Learning with Unlabeled Data

Zenglin Xu
Supervisors: Irwin King, Michael R. Lyu

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

November 17th, 2008

Outline

Outline

Machine Learning

- Learning from labeled data
  - Supervised learning
- Learning from unlabeled data
  - Unsupervised learning
- Learning from labeled and unlabeled data
  - Semi-supervised learning (SSL)
  - Self-taught learning
  - Learning with Universum

Semi-supervised learning and unlabeled data

Semi-supervised learning

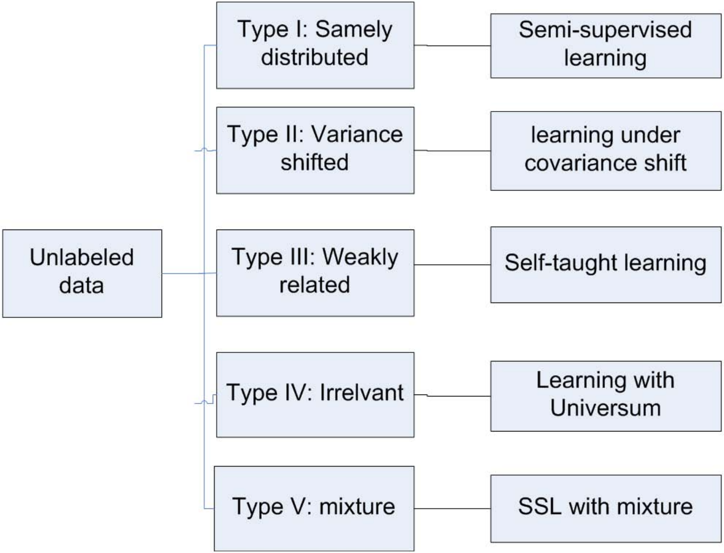
Unlabeled data and labeled data are assumed to be generated from the same distribution.

Unlabeled data

- Are not necessarily generated from the same distribution as labeled data
- May be from other tasks
- May be irrelevant

In this thesis, unlabeled data has a more general meaning than that in semi-supervised learning.

Types of unlabeled data



Types of unlabeled data (I)

Labeled


Unlabeled


same-distribution

- Unlabeled data and labeled data are drawn from the same distribution
- Share the same label
- Semi-supervised learning
- Manifold assumption or low density assumption
- E.g., Transductive Support Vector Machine (TSVM)
- Survey: [zhu, 2005], [Chapelle et al., 2006]

Types of unlabeled data (II)

Labeled

Unlabeled

Variance-shifted

- Drawn from a variance-drifted distribution
- Share the same label with labeled data
- Learning under covariance shift or sample bias correction
- E.g., [Shimodaira et al., 2000], [Zadrozny et al., 2004]

Types of unlabeled data (III)

Labeled

Unlabeled

Weakly-related

- Share no common labels with labeled data
- Structurally related
- Self-taught learning: transfer learning from unlabeled data
- E.g., [Raina et al., 2007]

Types of unlabeled data (IV)

Labeled


Unlabeled

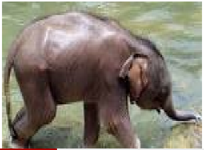

Irrelevant

- Unlabeled data are irrelevant data or background data
- Share no common labels
- Learning with universum
- E.g., [Weston et al., 2006]

Types of unlabeled data (V)

Labeled

Unlabeled



Mixture

- Mixture of two or more types of unlabeled data
- Relevant mixed with others
- Semi-supervised learning from a mixture
- E.g., [Zhang et al., 2008], [Huang et al., 2008]

Challenging issues in learning with unlabeled data

Challenges

- How to learn an efficient Convex relaxation for TSVM?
- How to efficiently learn a kernel?
- What is the relationships between the assumptions of semi-supervised learning?

Challenging issues in learning with unlabeled data

Challenges

- How to learn an efficient Convex relaxation for TSVM?
- How to efficiently learn a kernel?
- What is the relationships between the assumptions of semi-supervised learning?

Contributions

- An efficient convex relaxation model for Transductive SVM (NIPS 2007) (Chapter 3)
- An efficient method for multiple kernel learning (NIPS 2008) (Chapter 4)
- A unified framework for assumptions in semi-supervised learning (Chapter 5)

Challenging issues in learning with unlabeled data

Challenges

- How to better utilize the weakly-related unlabeled data?
- How to learn a model when irrelevant data are mixed with relevant data?
- How to actively find unlabeled data if they are not given?

Challenging issues in learning with unlabeled data

Challenges

- How to better utilize the weakly-related unlabeled data?
- How to learn a model when irrelevant data are mixed with relevant data?
- How to actively find unlabeled data if they are not given?

Contributions

- A supervised self-taught learning (SSTL) model that can deal with weakly-related unlabeled data (Chapter 6)
- A framework for learning with a mixture of relevant and irrelevant unlabeled data (ICDM 2008) (Chapter 7)
- A framework for semi-supervised text categorization that actively retrieves unlabeled documents from the Internet (CIKM 2008) (Chapter 8)

Presented topics

Topics

1. An efficient convex relaxation model for Transductive SVM (NIPS 2007)

Presented topics

Topics

1. An efficient convex relaxation model for Transductive SVM (NIPS 2007)
2. An efficient method for multiple kernel learning (NIPS 2008)

Presented topics

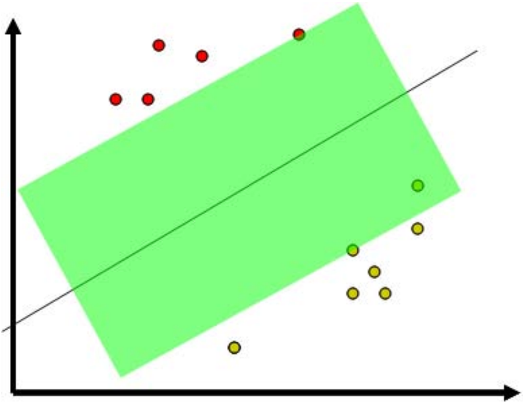
Topics

1. An efficient convex relaxation model for Transductive SVM (NIPS 2007)
2. An efficient method for multiple kernel learning (NIPS 2008)
3. A framework for semi-supervised text categorization that actively retrieves unlabeled documents from the Internet (CIKM 2008)

Outline

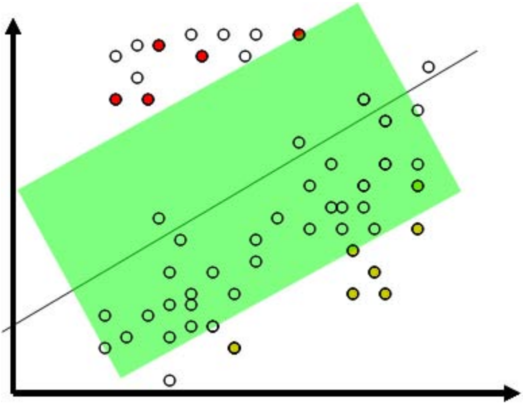
Transductive SVM

- SVM



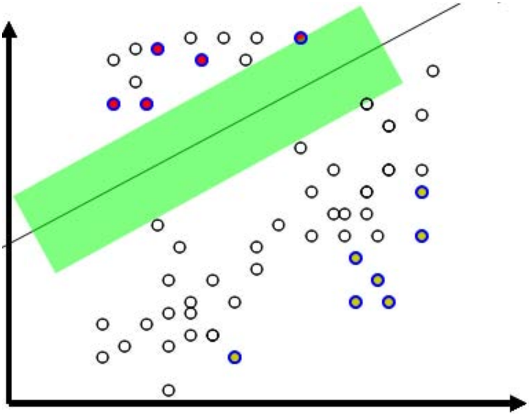
Transductive SVM

- SVM
- SVM with unlabeled data



Transductive SVM

- SVM
- SVM with unlabeled data
- Transductive SVM



Transductive SVM

TSVM: label $\mathbf{y}$ as a free variable

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

- $\{\mathbf{x}_i\}_{i=1}^n$: training data, l labeled, $n - l$ unlabeled
- $f = \mathbf{w}^\top\mathbf{x} - b$: decision function
- ξ: margin error
- C: tradeoff parameter

Primal form of TSVM

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

$$\min_{\mathbf{y}\in\{-1,+1\}^n,t,\nu,\delta,\lambda} \quad t \tag{2}$$

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

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

- $\mathbf{K}$: kernel matrix
- $\circ$: element-wise product; $\succeq$: $positivesemi - definite$
- $\mathbf{e}$: vector of all ones
- $\nu \in \mathbb{R}^n$: $\alpha \geq 0$
- $\delta \in \mathbb{R}^n$: $\alpha \leq C$
- λ: $\alpha^\top \mathbf{y} = 0$

Convex Relaxation of TSVM

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

Convex Relaxation of TSVM

$$
\begin{aligned}
\min_{\mathbf{M},t,\nu,\delta,\lambda} \quad & t \\
\text{s. t.} \quad & \begin{pmatrix} \mathbf{M} \circ \mathbf{K} & \mathbf{e} + \nu - \delta \\ (\mathbf{e} + \nu - \delta)^\top & t - 2C\delta^\top \mathbf{e} \end{pmatrix} \succeq 0 \\
& \nu \geq 0, \; \delta \geq 0, \\
& \mathbf{M} \succeq 0, \; M_{i,i} = 1, \; i = 1, 2, \ldots, n, \\
& M_{ij} = y_i^\ell y_j^\ell, \; 1 \leq i, j \leq l
\end{aligned} \tag{3}
$$

- y_i^ℓ, $i = 1, \ldots, l$: labels of labeled data

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 $\mathbf{y}^\top \mathbf{y}$
    - Not tight approximation

Our solution

TSVM in the dual form:

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

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

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

Our solution

$$
\begin{aligned}
\min_{\mathbf{w}} \quad & \mathbf{w}^\top \mathbf{P}^\top \mathbf{K}^{-1} \mathbf{P} \mathbf{w} \\
\text{s. t.} \quad & y_i^\ell w_i \geq 1,\ i = 1, 2, \ldots, l, \\
& w_i^2 \geq 1,\ i = l+1, l+2, \ldots, n, \\
& -\epsilon \leq \frac{1}{l} \sum_{i=1}^{l} w_i - \frac{1}{n-l} \sum_{i=l+1}^{n} w_i \leq \epsilon.
\end{aligned} \tag{4}
$$

- $\mathbf{w} = (\mathbf{z}, \lambda) \in \mathbb{R}^{n+1}$
- $\mathbf{P} = (\mathbf{I}_n, \mathbf{e}) \in \mathbb{R}^{n \times (n+1)}$
- $-\epsilon \leq \frac{1}{l} \sum_{i=1}^{l} w_i - \frac{1}{n-l} \sum_{i=l+1}^{n} w_i \leq \epsilon$: balance constraint

Our solution

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

- $\mathbf{a} = (\mathbf{y}^l, \mathbf{0}^{n-l}, 0) \in \mathbb{R}^{n+1}$
- $\mathbf{b} = (\mathbf{0}^l, \mathbf{1}^{n-l}, 0) \in \mathbb{R}^{n+1}$
- $\mathbf{c} = (\frac{1}{l}\mathbf{1}^l, -\frac{1}{u}\mathbf{1}^{n-l}, 0) \in \mathbb{R}^{n+1}$
- $\mathbf{A} = \mathbf{P}^\top \mathbf{K}^{-1} \mathbf{P}$
- $$\begin{aligned} \gamma = \arg\max_{\gamma, t} \quad & -\frac{1}{4}t + \sum_{i=1}^{n} \gamma_i - \epsilon(\alpha + \beta) \\ s.\ t. \quad & \begin{pmatrix} \mathbf{A} - \mathcal{D}(\gamma \circ \mathbf{b}) & \gamma \circ \mathbf{a} - (\alpha - \beta)\mathbf{c}, \\ (\gamma \circ \mathbf{a} - (\alpha - \beta)\mathbf{c})^\top & t \end{pmatrix} \succeq 0 \\ & \alpha \geq 0,\ \beta \geq 0,\ \gamma_i \geq 0,\ i = 1, 2, \ldots, n. \end{aligned}$$

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: it implements the conjugate of conjugate of the prediction function $f(\mathbf{x})$, which is the convex envelope of $f(\mathbf{x})$ [Hiriart et al., 1993].
- Related to the solution of the harmonic functions [Zhu et al., 2003]:

$$\mathbf{z} = \left(\mathbf{I}_n - \sum_{i=l+1}^{n} \gamma_i \mathbf{K} \mathbf{I}_n^i \right)^{-1} \left(\sum_{i=1}^{l} \gamma_i y_i^\ell \mathbf{K}(\mathbf{x}_i, \cdot) \right) \qquad (5)$$

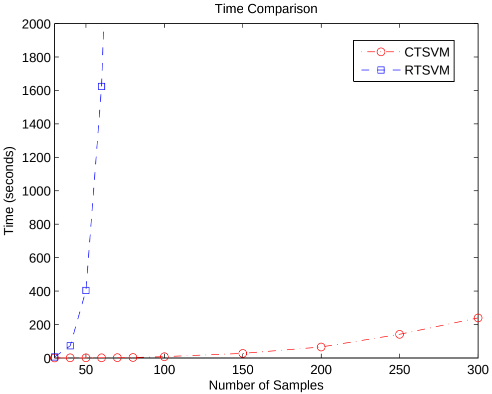
Data sets

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

Computation time comparison

- CTSVM: proposed [Xu et al., 2007]
- RTSVM: previous [Xu & Schuurmans, 2004]



- Course, labeled 20

Accuracy comparison

Table: The classification performance of Transductive SVMs on benchmark data sets.

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

Multiple kernel kearning (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 $\mathcal{P}$

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

Multiple kernel learning (MKL)

A generic approach to kernel learning

Typical applications of multiple kernel learning

- Multi-source data fusion (web classification, genome fusion)
- Image annotation
- Near duplicate frame detection in video
- Novelty detection

Multiple kernel learning

Multiple kernel learning

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

Properties

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

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

Available optimization methods for MKL

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

- Semi-definite Programming (SDP) [Lanckriet et al., 2004]: small scale
- Quadratically Constrained Quadratic Programming (QCQP) [Bach et al., 2004]: medium scale

Available optimization methods for MKL

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

- Semi-definite Programming (SDP) [Lanckriet et al., 2004]: small scale
- Quadratically Constrained Quadratic Programming (QCQP) [Bach et al., 2004]: medium scale
- Semi-Infinite Linear Programming (SILP) [Sonnenburg et al., 2006] : large scale
- Subgradient Descent (SD) [Rakotomamonjy et al., 2008] : large scale

A general framework for solving large-scale MKL

Convex-concave optimization

1. Initialize $\mathbf{p}^0 = \mathbf{e}/m$ and $i = 0$
2. *REPEAT*
3. Solve dual SVM with kernel $\mathbf{K} = \sum_{j=1}^{m} p_j^i \mathbf{K}_j$ for α^i
4. Update kernel weights by $\mathbf{p}^{i+1} = \arg\min\{f^i(\mathbf{p}) : \mathbf{p} \in \mathcal{P}\}$
5. Update $i = i + 1$ and calculate stopping criterion Δ^i
6. *UNTIL* $\Delta^i \leq \varepsilon$

A general framework for solving large-scale MKL

Convex-concave optimization

1. Initialize $\mathbf{p}^0 = \mathbf{e}/m$ and $i = 0$
2. *REPEAT*
3. Solve dual SVM with kernel $\mathbf{K} = \sum_{j=1}^{m} p_j^i \mathbf{K}_j$ for α^i
4. Update kernel weights by $\mathbf{p}^{i+1} = \arg\min\{f^i(\mathbf{p}) : \mathbf{p} \in \mathcal{P}\}$
5. Update $i = i + 1$ and calculate stopping criterion Δ^i
6. *UNTIL* $\Delta^i \leq \varepsilon$

- Methods differ in $f^i(\mathbf{p})$

Semi-Infinite Linear Programming (SILP) for MKL

$$f^i_{SILP}(\mathbf{p}) = \min_{\nu} \left\{ \nu : \nu \geq f(\mathbf{p}^j, \alpha^j) + (\mathbf{p} - \mathbf{p}^j)^\top \nabla_{\mathbf{p}} f(\mathbf{p}, \alpha^j), j = 0, \ldots, i \right\}$$

$f_{SILP}(\mathbf{p})$ is a cutting plane model

Pros and Cons

- Pro: utilize all $\{\mathbf{p}^j, \alpha^j\}_{j=0}^i$ obtained so far
- Con: inaccurate when $\mathbf{p}$ is far from $\{\mathbf{p}^j\}_{j=1}^i$ $\rightarrow$ oscillating solutions

Subgradient descent method (SD) for MKL

$$f_{SD}^{i}(\mathbf{p}) = \frac{1}{2}\|\mathbf{p} - \mathbf{p}^i\|_2^2 + \gamma_i(\mathbf{p} - \mathbf{p}^i)^\top \nabla_{\mathbf{p}} f(\mathbf{p}, \alpha^i)$$

Pros and Cons

- Pro: regularize by $\|\mathbf{p} - \mathbf{p}^i\|_2^2$, preventing $\mathbf{p}$ far from $\mathbf{p}^i$
- Con: only utilize the current solution $(\mathbf{p}^i, \alpha^i)$.
  - Require line search to determine optimal step size γ_i
  - Computationally expensive for convex-concave

Expected properties

Combining the strengths of SILP and SD

- Utilize all $\{(\mathbf{p}^j, \alpha^j)\}_{j=0}^{i}$ of previous solutions
- Keep the new solution not far from the current one $\mathbf{p}^i$

Expected properties

Combining the strengths of SILP and SD

- Utilize all $\{(\mathbf{p}^j, \alpha^j)\}_{j=0}^{i}$ of previous solutions
- Keep the new solution not far from the current one $\mathbf{p}^i$

$\Downarrow$

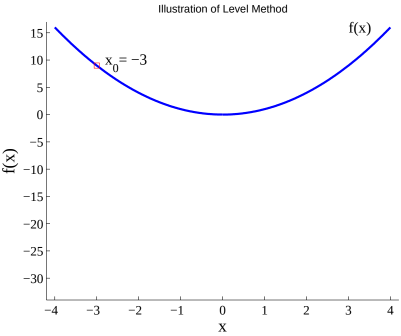
Level method

- Utilize all $\{(\mathbf{p}^j, \alpha^j)\}_{j=1}^{i}$ via constructing cutting plane models
- Adjust the new solution via projecting to level sets

Level Method

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

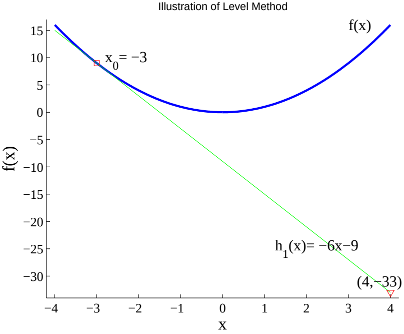
- Initialization: $x_0 = -3$, $\lambda = 0.9$



Level Method

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

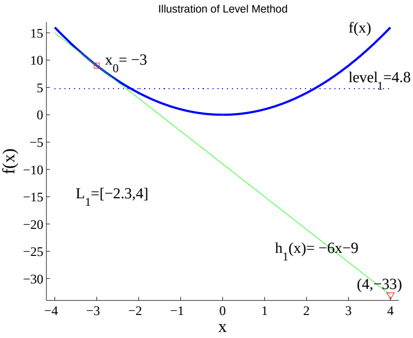
- Initialization: $x_0 = -3$, $\lambda = 0.9$
- Construct a cutting plane model $g_1(x)$



Level Method

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

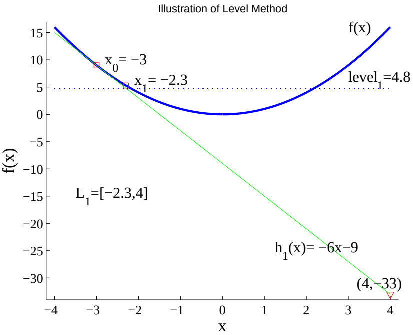
- Initialization: $x_0 = -3$, $\lambda = 0.9$
- Construct a cutting plane model $g_1(x)$
- Construct a level set $\mathcal{L}_1$
  $\text{level}_1 = \lambda \times f(x_0) + (1-\lambda) \times (-33)$
  $\mathcal{L}_1 = \{x \in \mathcal{X} : g_1(x) \le \text{level}_1\}$



Level Method

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

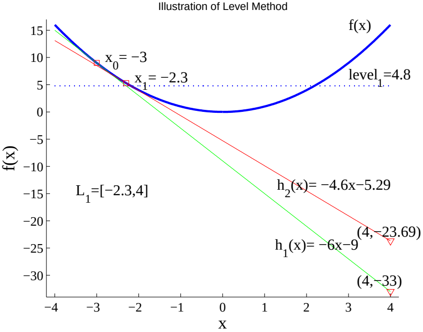
- Initialization: $x_0 = -3$, $\lambda = 0.9$
- Construct a cutting plane model $g_1(x)$
- Construct a level set $\mathcal{L}_1$
  $\text{level}_1 = \lambda \times f(x_0) + (1 - \lambda) \times (-33)$
  $\mathcal{L}_1 = \{x \in \mathcal{X} : g_1(x) \leq \text{level}_1\}$
- Project x_0 to level set $\mathcal{L}_1$,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



Level method

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

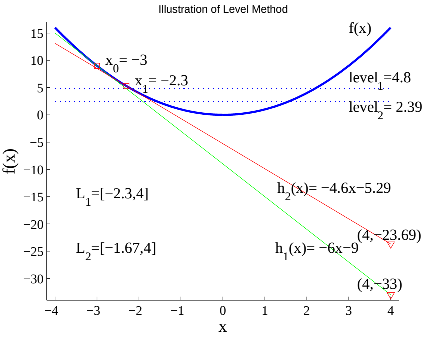
- Construct a new cutting plane model $g_2(x) = \min_x h_i(x)$



Level method

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

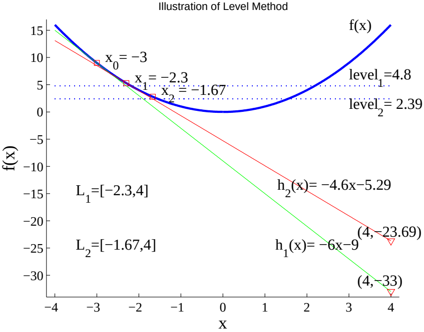
- Construct a new cutting plane model $g_2(x) = \min_{x} h_i(x)$
- Construct a new level set $\mathcal{L}_2$



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 $\mathcal{L}_2$



Key steps of level method for MKL

1. Build a cutting plane model

Key steps of level method for MKL

1. Build a cutting plane model
2. Construct a level set
   - Obtain an auxiliary solution by minimizing the cutting plane model
   - Estimate the lower and upper bounds for the optimal value of MKL
   - Compute the level value using the lower and upper bounds

Key steps of level method for MKL

1. Build a cutting plane model
2. Construct a level set
   - Obtain an auxiliary solution by minimizing the cutting plane model
   - Estimate the lower and upper bounds for the optimal value of MKL
   - Compute the level value using the lower and upper bounds
3. Obtain the new solution by projecting the existing solution to the level set

Cutting Plane Models

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

Proposition

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

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

Theorem

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

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

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 SILP method

Projection to level set

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

- Solve by efficient Quadratic Programming (QP)
  - Improve by using other distance metrics (e.g., L_1 norm)
- Projection ensures that the new solution $\mathbf{p}^{i+1}$ is close to $\mathbf{p}^i$
- The level set ensures a significant progress

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 $\mathbf{K} = \sum_{j=1}^{m} p_j^i \mathbf{K}_j$ 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 $\mathbf{K} = \sum_{j=1}^{m} p_j^i \mathbf{K}_j$ 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 $\mathbf{K} = \sum_{j=1}^{m} p_j^i \mathbf{K}_j$ 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 $\mathbf{K} = \sum_{j=1}^{m} p_j^i \mathbf{K}_j$ 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 $\mathbf{K} = \sum_{j=1}^{m} p_j^i \mathbf{K}_j$ 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 $\mathbf{K} = \sum_{j=1}^{m} p_j^i \mathbf{K}_j$ 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$\Delta^i \leq \varepsilon$

Convergence rate

Theorem

To obtain a solution $\mathbf{p}$ *that satisfies the stopping criterion, i.e.,*

$$|\max_{\alpha \in \mathcal{Q}} f(\mathbf{p}, \alpha) - f(\mathbf{p}^*, \alpha^*)| \leq \varepsilon,$$

the maximum number of iterations N *that the level method requires is bounded as follows*

$$N \leq \frac{2c(\lambda)L^2}{\varepsilon^2},$$

where $c(\lambda) = \frac{1}{(1-\lambda)^2\lambda(2-\lambda)}$ *and* $L = \frac{1}{2}\sqrt{m}nC^2 \max_{1 \leq i \leq m} \Lambda_{\max}(\mathbf{K}_i)$. $\Lambda_{\max}(M)$ *computes the maximum eigenvalue of matrix* M.

Convergence rate

- According to Information Based Complexity (IBC) theory, $\mathcal{O}(1/\varepsilon^2)$ is almost the optimal worst-case convergence rate when the optimization method is based on a black box first order oracle [Nemirovsky, 1983; Lemarechal, 1995]
- Real performance is usually far better

Experimental setup

- Base kernel matrices ([Rakotomamonjy et. al, 2008])
  - Gaussian kernels with 10 different widths ($\{2^{-3}, 2^{-2}, \dots, 2^{6}\}$) on all features and on each single feature
  - Polynomial kernels of degree 1 to 3 on all features and on each single feature.
- C set to be 100 for all experiments
- λ: initial value 0.9, increased to 0.99 when $\Delta_i/\ell_i \leq 0.01$
  - A larger λ accelerates the projection near to the convergence
- Stopping criterion
  - Duality gap ([Rakotomamonjy et. al, 2008])

Performance comparison

Table: n: number of training data, m: number of kernels.

	SD	SILP	Level
	Iono	$n = 175$	$m = 442$
Time(s)	33.5 ±11.6	1161.0 ±344.2	7.1 ±4.3
Accuracy (%)	92.1 ±2.0	92.0 ±1.9	92.1±1.9
#Kernel	26.9 ±4.0	24.4 ±3.4	25.4±3.9
	Breast	$n = 342$	$m = 117$
Time(s)	47.4 ±8.9	54.2 ±9.4	4.6 ±1.0
Accuracy (%)	96.6 ±0.9	96.6 ±0.8	96.6±0.8
#Kernel	13.1 ±1.7	10.6 ±1.1	13.3±1.5
	Pima	$n = 384$	$m = 117$
Time(s)	39.4 ±8.8	62.0 ±15.2	9.1 ±1.6
Accuracy (%)	76.9 ±1.9	76.9 ±2.1	76.9±2.1
#Kernel	16.6 ±2.2	12.0 ±1.8	17.6±2.6

Time-saving ratio

Table: Time-saving ratio(%) of the level method over the SILP and the SD method

	Iono	Breast	Pima	Sonar	Wpbc	Heart	Vote	Wdbc	Average
$\frac{SD-Level}{SD}$	78.9	90.4	77.0	58.7	32.5	54.7	82.8	87.4	70.3
$\frac{SILP-Level}{SILP}$	99.4	91.6	85.4	98.7	88.7	97.3	84.5	89.4	91.9

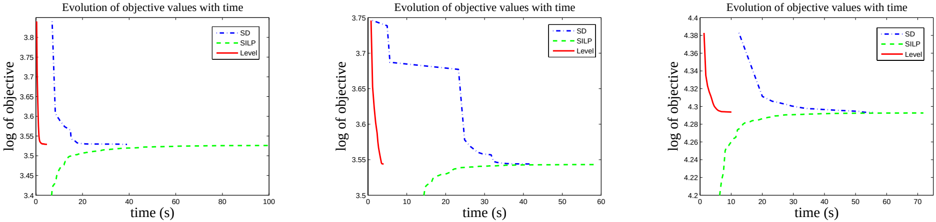
Experimental setup: semi-supervised setting

- 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,
  - linear kernel on each single feature.
- Graphs: 20 NN, cosine similarity
- Point-cloud-norm: [Sindhwani et al., 2005]
- Other settings similar to the supervised setting

Semi-supervised settings

	SD	SILP	Level
		1 vs 7	
Time(s)	13.7±10.7	511.6±698.9	**2.7**±1.1
Accuracy (%)	96.2±4.1	94.6±9.1	**96.5** ±3.6
#Kernel	8.4±2.8	7.2±2.7	9.4±2.8
		2 vs 3	
Time(s)	17.0± 27.8	1362.0±611.4	**2.4**±1.4
Accuracy (%)	86.9±2.9	86.9±3.1	**87.2**±3.0
#Kernel	13.1±2.9	11.7±1.9	14.4±2.9
		2 vs 7	
Time(s)	16.3±10.5	1249.5±684.3	**2.5**±1.0
Accuracy (%)	88.3±3.9	88.1±4.0	**88.6**±3.8
#Kernel	12.4±2.4	10.2±1.9	13.4± 2.9
		3 vs 8	
Time(s)	11.6±9.8	990.0±726.1	**2.4**±1.3
Accuracy (%)	85.4±4.5	85.5±4.6	**85.8**±4.5
#Kernel	13.6±2.6	11.7±1.7	14.7±2.5
		4 vs 7	
Time(s)	13.6±9.2	671.8±682.2	**1.7**±0.7
Accuracy (%)	86.9±5.7	87.0±5.6	**87.2**±5.8
#Kernel	11.3±2.0	9.9±1.6	13.2±2.7

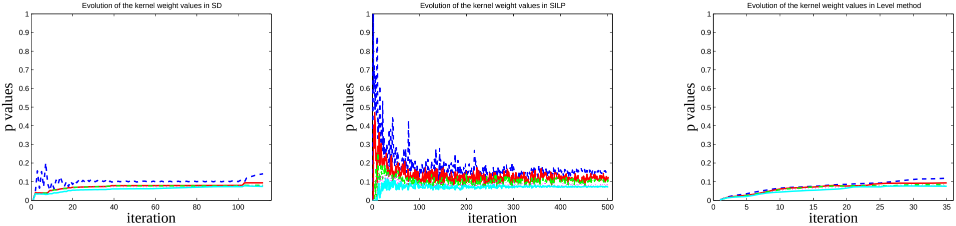
Objective evolution curves



(a) *Iono* (b) *Breast* (c) *Pima*

Figure: Evolution of objective values over time (seconds).

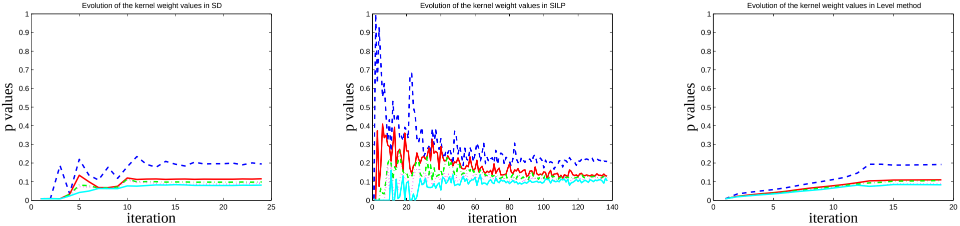
Kernel weights evolution curves for “Iono”



(a) *Iono/SD* (b) *Iono/SILP* (c) *Iono/Level*

Figure: The evolution curves of the five largest kernel weights for “Iono”

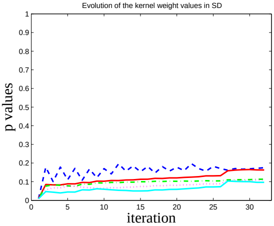
Kernel weights evolution curves for “Breast”



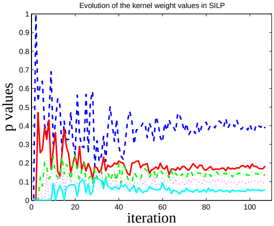
(d) *Breast/SD* (e) *Breast/SILP* (f) *Breast/Level*

Figure: The evolution curves of the five largest kernel weights for “Breast”

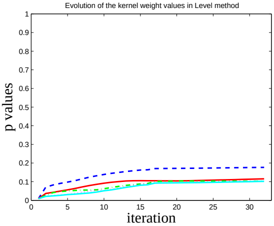
Kernel weights evolution curves for “Pima”



(g) *Pima/SD*



(h) *Pima/SILP*



(i) *Pima/Level*

Figure: The evolution curves of the five largest kernel weights for “Pima”

Analysis

- SILP
  - High computational cost due to the oscillation of solutions

Analysis

- SILP
  - High computational cost due to the oscillation of solutions
- SD
  - A large number of calls to SVM are required to compute the optimal step size via a line search
  - e.g., for "iono", 1231 times of calling to SVM for SD, while 47 for level method

Analysis

- SILP
  - High computational cost due to the oscillation of solutions
- SD
  - A large number of calls to SVM are required to compute the optimal step size via a line search
  - e.g., for "iono", 1231 times of calling to SVM for SD, while 47 for level method
- Level method
  - The cutting plane model utilizes the computational results of all iterations
  - The projection to level sets ensures the stability of solutions

Summary

- We propose an extended level method to efficiently solve the multiple kernel learning problem
- It utilizes the gradients of all the solutions that are obtained in past iterations
- It introduces a projection step to regularize the updated solution
- It saves on average 91.9% of computational time over the SILP method and 70.3% over the SD method.

Outline

Automated text categorization

Figure: Text categorization

Problems in automated text categorization

- bottleneck : sufficient numbers of labeled documents are expensive to collect

Problems in automated text categorization

- bottleneck : sufficient numbers of labeled documents are expensive to collect
- solution : exploiting unlabeled documents by so-called semi-supervised learning methods

What could we do when only a small amount of labeled documents are available?

Problems in automated text categorization

- bottleneck : sufficient numbers of labeled documents are expensive to collect
- solution : exploiting unlabeled documents by so-called semi-supervised learning methods

What could we do when only a small amount of labeled documents are available?

This study

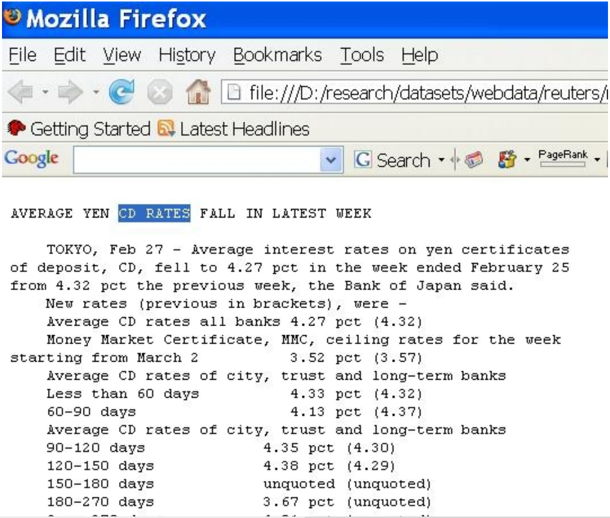
answers the questions:

- How to **collect** a multitude of unlabeled documents?
- How to **use** the unlabeled documents? (They might be in poor quality)

Collecting unlabeled data

One way to collect the unlabeled documents is through the web search engines.

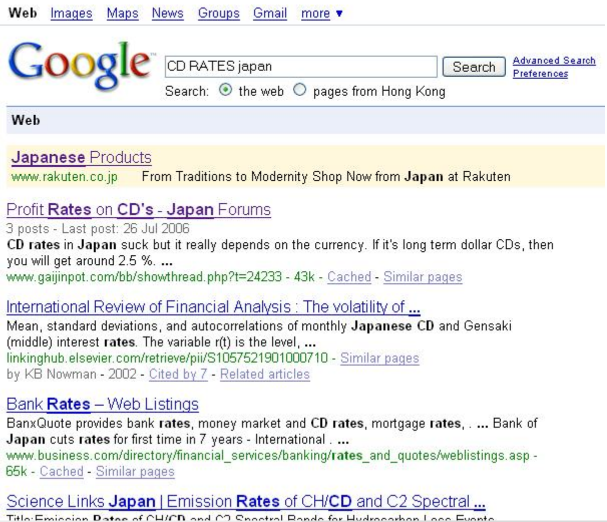
- Extract the keyword (query word)



Collecting unlabeled data

One way to collect the unlabeled documents is through the web search engines.

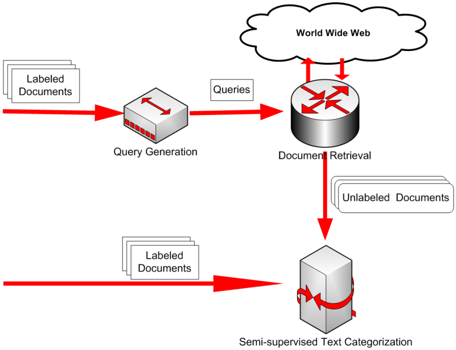
- Extract the keyword (query word)
- Retrieval the Internet



Semi-supervised text categorization framework

Framework

1. Query generation
2. Document retrieval
3. Semi-supervised text categorization



Query generation

Problems

- sparseness of words
- unrelated query words

$$\min_{\mathbf{w},\xi} \quad \sum_{j \in V_i} w_j + C \sum_{k=1}^{n_l} \xi_k \tag{6}$$

$$\text{s. t.} \quad y_k \left(\sum_{j \in V_i} w_j x_{k,j} + b \right) \geq 1 - \xi_k, \xi_k \geq 0, k = 1, \ldots, n_l \ ,$$

$$w_j \geq 0, \ \forall j \ , \quad w_j = 0, \ \forall j \notin V_i .$$

- Each document $\mathbf{x}_i$ generates a query q_i
- w: importance of a query word, ξ: classification error
- Word features with large weights will be selected to form a query.

Semi-supervised text categorization

- Auxiliary approach
  - All the unlabeled documents U_i (retrieved by q_i) share the same category label as x_i
  - Label vector $y*$ for retrieved data is not a free variable

Auxiliary approach

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \lambda\|\mathbf{w}\|_2^2 + \sum_{\mathbf{x}_i\in\mathcal{D}} \xi_i + \gamma \sum_{\mathbf{x}_j\in\mathcal{U}} \xi_j \\ \text{s. t.} \quad & y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \forall i\ \mathbf{x}_i \in \mathcal{D}\ , \\ & y_j^*(\mathbf{w}^\top\mathbf{x}_j + b) \geq 1 - \xi_j, \quad \forall j\ \mathbf{x}_j \in \mathcal{U}\ , \end{aligned} \tag{7}$$

Semi-supervised text categorization

- Semi-supervised approach
  - Does not assume any relationship between the class labels assigned to U_i and the class label of x_i
  - Label vector $y*$ for retrieved data is regarded as an optimization variable

Semi-supervised approach

$$
\begin{aligned}
\min_{\mathbf{w},b,\mathbf{y}^*} \quad & \lambda\|\mathbf{w}\|_2^2 + \sum_{\mathbf{x}_i\in\mathcal{D}} \xi_i + \gamma \sum_{\mathbf{x}_j\in\mathcal{U}} \xi_j \ , \\
\text{s. t.} \quad & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \forall i \ \mathbf{x}_i \in \mathcal{D} \ , \\
& y_j^*(\mathbf{w}^\top \mathbf{x}_j + b) \geq 1 - \xi_j, \quad \forall j \ \mathbf{x}_j \in \mathcal{U} \ ,
\end{aligned} \tag{8}
$$

Semi-supervised text categorization

Solving method:

- Auxiliary approach
  - SMO
- Semi-supervised approach
  - Convex-concave procedure (CCCP)

Convex-concave procedure

$$J_s(h) = \lambda\|\mathbf{w}\|_2^2 + \sum_{\mathbf{x}_i \in \mathcal{D}} \max(0, 1 - h(\mathbf{x}_i) y_i) + \gamma \sum_{\mathbf{x}_j \in \mathcal{U}} \left(L_s(h(\mathbf{x}_j), +1) + L_s(h(\mathbf{x}_j), -1)\right) .$$

- L_s: Ramp loss
- h: decision function

Experimental results

Table: The classification accuracy (%) of text categorization

Data set	SVM	Auxi-SVM	Semi-SVM
male vs. female	47.6	**76.1**	73.1
bacterial vs. virus	61.8	77.6	**78.3**
musculo vs. digestive	69.9	71.3	**77.0**
fourDisease	31.6	38.4	**58.0**
ship vs. trade	94.1	95.5	**95.9**
corn vs. wheat	69.2	69.0	**71.6**
money vs. trade	80.6	88.8	**88.9**
auto vs. motor	59.4	69.1	**69.2**
sci	35.5	56.1	**56.8**
average	61.1	71.3	**74.3**

Error reduction:

- 26.3% for Auxi-SVM
- 34.0% for Semi-SVM

Comparison among different search engines

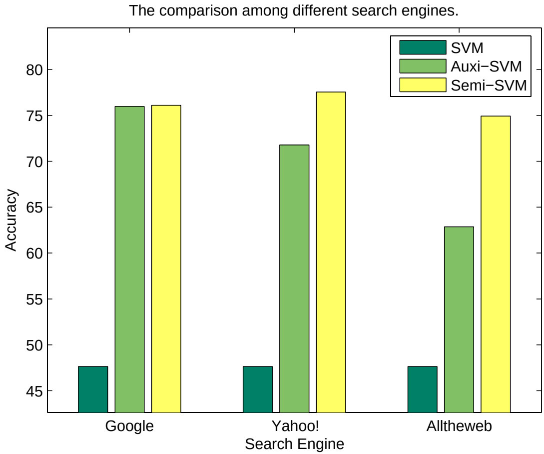
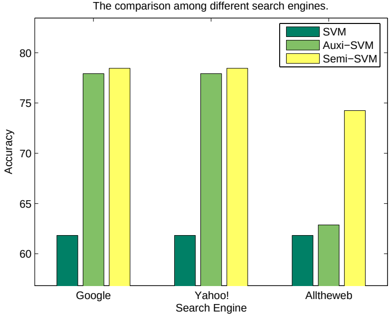
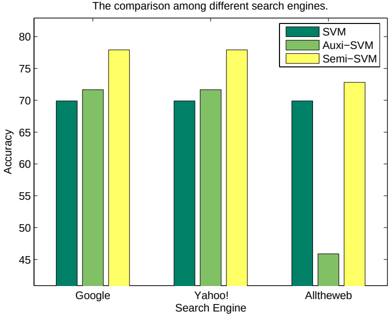


Figure: *bacterial vs. virus*

Comparison among different search engines



(a) *musculo vs. digestive*



(b) *male vs. female*

Figure: The classification accuracy of semi-supervised text categorization methods (i.e., Auxi-SVM and Semi-SVM) using different search engines (i.e., Google, Yahoo!, and Alltheweb) on two data sets of Ohmued.

Summary

Summary

1. A general framework for self-taught text categorization
2. A novel learning approach, named Discriminative Query Generation (DQG) method, for query generation
3. Reduce the classification error by 30% when compared with the state-of-the-art supervised text categorization method

Future work

- Online semi-supervised text categorization algorithms?

Outline

Conclusion

Presented

- An efficient convex relaxation model for Transductive SVM (NIPS 2007)
- An efficient method for multiple kernel learning (NIPS 2008)
- A framework for semi-supervised text categorization that actively retrieves unlabeled documents from the Internet (CIKM 2008)

Other contributions

- A unified framework for assumptions in semi-supervised learning
- A supervised self-taught learning (SSTL) model that can deal with weakly-related unlabeled data
- A framework for learning with a mixture of relevant and irrelevant unlabeled data (ICDM 2008)

Publications

- Semi-supervised learning
  1. Z. Xu, R. Jin, I. King, and M. R. Lyu, An Extended Level Method for Multiple Kernel Learning, *NIPS 2008*.
  2. Z. Xu, R. Jin, K. Huang, I. King, and M. R. Lyu. Semi-supervised text categorization by active search, *CIKM 2008*.
  3. K. Huang, Z. Xu, I. King, and Michael R. Lyu, Semi-supervised Learning from General Unlabeled Data, *ICDM 2008*.
  4. Z. Xu, R. Jin, J. Zhu, I. King, and M. R. Lyu. Efficient convex relaxation for transductive support vector machine, *NIPS 2007*.
  5. Z. Xu, J. Zhu, I. King, and M. R. Lyu. Maximum margin based semi-supervised spectral kernel learning, *IJCNN 2007*.
  6. Z. Xu, R. Jin, M. R. Lyu, and I. King. Semi-supervised Feature Selection via Manifold Regularization. Submited to *SDM 2009*.
  7. Z. Xu, R. Jin, K. Huang, I. King, and M. R. Lyuu. Semi-supervised text categorization by active search. Submitted to *Information Retrieval*.

Publications

- Supervised learning
  1. Z. Xu, K. Huang, J. Zhu, I. King, and M. R. Lyu. A Novel Kernel-based Maximum A Posteriori Classification Method. *Neural Networks*, Accepted.
  2. Z. Xu, R. Jin, J. Ye, I. King, and M. R. Lyu. Non-monotonic feature selection. Submited to *AIStats 2009*.
  3. J. Zhu, S. Hoi, Z. Xu and M. R. Lyu. An Effective Approach to 3D Deformable Surface Tracking, *ECCV 2008*.
  4. K. Huang, Z. Xu, I. King, M. R. Lyu, and Z. Zhou, A Novel Discriminative Naive Bayesian Network for Classification, in Bayesian Network Technologies: Applicatons and Graphical Models, 2007.
  5. Z. Xu, I. King, and M. R. Lyu, Web page classification with heterogeneous data fusion, WWW 2007 (poster).
  6. Z. Xu, I. King, and M. R. Lyu, Feature Selection Based on Minimum Error Minimax Probability Machine, IJPRAI, 2007.
  7. Z. Xu, K. Huang, J. Zhu, I. King, and M. R. Lyu, Kernel Maximum a Posteriori Classification with Error Bound Analysis, *ICONIP 2007*.

Publications

- 8 conference papers: 2 NIPS, 1 CIKM, 1 ICDM
- 2 journal papers
- 1 book chapter
- 3 submitted or under revision

QA

Thanks for your attention!

Acknowledgement (Coauthors)

- Rong Jin
- Kaizhu Huang
- Jianke Zhu