Classification
Lecture 3: Advanced Topics

Jing Gao
SUNY Buffalo

Outline

- **Basics**
  - Problem, goal, evaluation
- **Methods**
  - Decision Tree
  - Naïve Bayes
  - Nearest Neighbor
  - Rule-based Classification
  - Logistic Regression
  - Support Vector Machines
  - Ensemble methods
  - ………
- **Advanced topics**
  - Semi-supervised Learning
  - Multi-view Learning
  - Transfer Learning
  - ……

Multi-view Learning

- **Problem**
  - The same set of objects can be described in multiple different views
  - Features are naturally separated into *K* sets:

$$X = (X^1, X^2, \ldots, X^K)$$

  - Both labeled and unlabeled data are available
  - Learning on multiple views:
    - Search for labeling on the unlabeled set and target functions on *X*: $\{f_1, f_2, \ldots, f_k\}$ so that the target functions agree on labeling of unlabeled data

Learning from Two Views

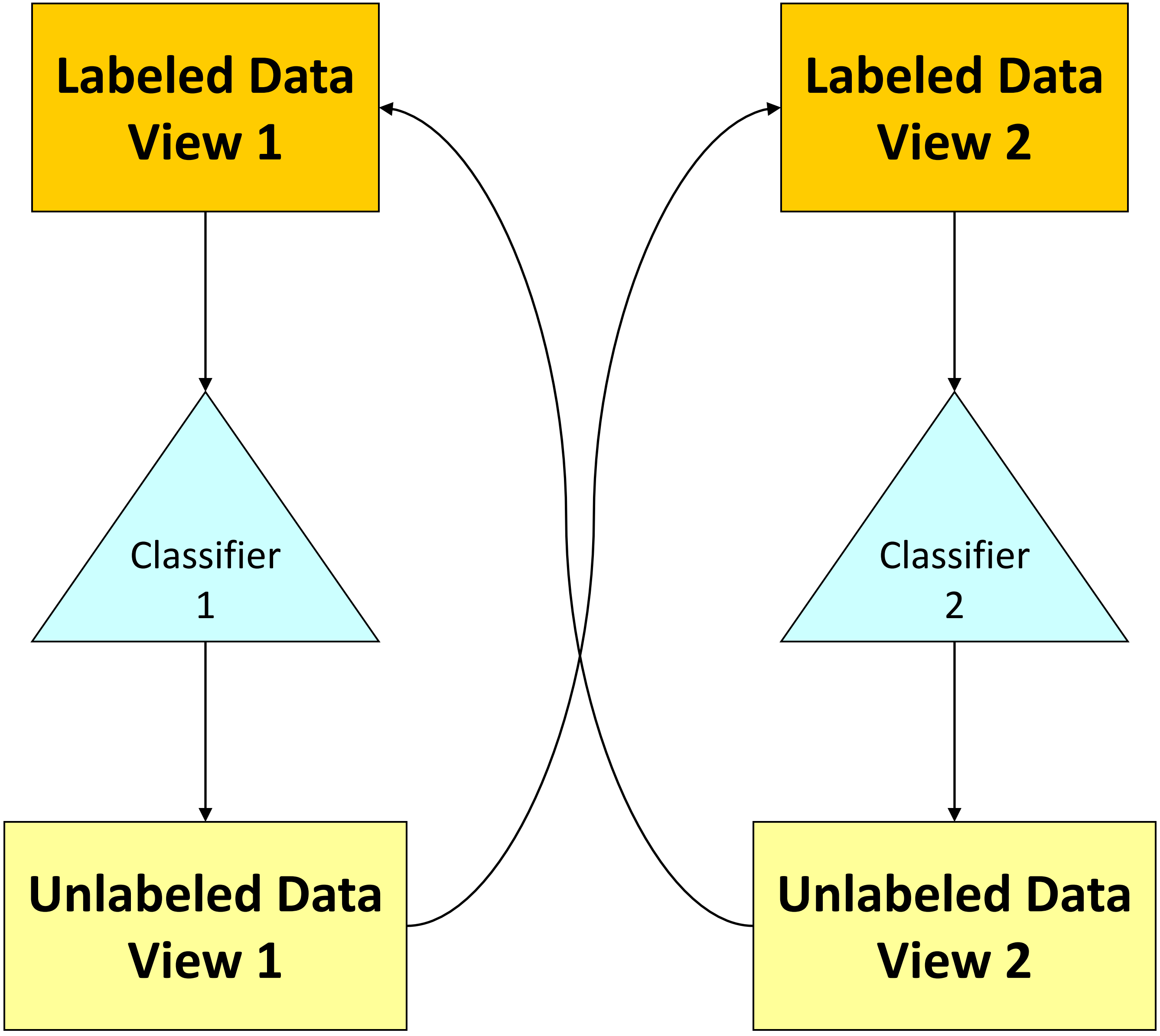
- **Input**
  - Features can be split into two sets: $X = X_1 \times X_2$
  - The two views are redundant but not completely correlated
  - Few labeled examples and relatively large amounts of unlabeled examples are available from the two views
- **Conditions**
  - Compatible --- all examples are labeled identically by the target concepts in each view
  - Uncorrelated --- given the label of any example, its descriptions in each view are independent

How It Works?

- **Conditions**
  - Compatible --- Reduce the search space to where the two classifiers agree on unlabeled data
  - Uncorrelated --- If two classifiers always make the same predictions on the unlabeled data, we cannot benefit much from multi-view learning
- **Algorithms**
  - Searching for compatible hypotheses
  - Canonical correlation analysis
  - Co-regularization

Searching for Compatible Hypotheses

- **Intuitions**
  - Two individual classifiers are learnt from the labeled examples of the two views
  - The two classifiers' predictions on unlabeled examples are used to enlarge the size of training set
  - The algorithm searches for "compatible" target functions
- **Algorithms**
  - Co-training [BlMi98]
  - Co-EM [NiGh00]
  - Variants of Co-training [GoZh00]


Labeled Data View 1
Labeled Data View 2
Classifier 1
Classifier 2
Unlabeled Data View 1
Unlabeled Data View 2

Co-Training*

Given:

- a set L of labeled training examples
- a set U of unlabeled examples

Train two classifiers from two views

Create a pool U' of

Loop for k iteratio

Select the top unlabeled examples with the most confident predictions from the other classifier

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'

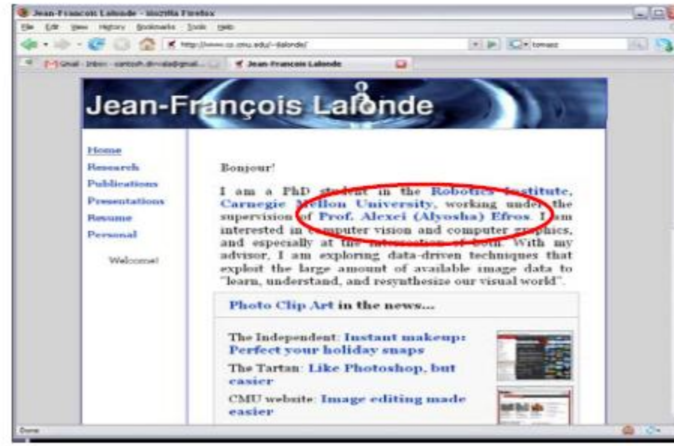
Add these self-labeled examples to L

Randomly choose $2p + 2n$ examples from U to replenish U'

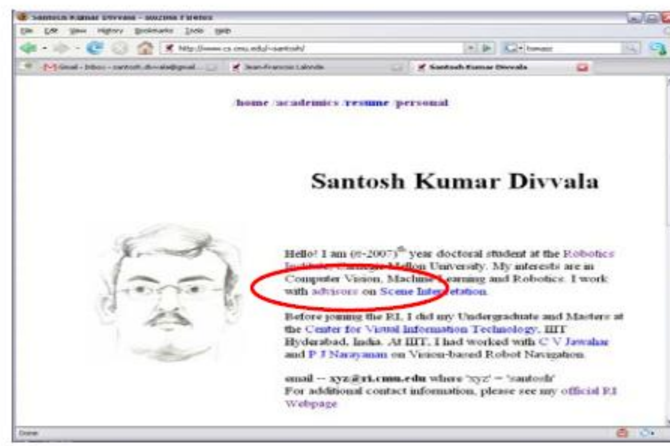
Add these self-labeled examples to the training set

*[BlMi98]

Applications: Faculty Webpages Classification



View1: Page Text



View2: Hyperlink Text

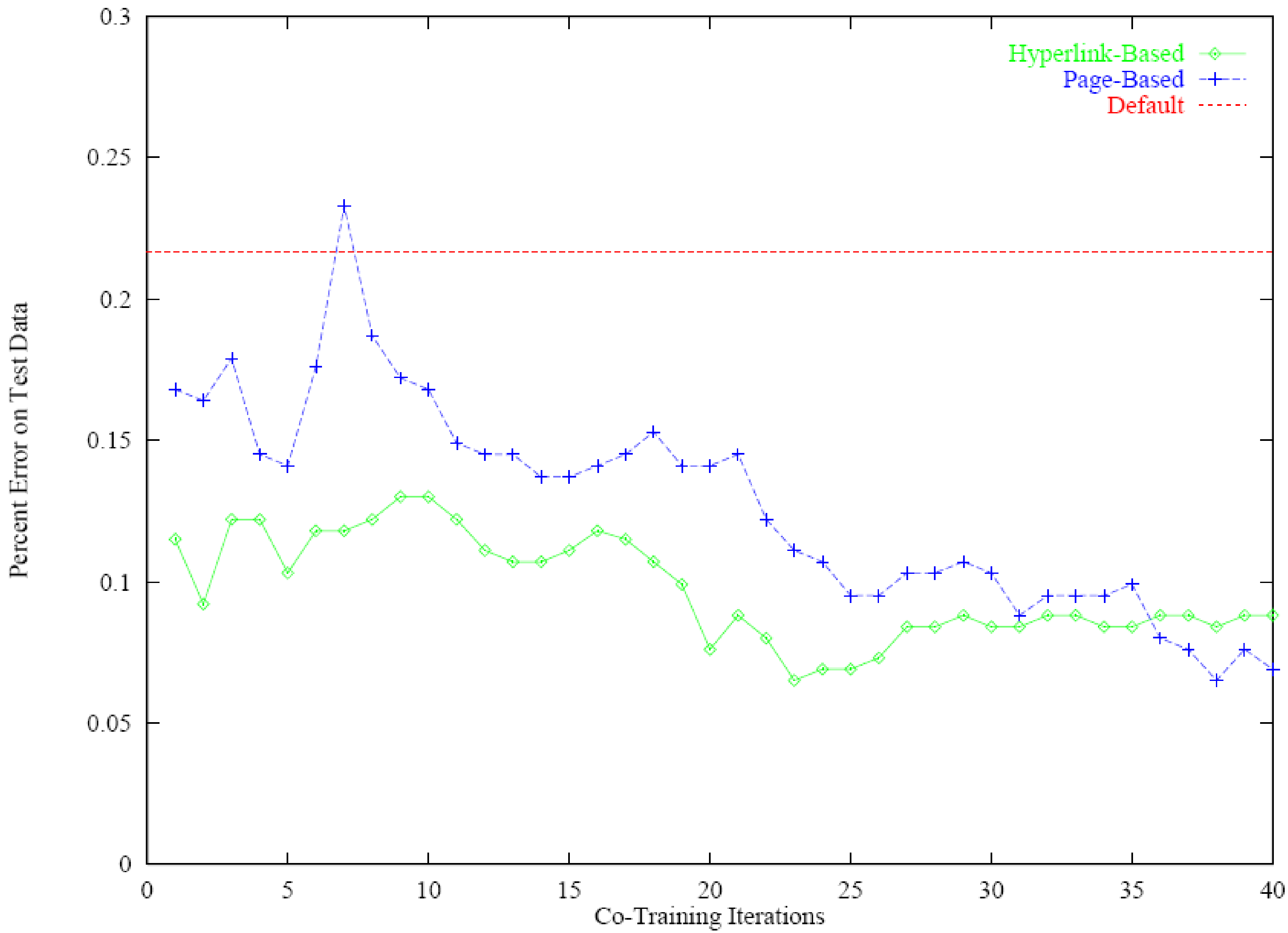


Figure 2: Error versus number of iterations for one run of co-training experiment.

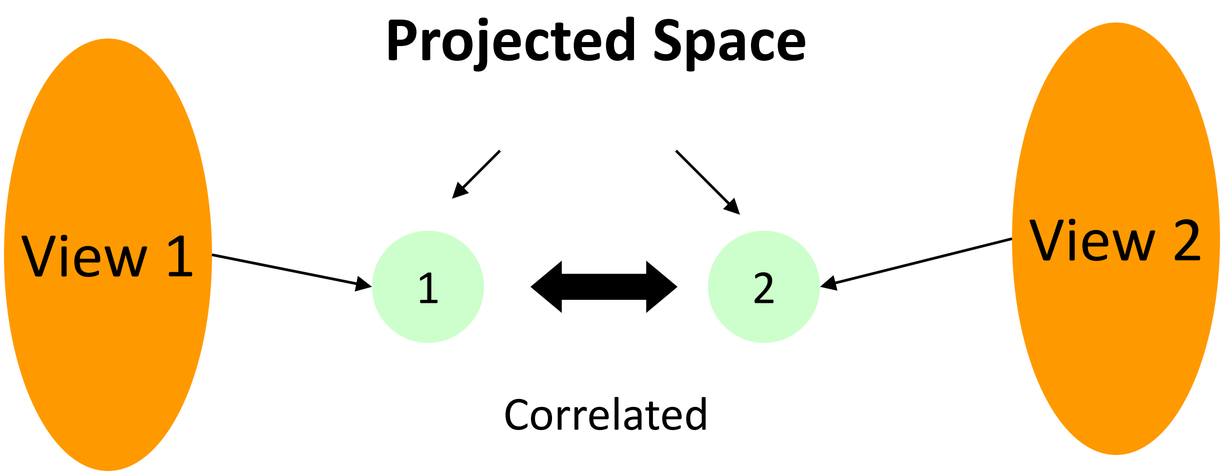
Co-EM*

- **Algorithm**
  - Labeled data set L, Unlabeled data set U, Let U_1 be empty, Let $U_2=U$
  - Iterate the following
    - Train a classifier h_1 from the feature set X_1 of L and U_1
    - Probabilistically label all the unlabeled data in U_2 using h_1
    - Train a classifier h_2 from the feature set X_2 of L and U_2
    - Let $U_1=U$, probabilistically label all the unlabeled data in U_1 using h_2
  - Combine h_1 and h_2
- **Co-EM vs. Co-Training**
  - Labeling unlabeled data: soft vs. hard
  - Selecting unlabeled data into training set: all vs. the top confident ones

*[NiGh00]

Canonical Correlation Analysis

- **Intuitions**
  - Reduce the feature space to low-dimensional space containing discriminative information
  - With compatible assumption, the discriminative information is contained in the directions that correlate between the two views
  - The goal is to maximize the correlation between the data in the two projected spaces



Algorithms

- **Co-training in the reduced spaces [ZZY07]**
  - Project the data into the low-dimensional spaces by maximizing correlations between two views
  - Compute probability of unlabeled data belonging to positive or negative classes using the distance between unlabeled data and labeled data in the new feature spaces
  - Select the top-confident ones to enhance the training set and iterate
- **SVM+Canonical Correlation Analysis [FHM+05]**
  - First reduce dimensions, then train SVM classifiers
  - Combine the two steps together

Co-Regularization Framework

- **Intuitions**
  - Train two classifiers from the two views simultaneously
  - Add a regularization term to enforce that the two classifiers agree on the predictions of unlabeled data

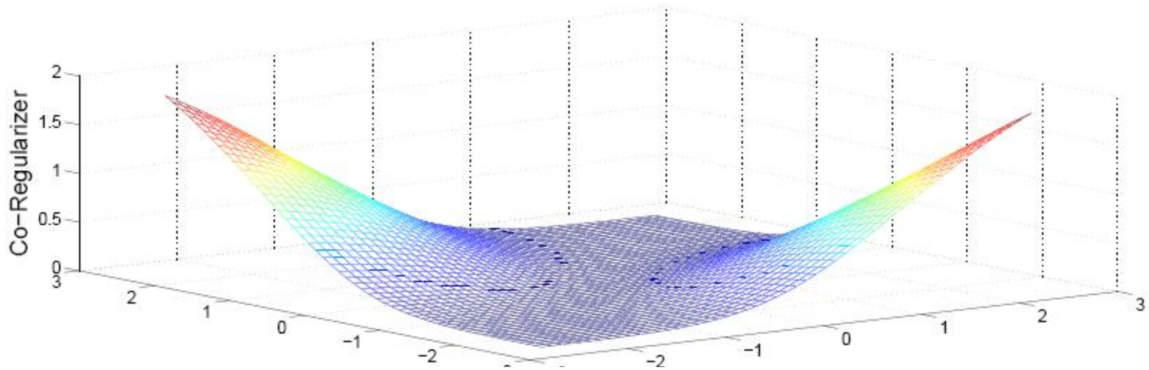
Risk of classifier 2 on view 2 of labeled data

$$\min \quad R(f_1; L_1) + R(f_2; L_2) + R(f_1, f_2; U_1, U_2)$$

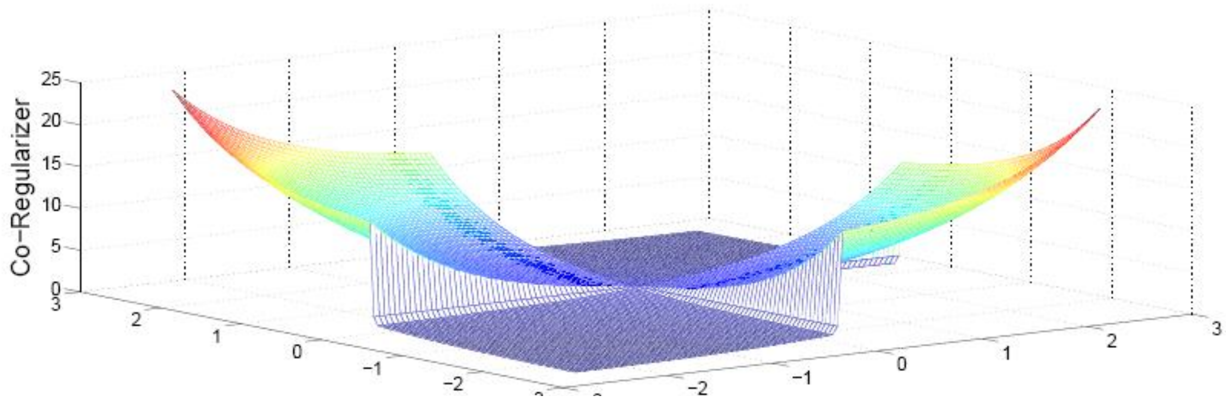
Risk of classifier 1 on view 1 of labeled data

Disagreement between two classifiers on unlabeled data

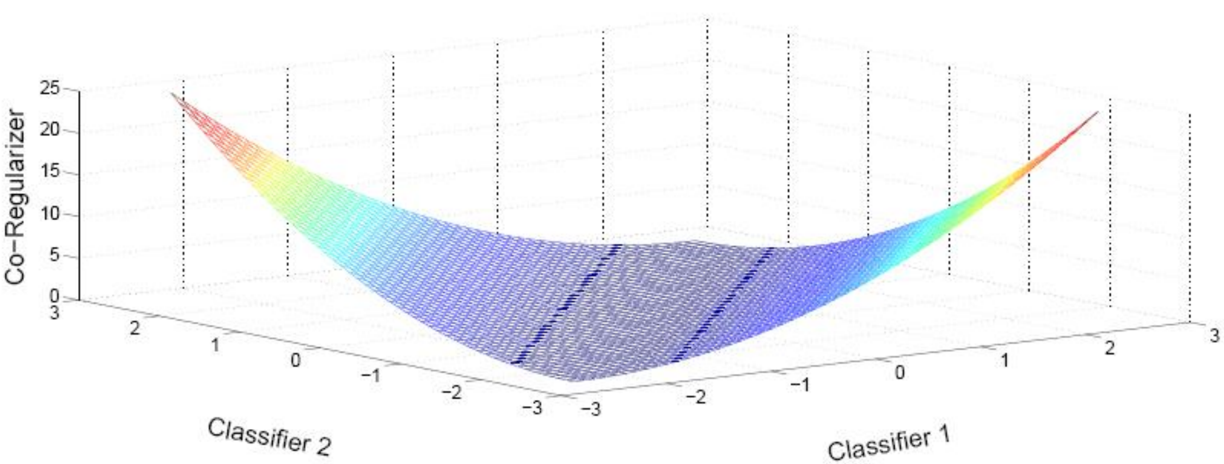
- **Algorithms**
  - Co-boosting [CoSi99]
  - Co-regularized least squares and SVM [SNB05]
  - Bhattacharyya distance regularization [GGB+08]



Bhattacharyya distance



Exponential loss



Least square

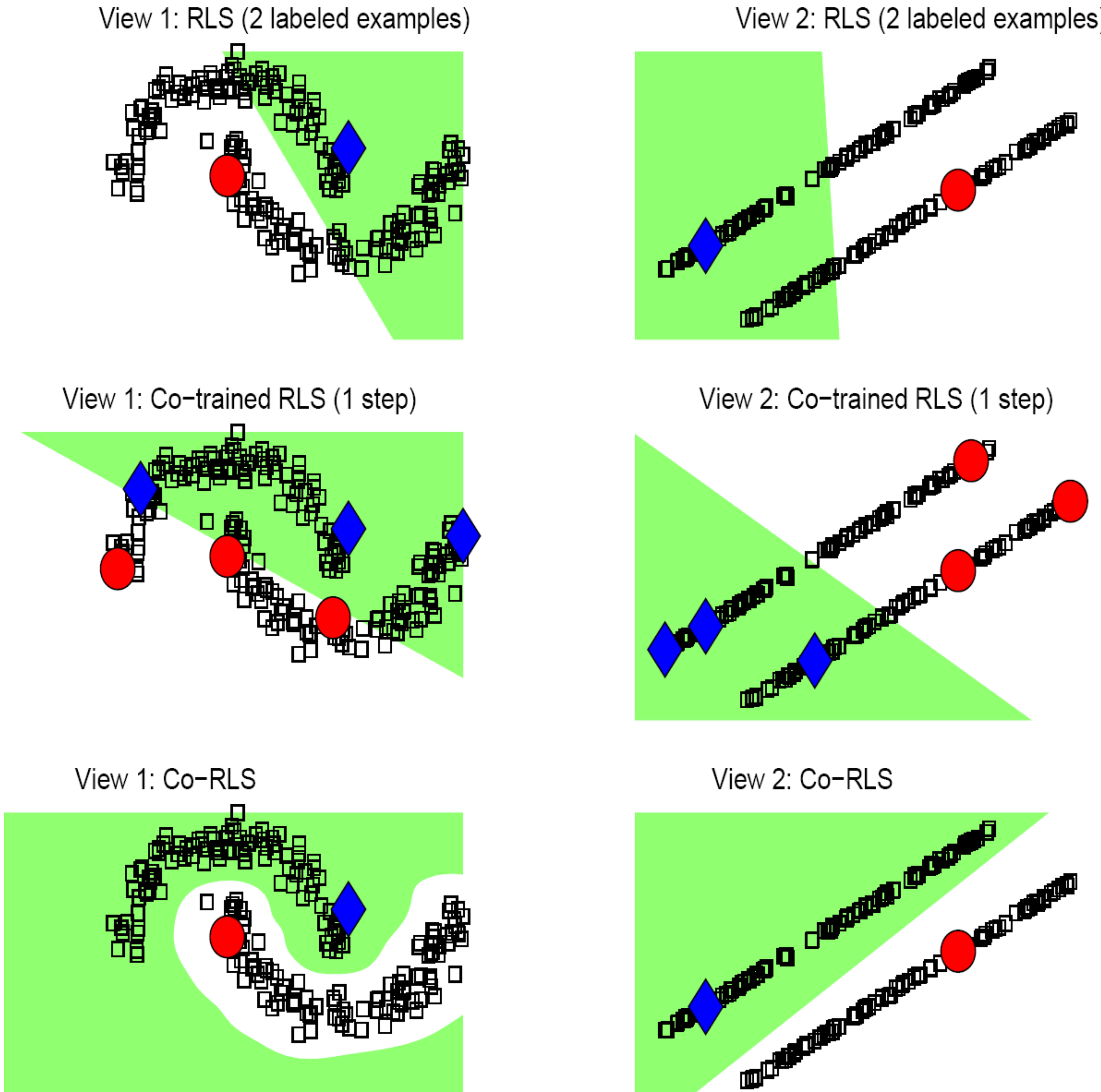
15

Comparison of Loss Functions

- **Loss functions**
  - Exponential: $\sum_{x \in U} \exp\left(-\tilde{y}_2 f_1(x)\right) + \exp\left(-\tilde{y}_1 f_2(x)\right)$
  - Least Square: $\sum_{x \in U} (f_1(x) - f_2(x))^2$
  - Bhattacharyya distance: $E_U(B(p_1, p_2))$

$$B(p_1, p_2) = -\log \sum_y \sqrt{p_1(y) p_2(y)}$$

- **When two classifiers don't agree**
  - Loss grows exponentially, quadratically, linearly
- **When two classifiers agree**
  - Little penalty ➡ Penalize the margin


View 1: RLS (2 labeled examples)
View 2: RLS (2 labeled examples)
View 1: Co−trained RLS (1 step)
View 2: Co−trained RLS (1 step)
View 1: Co−RLS
View 2: Co−RLS

[SNB05]

Semi-supervised Learning

- Learning from a mixture of labeled and unlabeled examples

Labeled Data **Unlabeled Data**

$$L = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \quad D = \{(x_{n+1}), (x_{n+2}), \ldots, (x_{n+m})\}$$

$$y = f(x)$$

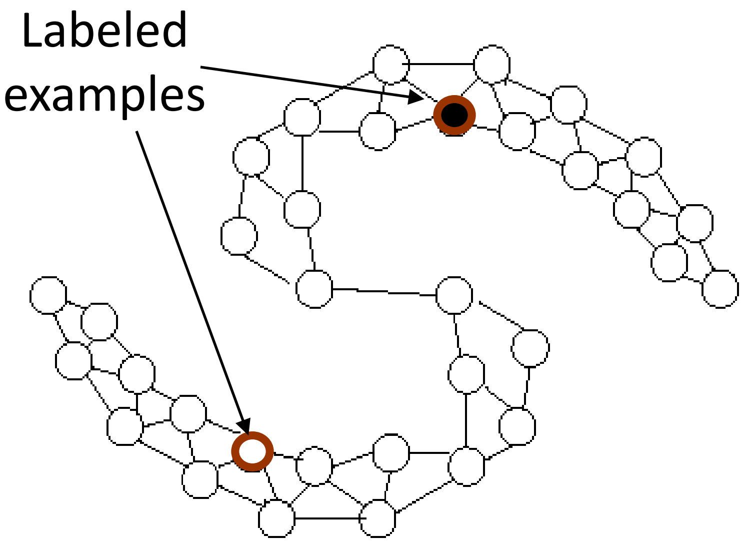
usage	supervised learning	semi-supervised learning	unsupervised learning
$\{(x, y)\}$ labeled data	yes	yes	no
$\{x\}$ unlabeled data	no	yes	yes

Why Semi-supervised Learning?

- **Labeling**
  - Expensive and difficult
  - Unreliable
- **Unlabeled examples**
  - Easy to obtain in large numbers
  - Ex. Web pages, text documents, etc.

Manifold Assumption

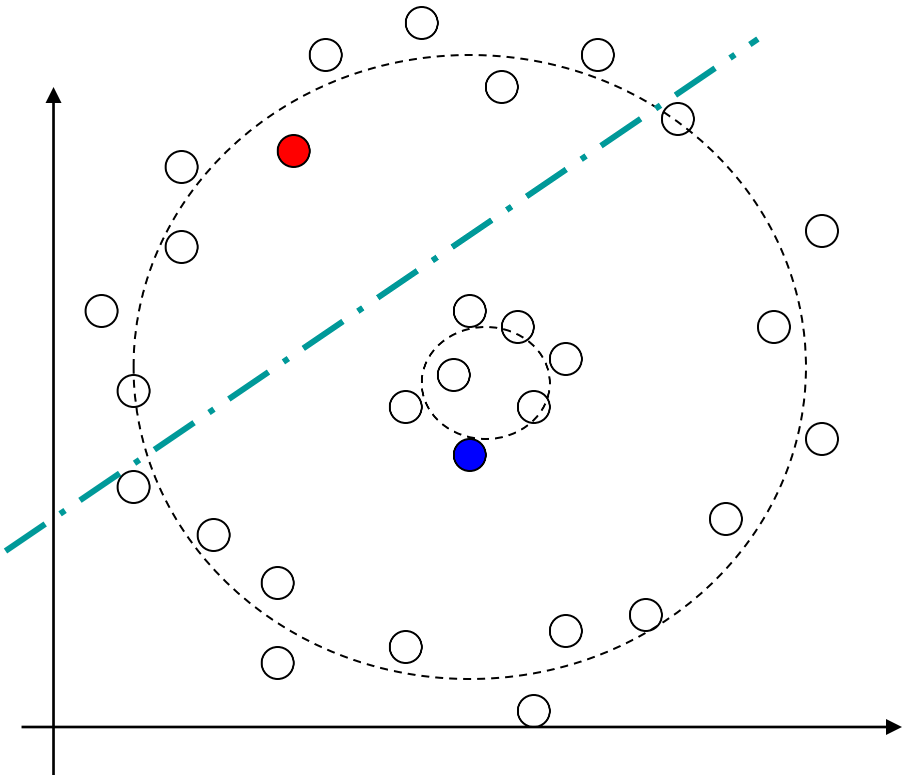
- **Graph representation**
  - Vertex: training example (labeled and unlabeled)
  - Edge: similar examples



- Regularize the classification function *f*(*x*)

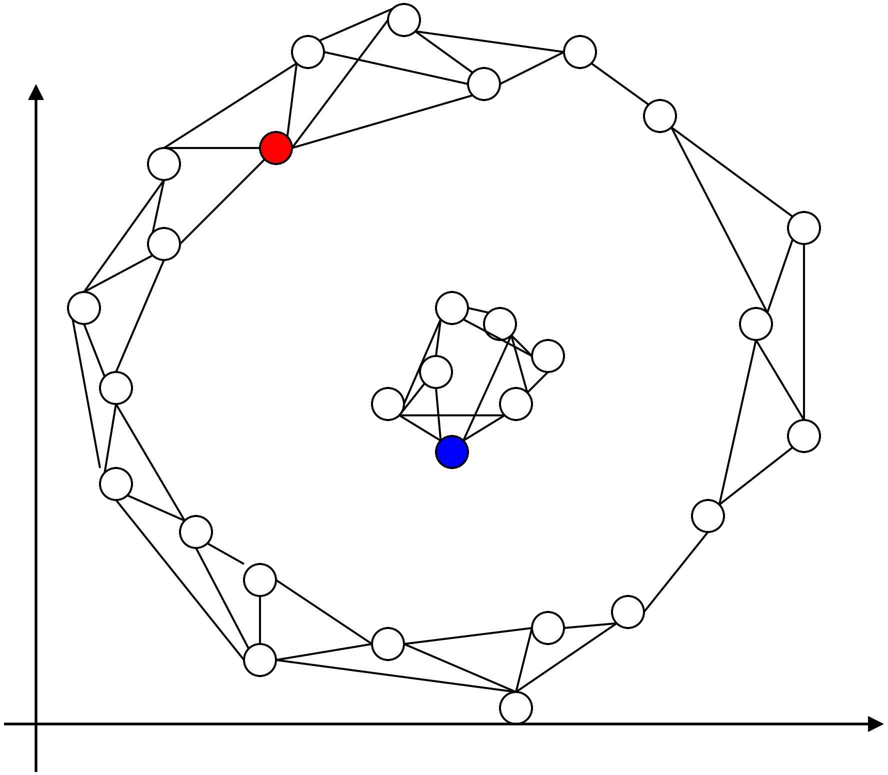
x_1 and x_2 are connected ->
distance between $f(x_1)$ and $f(x_2)$ is small

Label Propagation: Key Idea



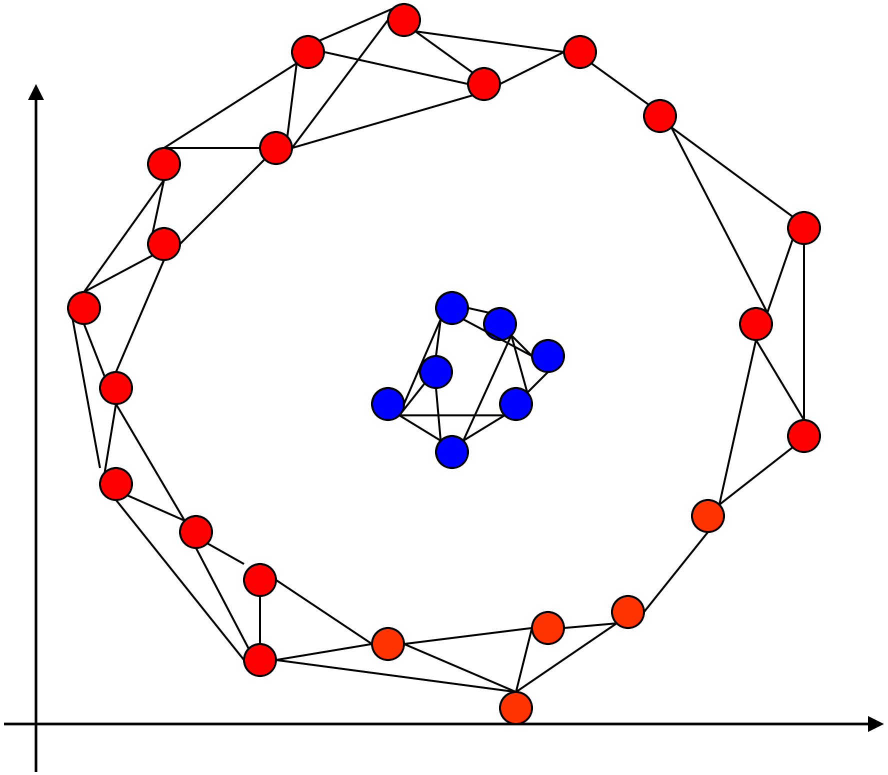
- A decision boundary based on the labeled examples is unable to take into account the layout of the data points
- How to incorporate the data distribution into the prediction of class labels?

Label Propagation: Key Idea



- Connect the data points that are close to each other

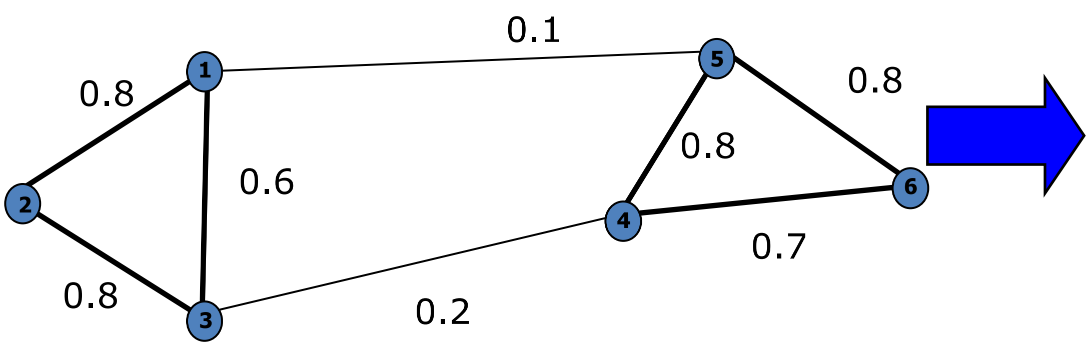
Label Propagation: Key Idea



- Connect the data points that are close to each other
- Propagate the class labels over the connected graph

Matrix Representations

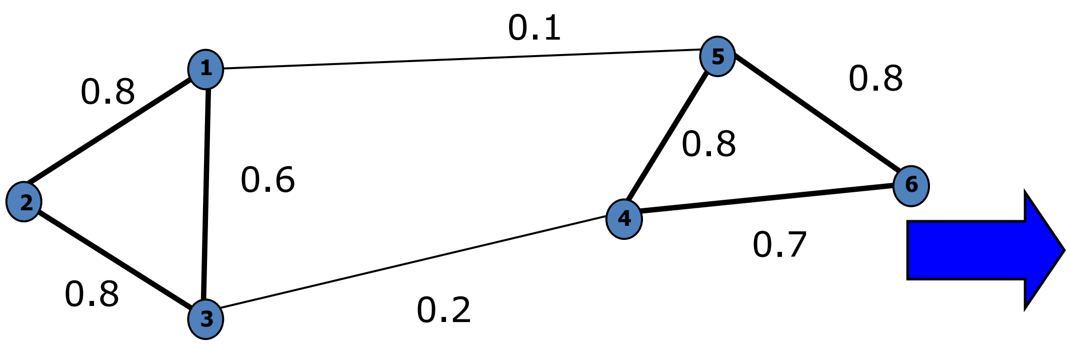
- **Similarity matrix (*W*)**
  - $n \times n$ matrix
  - $W = [w_{ij}]$: similarity between x_i and x_j



	x_1	x_2	x_3	x_4	x_5	x_6
x_1	0	0.8	0.6	0	0.1	0
x_2	0.8	0	0.8	0	0	0
x_3	0.6	0.8	0	0.2	0	0
x_4	0	0	0.2	0	0.8	0.7
x_5	0.1	0	0	0.8	0	0.8
x_6	0	0	0	0.7	0.8	0

Matrix Representations

- **Degree matrix (D)**
  - $n \times n$ diagonal matrix
  - $D(i,i) = \sum_j w_{ij}$: total weight of edges incident to vertex x_i

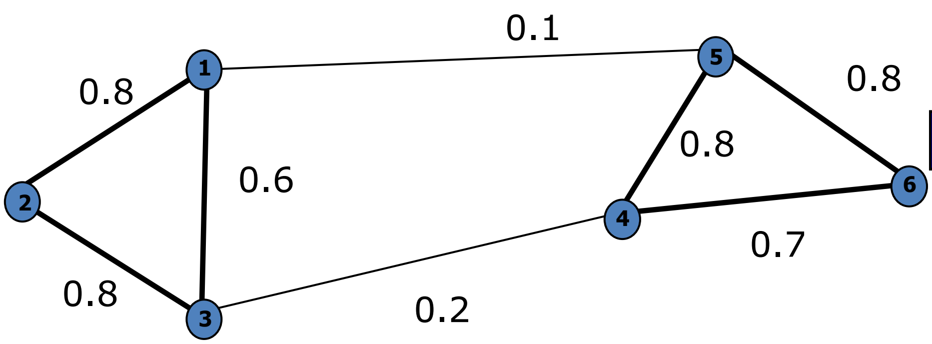


	x_1	x_2	x_3	x_4	x_5	x_6
x_1	1.5	0	0	0	0	0
x_2	0	1.6	0	0	0	0
x_3	0	0	1.6	0	0	0
x_4	0	0	0	1.7	0	0
x_5	0	0	0	0	1.7	0
x_6	0	0	0	0	0	1.5

Matrix Representations

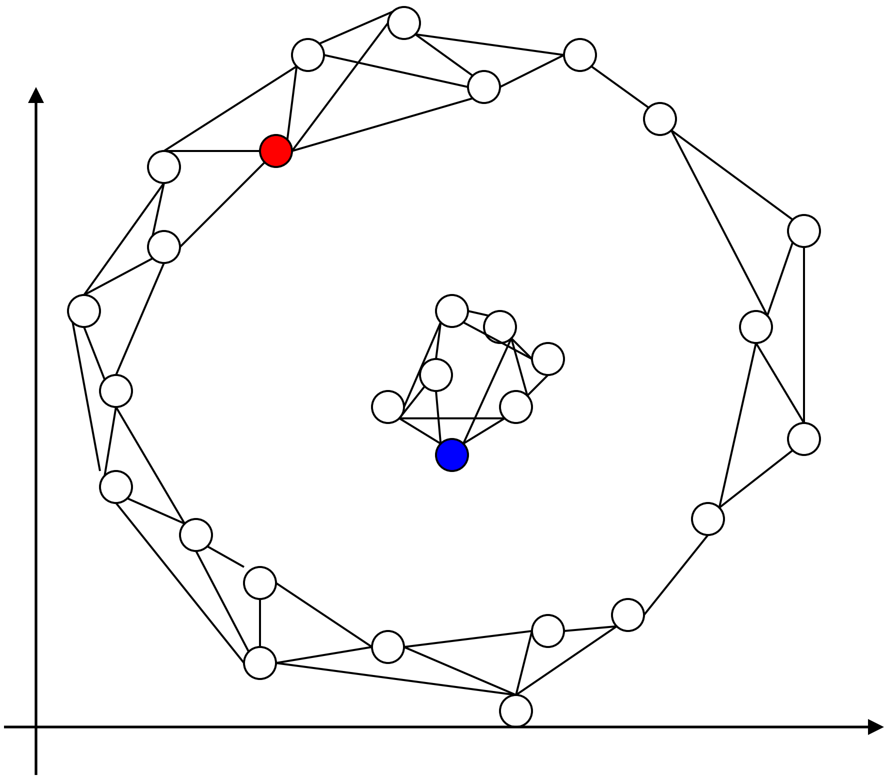
- **Normalized similarity matrix (*S*)**
  - $n\ x\ n$ symmetric matrix

$$S = D^{-0.5} W D^{-0.5}$$



	x_1	x_2	x_3	x_4	x_5	x_6
x_1	0	0.52	0.39	0	0.06	0
x_2	0.52	0	0.5		0	0
x_3	0.39	0.5	0	0.12	0	0
x_4	0	0	0.12	0	0.47	0.44
x_5	0.06	0	0	0.47	0	0.5
x_6	0	0	0	0.44	0.5	0

Normalized Similarity Matrix



$$S = D^{-0.5} W D^{-0.5}$$

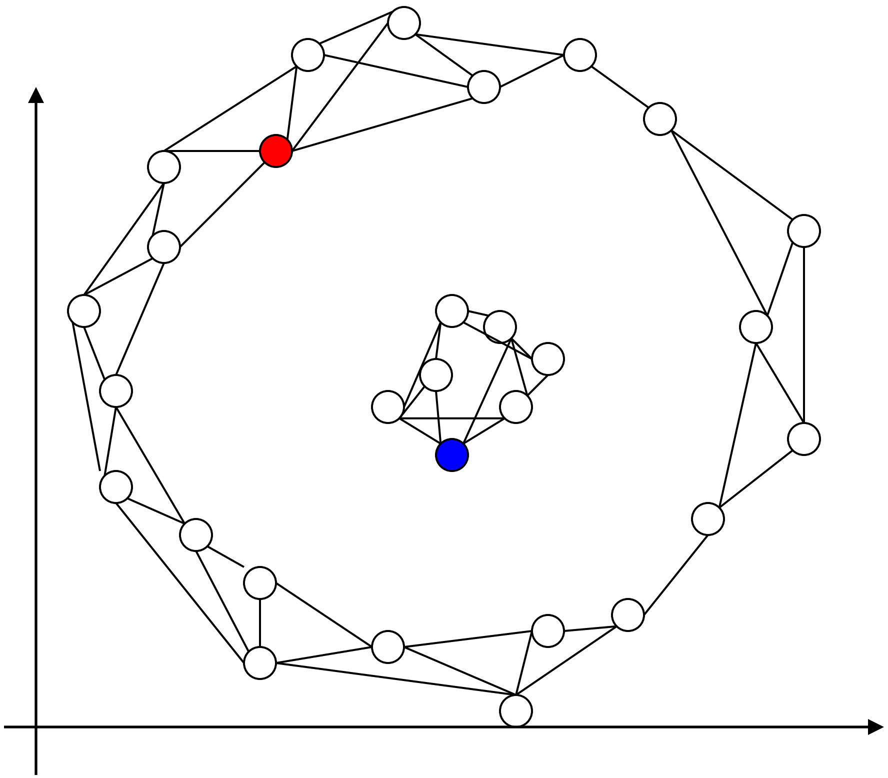
1 30

$$\begin{array}{c} 1 \\ \\ \cdots \\ \\ \\ \\ 30 \end{array} \begin{bmatrix} s_{11} & \cdots & s_{1f} & \cdots & s_{1n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ s_{i1} & \cdots & s_{if} & \cdots & s_{in} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ s_{n1} & \cdots & s_{nf} & \cdots & s_{nn} \end{bmatrix}$$

Initial Label and Prediction

- **Let Y be the initial assignment of class labels**
  - $y_i = 1$ when the i-th node is assigned to the positive class
  - $y_i = -1$ when the i-th node is assigned to the negative class
  - $y_i = 0$ when the i-th node is not initially labeled
- **Let F be the predicted class labels**
  - The i-th node is assigned to the positive class if $f_i > 0$
  - The i-th node is assigned to the negative class if $f_i < 0$

Initial Label and Prediction



Initial Label

Y

$$\begin{array}{c} 1 \\ \cdots \\ \\ \\ 30 \end{array} \begin{bmatrix} y_1 \\ \vdots \\ y_i \\ \vdots \\ y_n \end{bmatrix} \begin{array}{c} \\ -1 \\ \\ 1 \\ \\ \end{array}$$

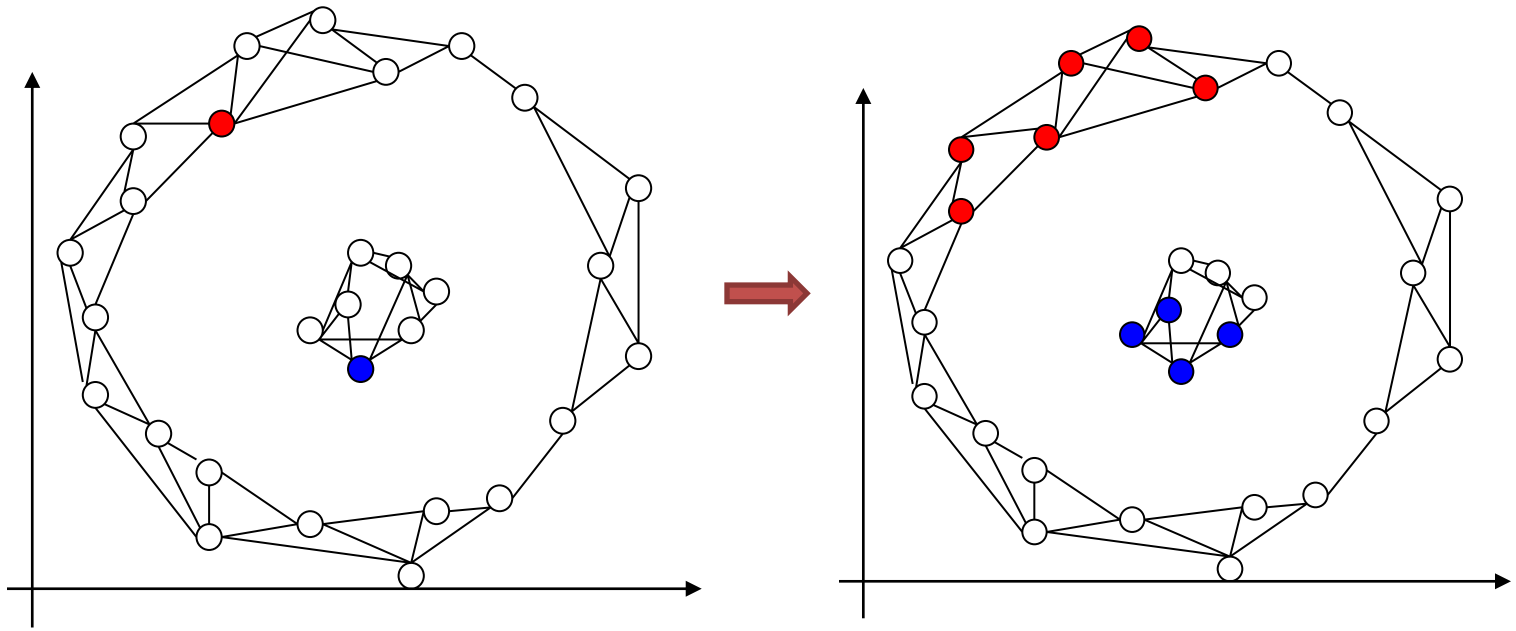
Prediction

F

$$\begin{array}{c} 1 \\ \cdots \\ \\ \\ 30 \end{array} \begin{bmatrix} f_1 \\ \vdots \\ f_i \\ \vdots \\ f_n \end{bmatrix}$$

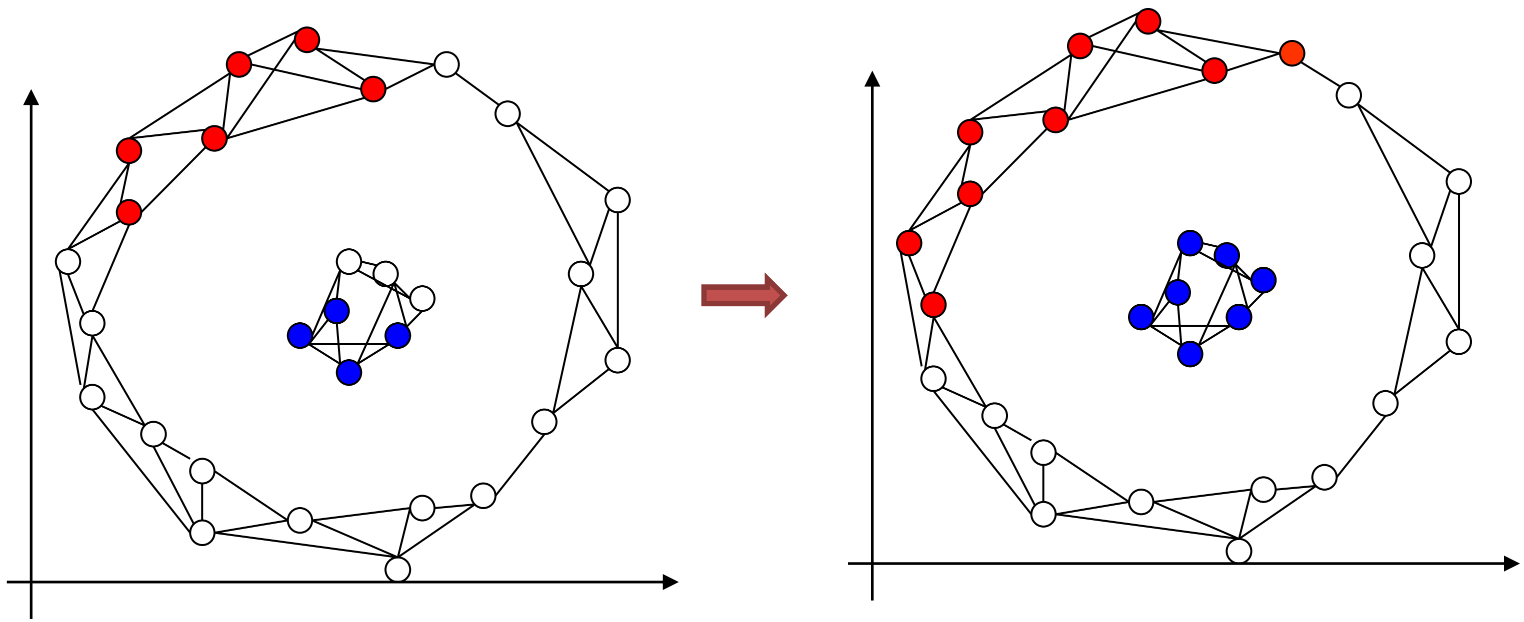
Label Propagation

- **One iteration**
  - $F = Y + \alpha \mathbf{S} Y = (I + \alpha \mathbf{S})Y$
  - α weights the propagation values



Label Propagation

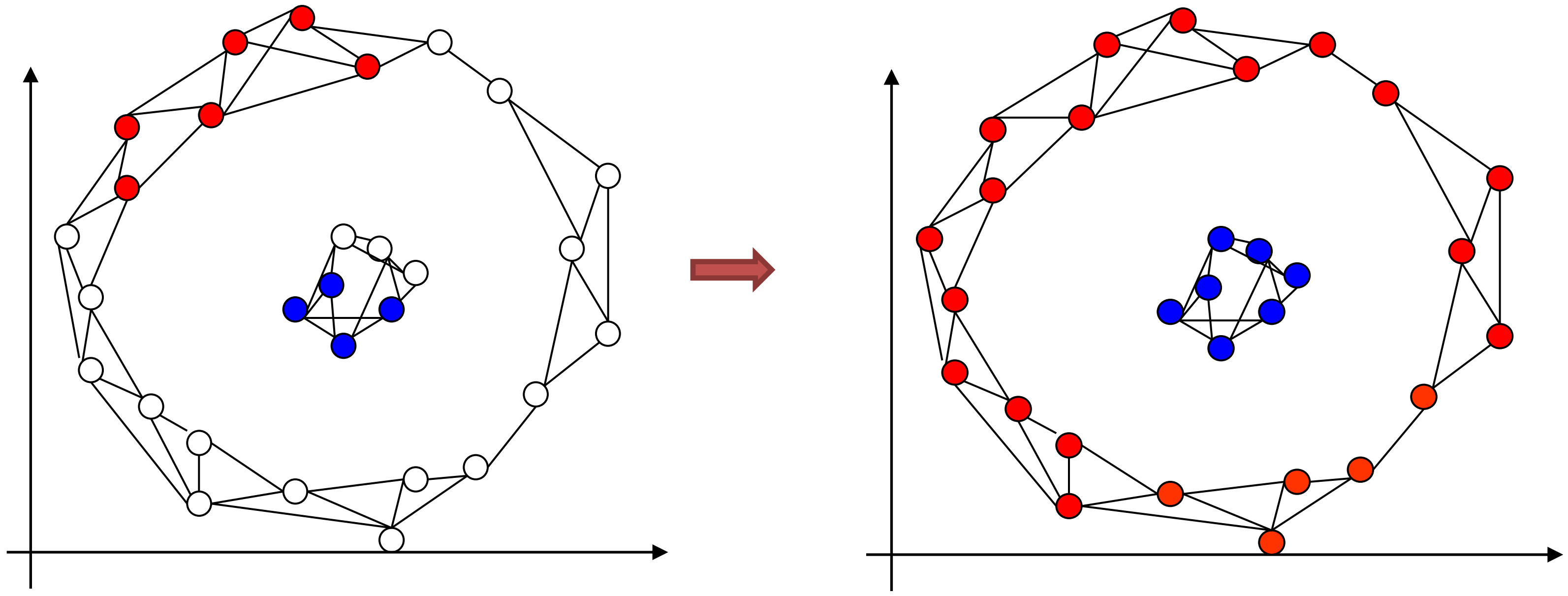
- **Two iteration**
  - $F = Y + \alpha \mathbf{S} Y + \alpha^2 \mathbf{S}^2 Y = (I + \alpha \mathbf{S} + \alpha^2 \mathbf{S}^2) Y$



Label Propagation

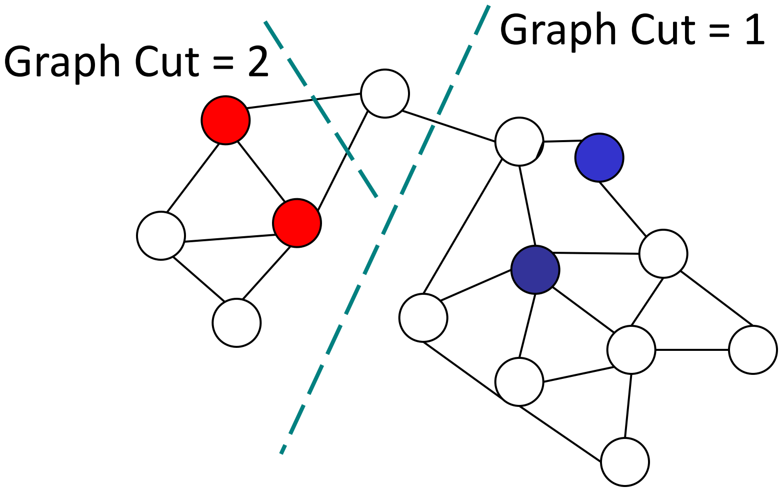
- **More iterations**

$$F = \left(\sum_{n=0}^{\infty} \alpha^n \mathbf{S}^n\right)Y = (I - \alpha\,\mathbf{S})^{-1}Y$$



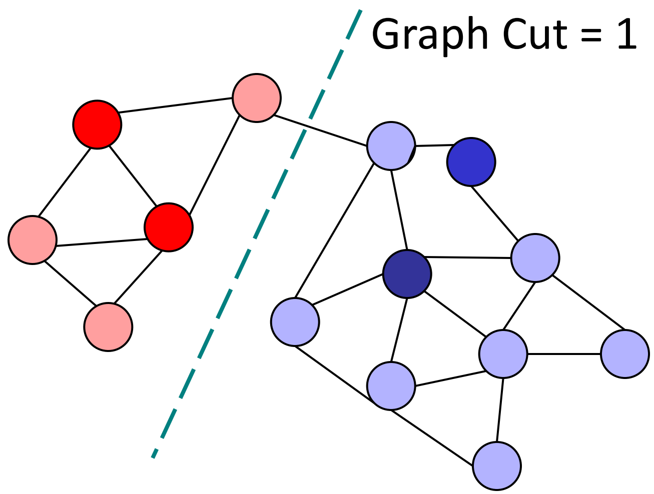
Graph Partitioning

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



Graph Partitioning

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



Review of Spectral Clustering

- Express a bi-partition (C_1,C_2) as a vector

$$f_i = \begin{cases} 1 & if\ \ x_i \in C_1 \\ -1 & if\ \ x_i \in C_2 \end{cases}$$

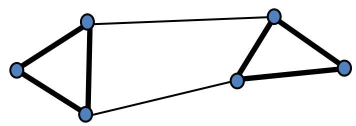
- We can minimise the cut of the partition by finding a non-trivial vector f that minimizes the function

$$g(f) = \sum_{i,j \in V} w_{ij}(f_i - f_j)^2 \ = f^T L\ f$$

Laplacian matrix

Spectral Bi-partitioning Algorithm

1. Pre-processing
   – Build Laplacian matrix L of the graph



	x_1	x_2	x_3	x_4	x_5	x_6
x_1	1.5	-0.8	-0.6	0	-0.1	0
x_2	-0.8	1.6	-0.8	0	0	0
x_3	-0.6	-0.8	1.6	-0.2	0	0
x_4	0	0	-0.2	1.7	-0.8	-0.7
x_5	-0.1	0	0	-0.8	1.7	-0.8
x_6	0	0	0	-0.7	-0.8	1.5

2. Decomposition
   – Find eigenvalues X and eigenvectors Λ of the matrix L

   – Map vertices to corresponding components of λ_2

$\Lambda =$

0.0
0.4
2.2
2.3
2.5
3.0

$X =$

0.4	0.2	0.1	0.4	-0.2	-0.9
0.4	0.2	0.1	-0.	0.4	0.3
0.4	0.2	-0.2	0.0	-0.2	0.6
0.4	-0.4	0.9	0.2	-0.4	-0.6
0.4	-0.7	-0.4	-0.8	-0.6	-0.2
0.4	-0.7	-0.2	0.5	0.8	0.9

x_1	0.2
x_2	0.2
x_3	0.2
x_4	-0.4
x_5	-0.7
x_6	-0.7

Semi-Supervised Learning

$$g(f) = \sum_{i,j \in V} w_{ij} (f_i - f_j)^2 = f^T L f$$

Method 1:
Fix y_l, solve for f_u

$$f = \begin{bmatrix} y_l \\ f_u \end{bmatrix} \qquad L = \begin{bmatrix} L_{ll} & L_{lu} \\ L_{ul} & L_{uu} \end{bmatrix}$$

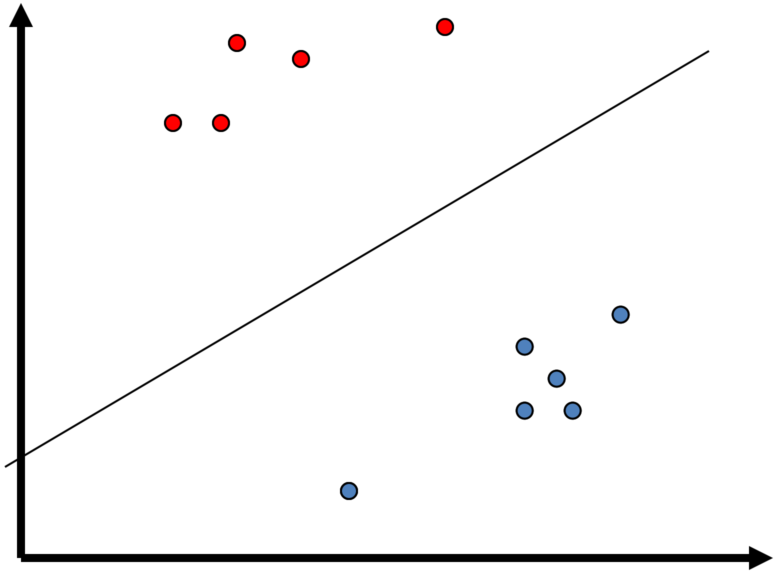
$$\min_{f_u} f^T L f$$

Method 2:
Solve for f

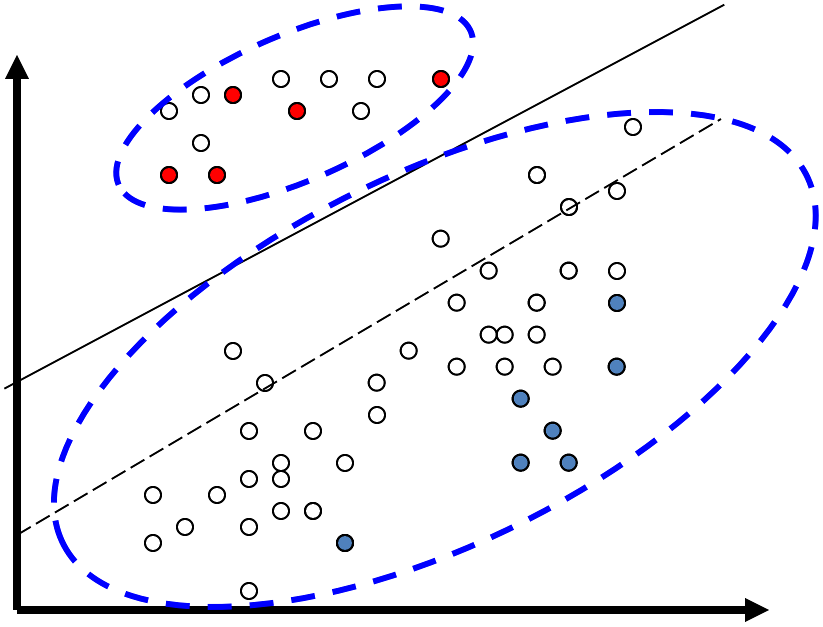
$$\min_f f^T L f + (f - y)^T C (f - y)$$

$$C_{ii} = 1 \qquad \text{if } x_i \text{ is labeled}$$

Clustering Assumption



Clustering Assumption

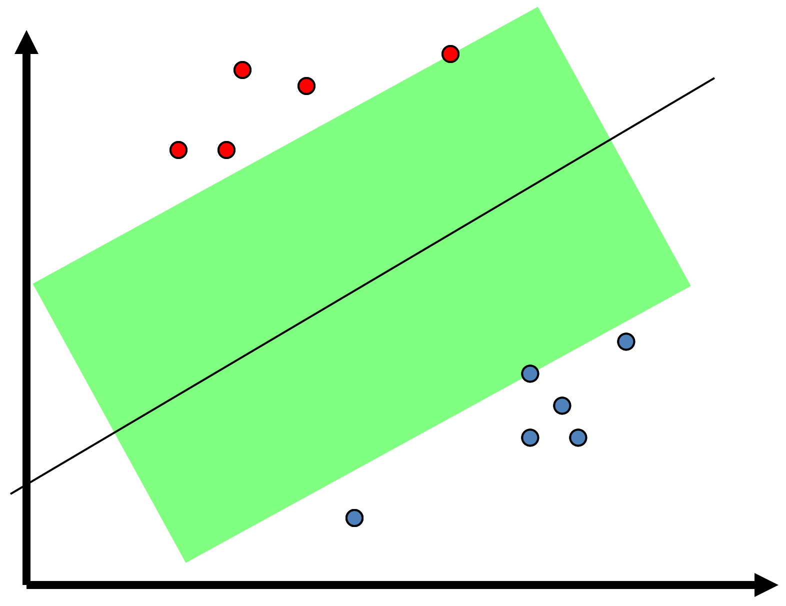


– Points with same label are connected through high density regions, thereby defining a cluster

– Clusters are separated through low-density regions

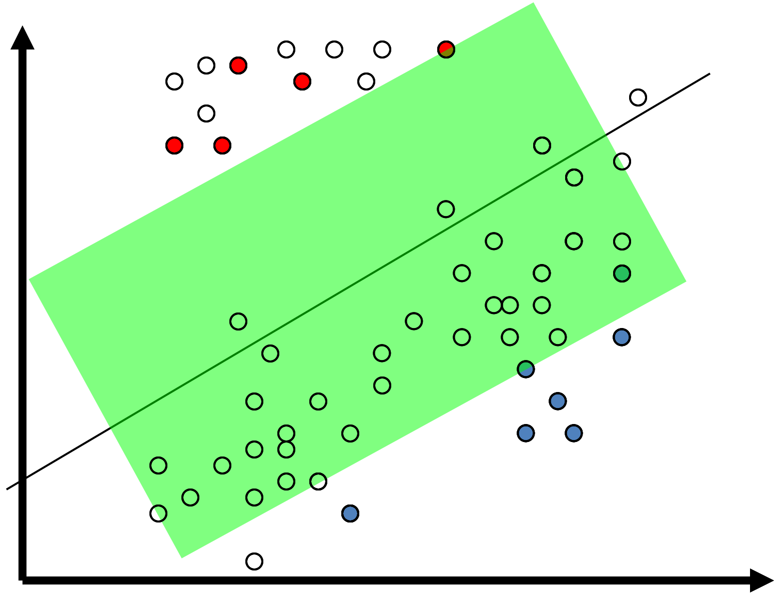
Transductive SVM

- Decision boundary given a small number of labeled examples



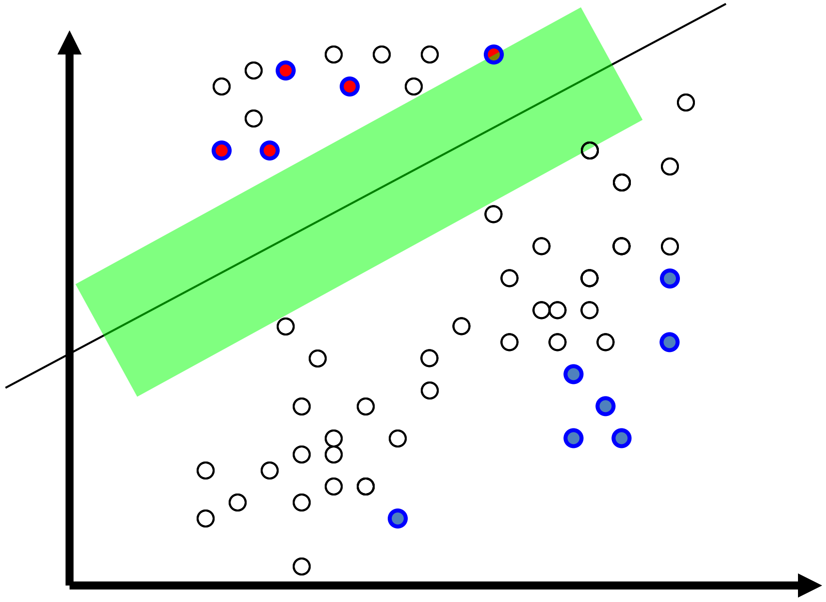
Transductive SVM

- Decision boundary given a small number of labeled examples
- How will the decision boundary change given both labeled and unlabeled examples?



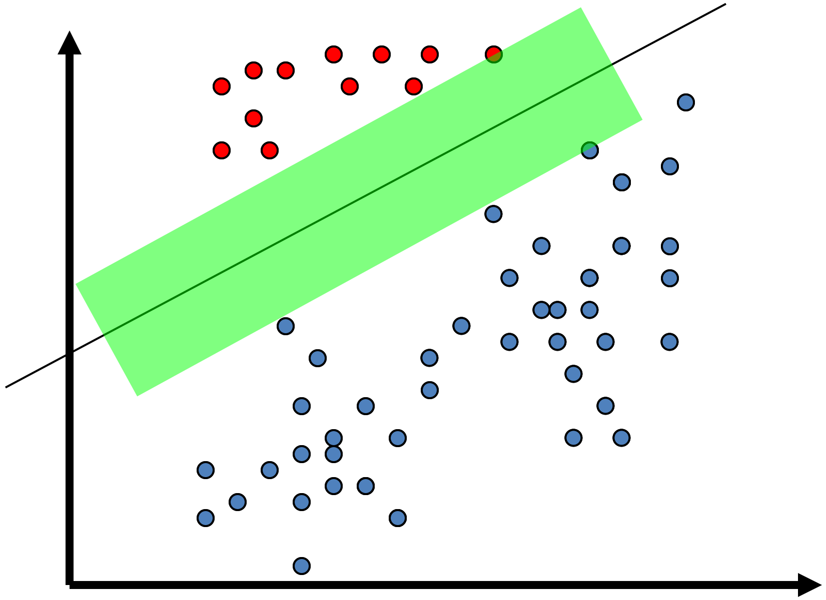
Transductive SVM

- Decision boundary given a small number of labeled examples
- Move the decision boundary to place with low local density



Transductive SVM

- Decision boundary given a small number of labeled examples
- Move the decision boundary to place with low local density
- Classification results
- How to formulate this idea?



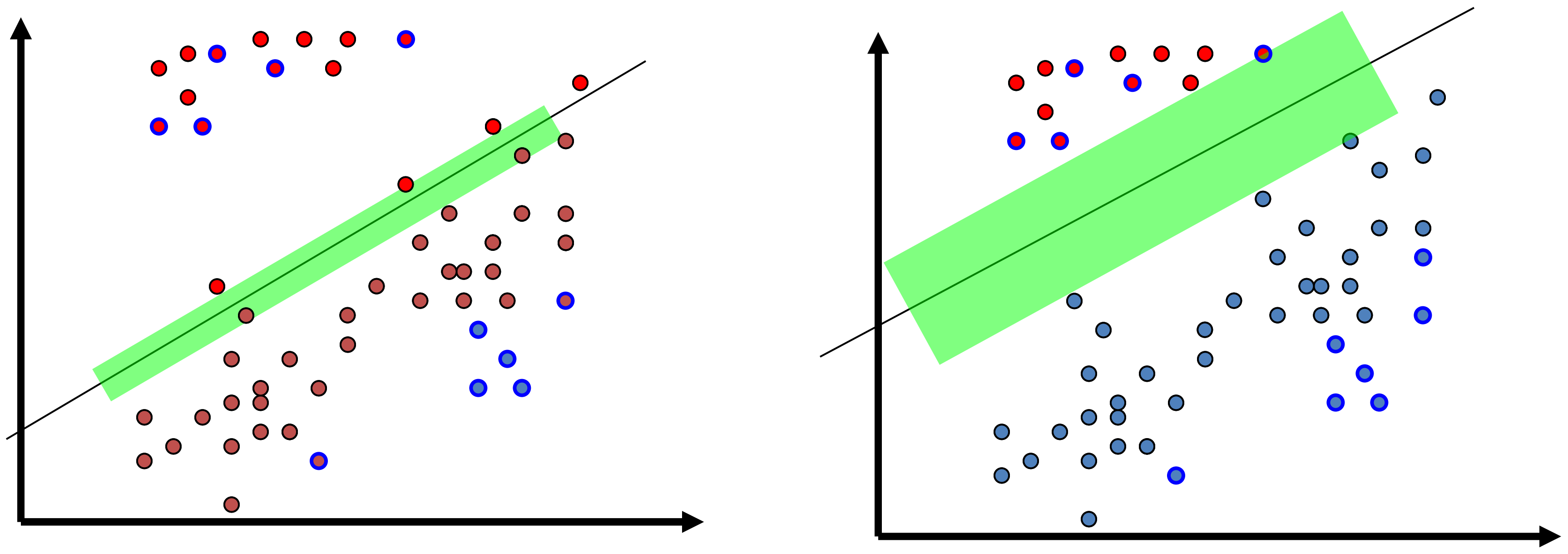
Transductive SVM: Formulation

- Labeled data L: $L = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$
- Unlabeled data D: $D = \{(x_{n+1}), (x_{n+2}), ..., (x_{n+m})\}$
- Maximum margin principle for mixture of labeled and unlabeled data
  - For each label assignment of unlabeled data, compute its maximum margin
  - Find the label assignment whose maximum margin is maximized

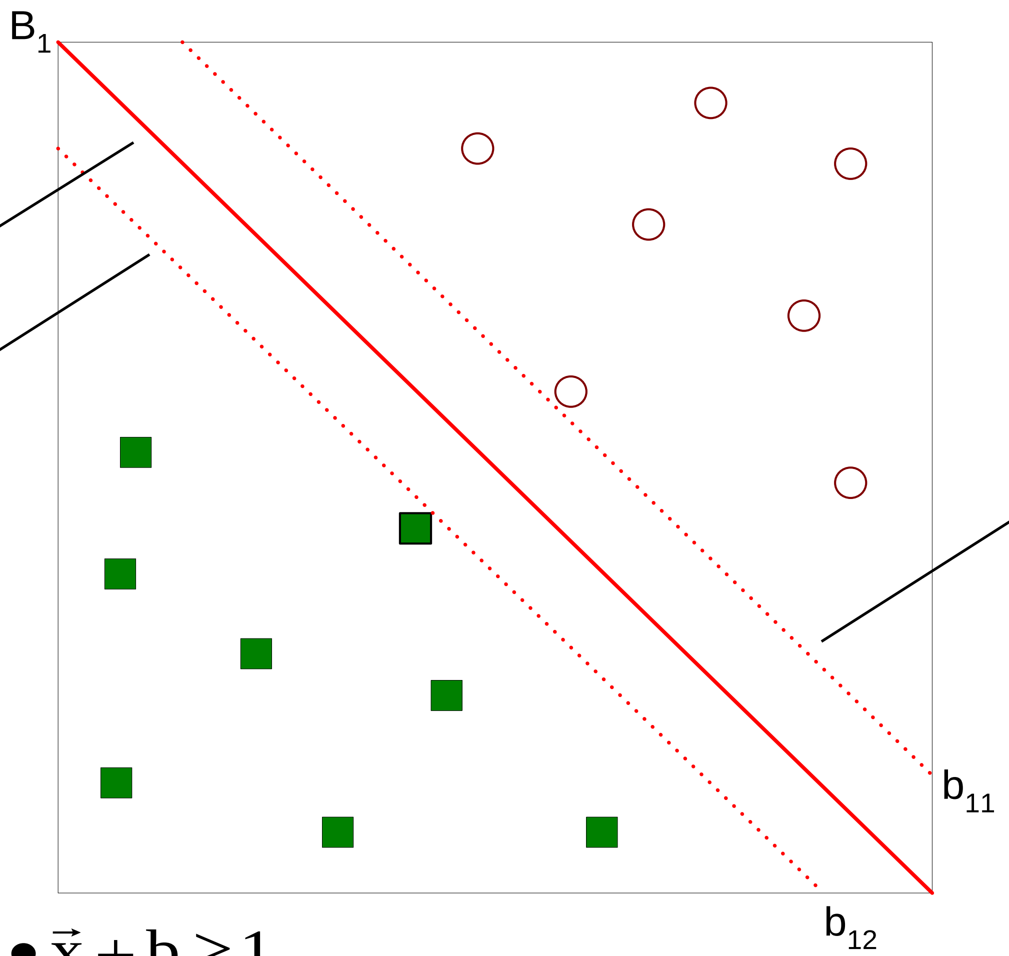
Tranductive SVM

Different label assignment for unlabeled data

→ different maximum margin



Traditional SVM



$$\vec{w} \bullet \vec{x} + b = 0$$

$$\vec{w} \bullet \vec{x} + b = -1$$

$$\vec{w} \bullet \vec{x} + b = +1$$

$$y = \begin{cases} 1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}} + \mathrm{b} \geq 1 \\ -1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}} + \mathrm{b} \leq -1 \end{cases}$$

$$\text{Margin} = \frac{2}{\| \vec{w} \|^2}$$

SVM Formulation

- We want to maximize: $\mathrm{Margin} = \dfrac{2}{\|\vec{w}\|^2}$
  - Which is equivalent to minimizing: $\|\vec{w}\|^2 = \vec{w} \cdot \vec{w}$
  - But subjected to the following constraints:

$$\vec{\mathrm{w}} \bullet \vec{\mathrm{x}}_{\mathrm{i}} + \mathrm{b} \geq 1 \ \text{ if } \ \mathrm{y}_{\mathrm{i}} = 1$$

$$\vec{\mathrm{w}} \bullet \vec{\mathrm{x}}_{\mathrm{i}} + \mathrm{b} \leq -1 \ \text{ if } \ \mathrm{y}_{\mathrm{i}} = \text{-}1$$

⇩

$$\mathrm{y}_{\mathrm{i}}(\vec{\mathrm{w}} \bullet \vec{\mathrm{x}}_{\mathrm{i}} + \mathrm{b}) \geq 1$$

Transductive SVM: Formulation

Original SVM

$$\{\vec{w}^*, b^*\} = \underset{\vec{w}, b}{\operatorname{argmin}}\ \vec{w} \cdot \vec{w}$$

$$\left.\begin{aligned} & y_1\left(\vec{w} \cdot \vec{x}_1 + b\right) \geq 1 \\ & y_2\left(\vec{w} \cdot \vec{x}_2 + b\right) \geq 1 \\ & \cdots\cdot \\ & y_n\left(\vec{w} \cdot \vec{x}_n + b\right) \geq 1 \end{aligned}\right\} \text{labeled examples}$$

A binary variables for label of each example

Transductive SVM

$$\{\vec{w}^*, b^*\} = \underset{y_{n+1}, \ldots, y_{n+m}}{\operatorname{argmin}}\ \underset{\vec{w}, b}{\operatorname{argmin}}\ \vec{w} \cdot \vec{w}$$

$$\left.\begin{aligned} & y_1\left(\vec{w} \cdot \vec{x}_1 + b\right) \geq 1 \\ & y_2\left(\vec{w} \cdot \vec{x}_2 + b\right) \geq 1 \\ & \cdots\cdot \\ & y_n\left(\vec{w} \cdot \vec{x}_n + b\right) \geq 1 \end{aligned}\right\} \text{labeled examples}$$

$$\left.\begin{aligned} & y_{n+1}\left(\vec{w} \cdot \vec{x}_{n+1} + b\right) \geq 1 \\ & \cdots\cdot \\ & y_{n+m}\left(\vec{w} \cdot \vec{x}_{n+m} + b\right) \geq 1 \end{aligned}\right\} \text{unlabeled examples}$$

Constraints for unlabeled data

Alternating Optimization

Transductive SVM

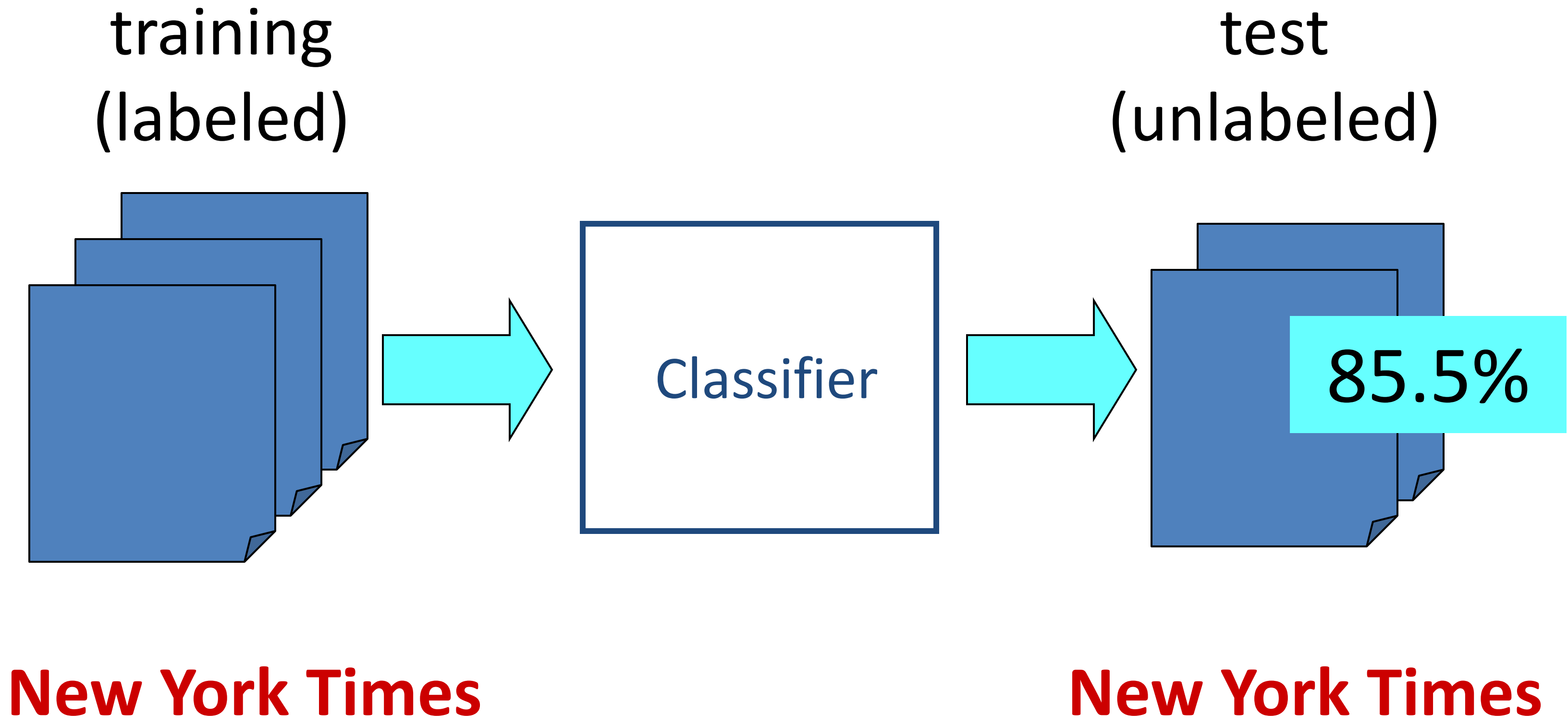
$$\{\vec{w}^*, b^*\} = \underset{y_{n+1},\ldots,y_{n+m}}{\operatorname{argmin}} \ \underset{\vec{w},b}{\operatorname{argmin}} \ \vec{w} \cdot \vec{w}$$

$$\left.\begin{aligned} & y_1\left(\vec{w} \cdot \vec{x}_1 + b\right) \geq 1 \\ & y_2\left(\vec{w} \cdot \vec{x}_2 + b\right) \geq 1 \\ & \ldots. \\ & y_n\left(\vec{w} \cdot \vec{x}_n + b\right) \geq 1 \end{aligned}\right\} \text{labeled examples}$$

$$\left.\begin{aligned} & y_{n+1}\left(\vec{w} \cdot \vec{x}_{n+1} + b\right) \geq 1 \\ & \ldots. \\ & y_{n+m}\left(\vec{w} \cdot \vec{x}_{n+m} + b\right) \geq 1 \end{aligned}\right\} \text{unlabeled examples}$$

- Step 1: fix $y_{n+1},\ldots, y_{n+m}$, learn weights **w**
- Step 2: fix weights w, try to predict $y_{n+1},\ldots, y_{n+m}$

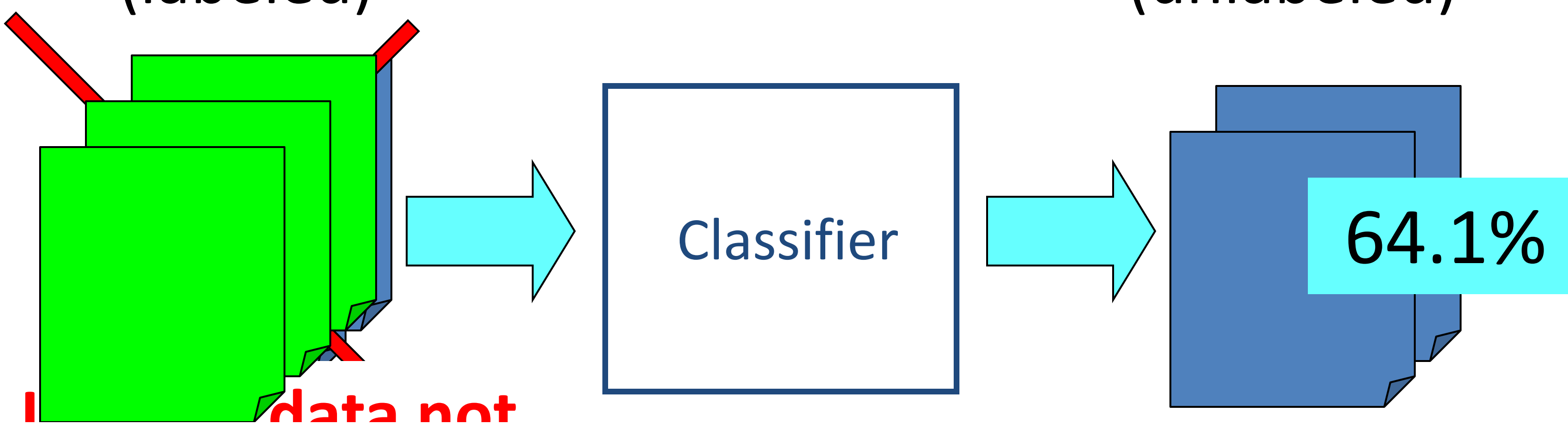
Standard Supervised Learning



In Reality……

training (labeled)

test (unlabeled)

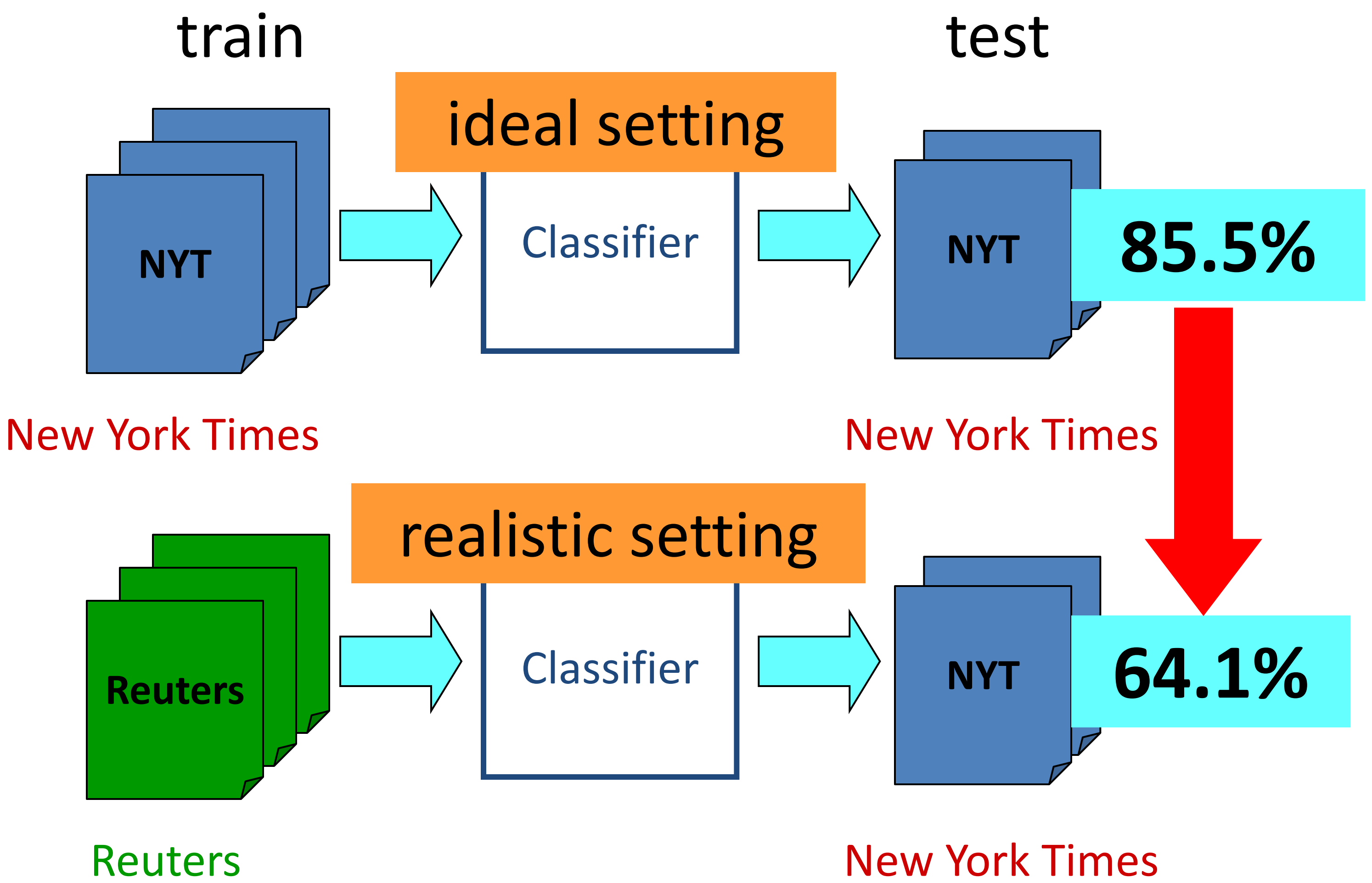


Reuters

New York Times

New York Times

Domain Difference → Performance Drop



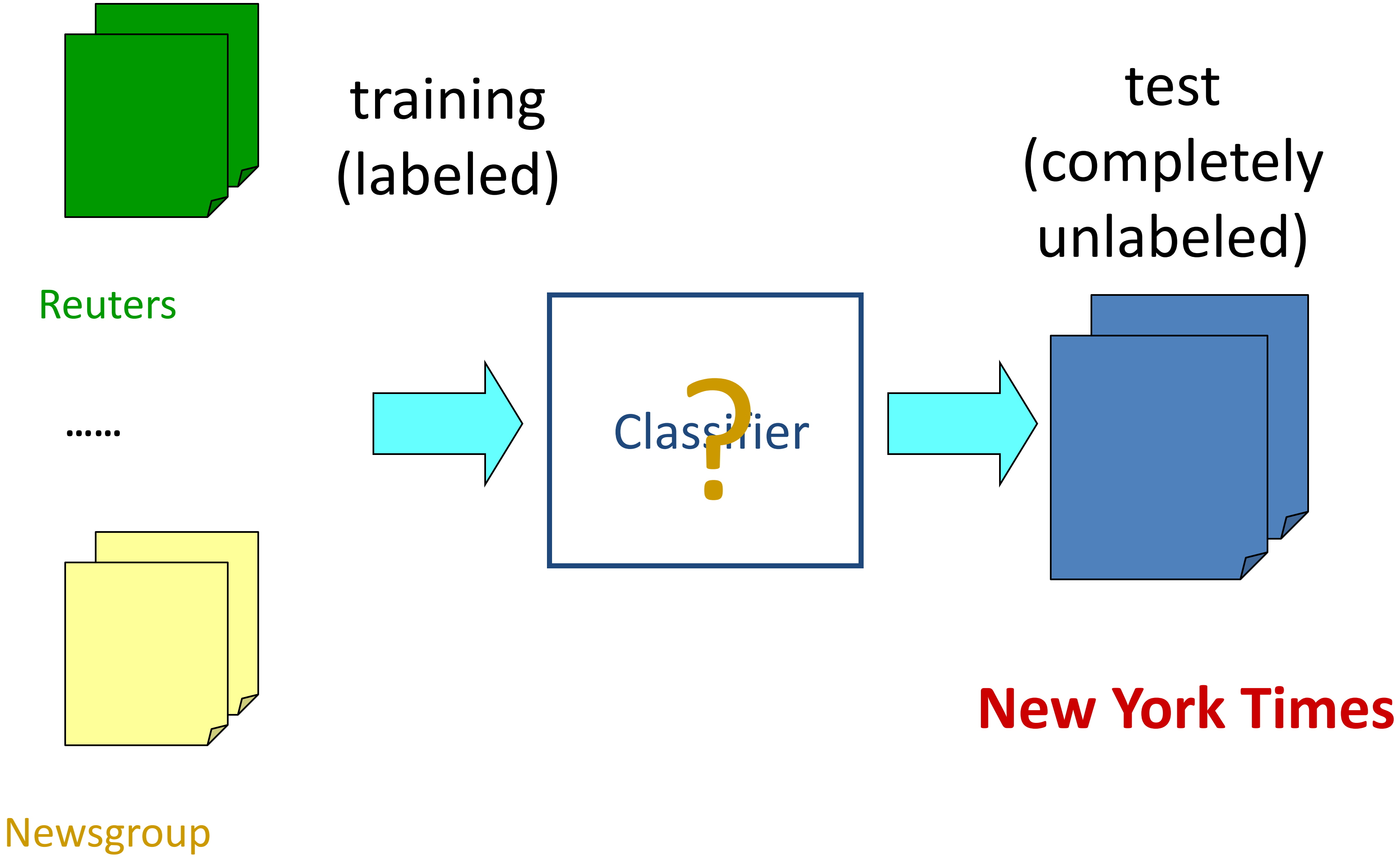
Other Examples

- **Spam filtering**
  - Public email collection → personal inboxes
- **Intrusion detection**
  - Existing types of intrusions → unknown types of intrusions
- **Sentiment analysis**
  - Expert review articles→ blog review articles
- **The aim**
  - To design learning methods that are aware of the training and test domain difference
- **Transfer learning**
  - Adapt the classifiers learnt from the source domain to the new domain

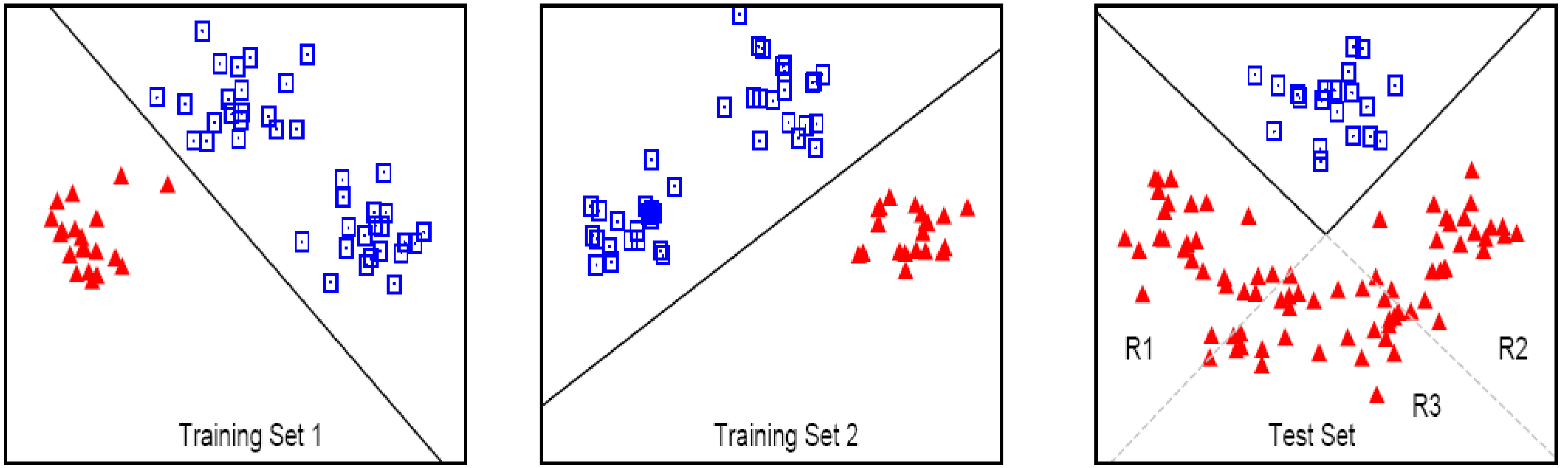
Approaches to Transfer Learning

Transfer learning approaches	**Description**
Instance-transfer	*To re-weight some labeled data in a source domain for use in the target domain*
Feature-representation-transfer	Find a "good" feature representation that reduces difference between a source and a target domain or minimizes error of models
Model-transfer	Discover shared parameters or priors of models between a source domain and a target domain
Relational-knowledge-transfer	Build mapping of relational knowledge between a source domain and a target domain.

All Sources of Labeled Information



A Synthetic Example

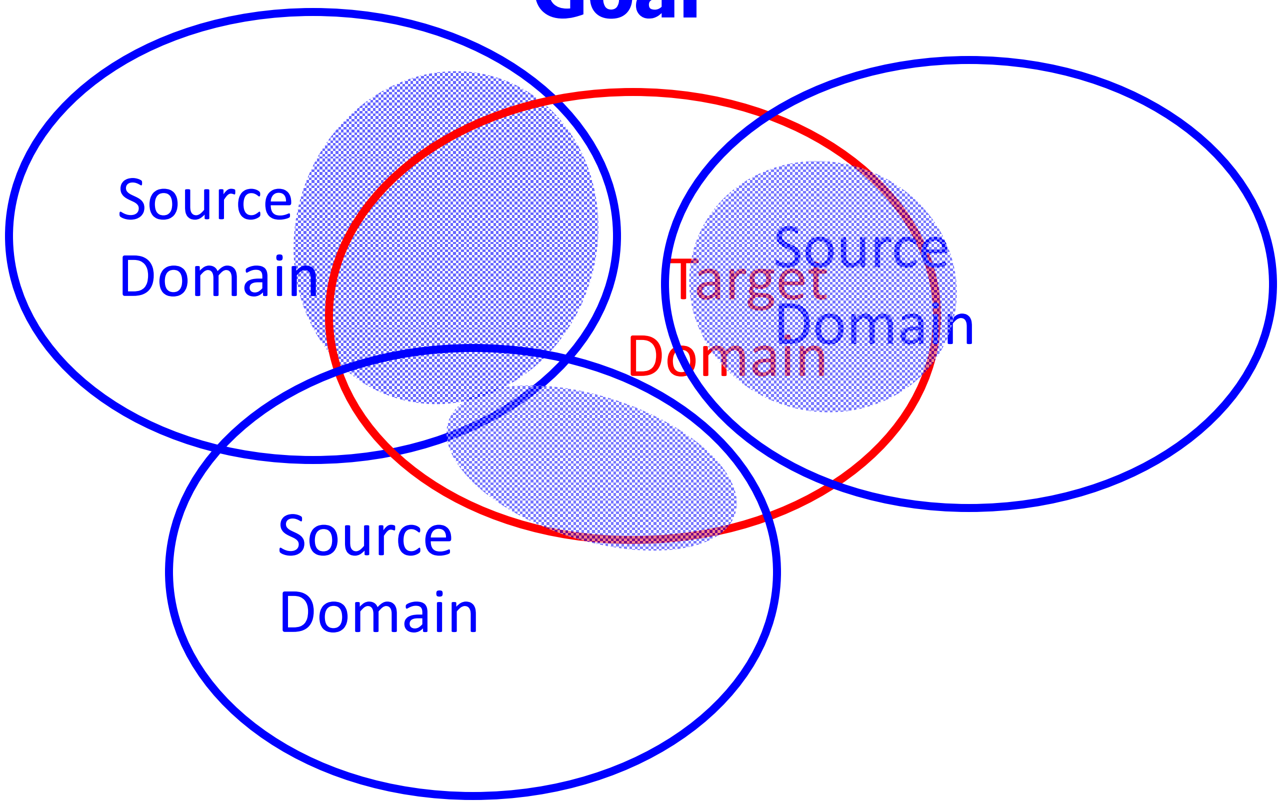


Training → **Test**

(have conflicting concepts)

Partially overlapping

Goal

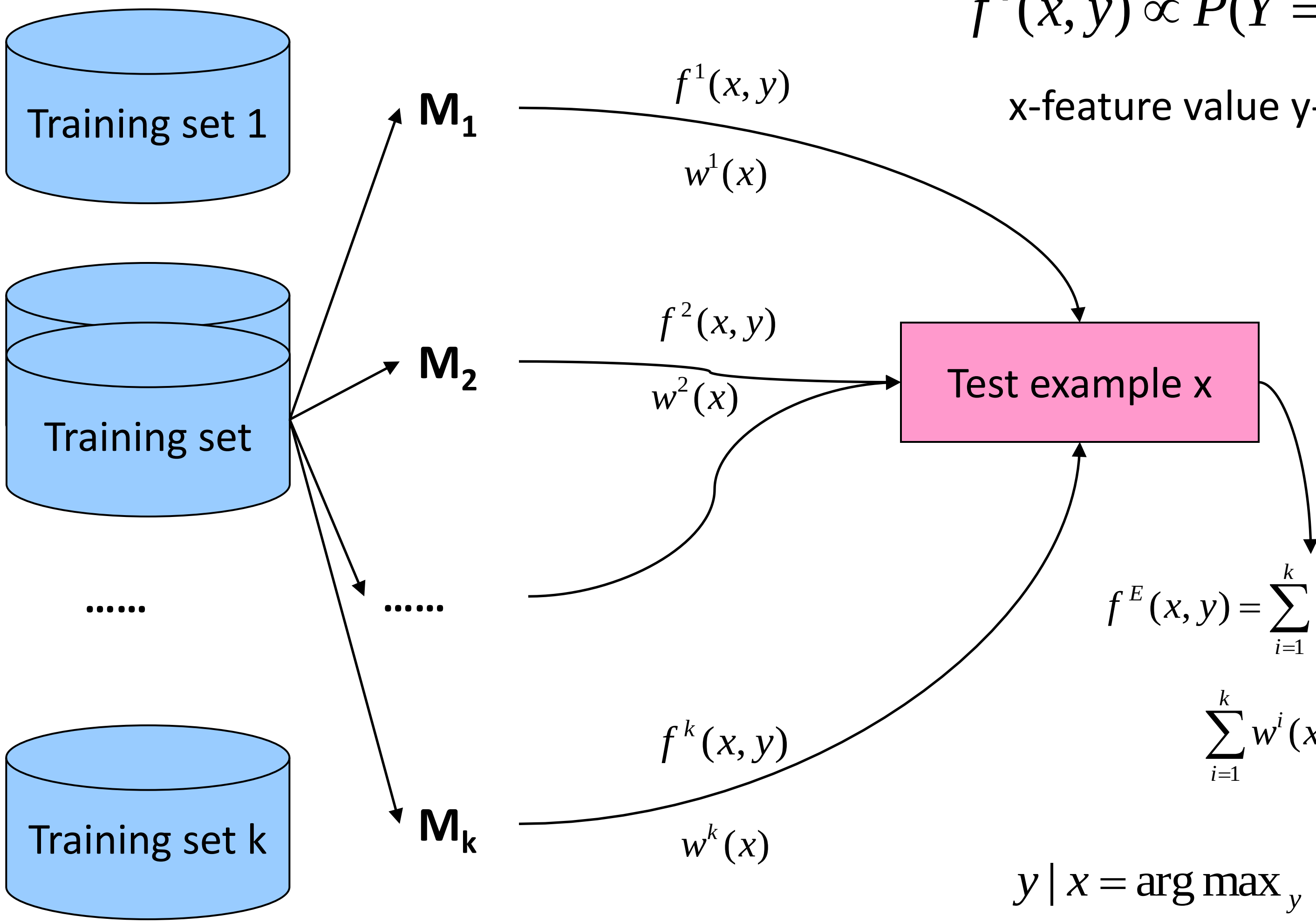


- To unify knowledge that are consistent with the test domain from multiple source domains (models)

Locally Weighted Ensemble

$$f^i(x,y) \propto P(Y=y \mid x, M_i)$$

x-feature value y-class label

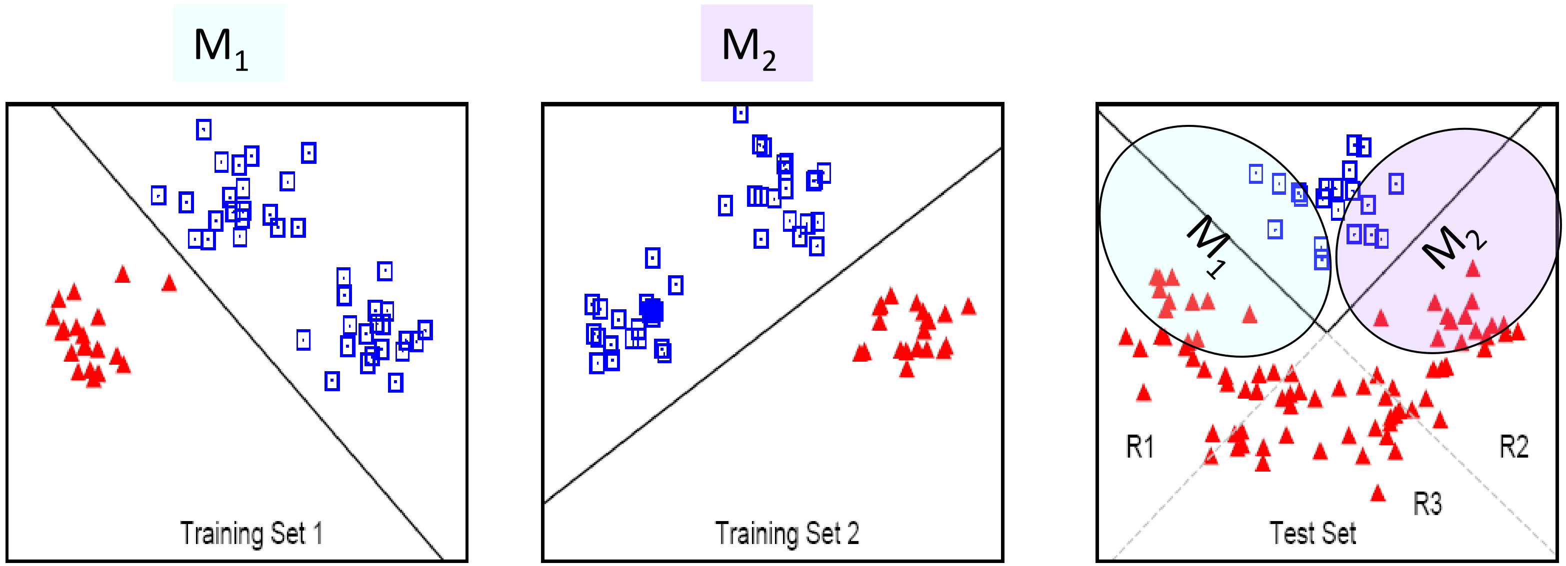


$$f^E(x,y) = \sum_{i=1}^{k} w^i(x) f^i(x,y)$$

$$\sum_{i=1}^{k} w^i(x) = 1$$

$$y \mid x = \arg\max_y f^E(x,y)$$

Synthetic Example Revisited



Training

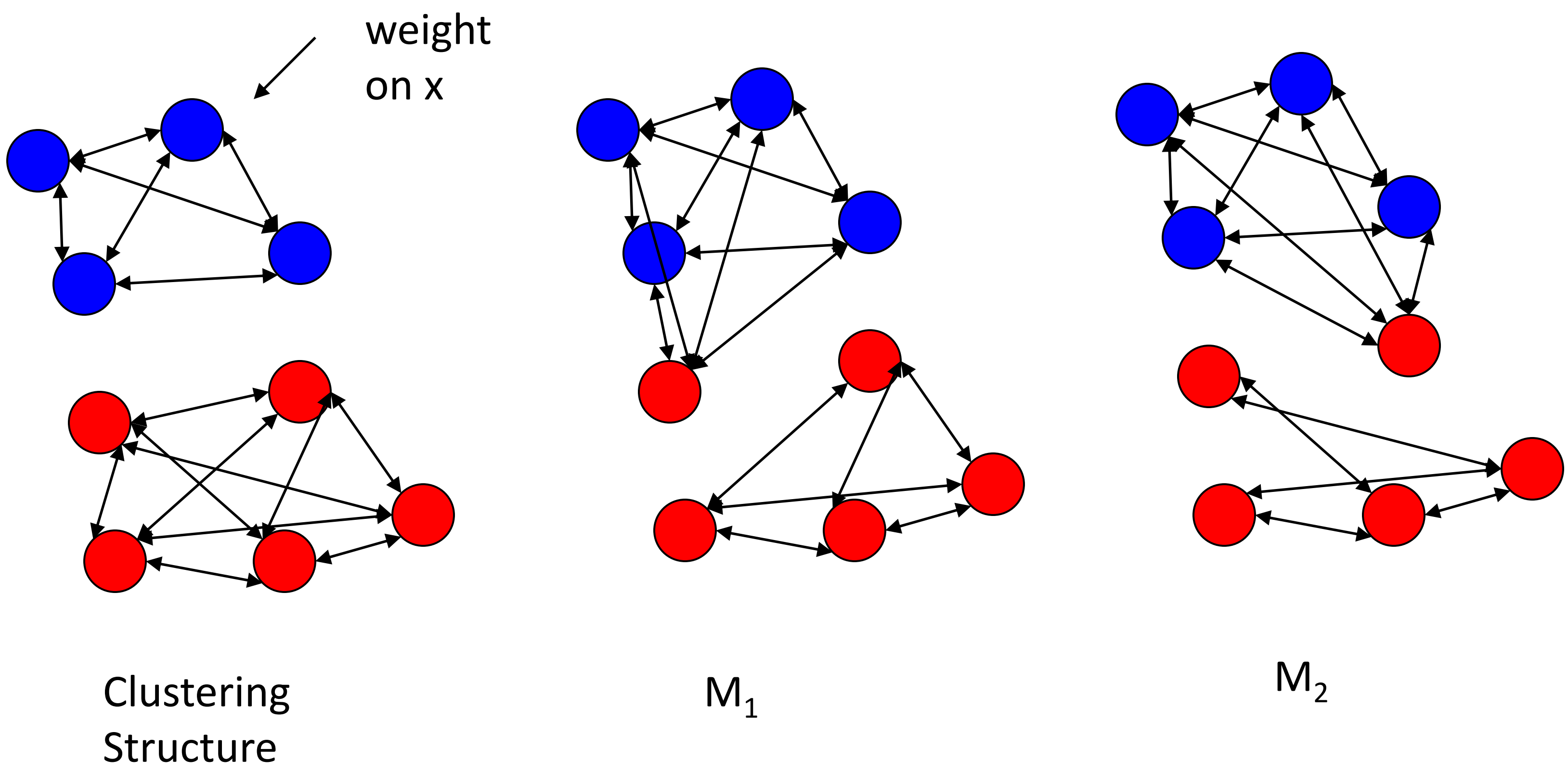
Test

(have conflicting concepts)

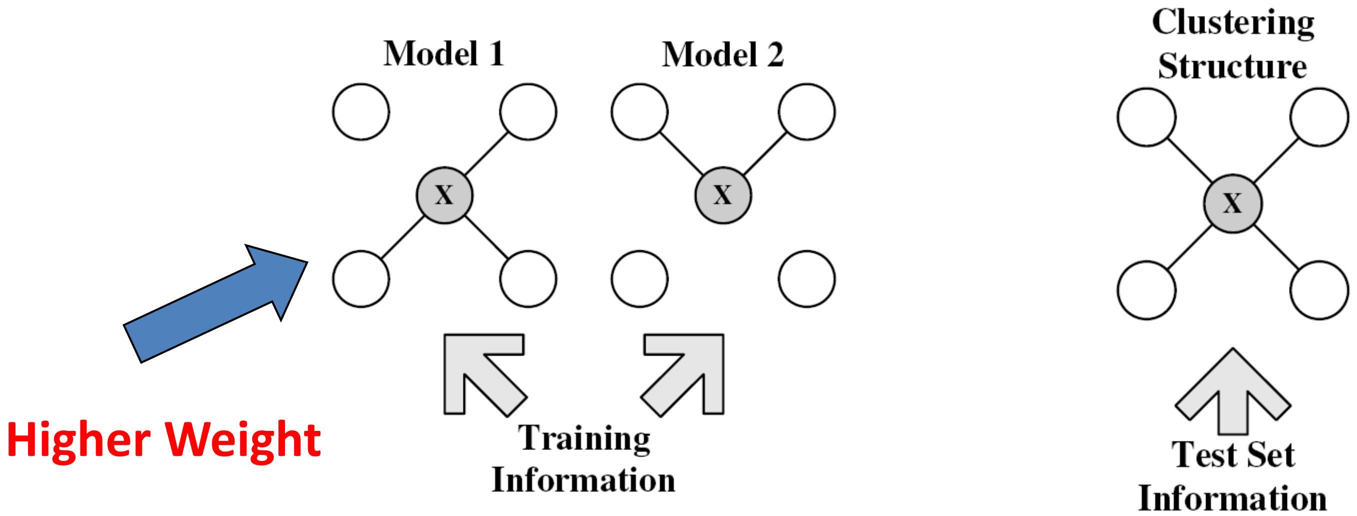
Partially overlapping

Graph-based Heuristics

- **Graph-based weights approximation**
  - Map the structures of models onto test domain



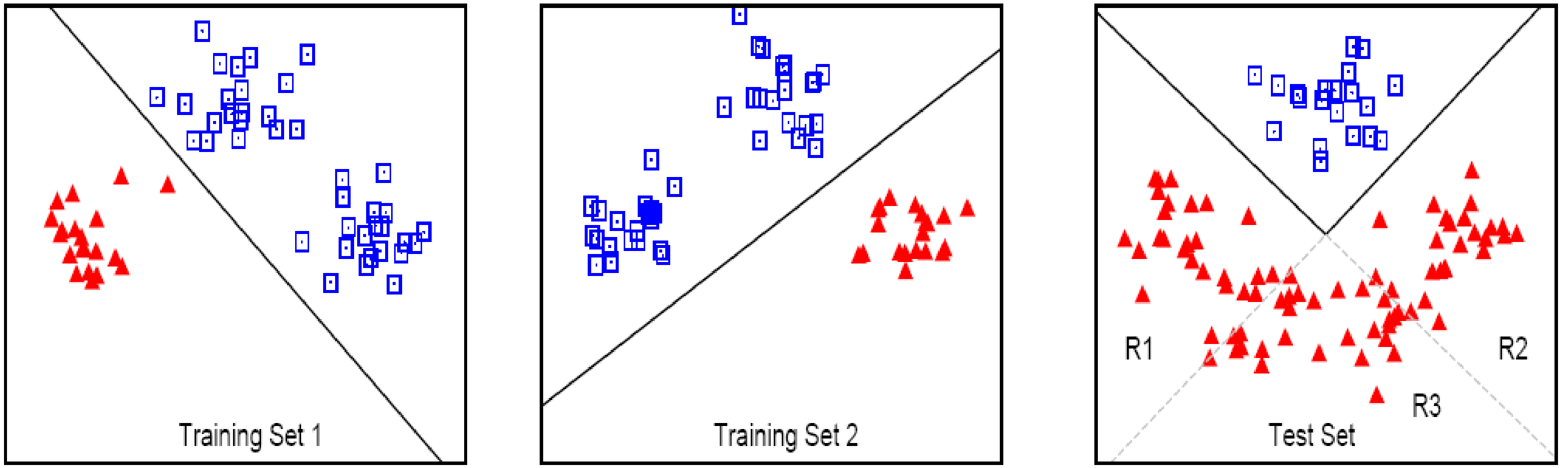
Graph-based Heuristics



- **Local weights calculation**
  - Weight of a model is proportional to the similarity between its neighborhood graph and the clustering structure around x.

$$w_{M,\mathbf{x}} \propto s(G_M, G_T; \mathbf{x}) = \frac{\sum_{v_1 \in V_M} \sum_{v_2 \in V_T} 1\{v_1 = v_2\}}{|V_M| + |V_T|}$$

A Synthetic Example



Training

Test

(have conflicting concepts)

Partially overlapping

Experiments on Synthetic Data