



# Lecture 5: Semi-supervised Learning

Pilsung Kang

School of Industrial Management Engineering

Korea University

# AGENDA

01 **Semi-supervised Learning: Overview**

---

02 Self-Training

---

03 Generative Models

---

04 Graph-based SSL

---

05 Multi-view Algorithms (Co-Training)

---

# Machine Learning Categories

- Category of learning
  - ✓ According to the existence of target

## Supervised Learning

A given dataset **X & Y**

	Var. 1	Var. 2	...	Var. d	→	Y
Ins. 1	..	..	...	..		..
Ins. 2	..	..	...	..	$y = f(x)$	..
...	...	...	...	...		...
Ins. N	..	..	..	..		..

## Semi-supervised Learning

A given dataset **X & Y**

	Var. 1	Var. 2	...	Var. d	→	Y
Ins. 1	..	..	...	..		..
Ins. 2	..	..	...	..	$y = f(x)$	..
...	...	...	...	...		...
Ins. N	..	..	..	..		..
...	...	...	...	...		...
...	...	...	...	...		...
...	...	...	...	...		...
...	...	...	...	...		...
Ins. M	..	..	..	..		..

## Unsupervised Learning

A given dataset **X**

	Var. 1	Var. 2	...	Var. d
Ins. 1	..	..	...	..
Ins. 2	..	..	...	..
...	...	...	...	...
Ins. N	..	..	..	..

# Backgrounds

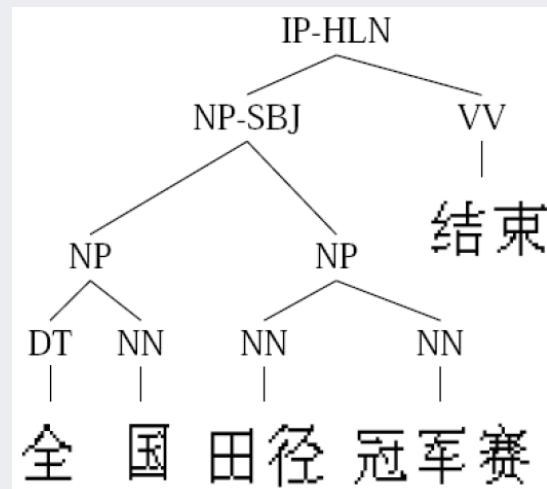
Zhu (2007)

- People want better performance for free

- ✓ Unlabeled data is cheap
- ✓ Labeled data can be hard to get
  - human annotation is boring
  - labels may require experts
  - labels may require special devices
  - **your graduate students are on vacation!!!**

- Natural language parsing task (Penn Chinese Treebank)

- ✓ 2 years for 4,000 sentences



# Backgrounds

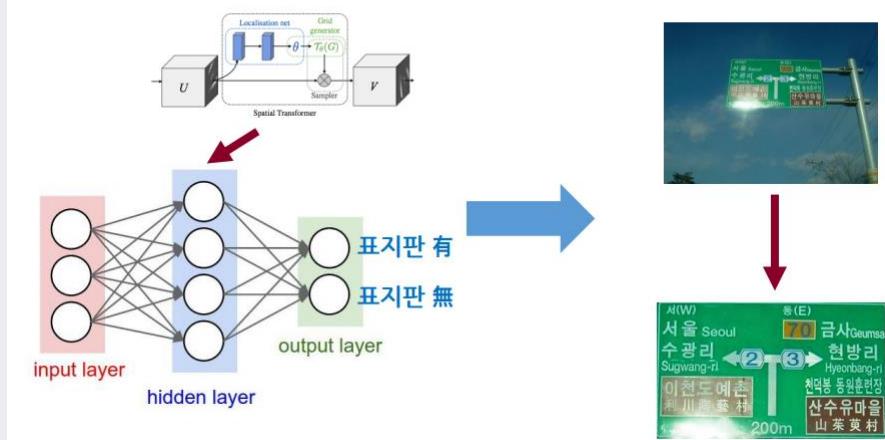
- Labeled data can be hard to get



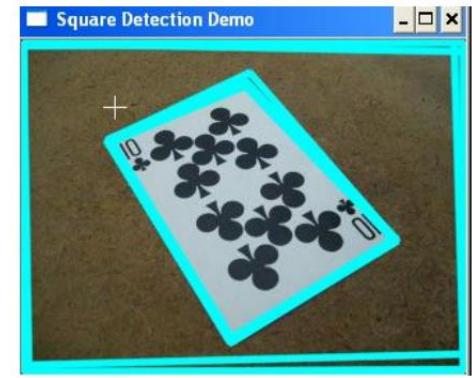
# Backgrounds

- Labeled data can be hard to get

- Spatial transformer를 이용하여 표지판 추출이 가능할까?



- Square detection in OpenCV

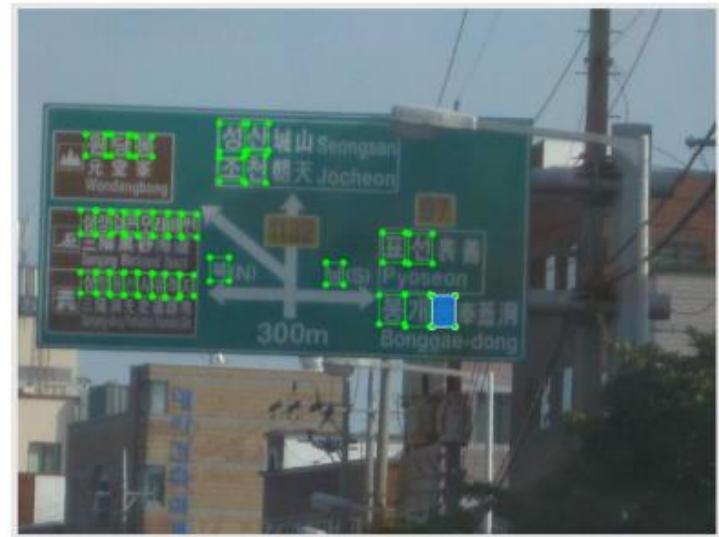


- Square detection in OpenCV



# Backgrounds

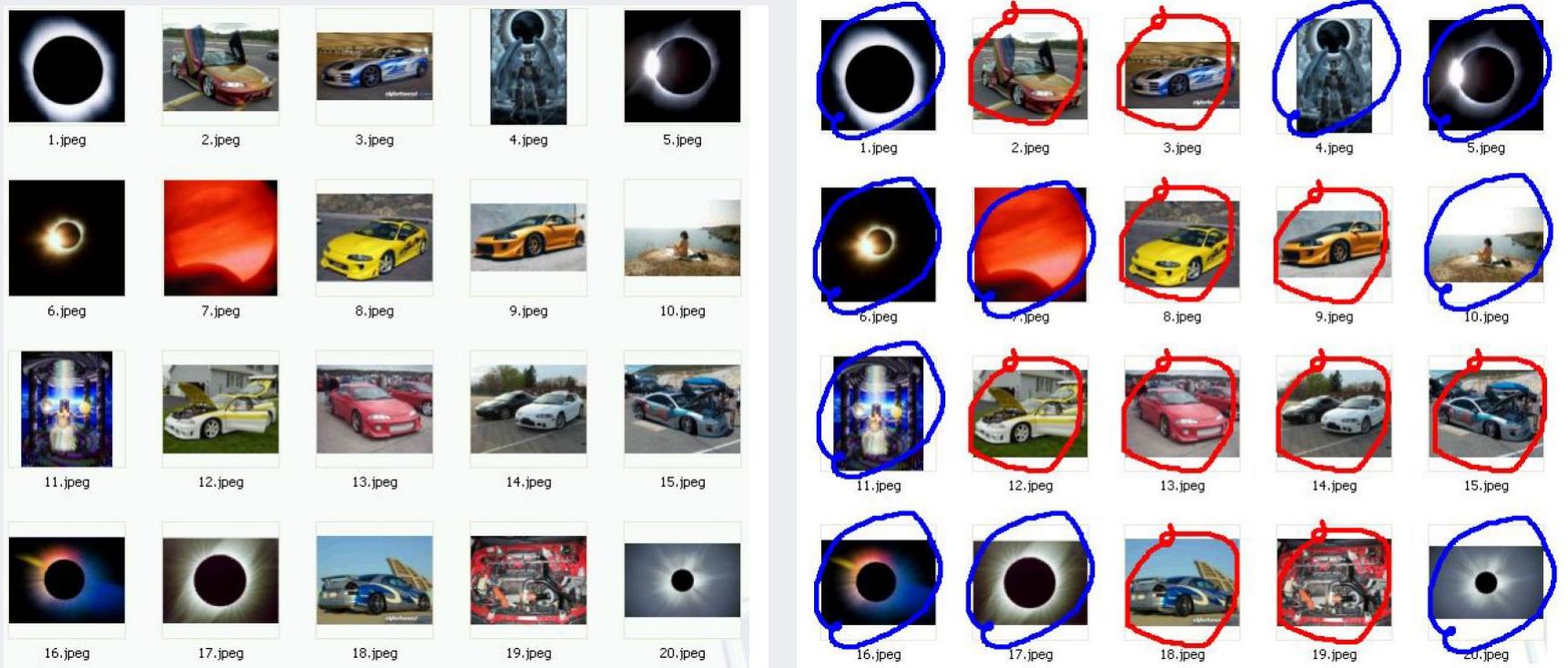
- Labeled data can be hard to get
  - 그래서 최후의 방법으로 남겨두었지만, 어쩔 수 없이 **직접 labelling** 하기로 하였습니다.
  - 하지만 711개 글자 class에 대하여 모두 하는 것은 비효율적일 것이라 판단하여, 글자를 모두 1개 class로 label을 할당하여 학습하기로 하였습니다.



# Backgrounds

Zhu (2007)

- For some tasks, it may not be too difficult to label 1000+ instances
  - ✓ Image categorization of “eclipse”

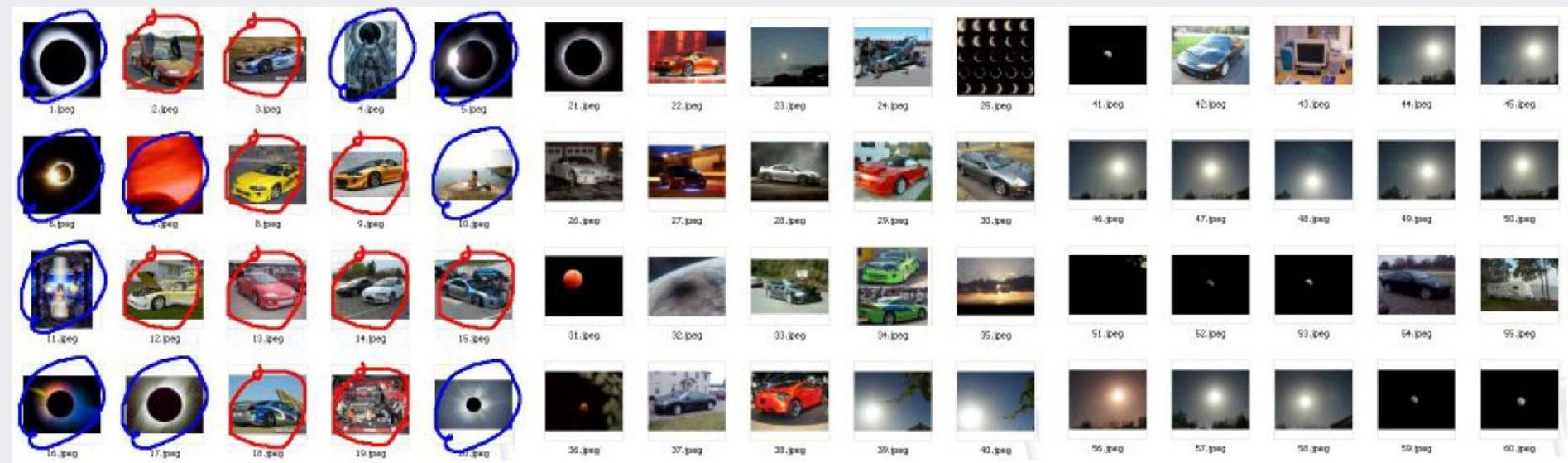


# Backgrounds

Zhu (2007)

- Example of not-so-hard-to-get labels

✓ Nonetheless...



✓ We still have bunch of other images that are not labeled yet...

How can we utilize those unlabeled data  
to improve the performance of supervised learning task?

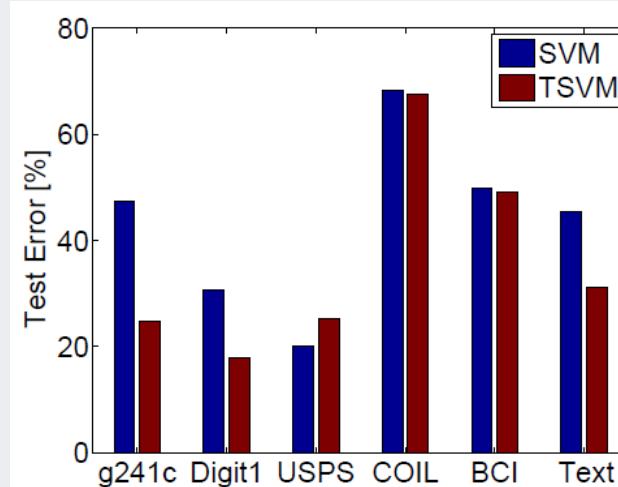
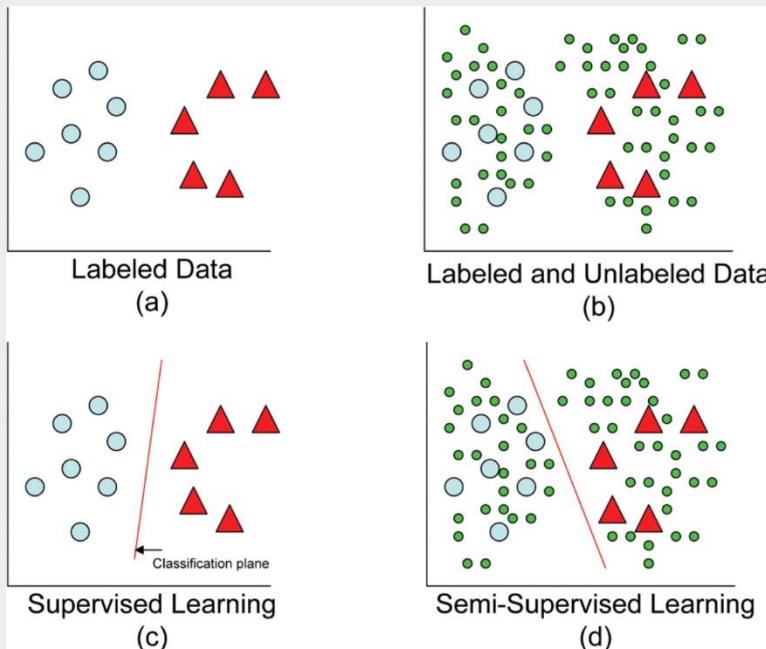
# Purpose

Zien (2008)

- Purpose

- ✓ Using both labeled and unlabeled data to build better learners, than using each of alone.

- Can it be possible?



10 labeled points  
~1400 unlabeled points

SVM: supervised  
TSVM:  
semi-supervised

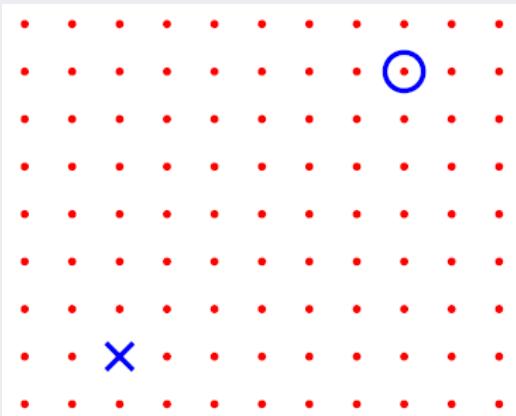
# Semi-supervised Learning

Zien (2008)

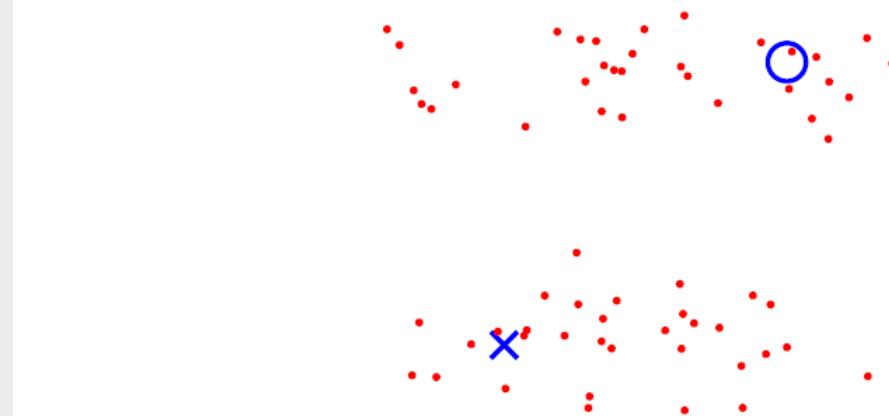
- Why would unlabeled data be useful at all?
  - ✓ Uniformly distributed data do not help
  - ✓ Must use properties of  $\Pr(x)$

## Cluster Assumption

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

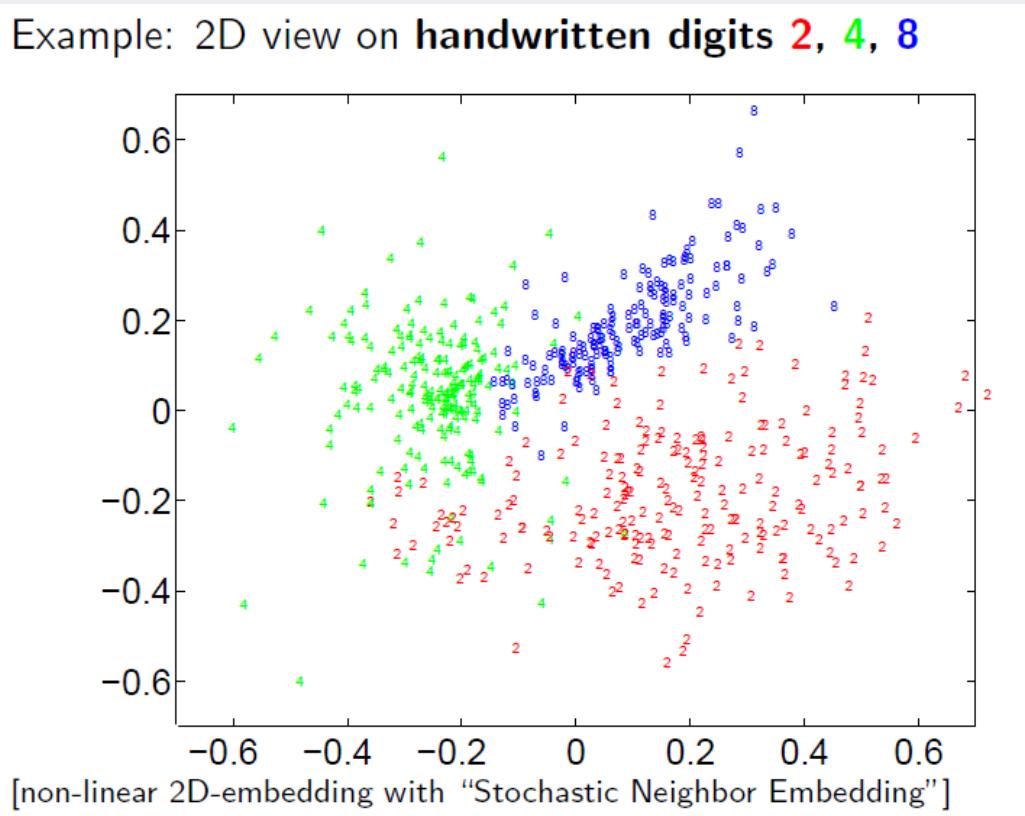


vs



# Semi-supervised Learning

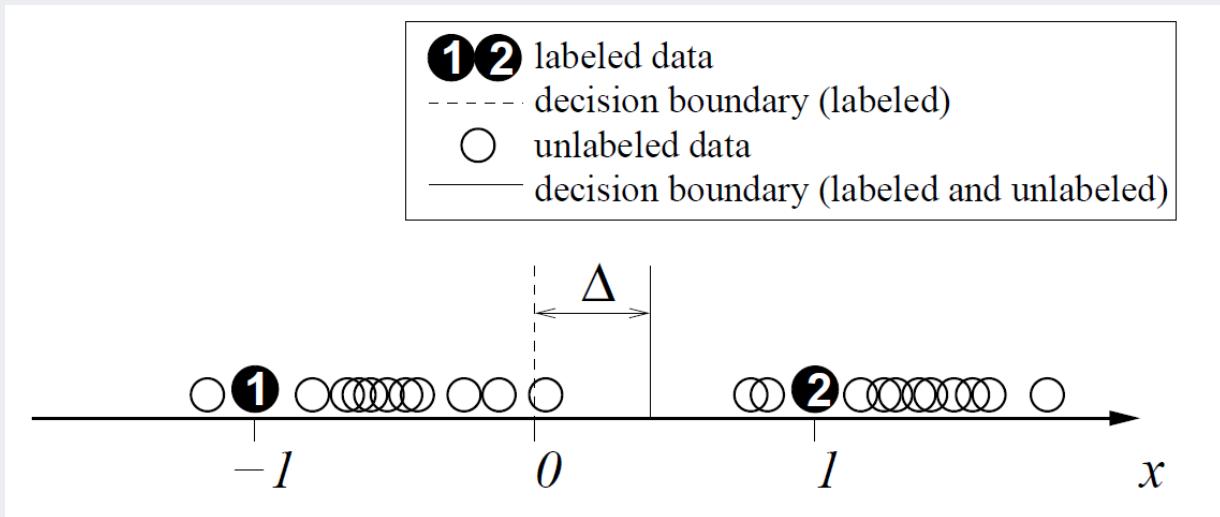
- Why would unlabeled data be useful at all?
  - ✓ The cluster assumption seems to hold for many real data sets
  - ✓ Many SSL algorithms (implicitly) make use of it



# Semi-supervised Learning

Zhu (2007)

- Why would unlabeled data be useful at all?
  - ✓ With and without unlabeled data: decision boundary shift



- Does unlabeled data always help?
  - ✓ Unfortunately, this is not the case, yet. (Ben-David et al. (2008) and Singl et al. (2008).)

# Semi-supervised Learning

Zhu (2007)

- Notations

- ✓ Input instance  $\mathbf{x}$ , label  $y$
- ✓ Learner  $f : \mathcal{X} \mapsto \mathcal{Y}$
- ✓ Labeled data  $(\mathbf{X}_l, \mathbf{y}_l) = \{(\mathbf{x}_{1:l}, y_{1:l})\}$
- ✓ Unlabeled data  $\mathbf{X}_u = \{(\mathbf{x}_{l+1:n})\}$  available during training
- ✓ Usually  $l \ll n$
- ✓ Test data  $\mathbf{X}_{test} = \{(\mathbf{x}_{n+1:})\}$  not available during training

- SSL vs. Transductive learning

Semi-supervised learning

is ultimately applied to the test data (inductive).

Transductive learning

is only concerned with the unlabeled data.

# Semi-supervised Learning

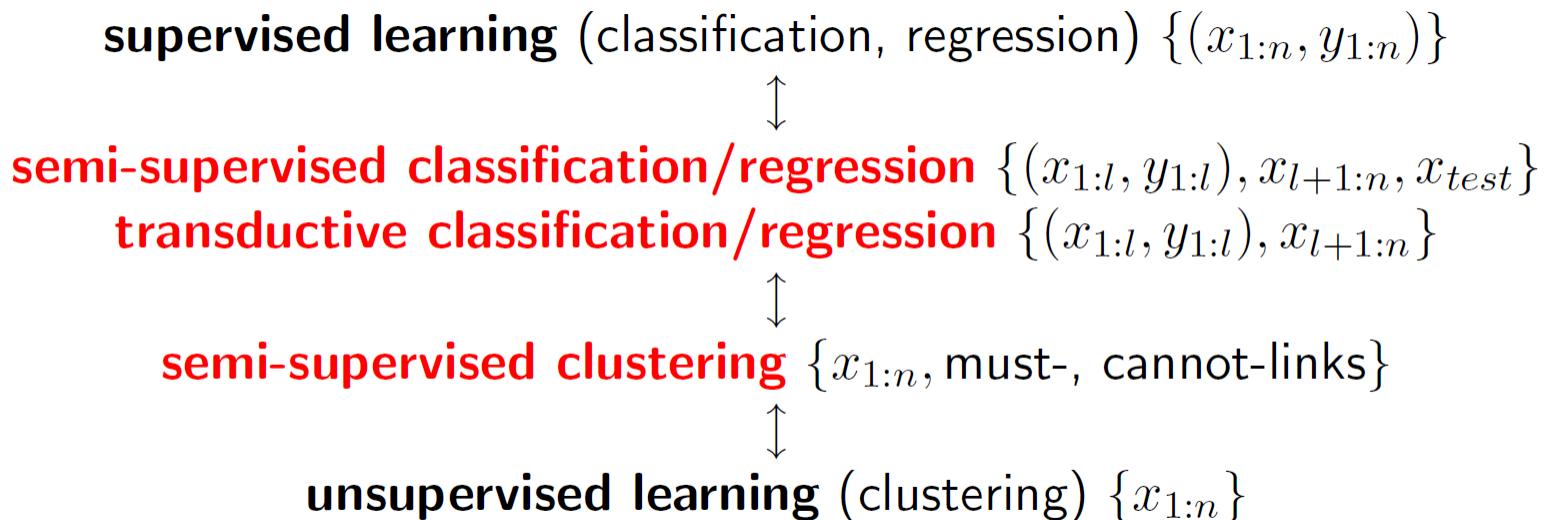
- Semi-supervised vs. Transductive

Setting	Input to the algorithm (Distribution $D$ )	Algorithm output	Performance of the algorithm	Examples
Supervised Learning	labeled examples	a function that maps points to labels	expected error on an unseen point	Support Vector Machines, AdaBoost
Unsupervised Learning	unlabeled examples	a function that maps points to labels	expected error on an unseen point	Clustering
Semi-supervised Learning (SSL)	labeled & unlabeled examples	a function that maps points to labels	expected error on an unseen point	
Transductive Learning	labeled & unlabeled examples	labels of the unlabeled examples	average error on unlabeled examples	Transductive SVMs, Graph regularization

# Semi-supervised Learning

Zhu (2007)

- Semi-supervised vs. Transductive



# AGENDA

01 Semi-supervised Learning: Overview

---

02 Self-Training

---

03 Generative Models

---

04 Graph-based SSL

---

05 Multi-view Algorithms (Co-Training)

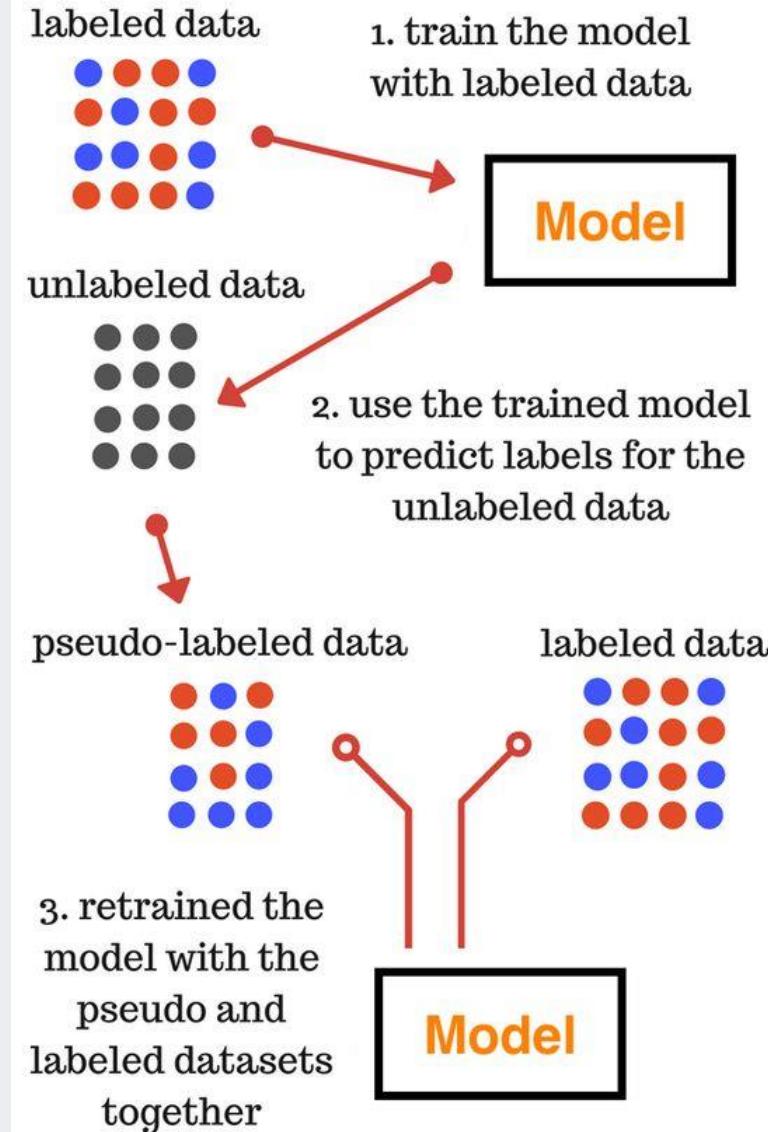
---

# Self-Training

- Assumption in the self-training
  - ✓ One's own high confidence predictions are correct
- Basic self-training algorithm
  - ✓ Train  $f$  from  $(\mathbf{X}_l, \mathbf{y}_l)$
  - ✓ Predict on  $\mathbf{x} \in \mathbf{X}_u$
  - ✓ Add  $(\mathbf{x}, f(\mathbf{x}))$  to labeled data
  - ✓ Repeat
- Variations in self-training
  - ✓ Add a few most confident  $(\mathbf{x}, f(\mathbf{x}))$  to labeled data
  - ✓ Add all  $(\mathbf{x}, f(\mathbf{x}))$  to labeled data
  - ✓ Add all  $(\mathbf{x}, f(\mathbf{x}))$  to labeled data, weigh each by confidence

# Self-Training

- Procedure



# Self-Training: Example I

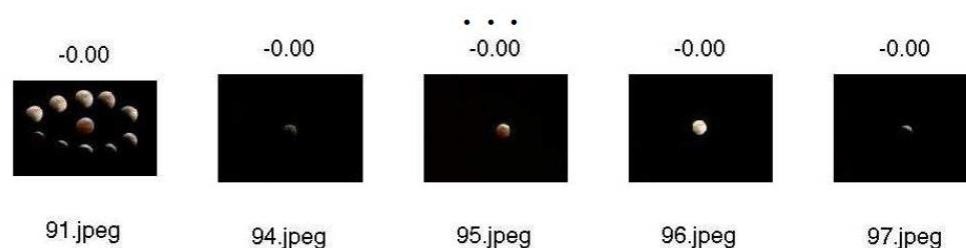
Zhu (2007)

- **Image Categorization**

1. Train a naïve Bayes classifier on the two initial labeled images



2. Classify unlabeled data, sort by confidence  $\log p(y = \text{astronomy} | x)$



# Self-Training: Example I

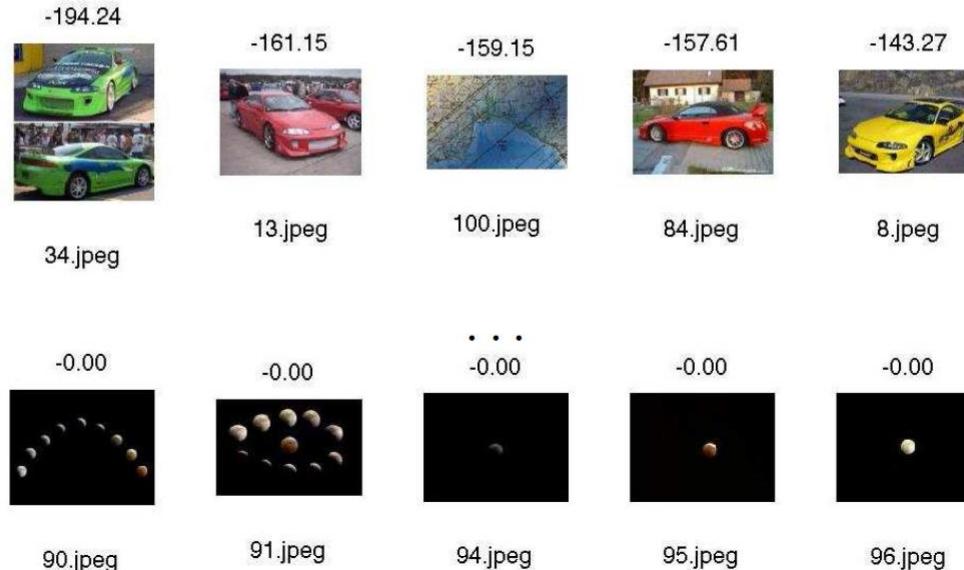
Zhu (2007)

- **Image Categorization**

3. Add the most confident images and **predicted** labels to labeled data



4. Re-train the classifier and repeat



# Self-Training: Example 2

Zhu (2009)

- Propagating 1-Nearest Neighbor

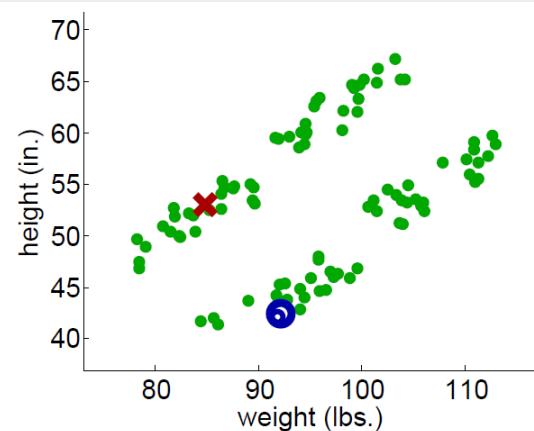
Input: labeled data  $\{(\mathbf{x}_i, y_i)\}_{i=1}^l$ , unlabeled data  $\{\mathbf{x}_j\}_{j=l+1}^{l+u}$ , distance function  $d()$ .

1. Initially, let  $L = \{(\mathbf{x}_i, y_i)\}_{i=1}^l$  and  $U = \{\mathbf{x}_j\}_{j=l+1}^{l+u}$ .
2. Repeat until  $U$  is empty:
  3. Select  $\mathbf{x} = \operatorname{argmin}_{\mathbf{x} \in U} \min_{\mathbf{x}' \in L} d(\mathbf{x}, \mathbf{x}')$ .
  4. Set  $f(\mathbf{x})$  to the label of  $\mathbf{x}$ 's nearest instance in  $L$ .  
Break ties randomly.
  5. Remove  $\mathbf{x}$  from  $U$ ; add  $(\mathbf{x}, f(\mathbf{x}))$  to  $L$ .

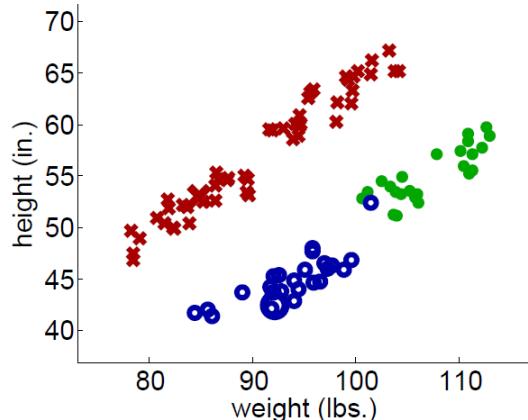
# Self-Training: Example 2

Zhu (2009)

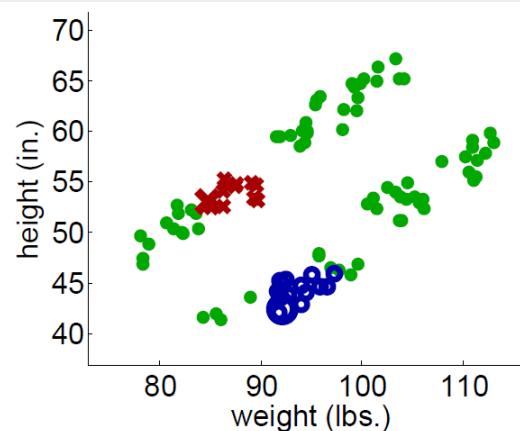
- Propagating 1-Nearest Neighbor



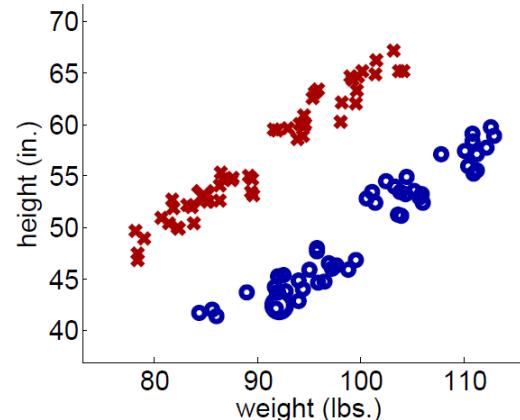
(a) Iteration 1



(c) Iteration 74



(b) Iteration 25

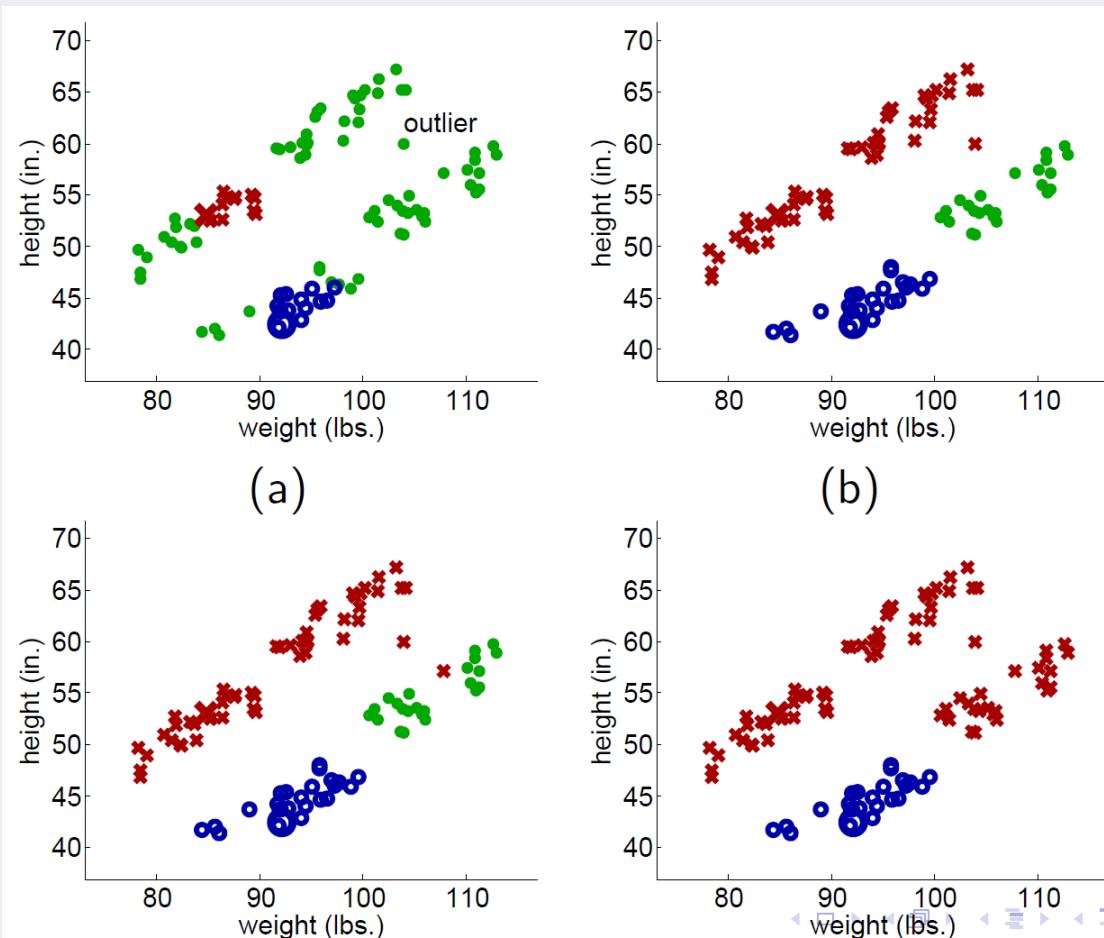


(d) Final labeling of all instances

# Self-Training: Example 2

Zhu (2009)

- Propagating 1-Nearest Neighbor (with a single outlier)



# Self-Training: Summary

- **Advantages**

- ✓ The simplest semi-supervised learning method
- ✓ A wrapper method, applies to existing (complex) classifiers
- ✓ Often used in real tasks like natural language processing

- **Disadvantages**

- ✓ Early mistakes could reinforce themselves
- ✓ Cannot say too much in terms of convergence

# AGENDA

01 Semi-supervised Learning: Overview

---

02 Self-Training

---

03 Generative Models

---

04 Graph-based SSL

---

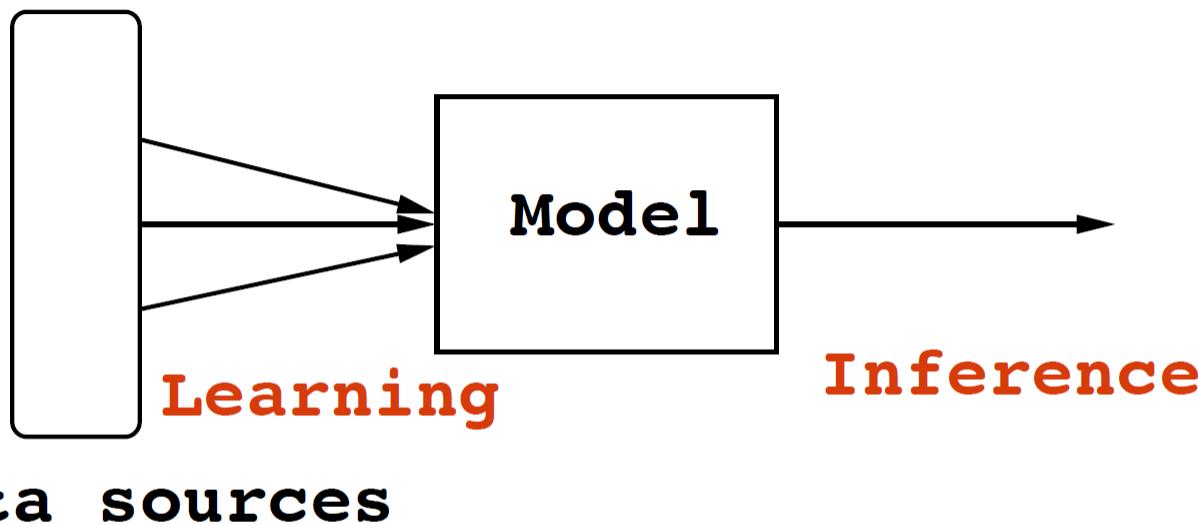
05 Multi-view Algorithms (Co-Training)

---

# Discriminative vs. Generative

Choi (2015)

## Environments



**Machine Learning:** A scientific discipline that is concerned with the design and development of **algorithms** that allow computers to **learn from empirical data** (sensor data or database) and to make **predictions**.

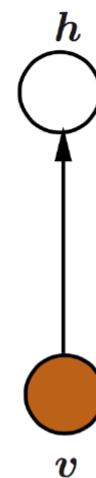
# Discriminative vs. Generative

Choi (2015)

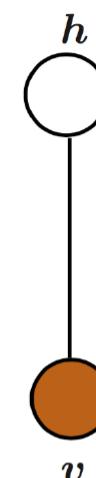
Generative model



Discriminative model



Undirected model



- ▶ Generative model: Joint distribution  $p(\mathbf{v}, \mathbf{h}) = p(\mathbf{v}|\mathbf{h})p(\mathbf{h})$
- ▶ Discriminative model: Directly model  $p(\mathbf{h}|\mathbf{v})$
- ▶ Undirected model: Energy-based model

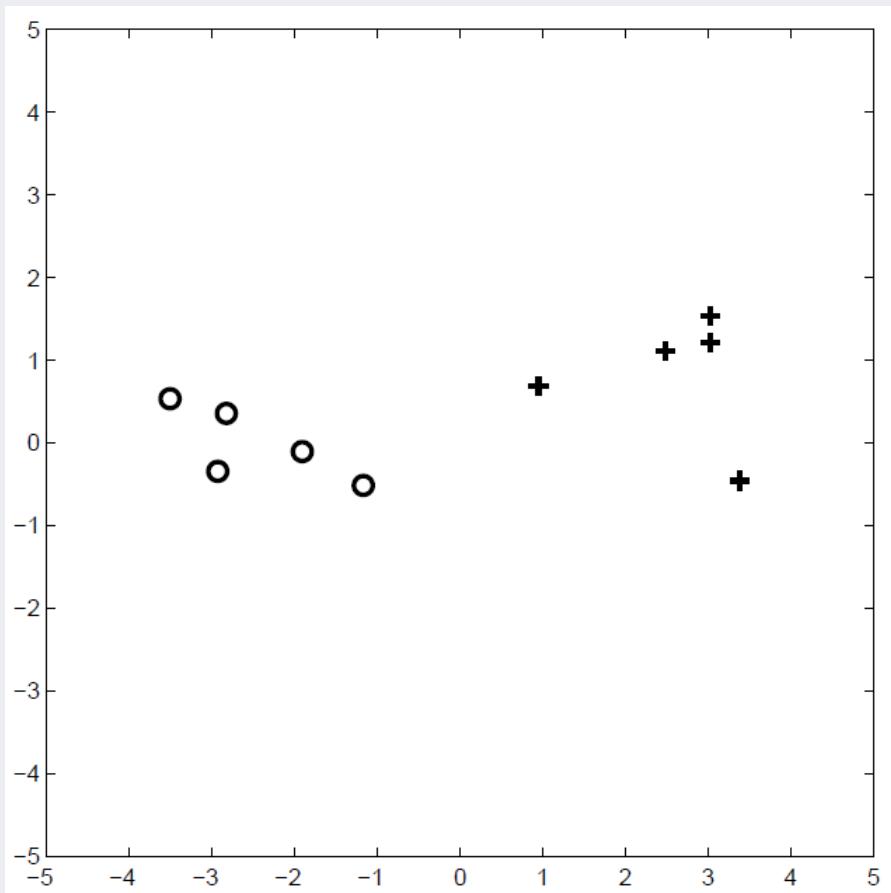
$$p(\mathbf{v}, \mathbf{h}) = \frac{\exp \{-E (\mathbf{v}, \mathbf{h})\}}{\sum_{\mathbf{v}', \mathbf{h}'} \exp \{-E (\mathbf{v}', \mathbf{h}')\}}$$

# Generative Models

Zhu (2007)

- Example

✓ Assuming each class has a Gaussian distribution, what is the decision boundary?



Model parameters :

$$\theta = \{w_1, w_2, \mu_1, \mu_2, \Sigma_1, \Sigma_2\}$$

$$p(x, y|\theta) = p(y|\theta)p(x|y, \theta)$$

$$= w_y \mathcal{N}(x; \mu_y, \Sigma_y)$$

Classification :

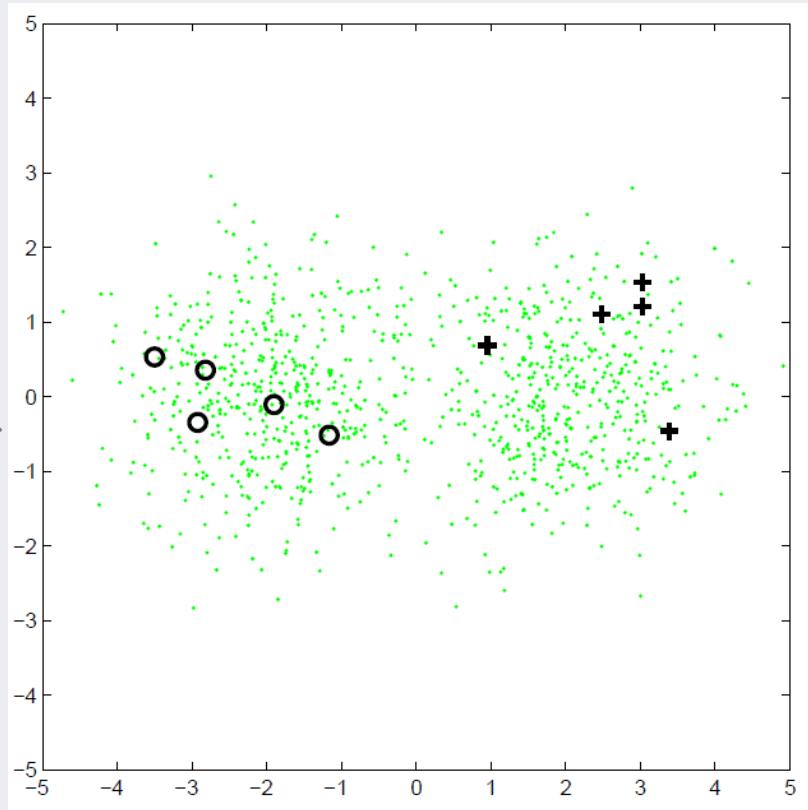
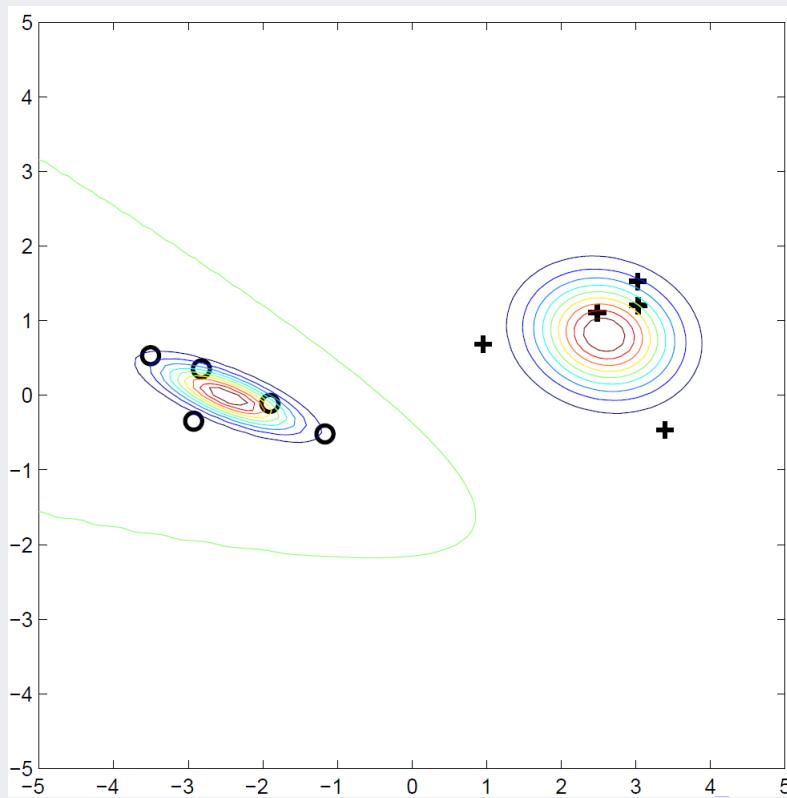
$$p(y|x, \theta) = \frac{p(x, y|\theta)}{\sum_{y'} p(x, y'|\theta)}$$

# Generative Models

Zhu (2007)

- Example

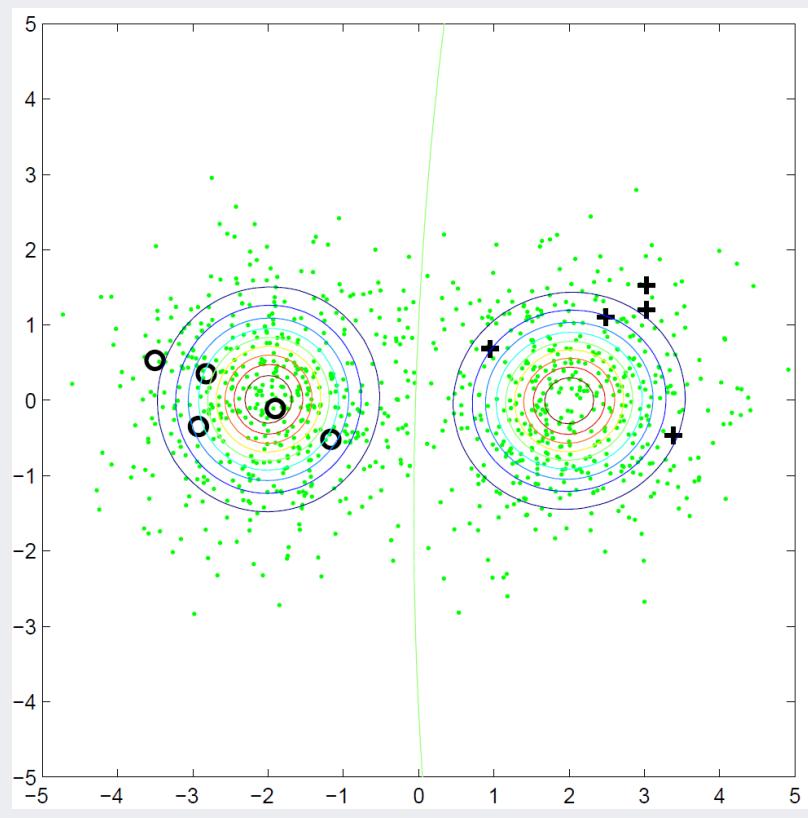
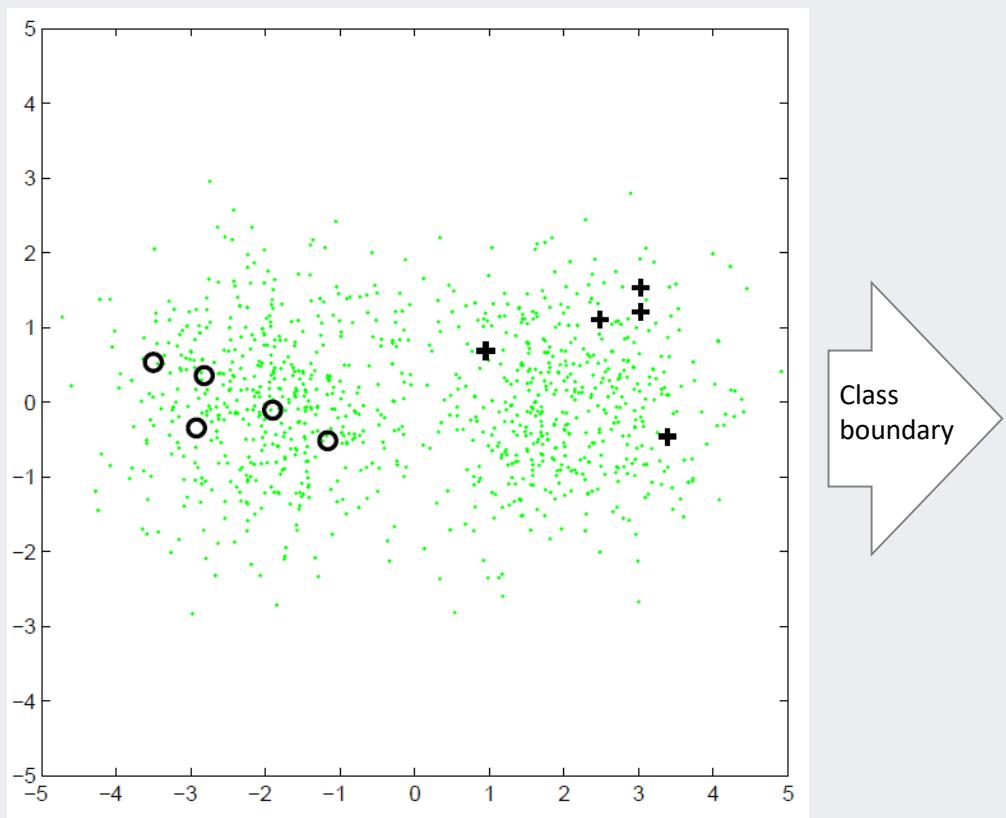
- ✓ The most likely model, and its decision boundary:



# Generative Models

Zhu (2007)

- Example
  - ✓ The most likely model, and its decision boundary:



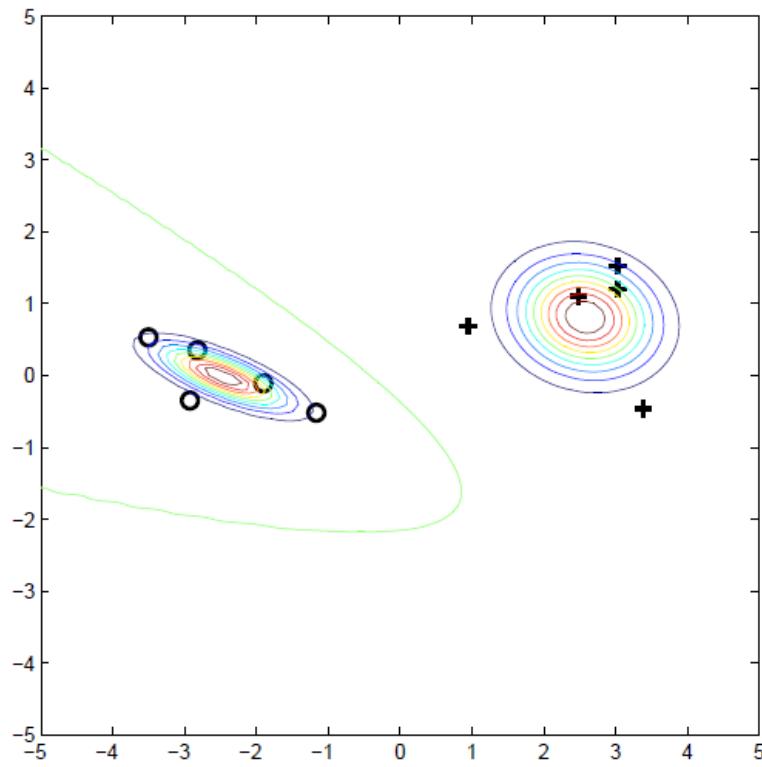
# Generative Models

Zhu (2007)

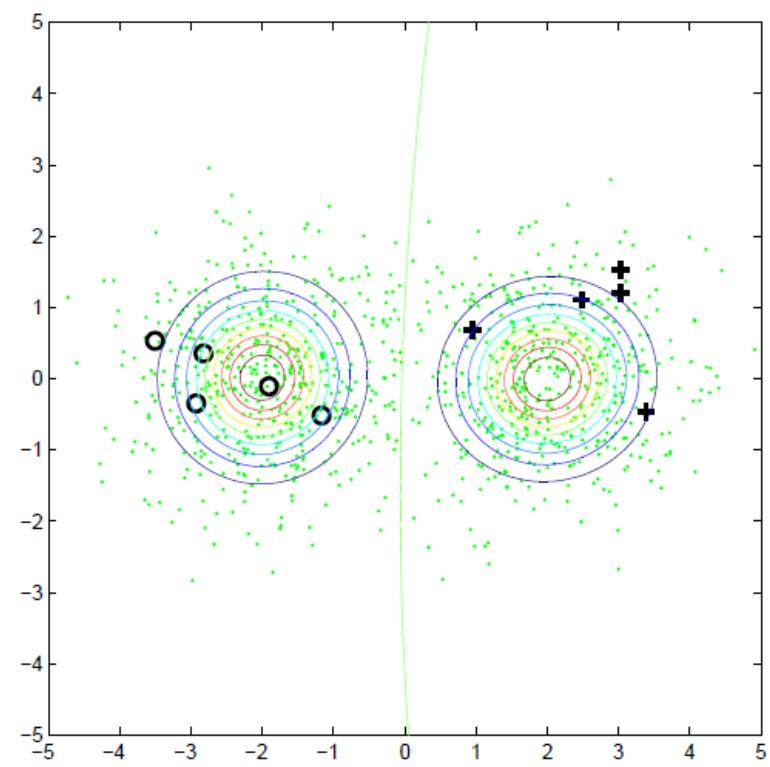
- Example

- ✓ They are different because they maximize different quantities

$$p(X_l, Y_l | \theta)$$



$$p(X_l, Y_l, X_u | \theta)$$



# Generative Models

Fox-Roberts and Rosten (2014), Kingma et al. (2014)

- Assumption: The full generative model  $p(\mathbf{X}, \mathbf{y} | \theta)$

- ✓ Generative model for semi-supervised learning:

- Quantity of interests:

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

- Find the maximum likelihood estimate (MLE) of  $\theta$ , the maximum a posteriori (MAP) estimate, or be Bayesian

- ✓ Examples of some generative models

- Mixture of Gaussian distribution (GMM): Image classification (EM algorithm)
    - Mixture of multinomial distribution (Naïve Bayes): Text categorization (EM algorithm)
    - Hidden Markov Models (HMM): Speech recognition (Baum-Welch algorithm)

# Generative Models

Zhu (2007)

- Gaussian Mixture Model

- ✓ For simplicity, consider binary classification with GMM using MLE

- Labeled data only

$$p(\mathbf{X}_l, \mathbf{y}_l | \theta) = \sum_{i=1}^l \log p(y_i | \theta) p(\mathbf{x}_i | y_i, \theta)$$

- MLE for  $\theta$  is trivial (frequency, sample mean, sample covariance)

- Labeled and unlabeled data

$$\begin{aligned} p(\mathbf{X}_l, \mathbf{y}_l, \mathbf{X}_u | \theta) &= \sum_{i=1}^l \log p(y_i | \theta) p(\mathbf{x}_i | y_i, \theta) \\ &\quad + \sum_{i=l+1}^{l+u} \log \left( \sum_{y=1}^2 p(y | \theta) p(\mathbf{x}_i | y, \theta) \right) \end{aligned}$$

- MLE is difficult because of hidden variables
    - The Expectation-Maximization (EM) algorithm is used to find a local optimum

# Generative Models

Zhu (2007)

- The EM algorithm for GMM

- ✓ Step 0: Start from MLE  $\theta = \{w, \mu, \Sigma\}_{1,2}$  on  $(\mathbf{X}_l, \mathbf{y}_l)$ , repeat
- ✓ Step 1: The **E-Step**: compute the expected label for all  $\mathbf{x} \in \mathbf{X}_u$

$$p(y|\mathbf{x}, \theta) = \frac{p(\mathbf{x}, y|\theta)}{\sum_{y'} p(\mathbf{x}, y'|\theta)}$$

- label  $p(y = 1|\mathbf{x}, \theta)$  -fraction of  $\mathbf{x}$  with class 1
  - label  $p(y = 2|\mathbf{x}, \theta)$  -fraction of  $\mathbf{x}$  with class 2
- ✓ Step 2: The **M-Step**: update MLE  $\theta$  with (now labeled)  $\mathbf{X}_u$ 
    - $w_c$  : proportion of class c
    - $\mu_c$  : sample mean of class c
    - $\Sigma_c$  : sample covariance of class c
  - Can be viewed as a special form of self-training

# Generative Models

Zhu (2007)

- The EM algorithm in general

- ✓ Set up:

- Obtain data  $\mathcal{D} = (\mathbf{X}_l, \mathbf{y}_l, \mathbf{X}_u)$
    - hidden data  $\mathcal{H} = \mathbf{y}_u$

$$p(\mathcal{D}|\theta) \sum_{\mathcal{H}} = p(\mathcal{D}, \mathcal{H}|\theta)$$

- ✓ Goal: find  $\theta$  to maximize  $p(\mathcal{D}|\theta)$

- ✓ Properties

- EM starts from an arbitrary  $\theta_0$
    - The E-step:  $q(\mathcal{H}) = p(\mathcal{H}|\mathcal{D}, \theta)$
    - The M-step: maximize  $\sum_{q(\mathcal{H})} \log p(\mathcal{D}, \mathcal{H}|\theta)$
    - EM iteratively improves  $p(\mathcal{D}|\theta)$
    - EM converges to a local maximum of  $\theta$

# Generative Models

Zhu (2007)

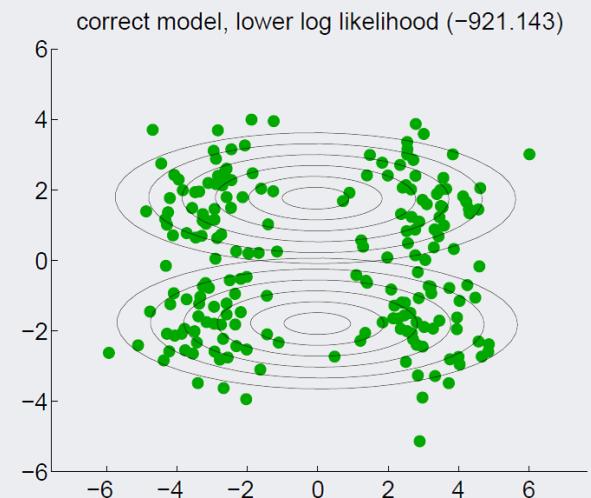
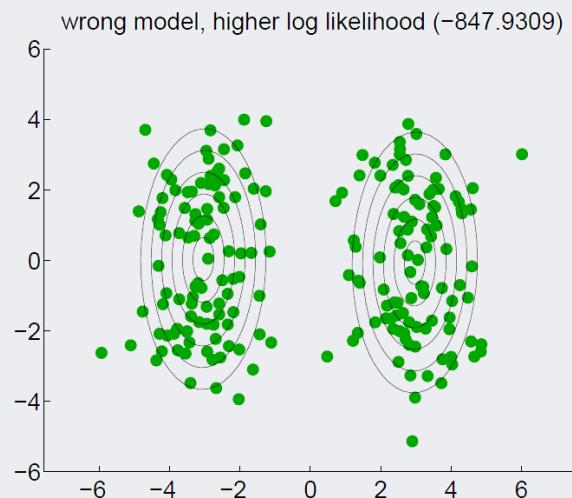
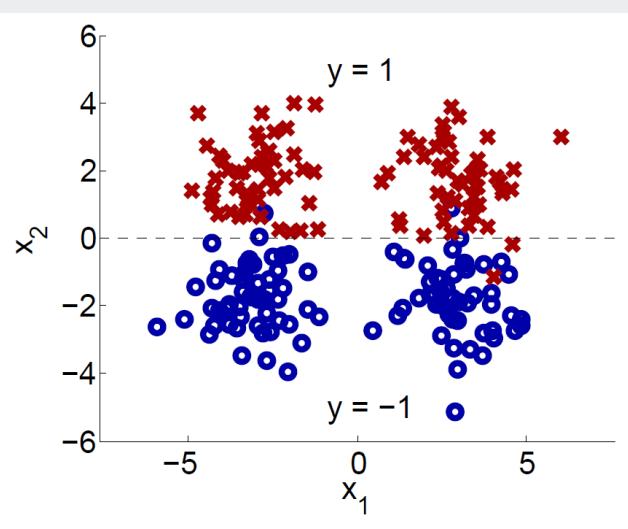
- The EM algorithm in general
  - ✓ Key is to maximize the posterior probability
  - ✓ EM is just one way to maximize it; other ways to find parameters are possible (variational approximation, direct optimization, etc.)
- Advantages
  - ✓ Clear, well-studied probabilistic framework
  - ✓ Can be extremely effective, if the model is close to correct

# Generative Models

Zhu (2007)

- Disadvantages

- ✓ Often difficult to verify the correctness of the model
- ✓ Model identifiability, local optima of the EM algorithm
- ✓ Unlabeled data may hurt if generative model is wrong



# AGENDA

01 Semi-supervised Learning: Overview

---

02 Self-Training

---

03 Generative Models

---

04 Graph-based SSL

---

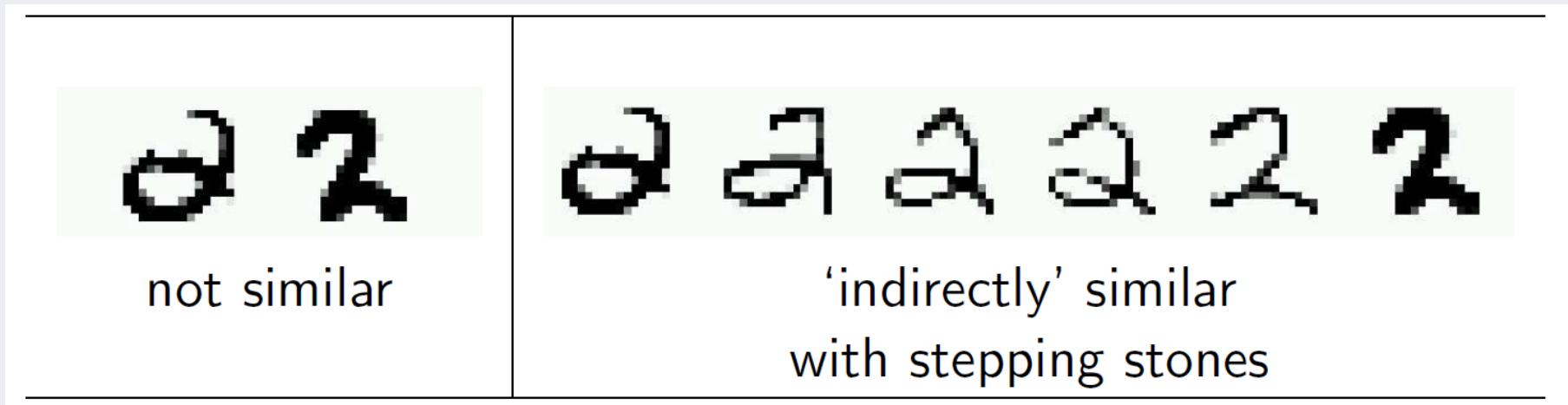
05 Multi-view Algorithms (Co-Training)

---

# Graph-based SSL

Zhu (2007)

- Example: Handwritten digit recognition with pixel-wise Euclidean distance



- Assumption
  - ✓ A graph is given on the labeled and unlabeled data
  - ✓ Instances connected by heavy edge tend to have the same label

# Graph-based SSL

Zhu (2007)

- **Graph construction**

- ✓ Nodes:  $\mathbf{X}_l \cup \mathbf{X}_u$
- ✓ Edges: similarity weights computed from features, e.g.,
  - k-nearest-neighbor graph, unweighted (0, 1 weights)
  - fully connected graph, weight decays with distance

$$w_{ij} = \exp\left(\frac{-||\mathbf{x}_i - \mathbf{x}_j||^2}{\sigma^2}\right)$$

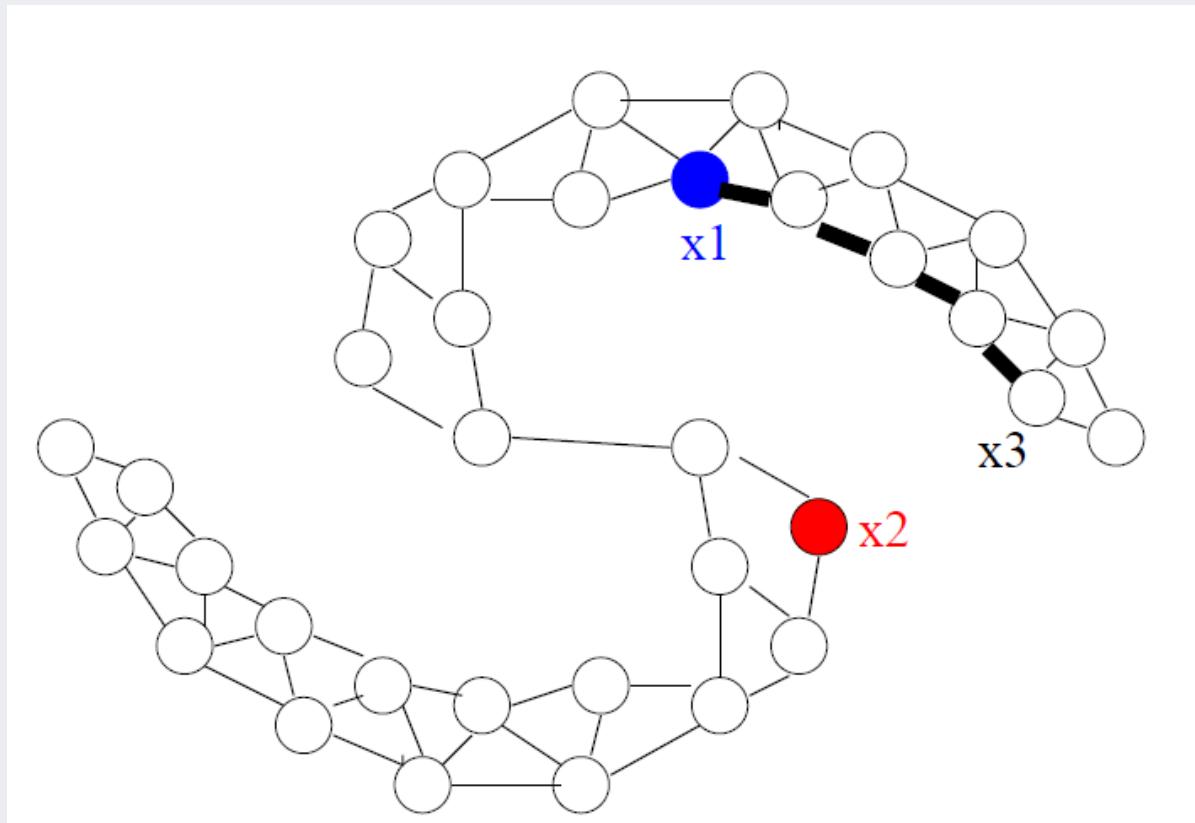
- $\varepsilon$ -radius graph

- ✓ **Assumption:** Instances connected by heavy edge tend to have the same label

# Graph-based SSL

Zhu (2007)

- **Assumption:** Instances connected by heavy edge tend to have the same label



# Graph-based SSL

Zhu (2007)

- The mincut algorithm

- ✓ Fix  $\mathbf{y}_l$ , find  $\mathbf{y}_u \in \{0, 1\}^{n-l}$  to minimize  $\sum_{i,j} w_{ij} |y_i - y_j|$

- ✓ Equivalently, solve the optimization problem

$$\min_{\mathbf{y} \in \{0,1\}^n} \infty \sum_{i=1}^l (y_i - \mathbf{y}_{li})^2 + \sum_{i,j} w_{ij} (y_i - y_j)^2$$

- ✓ Combinatorial problem, but has polynomial time solution

# Graph-based SSL

Zhu (2007)

- **Harmonic function**

✓ Relaxing discrete labels to continuous values in  $\mathbb{R}$ , the harmonic function  $f$  satisfies

$$f(\mathbf{x}_i) = y_i \text{ for } i = 1, \dots, l$$

- $f$  minimizes the energy

$$\sum_{i \sim j} w_{ij} (f(\mathbf{x}_i) - f(\mathbf{x}_j))^2$$

- the mean of a Gaussian random field
- average of neighbors

$$f(\mathbf{x}_i) = \frac{\sum_{j \sim i} w_{ij} f(\mathbf{x}_j)}{\sum_{j \sim i} w_{ij}}, \quad \forall \mathbf{x}_i \in \mathbf{X}_u$$

# Graph-based SSL

Zhu (2007)

- The graph Laplacian

- ✓ We can also compute  $f$  in closed form using the graph Laplacian

- $n \times n$  weight matrix  $\mathbf{W}$  on  $\mathbf{X}_l \cup \mathbf{X}_u$ 
  - Symmetric, non-negative
- Diagonal degree matrix:  $\mathbf{D} : \mathbf{D}_{ii} = \sum_{j=1}^n \mathbf{W}_{ij}$
- Graph **Laplacian** matrix  $\Delta = \mathbf{D} - \mathbf{W}$

Labeled graph	Degree matrix	Adjacency matrix	Laplacian matrix
	$\begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 2 & -1 & 0 & 0 & -1 & 0 \\ -1 & 3 & -1 & 0 & -1 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ -1 & -1 & 0 & -1 & 3 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix}$

# Graph-based SSL

Zhu (2007)

- The graph Laplacian

- ✓ We can also compute  $f$  in closed form using the graph Laplacian

- $n \times n$  weight matrix  $\mathbf{W}$  on  $\mathbf{X}_l \cup \mathbf{X}_u$ 
  - Symmetric, non-negative
- Diagonal degree matrix:  $\mathbf{D} : \mathbf{D}_{ii} = \sum_{j=1}^n \mathbf{W}_{ij}$
- Graph **Laplacian** matrix  $\Delta = \mathbf{D} - \mathbf{W}$
- The energy can be rewritten as

$$\frac{1}{2} \sum_{i \sim j} w_{ij} (f(\mathbf{x}_i) - f(\mathbf{x}_j))^2 = \mathbf{f}^T \Delta \mathbf{f}$$

# Graph-based SSL

Zhu (2007)

- Harmonic solution with Laplacian

- ✓ The harmonic solution minimizes energy subject to the given labels

$$\min_{\mathbf{f}} \infty \sum_{i=1}^l (f(\mathbf{x}_i) - y_i)^2 + \mathbf{f}^T \Delta \mathbf{f}$$

- Partition the Laplacian matrix  $\Delta = \begin{bmatrix} \Delta_{ll} & \Delta_{lu} \\ \Delta_{ul} & \Delta_{uu} \end{bmatrix}$

- Harmonic solution

$$\mathbf{f}_u = -\Delta_{uu}^{-1} \Delta_{ul} \mathbf{y}_l$$

- The normalized Laplacian is often used

$$\mathcal{L} = \mathbf{D}^{-\frac{1}{2}} \Delta \mathbf{D}^{-\frac{1}{2}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{W} \mathbf{D}^{-\frac{1}{2}}$$

# Graph-based SSL

Zhu (2007)

- Problems with harmonic solution

- ✓ It fixes the given labels  $\mathbf{y}_l$ 
  - What if some labels are wrong?
  - Want to be flexible and disagree with given labels occasionally
- ✓ It cannot handle new test points directly
  - $f$  is only defined on  $\mathbf{X}_u$
  - We have to add new test points to the graph, and find a new harmonic solution

- Allow  $f(\mathbf{X}_l)$  to be different from  $\mathbf{y}_l$  but penalize it
- Introduce a balance between labeled data fit and graph energy

$$\min_{\mathbf{f}} \sum_{i=1}^l (f(\mathbf{x}_i) - y_i)^2 + \lambda \mathbf{f}^T \Delta \mathbf{f}$$

# Graph-based SSL

- Solution

$$\min(\mathbf{f} - \mathbf{y})^T(\mathbf{f} - \mathbf{y}) + \lambda \mathbf{f}^T \Delta \mathbf{f} \quad \mathbf{y} = [\mathbf{y}_l; \underbrace{0; \dots; 0}_{N \text{.of unlabeled examples}}]$$

$$\frac{\partial E}{\partial \mathbf{f}} = (\mathbf{f} - \mathbf{y}) + \lambda \Delta \mathbf{f} = 0$$

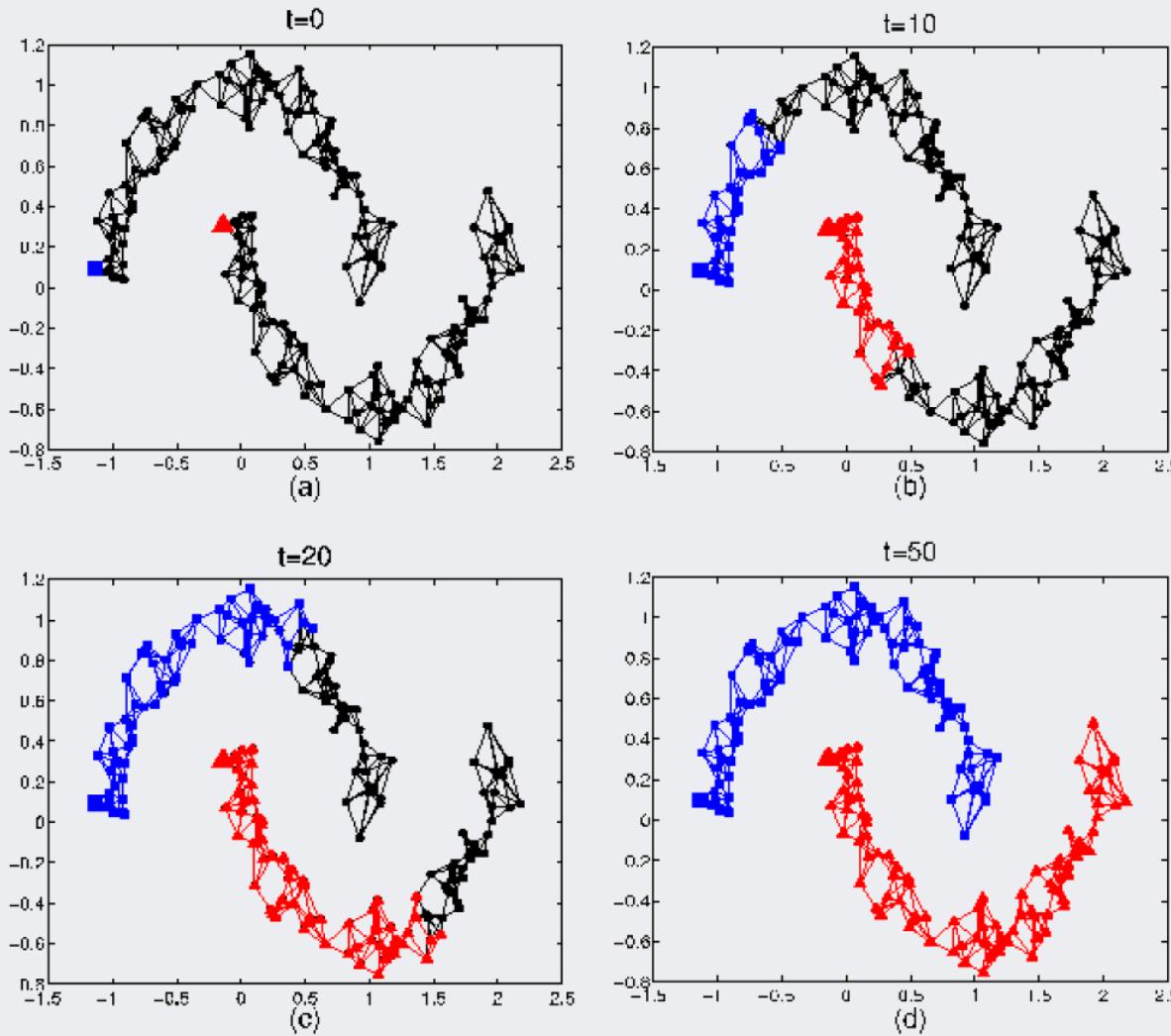
$$(\mathbf{I} + \lambda \Delta) \mathbf{f} = \mathbf{y} \Rightarrow \mathbf{f} = (\mathbf{I} + \lambda \Delta)^{-1} \mathbf{y}$$

- ✓ If  $\lambda$  is large, then the effect of  $\lambda \Delta$  increases  $\rightarrow$  more focused on the smoothness
- ✓ If  $\lambda$  is small, then the effect of  $\lambda \Delta$  decreases  $\rightarrow$  more focused on the accuracy of labeled data

# Graph-based SSL

Zien (2008)

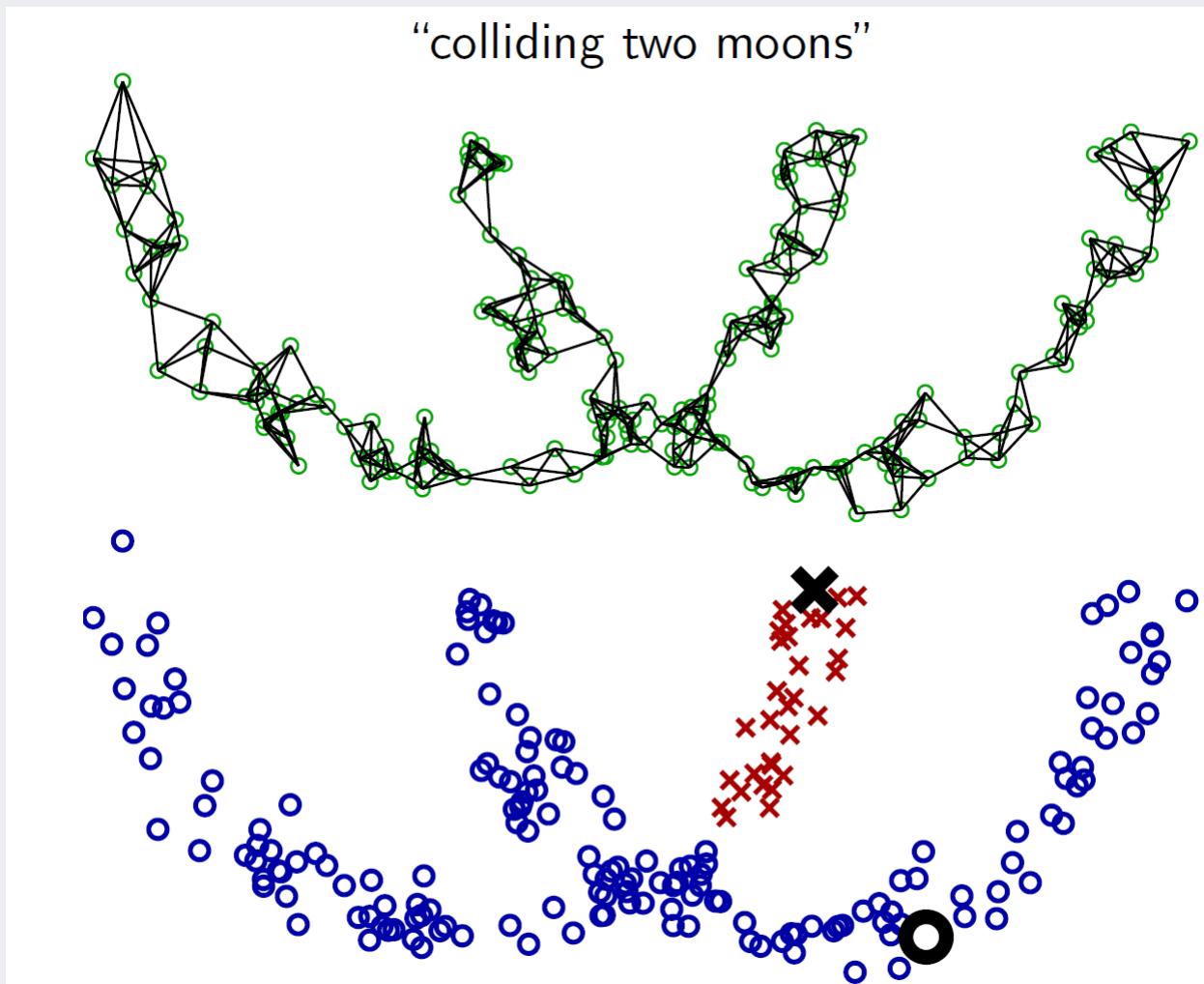
- Examples: Label Propagation



# Graph-based SSL

Zhu (2009)

- When the graph assumption is wrong



# AGENDA

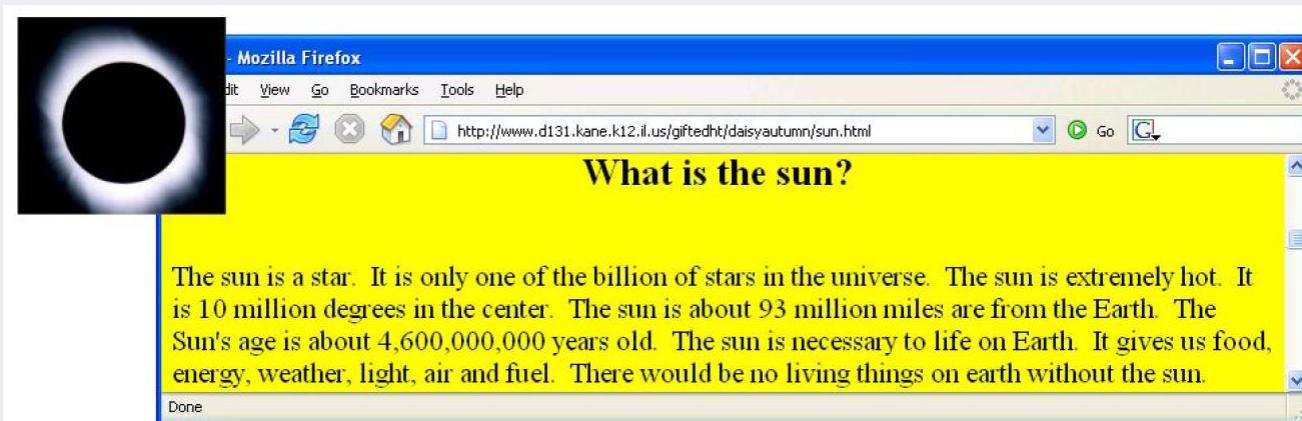
- 01 Semi-supervised Learning: Overview
- 02 Self-Training
- 03 Generative Models
- 04 Graph-based SSL
- 05 Multi-view Algorithms (Co-Training)

# Multi-view Algorithms

Blum and Mitchell (1998), Yu et al. (2011)

- Co-training

- ✓ Two views of an item: image and HTML text



# Multi-view Algorithms

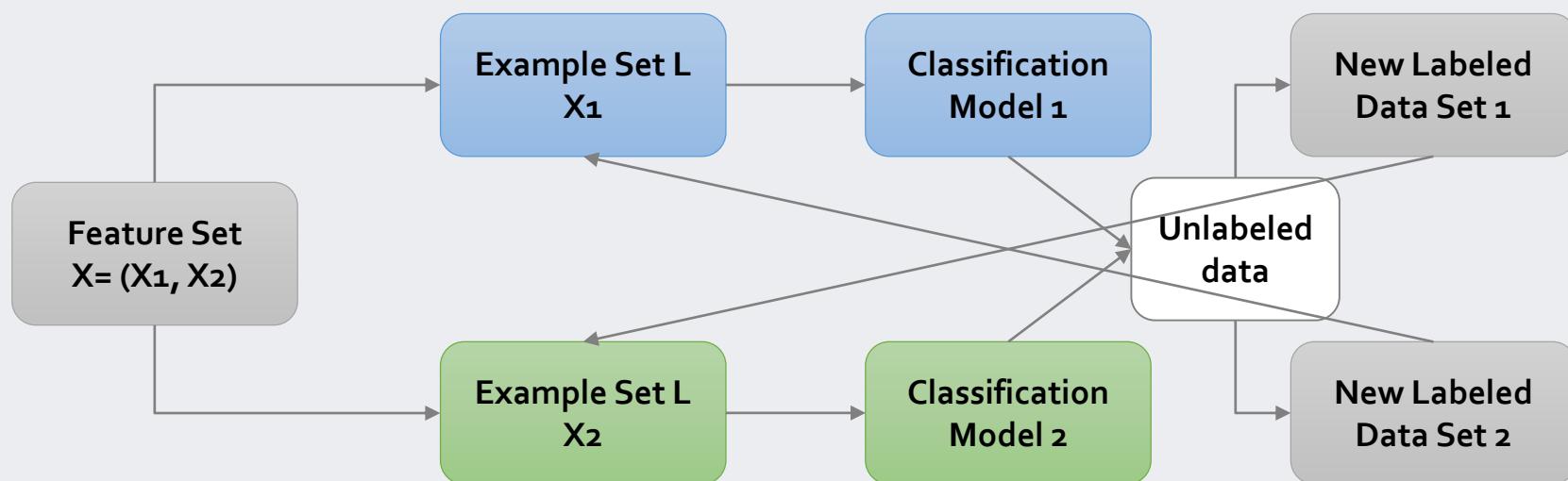
- Feature split

Each instance is represented by two sets of features  $x = [x^{(1)}; x^{(2)}]$

- $x^{(1)}$  = image features
- $x^{(2)}$  = web page text
- This is a natural feature split (or multiple views)

Co-training idea:

- Train an image classifier and a text classifier
- The two classifiers teach each other



# Multi-view Algorithms

Blum and Mitchell (1998)

- **Co-Training (Basic algorithm, for classification with Naïve Baye's Classifier)**

✓ Given

- A set  $L$  of labeled training examples
- A set of  $U$  of unlabeled examples

✓ Create a pool  $U'$  of example by choosing  $u$  examples at random from  $U$

✓ Loop for  $k$  iterations

- Use  $L$  to train a classifier  $h_1$  that considers only the  $x_1$  portion of  $x$
- Use  $L$  to train a classifier  $h_2$  that considers only the  $x_2$  portion of  $x$
- Allow  $h_1$  to label  $p$  positive and  $n$  negative examples from  $U'$
- Allow  $h_2$  to label  $p$  positive and  $n$  negative examples from  $U'$
- Add these self-labeled examples to  $L$
- Randomly choose  $2p+2n$  examples from  $U$  to replenish  $U'$

# Multi-view Algorithms

Zhou and Li (2005)

- Co-Training (with k-NN regression)

---

## ALGORITHM: COREG

INPUT: labeled example set  $L$ , unlabeled example set  $U$ , number of nearest neighbors  $k$ , maximum number of learning iterations  $T$ , distance orders  $p_1, p_2$

PROCESS:

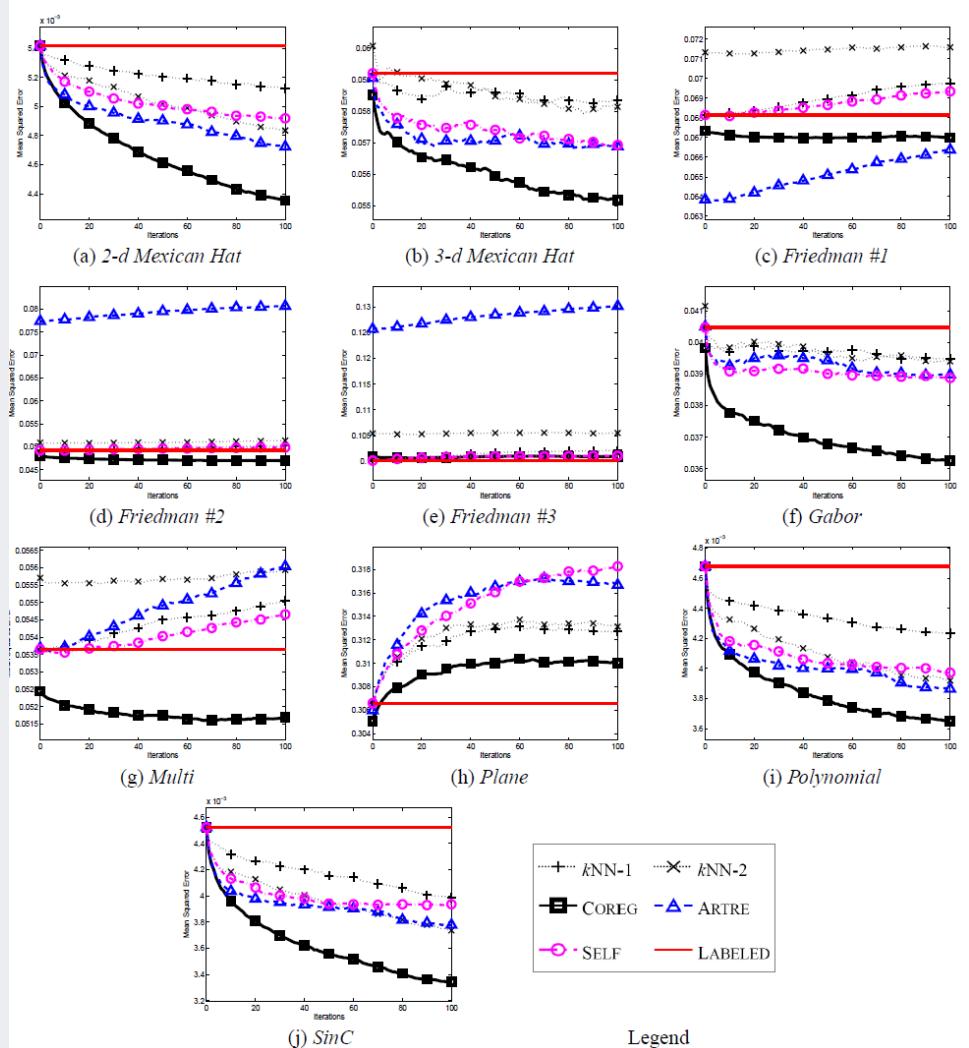
```

 $L_1 \leftarrow L; L_2 \leftarrow L$ 
Create pool  $U'$  by randomly picking examples from  $U$ 
 $h_1 \leftarrow kNN(L_1, k, p_1); h_2 \leftarrow kNN(L_2, k, p_2)$ 
Repeat for  $T$  rounds:
  for  $j \in \{1, 2\}$  do
    for each  $\mathbf{x}_u \in U'$  do
       $\hat{\mathbf{y}}_u \leftarrow h_j(\mathbf{x}_u)$ 
       $\Omega \leftarrow Neighbors(\mathbf{x}_u, k, L_j)$ 
       $h'_j \leftarrow kNN(L_j \cup \{(\mathbf{x}_u, \hat{\mathbf{y}}_u)\}, k, p_j)$ 
       $\Delta_{\mathbf{x}_u} \leftarrow \sum_{\mathbf{x}_i \in \Omega} ((\mathbf{y}_i - h_j(\mathbf{x}_i))^2 - (\mathbf{y}_i - h'_j(\mathbf{x}_i))^2)$ 
  end of for
  if there exists an  $\Delta_{\mathbf{x}_u} > 0$ 
    then  $\tilde{\mathbf{x}}_j \leftarrow \arg \max_{\mathbf{x}_u \in U'} \Delta_{\mathbf{x}_u}; \tilde{\mathbf{y}}_j \leftarrow h_j(\tilde{\mathbf{x}}_j)$ 
         $\pi_j \leftarrow \{(\tilde{\mathbf{x}}_j, \tilde{\mathbf{y}}_j)\}; U' \leftarrow U' - \pi_j$ 
    else  $\pi_j \leftarrow \emptyset$ 
  end of if
end of for
 $L_1 \leftarrow L_1 \cup \pi_2; L_2 \leftarrow L_2 \cup \pi_1$ 
if neither of  $L_1$  and  $L_2$  changes then exit
else
   $h_1 \leftarrow kNN(L_1, k, p_1); h_2 \leftarrow kNN(L_2, k, p_2)$ 
  Replenish  $U'$  by randomly picking examples from  $U$ 
end of Repeat

```

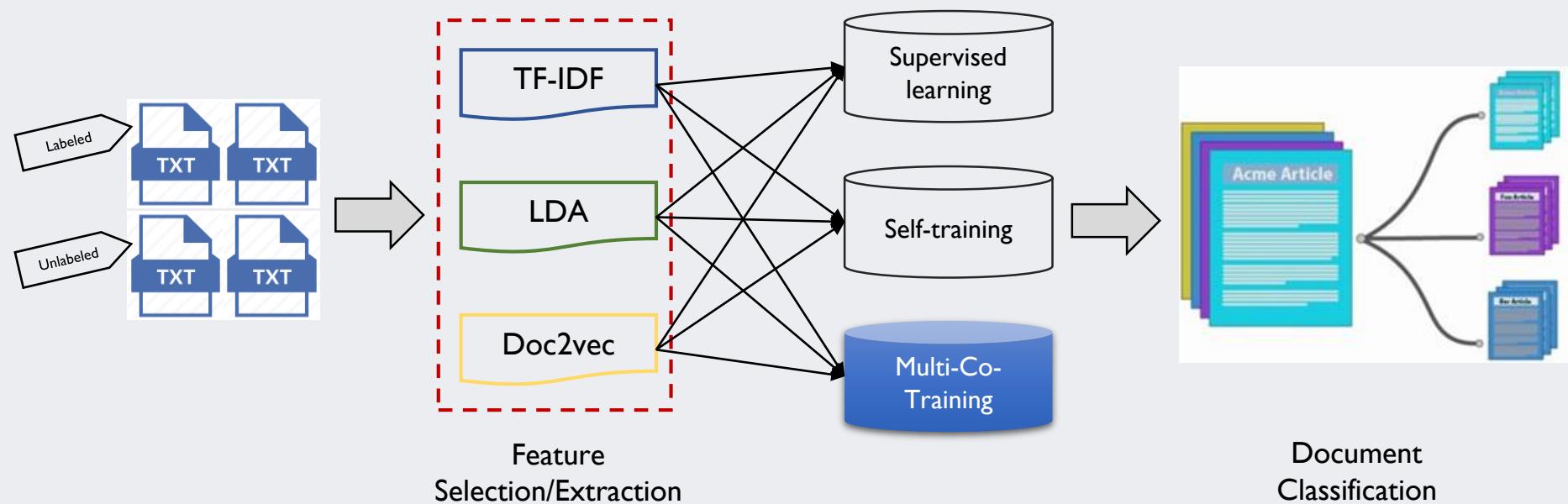
OUTPUT: regressor  $h^*(\mathbf{x}) \leftarrow \frac{1}{2} (h_1(\mathbf{x}) + h_2(\mathbf{x}))$

---



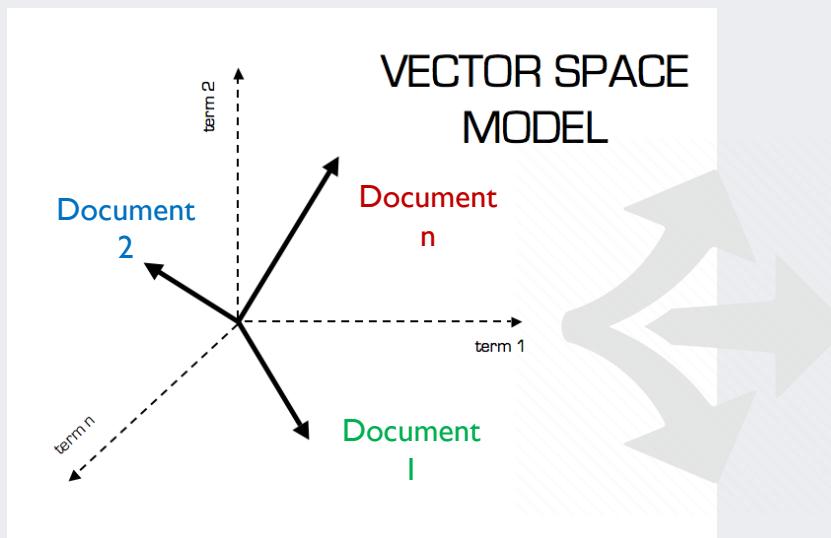
# Multi-view Algorithms

- Multi-Co-Training for Text Classification



# Multi-view Algorithms

- Multi-Co-Training for Text Classification



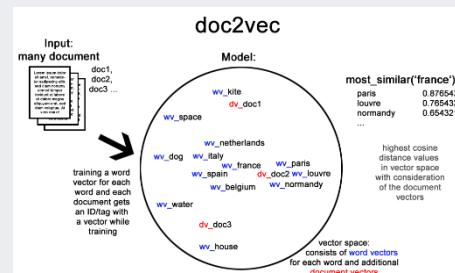
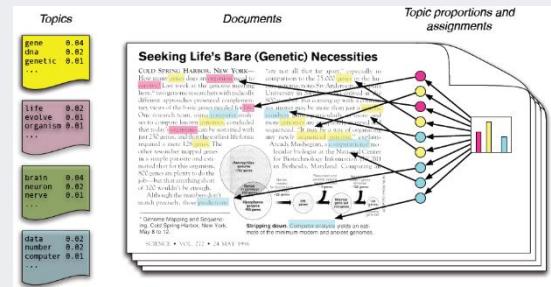
TF-IDF

LDA

Doc2vec

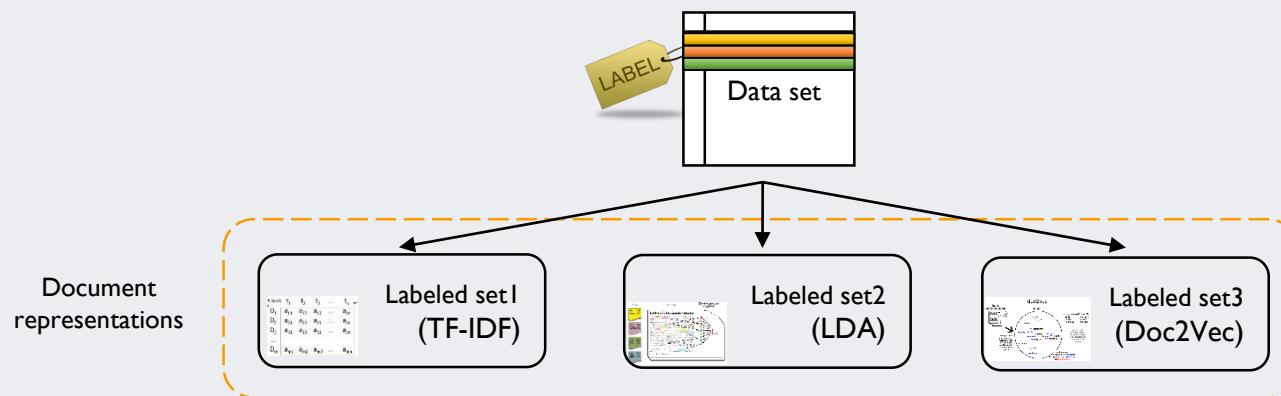
## Document1

Word	TF	N	DF	IDF	TF-IDF
This	4	3	3	1/3	1.33
is	5	2	3	1/3	1.67
an	3	3	3	1/3	1
Example	2	3	1	1	2



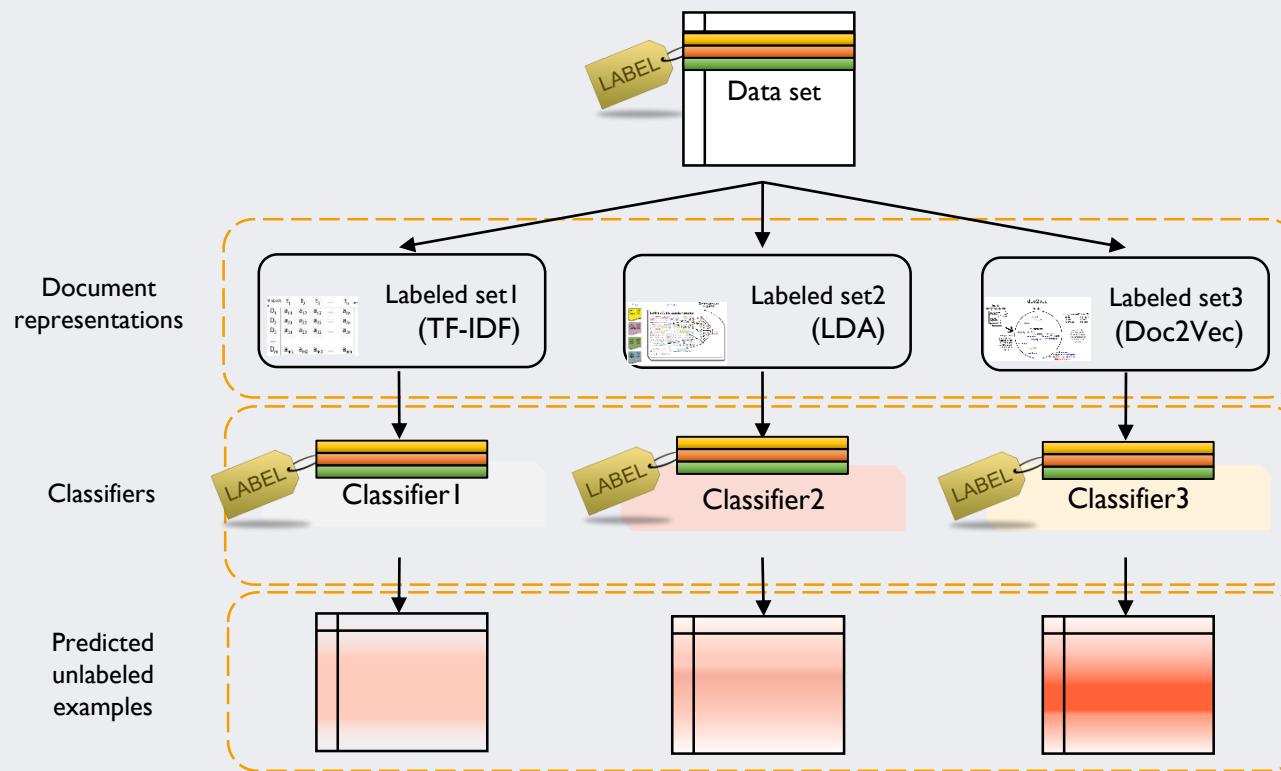
# Multi-view Algorithms

- Multi-Co-Training for Text Classification
  - ✓ Step 1) Create multi-views: TF-IDF, LDA and Doc2vec



# Multi-view Algorithms

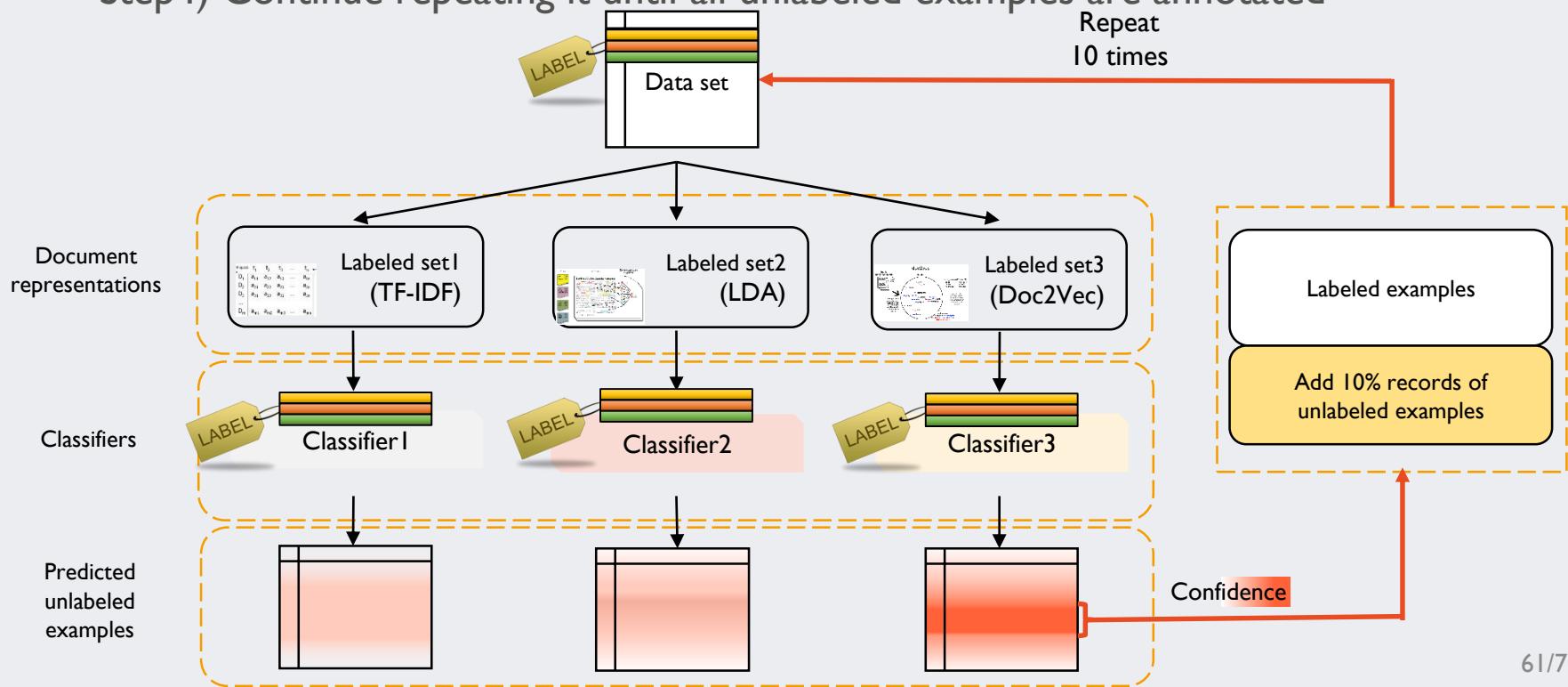
- Multi-Co-Training for Text Classification
  - ✓ Step 1) Create multi-views: TF-IDF, LDA and Doc2vec
  - ✓ Step 2) Build models and then predict unlabeled examples



# Multi-view Algorithms

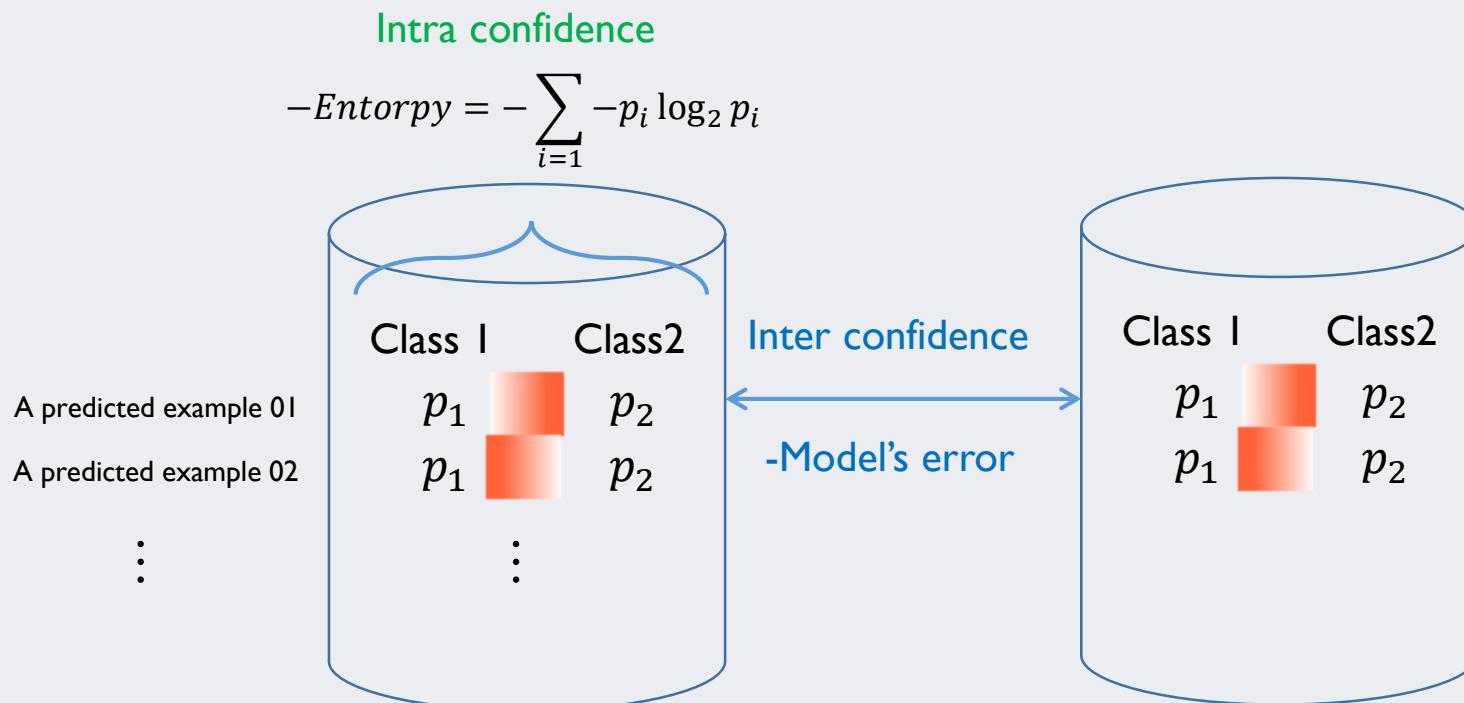
- Multi-Co-Training for Text Classification

- ✓ Step1) Create multi-views:TF-IDF, LDA and Doc2vec
- ✓ Step2) Build models and then predict unlabeled examples
- ✓ Step3) Add the predicted examples with high confidence to labeled examples
- ✓ Step4) Continue repeating it until all unlabeled examples are annotated



# Multi-view Algorithms

- Confidence measure with Naïve Bayesian
  - ✓ Intra confidence:  $-Entropy$
  - ✓ Inter confidence:  $-Training\ error$
  - ✓ Confidence measure =  $-Entropy \times -Training\ error$



# Multi-view Algorithms

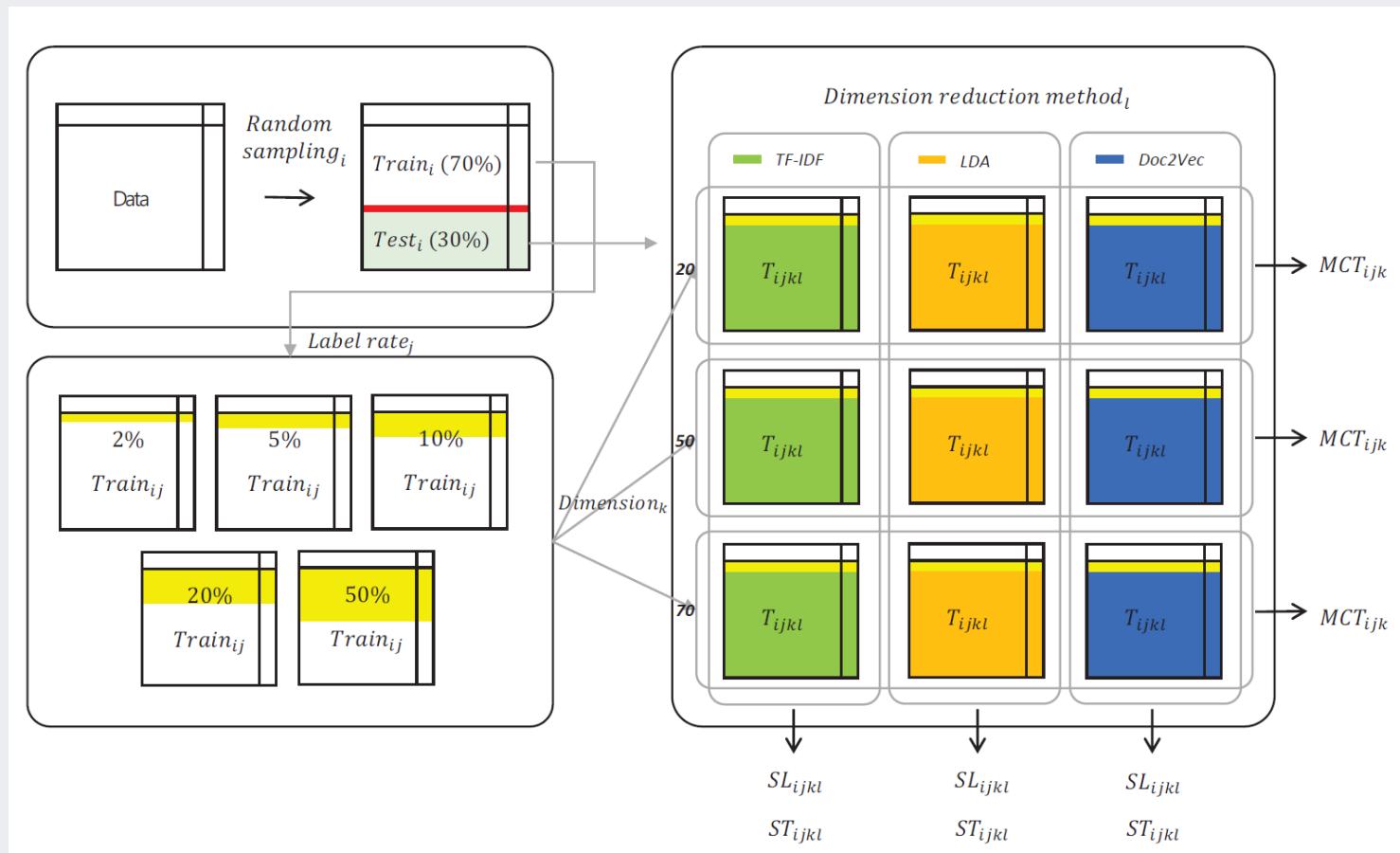
- Experiment: Data sets

Data	Description	Category	No. of documents
Economic	Whether a news article data is associated with the US economy	No : 6,458 (82.12%) Yes : 1,406 (17.88%)	7,864
20 Newsgroup	Data of 20,000 messages collected from 20 different news categories	Computer : 4,863 (30.40%) Recreation : 3,957 (24.74%) Science : 3,933 (24.59%) Talk : 3,244 (20.27%)	15,997
Ohsumed	Article-related abstracts of medical data	C04 : 2,630 (50.77%) C14 : 2,550 (49.23%)	5,180
Reuters	21,578 documents obtained from the Reuters news data	Earn : 3,953 (51.67%) Non-earn : 4,697 (48.33%)	8,650

# Multi-view Algorithms

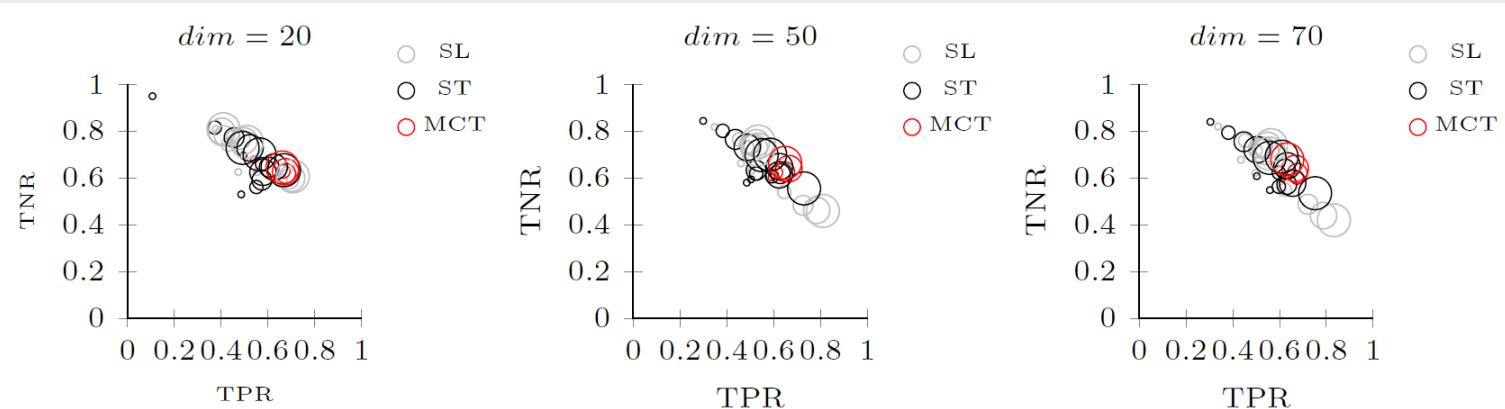
- Experiment: Evaluation procedure

- ✓ Evaluate the average and its standard deviation of Balanced Classification Rate (BCR)

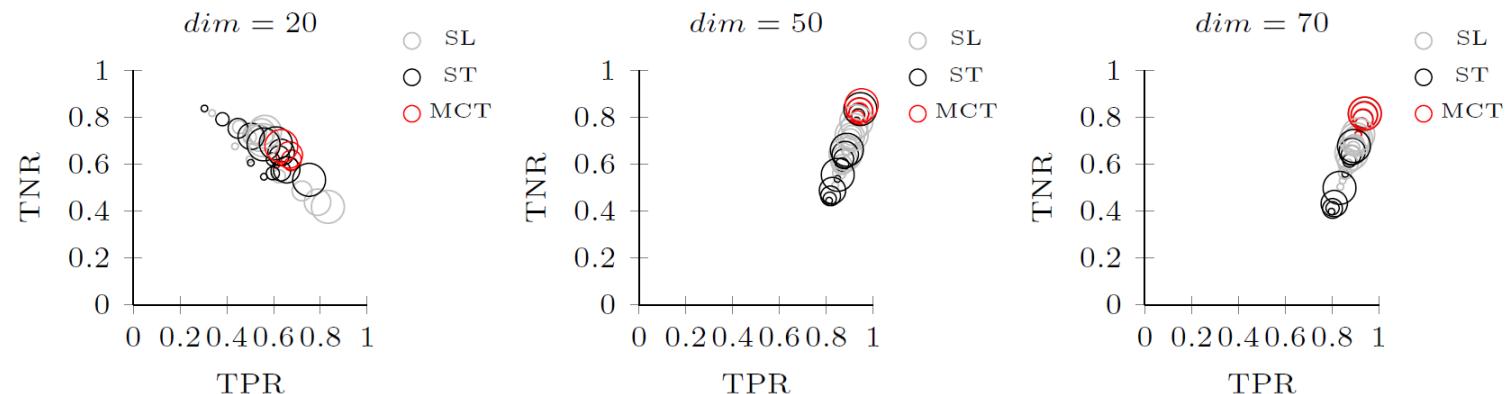


# Multi-view Algorithms

- **Experiment: Results**



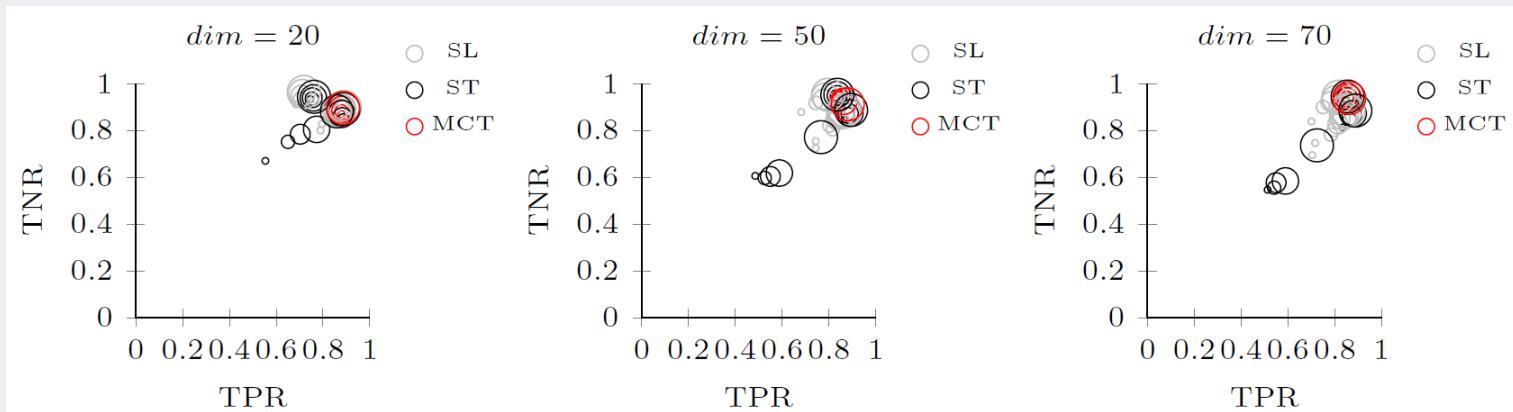
**Fig. 7** TPR—TNR plots for SL, ST, and MCT for Economic dataset (Size = label(%))



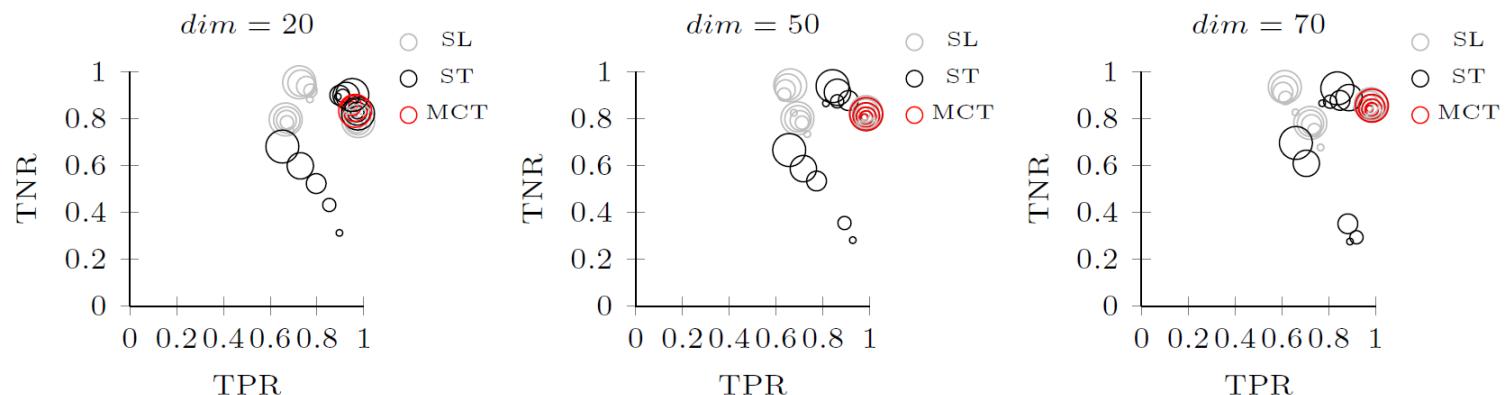
**Fig. 8** TPR—TNR plots for SL, ST, and MCT for Newsgroup dataset (Size = label(%))

# Multi-view Algorithms

- **Experiment: Results**



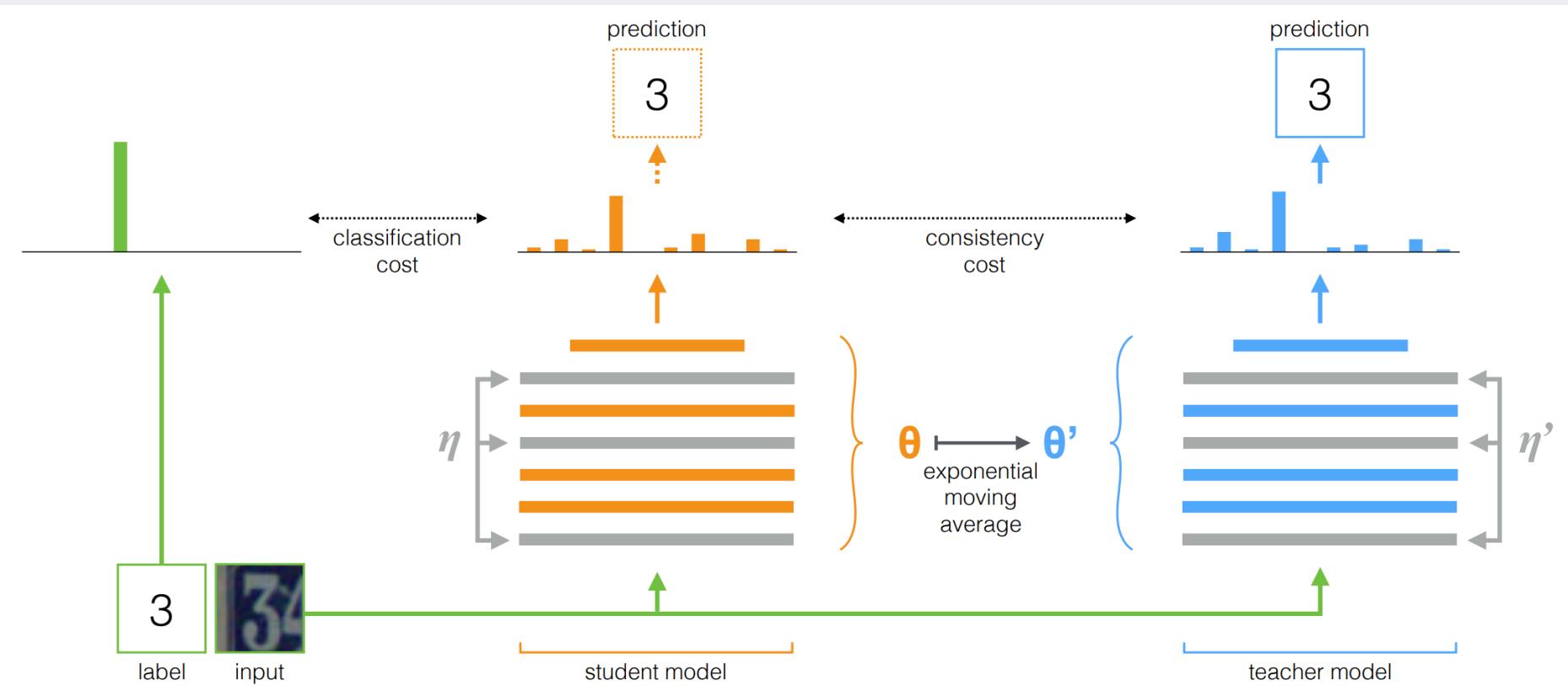
**Fig. 9** TPR—TNR plots for SL, ST, and MCT for Ohsumed dataset (Size = label(%))



**Fig. 10** TPR—TNR plots of SL, ST, and MCT for Reuters dataset (Size = label(%))

# Mean teachers are better role models

- Antti Tarvainen & Harri Valpola (2017) Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. NIPS 2017.





# References

## Research Papers

- Ben-David, S., Lu, T., & Pál, D. (2008, July). Does Unlabeled Data Provably Help? Worst-case Analysis of the Sample Complexity of Semi-Supervised Learning. In *COLT* (pp. 33-44).
- Bennett, K., & Demiriz, A. (1999). Semi-supervised support vector machines. *Advances in Neural Information processing systems*, 368-374.
- Blum, A., & Mitchell, T. (1998, July). Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory* (pp. 92-100). ACM.
- Fox-Roberts, P., & Rosten, E. (2014). Unbiased generative semi-supervised learning. *The Journal of Machine Learning Research*, 15(1), 367-443.
- Kim, D., Seo, D., Cho, S., & Kang, P. (2017+). Multi-co-training for document classification using various document representations:TF-IDF, LDA, and Doc2Vec. under review.
- Kingma, D. P., Mohamed, S., Rezende, D. J., & Welling, M. (2014). Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems* (pp. 3581-3589).
- Singh, A., Nowak, R., & Zhu, X. (2009). Unlabeled data: Now it helps, now it doesn't. In *Advances in neural information processing systems* (pp. 1513-1520).
- Yu, S., Krishnapuram, B., Rosales, R., & Rao, R. B. (2011). Bayesian co-training. *The Journal of Machine Learning Research*, 12, 2649-2680
- Zhou, Z. H., & Li, M. (2005, July). Semi-Supervised Regression with Co-Training. In *IJCAI* (Vol. 5, pp. 908-913).

# References

## Other materials

- Figures in the first page: 하상욱 단편시집 – 서울 시
- Zhu, X. (2007). Semi-Supervised Learning Tutorial. International Conference on Machine Learning (ICML 2007).
- Choi, S. (2015). Deep Learning:A Quick Overview. Deep Learning Workship. KIISE.
- Zien, A. (2008). Semi-Supervised Learning. Summer School on Neural Networks.
- Zhu, X. (2009). Tutorial on Semi-Supervised Learning. Theory and Practice of Computational Learning.