Basics and Advances of Semi-supervised Learning

Irwin King[1] and Zenglin Xu[2]

[1]Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, N. T., Hong Kong

[2]Department of Computer Science
Purdue University
West Lafayette, IN 47906 US

ICONIP 2011

Outline

1. Basics of Semi-supervised Learning
2. Advanced Topics
3. An Empirical Example
4. Conclusion

Outline

1. Basics of Semi-supervised Learning
   - Semi-supervised Learning
   - Probabilistic Methods
   - Co-training
   - Graph-based Semi-supervised Learning
   - Semi-supervised Support Vector Machine
2. Advanced Topics
   - Theory of semi-supervised learning
   - Advanced algorithms of semi-supervised learning
     - Variational setting
     - Large scale learning
3. An Empirical Example
4. Conclusion

A problem example

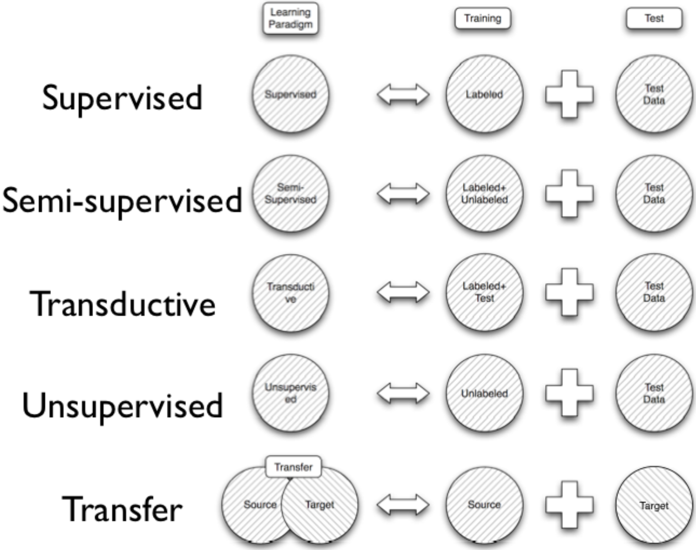
What is semi-supervised learning

Semi-supervised learning

Semi-supervised learning (SSL) is a class of machine learning techniques that make use of both labeled and unlabeled data for training.

- Supervised learning
- Unsupervised learning

Learning paradigms



Types of semi-supervised learning

Semi-supervised Classification

Given l labeled instances, $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{l}$, and u unlabeled instances, $\{\mathbf{x}_i\}_{i=l+1}^{l+u}$ for training

Constrained clustering

Given unlabeled instances $\{\mathbf{x}_i\}_{i=1}^{n}$, and "supervised information", e.g., must-links, cannot-links.

Concepts

labeled


unlabeled


semi-supervised learning

- Drawn from the same distribution
- Share the same label
- Surveys: [Zhu, 2005], [Chapelle et al., 2006]

Why we need semi-supervised learning?

- Unlabeled data are usually abundant
- Unlabeled data are usually easy to get
- Labeled data can be hard to get
    - Labels may require human efforts
    - Labels may require special devices
- Results can also be good

Why we need semi-supervised learning?

Some applications of SSL

- Web page classification:
  - Easy to crawl web pages
  - Require human experts to label them, e.g., DMOZ
- Telephone conversation transcription
  - 400 hours annotation time for each hour of speech

Semi-supervised learning vs, transductive learning

Inductive semi-supervised learning

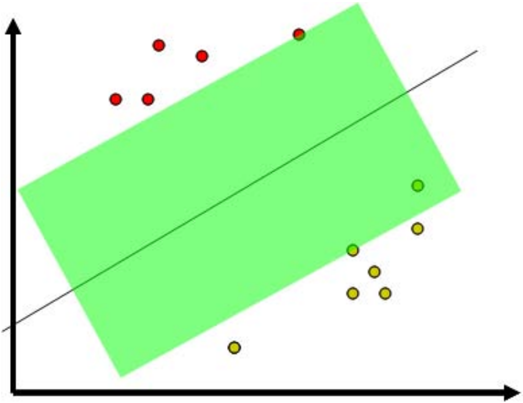
Given $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{l}$ and $\{\mathbf{x}_i\}_{i=l+1}^{l+u}$, learn a function $f : \mathcal{X} \longrightarrow \mathcal{Y}$ so that f is expected to be a good predictor on future data.

Transductive learning

Given $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{l}$ and $\{\mathbf{x}_i\}_{i=l+1}^{l+u}$, learn a function $f : \mathcal{X}^{l+u} \longrightarrow \mathcal{Y}^{l+u}$ so that f is expected to be a good predictor on the unlabeled data $\{\mathbf{x}_i\}_{i=l+1}^{l+u}$.

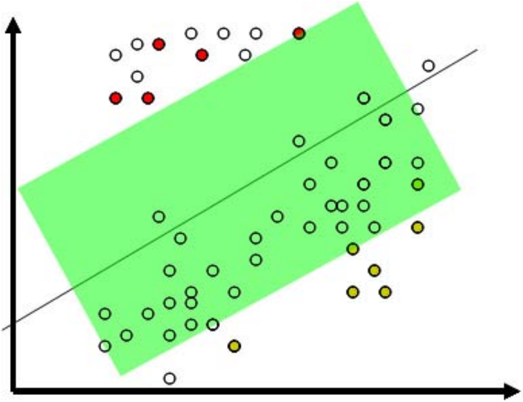
How semi-supervised learning is helpful

- SVM



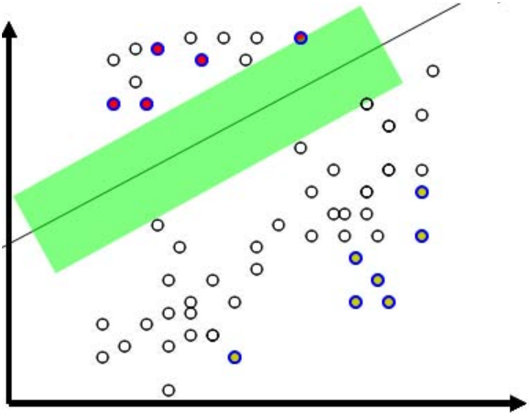
How semi-supervised learning is helpful

- SVM
- SVM with unlabeled data

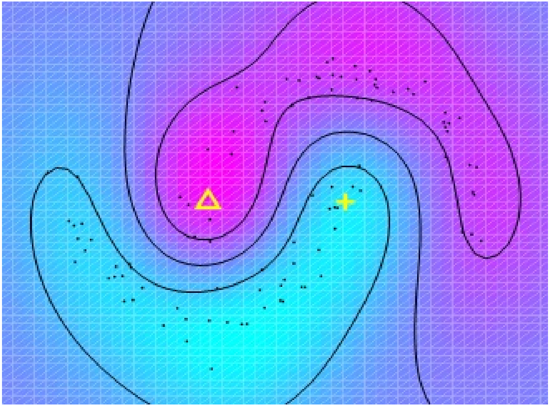


How semi-supervised learning is helpful

- SVM
- SVM with unlabeled data
- Semi-supervised SVM



How semi-supervised learning is helpful



- $p(\mathbf{x})$ carries information that is helpful for the inference of $p(y|\mathbf{x})$

Applications

- Natural language processing
  - **X**: sentence
  - **y**: parse tree
- Spam filtering
  - **X**: email
  - **y**: decision(spam or not spam)
- Video surveillance
  - **X**: video frame
  - **y**: decision(spam or not spam)
- Protein 3D structure prediction
  - **X**: DNA sequence
  - **y**: structure

How semi-supervised learning is possible?

- Assumptions or intuitions?
  - Cluster assumption (similarity)
  - Manifold assumption (structural)
  - Others
- Which one is correct?

Models

- Self-training
- Co-training
- Probabilistic generative models
- Graph-based models
- Large margin based methods
- Which one is good?

Self-training

Maybe a simple way of using unlabeled data

- Initialize $L = \{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$ and $U = \{\mathbf{x}_j\}_{i=l+1}^{n}$
- Repeat
  1. Train f from L using supervised learning
  2. Apply f to the unlabeled instances in U
  3. Remove a subset S from U ; add $\{(\mathbf{x}, f(\mathbf{x}))|\mathbf{x} \in S\}$ to L
- Until $U = \phi$

Self-training

- A wrapper method
- The choice of learner for f in step 3 is open
- Good for many real world tasks, e.g., natural language processing
- But mistake in choosing the f can reinforce itself

A simple example of generative model

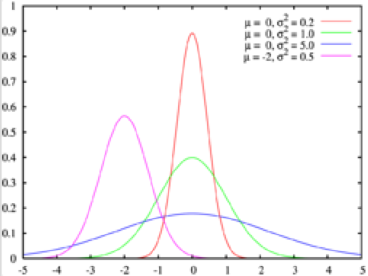
Gaussian mixture model (GMM)

- Model parameters: $\theta = \{\pi_i, \mu_i, \Sigma_i\}_{i=1}^K$, π_i: class priors, μ_i:Gaussian means, Σ_i:covariance matrices
- Joint distribution

$$
\begin{aligned}
p(\mathbf{x}, \mathbf{y}|\theta) &= p(\mathbf{y}|\theta) p(\mathbf{x}|\mathbf{y}, \theta) \\
&= \sum_{i=1}^{K} \pi_i \mathcal{N}(\mathbf{x}; \mu_i, \Sigma_i)
\end{aligned}
$$

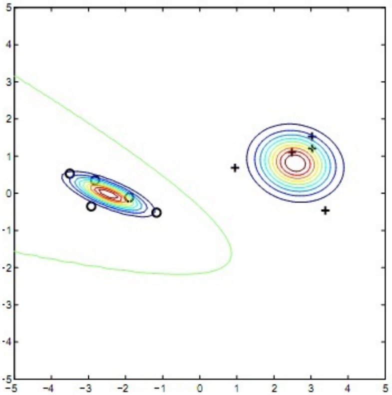
- Classification:

$$
p(\mathbf{y}|\mathbf{x}, \theta) = \frac{p(x, y|\theta)}{\sum_{i=1}^{K} p(\mathbf{x}, y_i|\theta)}
$$

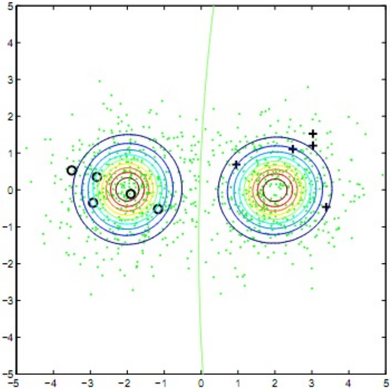


Effect of unlabeled data in GMM

$P(\mathbf{X}_l, \mathbf{Y}_l|\theta)$



$P(\mathbf{X}_l, \mathbf{Y}_l, \mathbf{X}_u|\theta)$



Generative model for semi-supervised learning

- Assumption: knowledge of $P(\mathbf{x}, \mathbf{y}|\theta)$
- Joint and marginal distribution

$$p(\mathbf{X}_l, \mathbf{Y}_l, \mathbf{X}_u|\theta) = \sum_{\mathbf{Y}_u} p(\mathbf{X}_l, \mathbf{Y}_l, \mathbf{X}_u, \mathbf{Y}_u|\theta)$$

- Objective: find the maximum likelihood estimate (MLE) of θ, the maximum a posteriori (MAP) estimate, or be Bayesian
- Optimization: Expectation Maximization (EM)
- Applications:
  - Mixture of Gaussian distributions (GMM): image classification
  - Mixture of multinomial distributions (Naïve Bayes): text categorization
  - Hidden Markov Models (HMM): speech recognition

Classification with GMM using MLE

- With only labeled data (the supervised case)
  - $\log p(\mathbf{X}_l, \mathbf{Y}_l|\theta) = \sum_{i=1}^{l} \log p(y_i|\theta)p(\mathbf{x}_i|y_i, \theta)$
  - MLE for θ trivial (sample mean and covariance)
- With both labeled and unlabeled data (the semi-supervised case)
  - $\log p(\mathbf{X}_l, \mathbf{Y}_l, \mathbf{X}_u|\theta) =$
    $\sum_{i=1}^{l} \log p(y_i|\theta)p(\mathbf{x}_i|y_i, \theta) + \sum_{i=l+1}^{l+u} \log \left(\sum_y p(y|\theta)p(\mathbf{x}_i|y, \theta)\right)$
  - MLE for θ not easy (hidden variables): EM

EM for GMM

1. Initialize $\theta^0 = \{\pi, \mu, \Sigma\}$ on $(\mathbf{X}_l, \mathbf{Y}_l)$,
2. The E-step:
   - for all $\mathbf{x} \in \mathbf{X}_u$, compute the expected label

$$p(\mathbf{y}|\mathbf{x}, \theta) = \frac{p(x, y|\theta)}{\sum_{i=1}^{K} p(\mathbf{x}, y_i|\theta)}$$

   - label all $\mathbf{x} \in \mathbf{X}_u$ according with $p(\mathbf{y}|\mathbf{x}, \theta)$
3. The M-step: update MLE θ with both $\mathbf{X}_l$ and $\mathbf{X}_u$

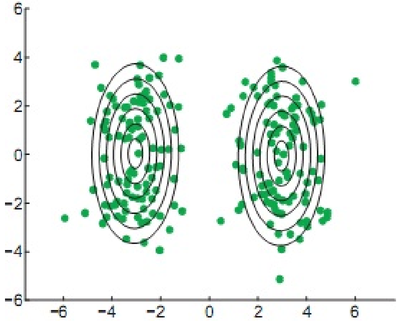
The assumption of mixture models

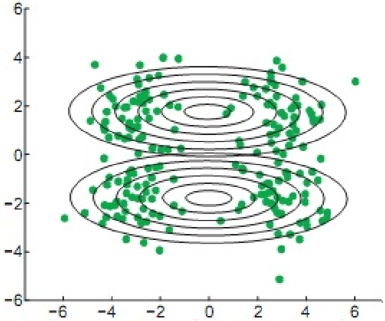
The assumption of mixture models

Data actually comes from the mixture model, where the number of components, prior $p(y)$, and conditional $p(\mathbf{x}|y)$ are all correct.

- This assumption could be WRONG!

Which one is correct?





The assumption of mixture models

Heuristics

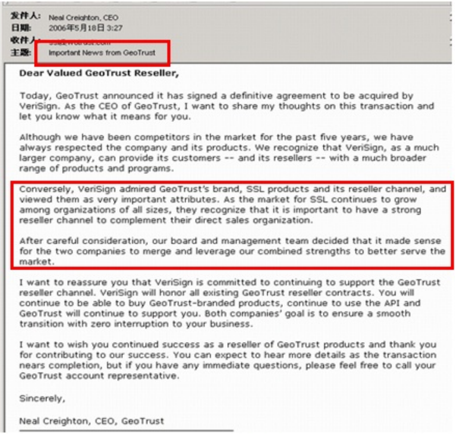
- Carefully construct the generative model, e.g., multiple Gaussian distributions per class
- Down-weight the unlabeled data $0 \leq \lambda < 1$

$$
\begin{aligned}
\log p(\mathbf{X}_l, \mathbf{Y}_l, \mathbf{X}_u|\theta) &= \sum_{i=1}^{l} \log p(y_i|\theta)p(\mathbf{x}_i|y_i,\theta) \\
&+ \lambda \sum_{i=l+1}^{l+u} \log\left(\sum_y p(y|\theta)p(\mathbf{x}_i|y,\theta)\right)
\end{aligned}
$$

Summary

- Assume a distribution for data
- Unlabeled data are used to help to identify parameters in $P(\mathbf{X}_l, \mathbf{Y}_l, \mathbf{X}_u|\theta)$
- Incorrect assumption would degrade performance
- Prior knowledge on data distribution is necessary
- Would be helpful to combine with discriminative models

Two views



发件人: Neal Creighton, CEO
日期: 2006年5月18日 3:27
收件人: [illegible]
主题: Important News from GeoTrust

Dear Valued GeoTrust Reseller,

Today, GeoTrust announced it has signed a definitive agreement to be acquired by VeriSign. As the CEO of GeoTrust, I want to share my thoughts on this transaction and let you know what it means for you.

Although we have been competitors in the market for the past five years, we have always respected the company and its products. We recognize that VeriSign, as a much larger company, can provide its customers -- and its resellers -- with a much broader range of products and programs.

Conversely, VeriSign admired GeoTrust's brand, SSL products and its reseller channel, and viewed them as very important attributes. As the market for SSL continues to grow among organizations of all sizes, they recognize that it is important to have a strong reseller channel to complement their direct sales organization.

After careful consideration, our board and management team decided that it made sense for the two companies to merge and leverage our combined strengths to better serve the market.

I want to reassure you that VeriSign is committed to continuing to support the GeoTrust reseller channel. VeriSign will honor all existing GeoTrust reseller contracts. You will continue to be able to buy GeoTrust-branded products, continue to use the API and GeoTrust will continue to support you. Both companies' goal is to ensure a smooth transition with zero interruption to your business.

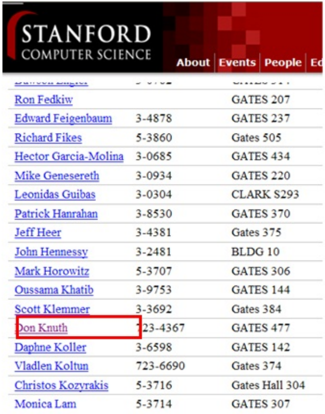
I want to wish you continued success as a reseller of GeoTrust products and thank you for contributing to our success. You can expect to hear more details as the transaction nears completion, but if you have any immediate questions, please feel free to call your GeoTrust account representative.

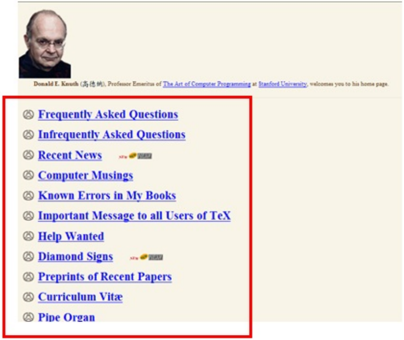
Sincerely,

Neal Creighton, CEO, GeoTrust

- Two views for email classification:
  - Title
  - Body

Two views





- Classify web pages into category for students and category for professors
- Two views of web page
  - Content: I am currently a professor of ...
  - Hyperlinks: a link to the faculty list of computer science department

Why co-training?

- Learners can learn from each other
- Implied agreement between two learners

Co-training algorithm

Input:

- Labeled data $(\mathbf{X}_l, \mathbf{Y}_l)$, unlabeled data $\mathbf{X}_u$
- Each instance has two views $\mathbf{x} = [\mathbf{x}^{(1)}, \mathbf{x}^{(2)}]$
- A learning speed k

Algorithm:

1. let $L_1 = L_2 = (\mathbf{X}_l, \mathbf{Y}_l)$.
2. Repeat until unlabeled data $U = \emptyset$:
    1. Train view-1 $f^{(1)}$ from L_1, view-2 $f^{(2)}$ from L_2.
    2. Classify unlabeled data with $f^{(1)}$ and $f^{(2)}$ separately
    3. Add $f^{(1)}$s top k most-confident predictions $(\mathbf{x}, f^{(1)}(\mathbf{x}))$ to L_2
    4. Add $f^{(2)}$s top k most-confident predictions $(\mathbf{x}, f^{(2)}(\mathbf{x}))$ to L_1
    5. Remove these $2k$ instances from the unlabeled data U.

Schematic of a co-training algorithm

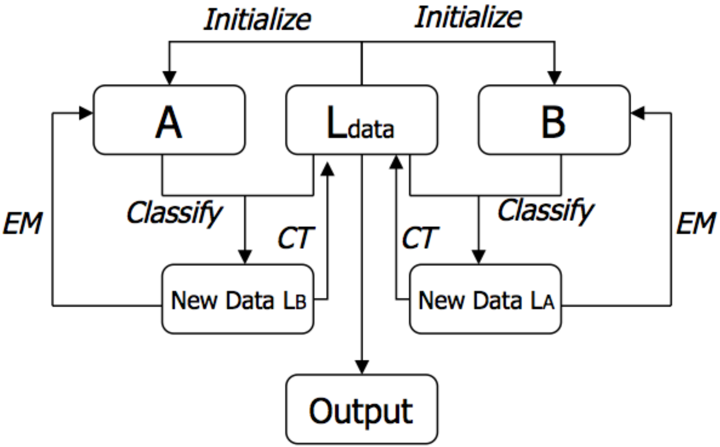


Illustration of co-training

1. Train a content-based classifier using labeled examples

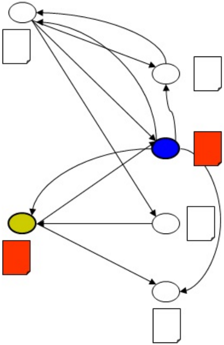


Illustration of co-training

1. Train a content-based classifier using labeled examples
2. Label the unlabeled examples that are confidently classified

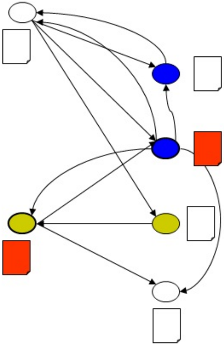


Illustration of co-training

1. Train a content-based classifier using labeled examples
2. Label the unlabeled examples that are confidently classified
3. Train a hyperlink-based classifier

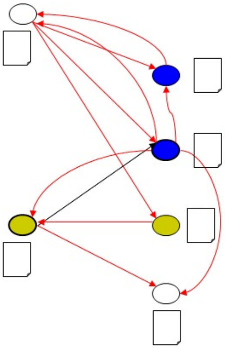


Illustration of co-training

1. Train a content-based classifier using labeled examples
2. Label the unlabeled examples that are confidently classified
3. Train a hyperlink-based classifier
4. Label the unlabeled examples that are confidently classified

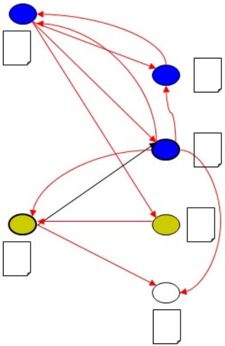
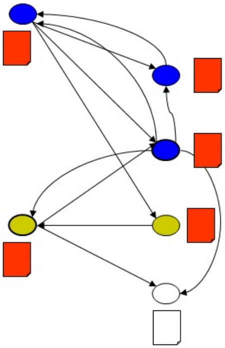


Illustration of co-training

1. Train a content-based classifier using labeled examples
2. Label the unlabeled examples that are confidently classified
3. Train a hyperlink-based classifier
4. Label the unlabeled examples that are confidently classified
5. Next iteration



Assumptions of co-training

Assumptions of co-training

- Each view alone is sufficient to make good classifications
- The two views are conditionally independently given the class label

Summary

- Key idea
  - Augment training examples of one view by exploiting the classifier of the other view
- Extension to multiple views
- Problem: how to find equivalent views

Graph-based semi-supervised learning

- Introduction
- Label propagation
- Graph partition
- Harmonic function
- Manifold regularization

Graph-based semi-supervised learning

Key idea

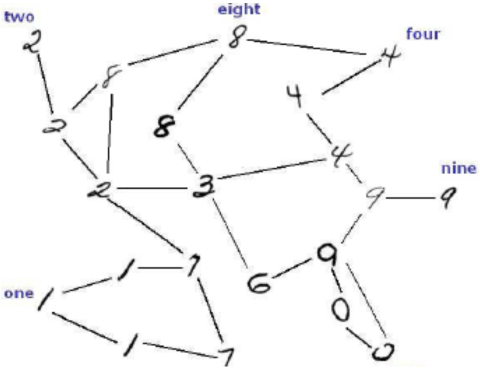
- Construct a graph with nodes being instances and edges being similarity measures among instances
- Look for some techniques to cut the graph
    - Labeled instances
    - Some heuristics, e.g., minimum cut

Graph-based semi-supervised learning

Graph construction

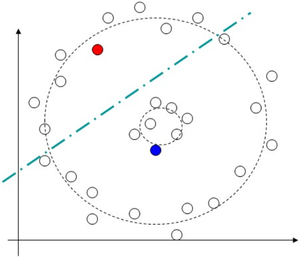
- $\mathcal{G} =< \mathcal{V}, \mathcal{E} >$, where $\mathcal{V} = \{\mathbf{x}_i\}_{i=1}^n$
- Build adjacency graph using a heuristic
  - ϵ-NN. $\epsilon \in \mathbb{R}^+$. Nodes $\mathbf{x}_i$ and $\mathbf{x}_j$ are connected if $\mathrm{dist}(\mathbf{x}_i, \mathbf{x}_j) \leq \epsilon$
  - k-NN. $k \in \mathbb{N}^+$. 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-based semi-supervised learning



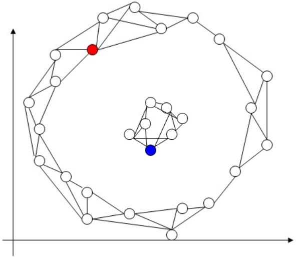
- $\mathcal{G} =< \mathcal{V}, \mathcal{E} >$
- W_{ij}: weights on edge $(\mathbf{x}_i, \mathbf{x}_j)$
- $D_{ii} = \sum_{j=1}^{n} W_{ij}$
- Graph Laplacian: $\mathbf{L} = \mathbf{D} - \mathbf{W}$
- Weighted graph Laplacian: $\mathbf{L} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{D} - \mathbf{W})\mathbf{D}^{-\frac{1}{2}}$

Label propagation



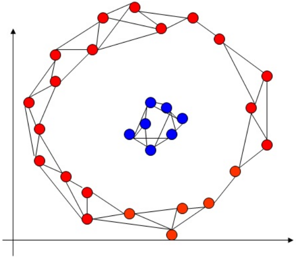
1. Supervised case: not consider the data distribution
2. How to include unlabeled data into the prediction of class labels?

Label propagation



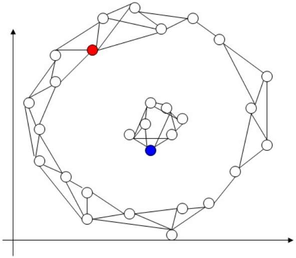
1. Supervised case: not consider the data distribution
2. How to include unlabeled data into the prediction of class labels?
3. Connect the data points that are close to each other

Label propagation



1. Supervised case: not consider the data distribution
2. How to include unlabeled data into the prediction of class labels?
3. Connect the data points that are close to each other
4. Propagate the class labels over the connected graph

Label propagation



Input:

- Given adjacency matrix W, degree matrix $\mathbf{D} = \mathrm{diag}(d_1, \ldots, d_n)$, $d_i = \sum_{j \neq i} W_{ij}$
- or normalized adjacency matrix: $\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$
- labels $\mathbf{Y}_l$
- decay parameter: α

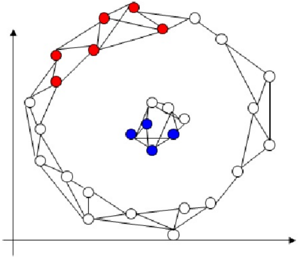
Label Propagation

- Initial class assignments $\hat{\mathbf{y}} = \{-1, 0, +1\}^n$

$$\hat{y}_i = \begin{cases} \pm 1 & \forall \mathbf{x}_i \in \mathbf{X}_l \\ 0 & \forall \mathbf{x}_i \in \mathbf{X}_u \end{cases}$$

- Predicted class assignments
  1. Predict the confidence scores $\mathbf{f} = (f_1, \ldots, f_n)$
  2. Predict the class assignments $y_i = \mathrm{sign}\,(f_i)$

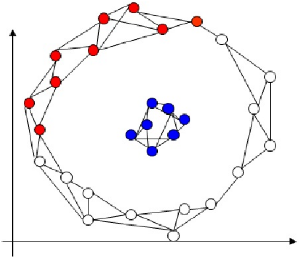
Label propagation



One round of propagation

- $$f_i = \begin{cases} \hat{y}_i & \forall \mathbf{x}_i \in \mathbf{X}_l \\ \alpha \sum_{j=1}^{n} W_{ij} \hat{y}_i & \forall \mathbf{x}_i \in \mathbf{X}_u \end{cases}$$
- $\mathbf{f}^{(1)} = \hat{\mathbf{y}} + \alpha W \hat{\mathbf{y}}$

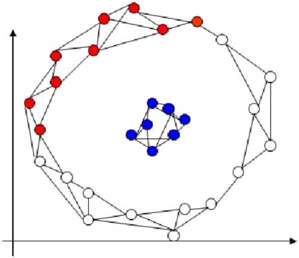
Label propagation



Two rounds of propagation

$$\begin{aligned} \mathbf{f}^{(2)} &= \mathbf{f}^{(1)} + \alpha W \mathbf{f}^{(1)} \\ &= \hat{\mathbf{y}} + \alpha W \hat{\mathbf{y}} + \alpha^2 W^2 \hat{\mathbf{y}} \end{aligned}$$

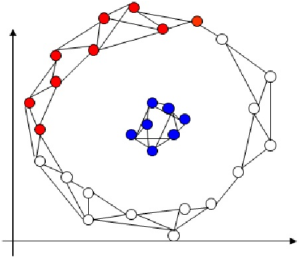
Label propagation



Any rounds of propagation

$$\mathbf{f}^{(t)} = \hat{\mathbf{y}} + \sum_{k=1}^{t} \alpha^k W^k \hat{\mathbf{y}}$$

Label propagation



Infinite rounds of propagation

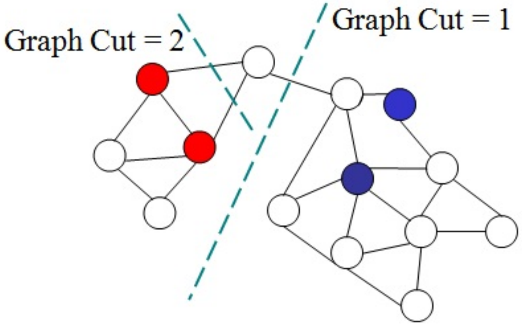
$$\mathbf{f}^{(\infty)} = \hat{\mathbf{y}} + \sum_{k=1}^{\infty} \alpha^k W^k \hat{\mathbf{y}}$$

Or equivalently

$$\mathbf{f}^{(\infty)} = (\mathbf{I} - \alpha W)^{-1} \hat{\mathbf{y}}$$

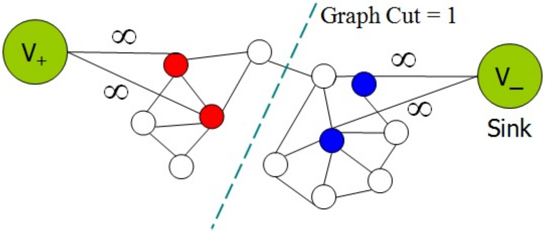
Graph partition

- Key idea
  - Classification as graph partitioning
- Search for a classification boundary
  - Consistent with labeled examples
  - Partition with small graph cut



Min-cuts

- V_+ : source, V_-: sink
- Infinite weights connecting sinks and sources

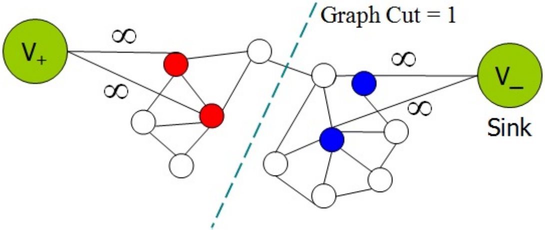


Min-cuts

- Fix $\mathbf{f}_l$, search for $\mathbf{f}_u$ to minimize $\sum_{i=1}^{n}\sum_{j=1}^{n} W_{ij}(f_i - f_j)^2$
- Equivalently, solve

$$\mathcal{C}(f) = \sum_{i=1}^{n}\sum_{j=1}^{n}\frac{W_{ij}(f_i - f_j)^2}{4} + \infty\sum_{i=1}^{l}(f_i - y_i)^2$$

- Loss function: $\infty\sum_{i=1}^{l}(f_i - y_i)^2$ (constraint)
- Combinatorial problem, but have polynomial time solution



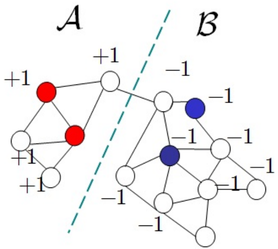
Harmonic Function

- Weight matrix $\mathbf{W}$
- membership function

$$f_i = \begin{cases} +1 & \forall \mathbf{x}_i \in \mathcal{A} \\ -1 & \forall \mathbf{x}_i \in \mathcal{B} \end{cases}$$

- Graph cut (energy function)

$$\begin{aligned} \mathcal{C}(f) &= \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{W_{ij}(f_i - f_j)^2}{4} \\ &= \frac{1}{4}\mathbf{f}^\top(\mathbf{D} - \mathbf{W})\mathbf{f} = \frac{1}{4}\mathbf{f}^\top \mathbf{L}\mathbf{f} \end{aligned}$$

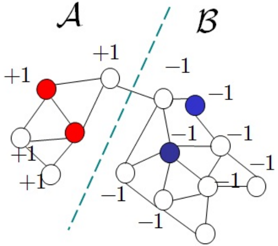


- Graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{W}$
  - Pairwise relationships among data
  - Manifold geometry of data

Harmonic Function

$$\min_{\mathbf{f}\in\{-1,+1\}^n} \quad \mathcal{C}(\mathbf{f}) = \frac{1}{4}\mathbf{f}^\top \mathbf{L}\mathbf{f}$$
$$\text{s. t.} \quad f_i = y_i,\ i = 1, \ldots, l$$

Challenge: combinatorial optimization?

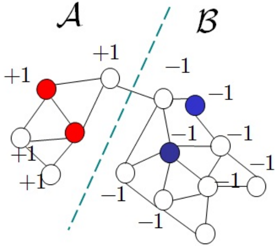


Harmonic Function

Relaxation to continuous space

$$\min_{\mathbf{f}\in\mathbb{R}^n} \quad \mathcal{C}(\mathbf{f}) = \frac{1}{4}\mathbf{f}^\top \mathbf{L}\mathbf{f}$$
$$\text{s. t.} \quad f_i = y_i,\ i = 1, \ldots, l$$

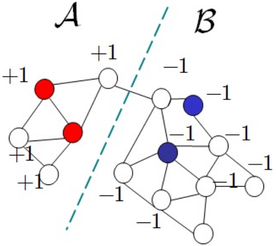
- $f(\mathbf{x}_i) = y_i$ for $i = 1, \ldots, l$
- f minimizes the energy function $\sum_{i=1}^n \sum_{j=1}^n \frac{W_{ij}(f_i - f_j)^2}{4}$
- average of neighbors $f(\mathbf{x}_i) = \frac{\sum_{j\sim i} W_{ij} f(\mathbf{x}_j)}{\sum_{j\sim i} W_{ij}}$



Harmonic Function

An alternative algorithm

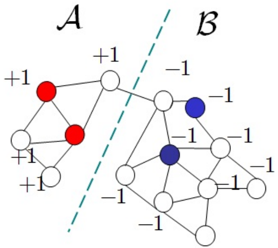
1. Fix $f(\mathbf{x}_i) = y_i$ for $\mathbf{x}_i \in \mathbf{X}_l$ and initialize $f(\mathbf{x}_i) = 0$ for $\mathbf{x}_i \in \mathbf{X}_u$
2. Repeat until convergence $f(\mathbf{x}_i) = \frac{\sum_{j\sim i} W_{ij} f(\mathbf{x}_j)}{\sum_{j\sim i} W_{ij}}$ for $\mathbf{x}_i \in \mathbf{X}_u$



Harmonic Function

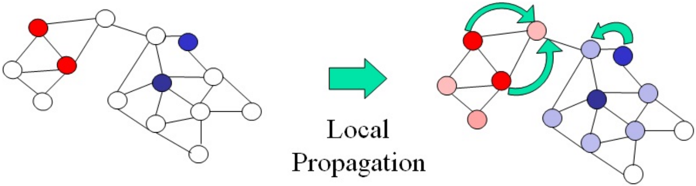
Analytical solution from the optimization perspective

$$\begin{aligned} \mathbf{f}_u &= -\mathbf{L}_{u,u}^{-1}\mathbf{L}_{u,l}\mathbf{y}_l \text{ where} \\ \mathbf{L} &= \begin{bmatrix} \mathbf{L}_{l,l} & \mathbf{L}_{l,u} \\ \mathbf{L}_{u,l} & \mathbf{L}_{u,u} \end{bmatrix} \\ \mathbf{f} &= (\mathbf{f}_l, \mathbf{f}_u) \end{aligned}$$



Harmonic function

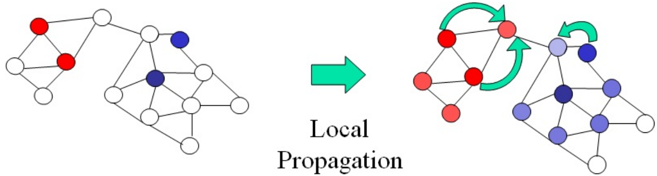
Connection to label propagation (learning with local and global consistency)

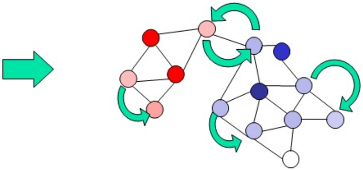


$$\mathbf{f}_u = -\mathbf{L}_{u,u}^{-1}\boxed{\mathbf{L}_{u,l}\mathbf{y}_l}$$

Harmonic function

Connection to label propagation (learning with local and global consistency)





$$\mathbf{f}_u = -\mathbf{L}_{u,u}^{-1}\mathbf{L}_{u,l}\mathbf{y}_l$$

Manifold regularization

Manifold regularization is inductive

- Define a function in a RKHS: $f(\mathbf{x}) = h(\mathbf{x}) + b$, $h(\mathbf{x}) \in \mathcal{H}_k$
- Flexible loss function: e.g., the hinge loss
- Regularizer prefers low energy $\mathbf{f}^\top \mathbf{L}\mathbf{f}$

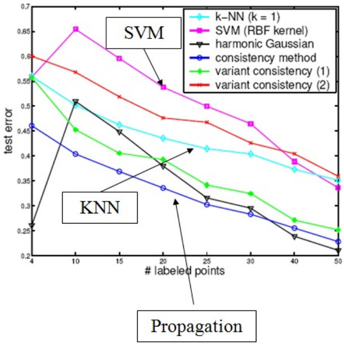
$$\min_f \sum_{i=1}^{l} (1 - y_i f(\mathbf{x}_i))_+ + \lambda_1 \|h\|_{\mathcal{H}_k} + \lambda_2 \mathbf{f}^\top \mathbf{L}\mathbf{f}$$

where

- λ_1 and λ_2 are non-negative tradeoff constants

Application

Label propagation (learning with local and global consistency)

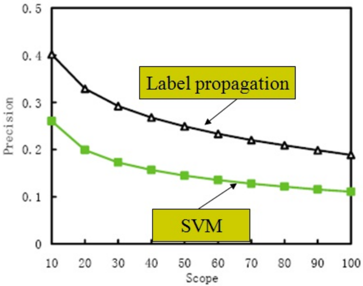


[Zhou et al., NIPS 2003]

- 20-newsgroups: autos, motorcycles, baseball, and hockey under rec
- Pre-processing: stemming, remove stopwords & rare words, and skip header
- #Docs: 3970, #word: 8014

Application

Label propagation (learning with local and global consistency)



[Wang et al., ACM MM 2004]

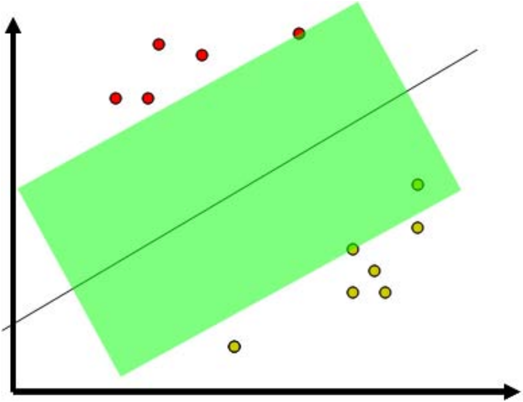
- 5,000 images
- Relevance feedback for the top 20 ranked images
- Classification problem
  - Relevant or not?
  - $f(\mathbf{x})$: degree of relevance Learning
- SVM vs. Label propagation

Summary of graph-based methods

- Construct a graph using pairwise similarity
- Key quantity: graph Laplacian
  - Captures the geometry of the graph
- Decision boundary is consistent
  - Graph structure
  - Labeled examples
- Parameters related to graph structure are important

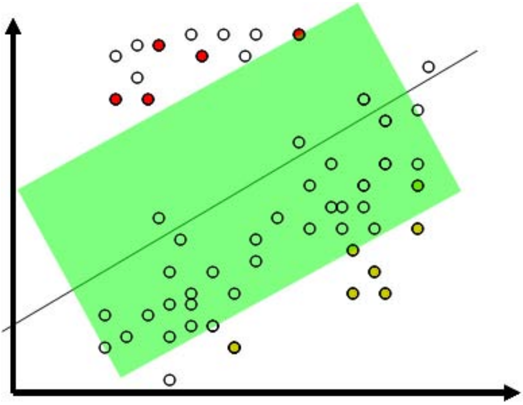
Semi-supervised SVM

- SVM



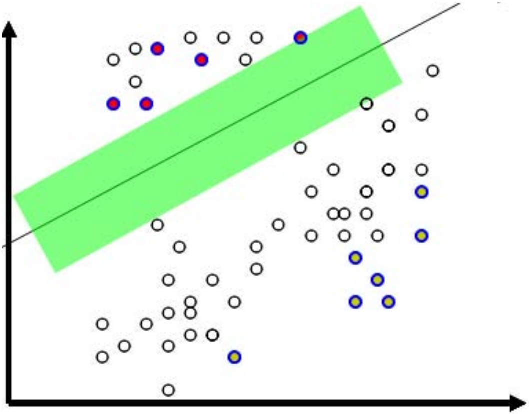
Semi-supervised SVM

- SVM
- SVM with unlabeled data



Semi-supervised SVM

- SVM
- SVM with unlabeled data
- Semi-supervised SVM (S3VM)



Assumptions of semi-supervised SVM

Low Density Separation Assumption

The decision boundary should lie in a low-density region, that is the decision boundary does not cut through dense unlabeled data.

Also known as cluster assumption

Semi-supervised SVM

S3VM: $\mathbf{y}_u$ for unlabeled data as a free variable

S3VM

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

- No longer convex optimization problem
- Alternating optimization

Semi-supervised SVM

Equivalently, unconstrained form:

S3VM

$$\min_{f} \min_{\mathbf{y}_u} \quad \|\mathbf{w}\|_2^2 + C_l \sum_{i=1}^{l} (1 - y_i f(\mathbf{x}_i))_+ + C_u \sum_{i=l+1}^{l+u} (1 - y_i f(\mathbf{x}_i))_+$$

where $(1 - y_i f(\mathbf{x}_i))_+ = \max(0, 1 - y_i f(\mathbf{x}_i))$

Optimize over $\mathbf{y}^u = (y_{l+1}^u, \ldots, y_n^u)$, we have

$$\min_{y_i^u} (1 - y_i f(\mathbf{x}_i))_+ = (1 - \mathrm{sign}\,(f(\mathbf{x}_i)) f(\mathbf{x}_i))_+ = (1 - |f(\mathbf{x}_i)|)_+$$

Semi-supervised SVM

S3VM objective

$$\min_{f} \quad \|\mathbf{w}\|_2^2 + C_l \sum_{i=1}^{l} (1 - y_i f(\mathbf{x}_i))_+ + C_u \sum_{i=l+1}^{l+u} (1 - |f(\mathbf{x}_i)|)_+$$

- Non-convex problem
- Optimization methods?

Representative optimization methods for S3VM

- label-switch-retraining [Joachims, 1999]
- gradient descent [Chapelle and Zien, 2005]
- continuation [Chapelle et al., 2006]
- concave-convex procedure [Collobert et al, 2006]
- semi-definite programming [Bie and Cristiannini, 2004; Xu et al., 2004; Xu et al., 2007]
- deterministic annealing [Sindhwani et al., 2006]
- branch-and-bound [Chapelle et al., 2006]
- non-differentiable method [Astorino and Fuduli, 2007]

Experiments

Experimental data

data set	classes	dims	points	labeled
g50c	2	50	550	50
Text	2	7511	1946	50
Uspst	10	256	2007	50
Isolet	9	617	1620	50
Coil20	20	1024	1440	40
Coil3	3	1024	216	6
2moons	2	102	200	2

Figure: Data sets.

Data and results are from [Chapelle et al., 2008]

Quality of performance

Quality of minimization

∇S^3VM	cS^3VM	CCCP	S^3VM^{light}	∇DA	Newton
1.7	1.9	4.5	4.9	4.3	3.7

Figure: Average objective values.

Quality of prediction

	∇S^3VM	cS^3VM	CCCP	S^3VM^{light}	∇DA	Newton	SVM	SVM-5cv
g50c	6.7	6.4	6.3	6.2	7	6.1	8.2	4.9
Text	5.1	5.3	8.3	8.1	5.7	5.4	14.8	2.8
Uspst	15.6	36.2	16.4	15.5	27.2	18.6	20.7	3.9
Isolet	25.8	59.8	26.7	30	39.8	32.2	32.7	6.4
Coil20	25.6	30.7	26.6	25.3	12.3	24.1	24.1	0

Figure: Errors on unlabeled data.

Combine with graph-based methods

		Exact r	(Table 8 setting)	Estimated r	(Table 13 setting)
	LapSVM	S^3VM^{light}	LapSVM-S^3VM^{light}	S^3VM^{light}	LapSVM-S^3VM^{light}
g50c	6.4	6.2	4.6	7.5	6.1
Text	11	8.1	8.3	9.2	9.0
Uspst	11.4	15.5	8.8	24.4	19.6
Isolet	41.2	30.0	46.5	36.0	49.0
Coil20	11.9	25.3	12.5	25.3	12.5
Coil3	20.6	56.7	17.9	56.7	17.9
2moons	7.8	68.8	5.1	68.8	5.1

Figure: Errors on unlabeled data.

- Seem to have better performance

Summary

Semi-supervised SVM

- Based on maximum margin principle
- Low density assumption
- Extend SVM by pushing the decision boundary traversing low density regions
- Classification margin is decided by
  - Class labels assigned to unlabeled data
  - Labeled examples
- Problem: non-convex optimization
  - Solvers: $\Delta S3VM$, $\mathrm{SVM}^{\mathrm{light}}$, CCCP, etc
  - No one is the best?
  - Sensitive to data

Outline

1. Basics of Semi-supervised Learning
   - Semi-supervised Learning
   - Probabilistic Methods
   - Co-training
   - Graph-based Semi-supervised Learning
   - Semi-supervised Support Vector Machine
2. Advanced Topics
   - Theory of semi-supervised learning
   - Advanced algorithms of semi-supervised learning
     - Variational setting
     - Large scale learning
3. An Empirical Example
4. Conclusion

Theory of semi-supervised learning

Questions

- Whether unlabeled data can help?
- If yes, why unlabeled data can help?
- If yes, how much unlabeled do we need?
- Which assumption of SSL should we take?

Theory of semi-supervised learning

- Complexity analysis of SSL
  - E.g., Augmented PAC model (Balcan & Blum, 2008)
  - Analyze the compatibility between data distributions and learning functions
  - Unlabeled data are potentially helpful in estimating the compatibility
  - Bounds for the number of unlabeled data are proposed

Theory of semi-supervised learning

- Complexity analysis of SSL
    - E.g., Augmented PAC model (Balcan & Blum, 2008)
    - Analyze the compatibility between data distributions and learning functions
    - Unlabeled data are potentially helpful in estimating the compatibility
    - Bounds for the number of unlabeled data are proposed
- Assumption analysis in SSL
    - E.g., manifold or smoothness? (Lafferty & Wasserman, 2007)
    - Manifold assumption is more important than smoothness assumption

Theory of semi-supervised learning

- Complexity analysis of SSL
    - E.g., Augmented PAC model (Balcan & Blum, 2008)
    - Analyze the compatibility between data distributions and learning functions
    - Unlabeled data are potentially helpful in estimating the compatibility
    - Bounds for the number of unlabeled data are proposed
- Assumption analysis in SSL
    - E.g., manifold or smoothness? (Lafferty & Wasserman, 2007)
    - Manifold assumption is more important than smoothness assumption
- Whether or to what extent unlabeled data can help?
    - E.g., Finite sample analysis (Singh, Nowak, & Zhu, 2008)
    - Bridge the gap between theory analysis and practice
    - Cases where SSL can help are given based on the analysis of margins

New directions of semi-supervised learning

- Probabilistic methods: hybrids of generative models and discriminative models (Lasserre et. al, 2006; Fujino et. al, 2008)
- Extensions of multiview learning: view disagreement, structured output, information theoretic framework
- Graph-based methods: how to construct the graph?
- Semi-supervised SVM: new optimization methods?

Extensions

- Variational settings of SSL (different distributions)
  - Variance shift for test data
  - Unlabeled data are irrelevant
  - Mixture of relevant and irrelevant unlabeled data

Extensions

- Scalability
  - Online learning, e.g., online manifold regularization
  - Efficient optimization algorithms, like CCCP

Variational settings of SSL (different distributions)

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]

Learning with irrelevant data

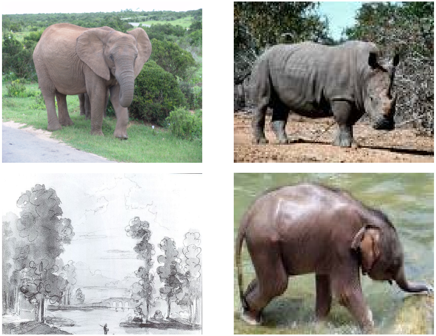
Unlabeled

Irrelevant

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

SSL with mixture of relevant and irrelevant data

Unlabeled



Mixture

- Relevant mixed with others
- Semi-supervised learning from a mixture
- E.g., [Zhang et al., 2008], [Huang et al., 2008]

Large scale semi-supervised learning

Perspective:

- Efficient algorithms
- Online learning
  - Examples arrive sequentially, no need to store them all

Online semi-supervised learning

Online semi-supervised learning:

1. At time t, adversary picks $\mathbf{x}_t \in \mathcal{X}$, $y_t \in \mathcal{Y}$ shows $\mathbf{x}_t$
2. Learner builds a classifier $f_t : \mathcal{X} \to R$, and predicts $f_t(\mathbf{x}_t)$
3. With small probability, adversary reveals y_t
4. Learner updates to f_{t+1} based on $\mathbf{x}_t$ and y_t (if given)

Online manifold regularization

- Bach mode manifold regularization

$$\mathcal{J}(f) = \frac{1}{l}\delta(y_t)\ell(f(\mathbf{x}_t, y_t)) + \frac{\lambda_1}{2}\|f\|^2_{\mathcal{H}} + \frac{\lambda}{2T}\sum_{i=1}^{T}\sum_{j=1}^{T}(f(\mathbf{x}_i) - f(\mathbf{x}_j))^2 W_{ij}$$

  - $\delta(y_t)$: indicator of whether $\mathbf{x}_t$ is labeled
- Instantaneous risk

$$\mathcal{J}_t(f) = \frac{T}{l}\delta(y_t)\ell(f(\mathbf{x}_t, y_t)) + \frac{\lambda_1}{2}\|f\|^2_{\mathcal{H}} + \lambda_2\sum_{i=1}^{T}(f(\mathbf{x}_i) - f(\mathbf{x}_t))^2 W_{ij}$$

  - Involves graph edges between $\mathbf{x}_t$ and all previous examples
- $\mathcal{J}(f) = \sum_{t=1}^{T}\mathcal{J}_t(f)$

Online manifold regularization

Use gradient descent to update

$$f_{t+1} = f_t - \eta_t \frac{\partial \mathcal{J}_t(f)}{\partial f} \mid f_t$$

- $\eta_t = 1/\sqrt{(t)}$
- Iteratively update
  1. $f_t = \sum_{i=1}^{t-1} \alpha_i^{(t)} K(\mathbf{x}_i, \dot{})$
  2. update $\alpha^{(t+1)}$ by

$$\begin{aligned} \alpha_i^{(t+1)} &= (1 - \eta_t \lambda_1)\alpha_i^{(t)} - 2\eta_t \lambda_2 (f_t(\mathbf{x}_i) - f_t(\mathbf{x}_t)) W_{i,t}, \ i = 1, \ldots, t-1 \\ \alpha_t^{(t+1)} &= 2\eta_t \lambda_2 \sum_{i=1}^{t-1} (f_t(\mathbf{x}_i) - f_t(\mathbf{x}_t)) W_{i,t} - \eta_t \frac{T}{l} \delta(y_t) \ell'(f(\mathbf{x}_t, y_t)) \end{aligned}$$

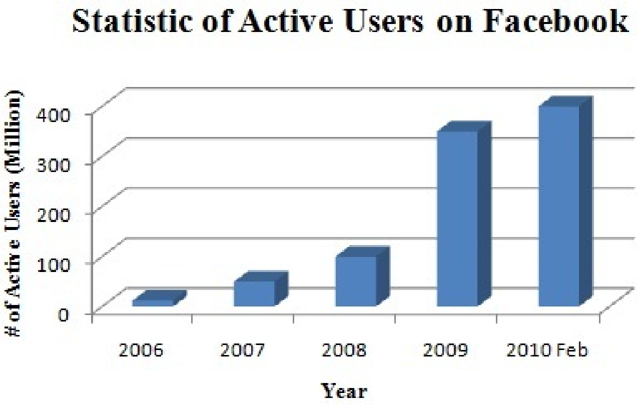
- Space $\mathcal{O}(T)$: stores all previous examples
- Time $\mathcal{O}(T^2)$: each new instance connects to all previous ones
- Can be further reduced by approximation techniques

Outline

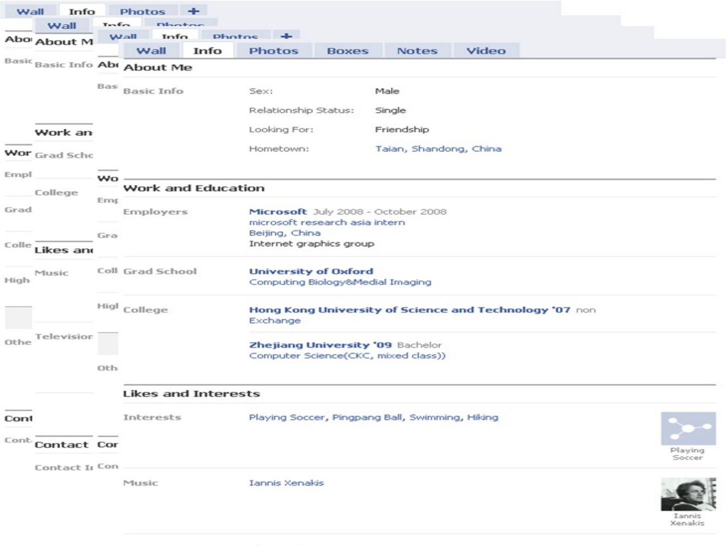
1. Basics of Semi-supervised Learning
   - Semi-supervised Learning
   - Probabilistic Methods
   - Co-training
   - Graph-based Semi-supervised Learning
   - Semi-supervised Support Vector Machine
2. Advanced Topics
   - Theory of semi-supervised learning
   - Advanced algorithms of semi-supervised learning
     - Variational setting
     - Large scale learning
3. An Empirical Example
4. Conclusion

Privacy exposure in social networks

- Number of users in Facebook

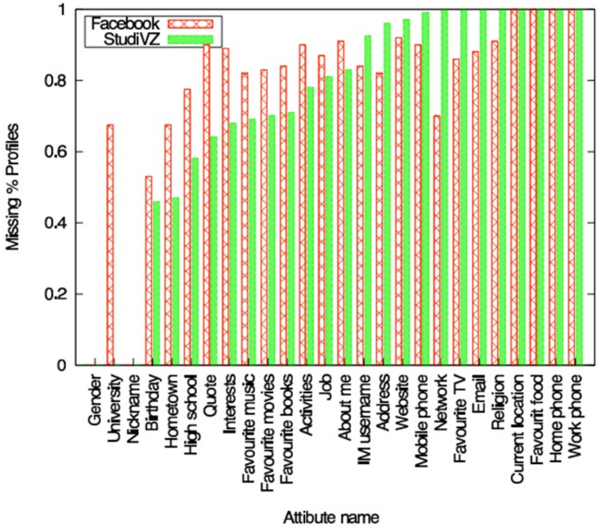


Privacy exposure in social networks



Privacy exposure in social networks

User profiles are not complete



Privacy exposure in social networks

- Friends (linked persons) may share similar property
- Information of friends may expose his information

$$\Downarrow$$

- How much of these context information can be exposed?

$$\Downarrow$$

- Semi-supervised methods seem to suite our scenario

Experiment

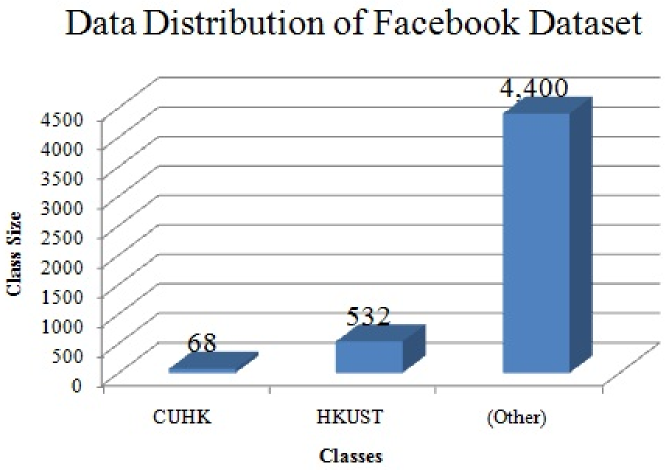
- Objective: to expose which university a user comes from
- Methods: SSL framework
- Datasets: real-world data from Facebook and StudiVZ

Experiment

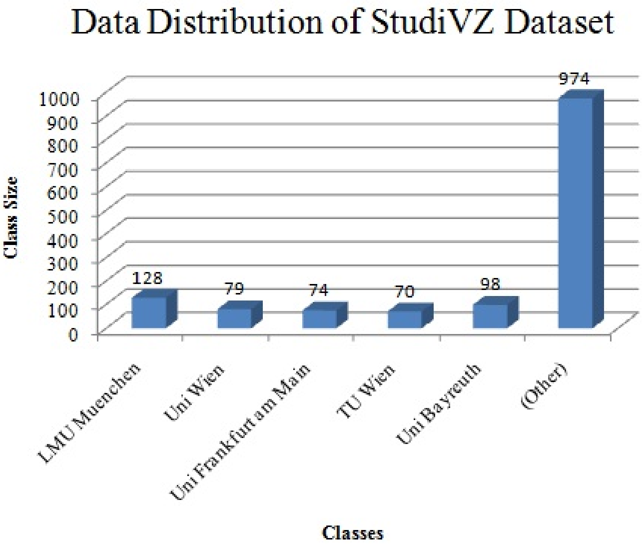
- Objective: to expose which university a user comes from
- Methods: SSL framework
- Datasets: real-world data from Facebook and StudiVZ

Dataset	Facebook	StudiVZ
Vertices	5,000	1,423
Edges	31,442	7,769
Groups	61	406
Networks	78	0
Classes	3	6

Experiment


Data Distribution of Facebook Dataset
Class Size
4500
4000
3500
3000
2500
2000
1500
1000
500
0
68
532
4,400
CUHK
HKUST
(Other)
Classes

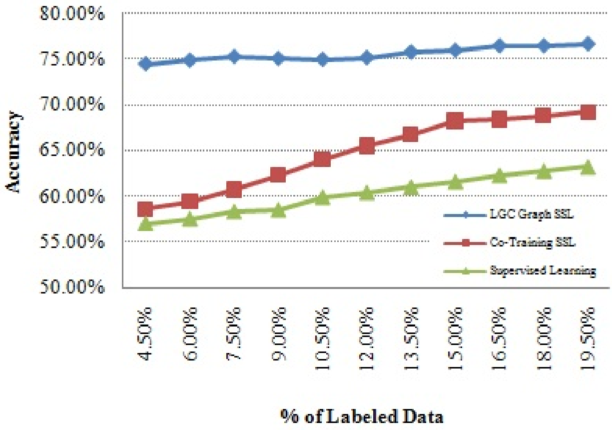
Experiment


Data Distribution of StudiVZ Dataset
Class Size
1000
900
800
700
600
500
400
300
200
100
0
128
79
74
70
98
974
LMU Muenchen
Uni Wien
Uni Frankfurt am Main
TU Wien
Uni Bayreuth
(Other)
Classes

Experiment

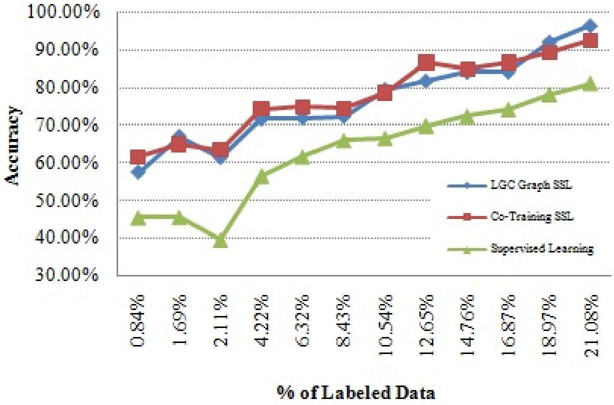
- Feature Selection
  - Users' profile: top 3 completeness
  - Relational information
- Data Translation
  - Missing Value: average value
  - Similarity: cosine similarity

Experiment



Experiment Result on Facebook Dataset with 5,000 Users

Experiment



Experiment Result on StudiVZ Dataset with 1,423 Users

Summary

- Learn hidden users' attributes based on relational information and profile similarity among users
- SSL predicts sensitive information more accurately than supervised learning
- Users' security is never secure and protections are needed

Outline

1. Basics of Semi-supervised Learning
   - Semi-supervised Learning
   - Probabilistic Methods
   - Co-training
   - Graph-based Semi-supervised Learning
   - Semi-supervised Support Vector Machine
2. Advanced Topics
   - Theory of semi-supervised learning
   - Advanced algorithms of semi-supervised learning
     - Variational setting
     - Large scale learning
3. An Empirical Example
4. Conclusion

Conclusion

Presented

- A brief introduction to semi-supervised learning
  - Generative models
  - Co-training
  - Graph-based methods
  - Semi-supervised support vector machine
- Advance topics in semi-supervised learning
- An empirical evaluation of semi-supervised learning in online social network analysis

References and therein

1. O. Chapelle, B. Schölkopf, and A. Zien. Semi-Supervised Learning. MIT Press, Cambridge, MA, 2006.
2. O. Chapelle and A. Zien. Semi-supervised classification by low density separation. Tenth International Workshop on Artificial Intelligence and Statistics, 2005.
3. R. Collobert, F. Sinz, J. Weston, and L. Bottou. Large scale transductive SVMs. Journal of Machine Learning Reseaerch, 2006.
4. T. Joachims. Transductive inference for text classification using support vector machines, ICML 1999.
5. X. Zhu. Semi-supervised learning literature survey. Technical report, Computer Sciences, University of Wisconsin-Madison, 2005.
6. X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions, ICML 2003
7. Xiaojin Zhu and Andrew B. Goldberg. Introduction to Semi-Supervised Learning. Morgan & Claypool, 2009.
8. `http://pages.cs.wisc.edu/~jerryzhu/pub/sslchicago09.pdf`
9. `http://www.cse.msu.edu/~cse847/slides/semisupervised-1.ppt`

QA

Thanks for your attention!