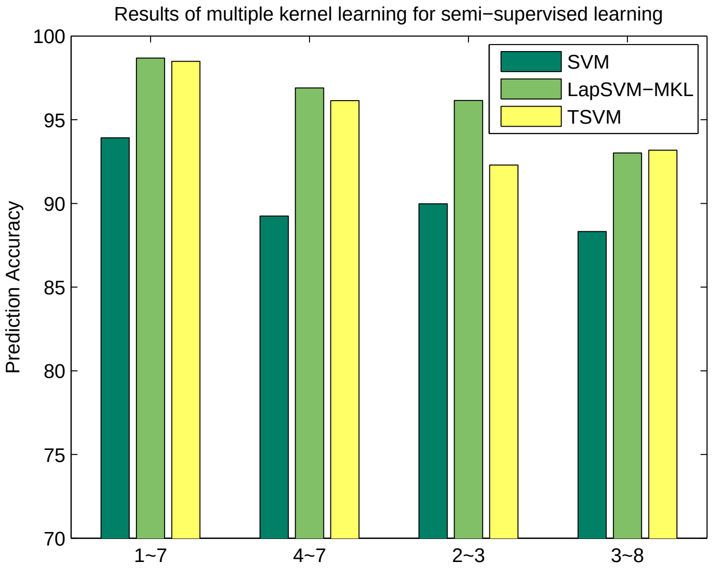
Competitive algorithms:

- baseline: SVM
- TSVM: Convex Concave Procedure (CCCP)
- LapSVM-MKL: proposed

Dataset:

- USPS (US Postal Service's handwritten digits of 400 images and 20 labelled images)

Performance comparison



Summary

Semi-supervised kernel selection

- learning graph structure and base kernels at the same time
- convex optimization
- good performance
- efficient optimization via level method

Outline

Feature selection

Feature selection

Given the number of required features, denoted by m, the goal of feature selection is to choose a subset of m features, denoted by $\mathcal{S}$, that maximizes a generalized performance criterion $\mathcal{Q}$.

Combinatorial optimization:

$$\mathcal{S}^* = \arg\max \mathcal{Q}(\mathcal{S}) \quad \text{s. t.} \quad |\mathcal{S}| = m. \tag{14}$$

How many features do we need?

The number of required features is

- dependant on learning tasks, e.g., data visualization
- dependant on computational resources, e.g., sensor networks, embedded system
- a model selection problem

We assume that an **external oracle** decides the number of selected features.

Feature selection

Feature selection criterion

- mutual information (Koller & Sahami, 1996)
- maximum margin (Weston et al., 2000; Guyon et al., 2002)
- kernel alignment (Cristianini et al., 2001; Neumann et al., 2005)
- Hilbert Schmidt independence criterion (Song et al., 2007)

Feature selection methods

SVM-based methods

Calculate weight/score **w** for each feature, and then select features with the largest weights

- L2-SVM (Vapnik, 1998; Guyon et al.,2002)
- L1-SVM (Fung & Mangasarian, 2000; Ng, 2004)
- Lasso/LARS (Tibshirani,1996; Efron et al., 2004)
- L0-SVM (Bradley & Mangasarian, 1998; Weston et al., 2003; Neumann et al., 2005; Chan et al.,2007)

Feature selection

- supervised
  - not work well when the number of labeled samples is small
- unsupervised
  - unable to identify the discriminative features
- semi-supervised
  - avoid the high cost in manually labeling data
  - exploit abundant unlabeled data

Semi-supervised feature selection

Semi-supervised feature selection based on manifold regularization

- maximum margin
  - discriminative
  - incorporating the interaction of features
- manifold regularization
  - better exploits the underlying structural information of the unlabeled data
- convex-concave optimization
  - optimality
  - efficient solver (e.g., level method)

SFS

Notations

- labeled data: $\mathbf{X}_\ell = (\mathbf{x}_1, \ldots, \mathbf{x}_l)$
- labels: $\mathbf{y} = (y_1, y_2, \ldots, y_l)$
- unlabeled examples: $\mathbf{X}_u$
- training data: $\mathbf{X} = (\mathbf{X}_\ell, \mathbf{X}_u)$
- feature indicator: $\mathbf{p} = (p_1, \ldots, p_d)^\top$ and $p_i \in \{0, 1\}$, $i = 1, \ldots, d$
- kernel matrix: $\mathbf{K}$
- kernel defined on each feature: $\mathbf{K}_i = \mathbf{x}_i \mathbf{x}_i^\top$

Semi-supervised learning

Manifold regularization

$$\|\mathbf{f}\|_I^2 = \sum_{i=1}^{n}\sum_{j=1}^{n}(f(\mathbf{x}_i) - f(\mathbf{x}_j))^2 W_{ij} = \mathbf{f}^\top \mathcal{L}\mathbf{f}, \qquad (15)$$

- W_{ij}: weights on edge $(\mathbf{x}_i, \mathbf{x}_j)$
- $D_{ii} = \sum_{j=1}^{n} W_{ij}$
- $\mathcal{L} = \mathbf{D} - \mathbf{W}$

Semi-supervised SVM based on manifold regularization

Semi-supervised SVM

$$
\begin{aligned}
\min_{\mathbf{w},b,\xi} \quad & \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{l}\xi_i + \frac{\rho}{2}\mathbf{w}^\top \mathbf{X}^\top \mathcal{L}\mathbf{X}\mathbf{w} \\
\text{s. t.} \quad & y_i(\mathbf{w}^\top \mathbf{x}_i - b) \geq 1 - \xi_i, \ i = 1, \ldots, l, \\
& \xi_i \geq 0, \ i = 1, \ldots, l,
\end{aligned}
\tag{16}
$$

- ξ: margin error
- ρ: trade-off parameter

Dual form

Semi-supervised SVM

$$\max_{\alpha \in \mathcal{Q}} \quad \alpha^\top \mathbf{e} - \frac{1}{2}(\alpha \circ \mathbf{y})^\top \mathbf{X}_\ell (\mathbf{I} + \rho \mathbf{X}^\top \mathcal{L} \mathbf{X})^{-1} \mathbf{X}_\ell^\top (\alpha \circ \mathbf{y})$$

- $\mathcal{Q} = \{\alpha \in [0, C]^l | \alpha^\top \mathbf{y} = 0\}$
- $\mathbf{I} \in \mathbb{R}^{n \times n}$: identity matrix
- $\circ$: element-wise product.

Semi-supervised feature selection

Semi-supervised feature selection

$$\min_{\mathbf{w},b,\xi,\mathbf{p}\in\mathcal{P}} \quad \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{l}\xi_i + \frac{\rho}{2}\mathbf{w}^\top \mathrm{D}(\mathbf{p})\mathbf{X}^\top \mathcal{L}\mathbf{X}\mathrm{D}(\mathbf{p})\mathbf{w} \tag{17}$$

$$\text{s. t.} \quad y_i(\mathbf{w}^\top \mathrm{D}(\mathbf{p})\mathbf{x}_i - b) \geq 1 - \xi_i,\ i = 1, \ldots, l,$$

$$\xi_i \geq 0,\ i = 1, \ldots, l,$$

- $\mathcal{P} = \{\mathbf{p} \in [0,1]^d | \mathbf{p}^\top \mathbf{e} = m\}$.
- $\mathrm{D}()$: diagonal matrix

Dual problem of semi-supervised feature selection

Semi-supervised feature selection

The dual of (17) is equivalent to the following min-max optimization problem

$$\min_{\mathbf{p}\in\mathcal{P}} \max_{\alpha\in\mathcal{Q}} \alpha^\top \mathbf{e} - \frac{1}{2}(\alpha \circ \mathbf{y})^\top \mathbf{X}_\ell \Gamma \mathbf{X}_\ell^\top (\alpha \circ \mathbf{y}) \quad (18)$$

$$\Gamma = \mathrm{D}(\mathbf{p})\left(\mathbf{I} + \rho\mathbf{Z}\right)^{-1}\mathrm{D}(\mathbf{p}) \quad (19)$$

$$\mathbf{Z} = \mathbf{X}^\top \mathcal{L} \mathbf{X} \quad (20)$$

Equivalent form

Equivalent form

$$\min_{\mathbf{p}\in\mathcal{P}} \max_{\alpha\in\mathcal{Q},\tau\in[0,1]} \alpha^\top \mathbf{e} - \frac{1}{2}(\alpha \circ \mathbf{y})^\top \mathbf{X}_\ell \mathbf{A} \mathbf{X}_\ell^\top (\alpha \circ \mathbf{y}) \quad (21)$$

- reduce the quadratic optimization of **p** to linear optimization

$$\mathbf{A} = (1-\tau)^2 \mathrm{D}(\mathbf{p}) + \frac{\tau^2}{\rho}\mathbf{Z}^{-1} \quad (22)$$

- $\mathbf{A} \succeq \Gamma$ for any $\tau \in [0, 1]$.

Connection to multiple kernel learning

Linear kernel

$$\mathbf{K} = \mathbf{X}_\ell \mathbf{X}_\ell^\top = \sum_{i=1}^{d} \mathbf{v}_i \mathbf{v}_i^\top = \sum_{i=1}^{d} \mathbf{K}_i,$$

Semi-supervised feature selection as MKL

$$\min_{\mathbf{p}\in\mathcal{P}} \max_{\alpha\in\mathcal{Q}} \quad \alpha^\top \mathbf{e} - \frac{1}{2}(\alpha \circ \mathbf{y})^\top \mathbf{M} (\alpha \circ \mathbf{y}) \tag{23}$$

- $\mathbf{M} = (1-\tau)^2 \sum_{i=1}^{d} p_i \mathbf{K}_i + \frac{\tau^2}{\rho}\mathbf{H}$.
- $\mathbf{v}_i$: i-th feature of $\mathbf{X}$
- $\mathbf{H} = \mathbf{X}_\ell^\top (\mathbf{X}^\top \mathcal{L} \mathbf{X})^{-1} \mathbf{X}_\ell$

Level method for semi-supervised feature selection

- cutting plane model

$$g^i(\mathbf{p}) = \max_{1\le j\le i} \varphi(\mathbf{p}^j, \alpha^j) + (\mathbf{p} - \mathbf{p}^j)^\top \nabla_{\mathbf{p}}(\mathbf{p}^j, \alpha^j) \tag{24}$$

- lower bound and upper bound

$$\underline{\varphi}^i = \min_{\mathbf{p}\in\mathcal{P}} g^i(\mathbf{p}), \quad \overline{\varphi}^i = \min_{1\le j\le i} \varphi(\mathbf{p}^j, \alpha^j). \tag{25}$$

- projection

$$\min_{\mathbf{p}\in\mathcal{L}^i} \|\mathbf{p} - \mathbf{p}^i\|_2^2 \tag{26}$$

- gap

$$\Delta^i = \overline{\varphi}^i - \underline{\varphi}^i. \tag{27}$$

Level method for semi-supervised feature selection

1. Initialize $\mathbf{p}^0 = \frac{m}{d}\mathbf{e}$ and $i = 0$
2. REPEAT
3. Obtain α^i by solving SVM with $\mathbf{M} = (1-\tau)^2 \mathbf{X}_\ell \mathrm{D}(\mathbf{p}^i) \mathbf{X}_\ell^\top + \frac{\tau^2}{\rho}\mathbf{H}$
4. Construct the cutting plane model $g^i(\mathbf{p})$ in (24)
5. Calculate the lower bound $\underline{\varphi}^i$ and the upper bound $\overline{\varphi}^i$ in (25), and the gap Δ^i in (27)
6. Obtain $\mathbf{p}^{i+1}$ via the projection step (26)
7. UNTIL$\Delta^i \leq \varepsilon$

Experimental setup

Comparison algorithms

- Fisher that calculates a Fisher/Correlation score for each feature (Bishop, 1995).
- L_0-appr that approximates the L_0-norm by minimizing a logarithm function (Weston et al., 2003).
- L_1-SVM that replaces L_2-norm of $\mathbf{w}$ with L_1-norm in SVM (Fung & Mangasarian, 2000).

Experimental setup

Comparison algorithms

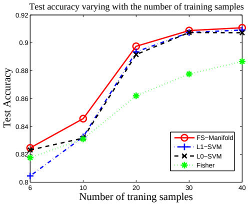
- 10% of data are employed for training
- normalize each feature to be a Gaussian distribution with zero mean and unit standard deviation, based on the training data
- C in all SVM-based feature selection methods is chosen by a 5-fold cross validation

Results on text data

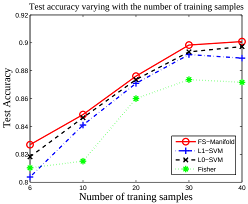
Table: The classification accuracy (%) on text data sets. The best result, and those not significantly worse than it (t-test with 95% confidence level), are highlighted.

Data	#F	FS-Manifold	L_1-SVM	L_0-SVM	Fisher
DS1	50	**82.9**±2.4	82.2±2.9	82.3±2.9	82.3±3.5
	100	**83.5**±2.2	82.9±2.6	**83.2**±2.6	**83.4**±2.6
DS2	50	**89.7**±3.9	88.7±8.6	89.1±4.9	**89.8**±6.9
	100	**91.1**±3.4	**90.9**±5.8	90.3±3.7	90.3±5.6
DS3	50	**84.2**±4.3	82.0±4.4	82.9±4.3	81.3±4.7
	100	**85.8**±3.9	84.1±4.2	85.2±4.4	84.3±4.1

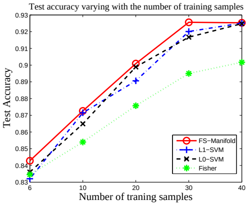
Results on USPS data



(a) *2 vs 3*



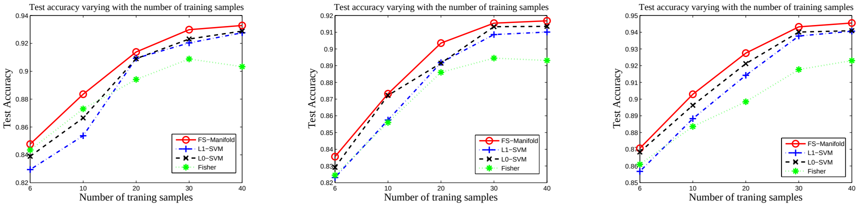
(b) *3 vs 8*



(c) *4 vs 7*

Figure: The comparison among different feature selection algorithms when the number of selected features is equal to 10.

Results on USPS data



(a) *2 vs 3* (b) *3 vs 8* (c) *4 vs 7*

Figure: The comparison among different feature selection algorithms when the number of selected features is equal to 20.

Outline

Conclusion

Presented

- a brief introduction to semi-supervised learning
- three semi-supervised methods
  - an efficient convex relaxation model for Transductive SVM
  - an effective method for semi-supervised kernel learning
  - an effective method for semi-supervised feature selection

Future topics

- when semi-supervised learning will be helpful?
- what is the connection between the low-density assumption and manifold assumption in semi-supervised learning?
- how to obtain or better approximate the optimal solution of semi-supervised models?

Recent publications of our lab in machine learning

- Conference papers
  1. Z. Xu, R. Jin, J. Ye, I. King, and M. R. Lyu. Non-monotonic feature selection, *ICML 2009*.
  2. Z. Xu, R. Jin, M. R. Lyu, and I. King. Semi-supervised Feature Selection via Manifold Regularization. *IJCAI 2009*.
  3. Z. Xu, R. Jin, I. King, and M. R. Lyu, An Extended Level Method for Multiple Kernel Learning, *NIPS 2008*.
  4. Z. Xu, R. Jin, K. Huang, I. King, and M. R. Lyu. Semi-supervised text categorization by active search, *CIKM 2008*.
  5. K. Huang, Z. Xu, I. King, and Michael R. Lyu, Semi-supervised Learning from General Unlabeled Data, *ICDM 2008*.
  6. H. Yang, I. King, and M. Lyu, Learning with Consistency between Inductive Functions and Kernels, *NIPS*, 2008
  7. Z. Xu, R. Jin, J. Zhu, I. King, and M. R. Lyu. Efficient convex relaxation for transductive support vector machine, *NIPS 2007*.

Publications of our lab in machine learning

- Journal papers
  1. Z. Xu, K. Huang, J. Zhu, I. King, and M. R. Lyu. A Novel Kernel-based Maximum A Posteriori Classification Method. *Neural Networks*, 2009.
  2. K. Huang, Z. Xu, I. King, M. R. Lyu, and Z. Zhou, A Novel Discriminative Naive Bayesian Network for Classification, in Bayesian Network Technologies: Applications and Graphical Models, 2007.
  3. Z. Xu, I. King, and M. R. Lyu, Feature Selection Based on Minimum Error Minimax Probability Machine, IJPRAI, 2007.
  4. K. Huang, D. Zheng, I. King, and M. R. Lyu, Arbitrary Norm Support Vector Machines, Neural Computation, 2009.
  5. K. Huang, H. Yang, M. R. Lyu, and I. King, Maxi-Min Margin Machine: Learning Large Margin Classifiers Locally and Globally, IEEE Transactions on Neural Networks, 2008.

References

1. O. Chapelle, M. Chi, and A. Zien. A continuation method for semi-supervised SVMs, ICML 2006.
2. O. Chapelle, B. Schölkopf, and A. Zien. Semi-Supervised Learning. MIT Press, Cambridge, MA, 2006.
3. O. Chapelle, V. Sindhwani, and S. Keerthi. Branch and bound for semi-supervised support vector machines, NIPS 2006.
4. O. Chapelle and A. Zien. Semi-supervised classification by low density separation. Tenth International Workshop on Artificial Intelligence and Statistics, 2005.
5. R. Collobert, F. Sinz, J. Weston, and L. Bottou. Large scale transductive SVMs. Journal of Machine Learning Reseaerch, 2006.
6. T. Joachims. Transductive inference for text classification using support vector machines, ICML 1999.
7. G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. Journal of Machine Learning Research, 2004.
8. L. Xu and D. Schuurmans. Unsupervised and semi-supervised multi-class support vector machines. AAAI 2005.
9. X. Zhu. Semi-supervised learning literature survey. Technical report, Computer Sciences, University of Wisconsin-Madison, 2005.
10. X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions, ICML 2003

References

1. G. Fung and O. L. Mangasarian. Data selection for support vector machine classifiers. KDD 2000
2. I. Guyon and A. Elisseeff. An introduction to variable and feature selection. Journal of Machine Learning Research, 2003
3. I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. Machine Learning, 2002
4. D. Koller and M. Sahami. Toward optimal feature selection. ICML 1996
5. A. Rakotomamonjy. Variable selection using svm-based criteria. Journal of Machine Learning Research, 2003
6. Y. Ng. Feature selection, L1 vs. L2 regularization, and rotational invariance. ICML 2004
7. L. Song, A. Smola, A. Gretton, K. M. Borgwardt, and J. Bedo. Supervised feature selection via dependence estimation. ICML 2007
8. J. Weston, A. Elisseeff, B. Scholkopf, and M. Tipping. Use of the zero norm with linear models and kernel methods. Journal of Machine Learning Research, 2003
9. J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature selection for SVMs. NIPS 2000
10. Z. Zhao and H. Liu, Semi-supervised Feature Selection via Spectral Analysis, SDM 2007

QA

Thanks for your attention!