SDM'2010
Columbus, OH

On the Power of Ensemble: Supervised and Unsupervised Methods Reconciled*

Jing Gao[1], Wei Fan[2], Jiawei Han[1]

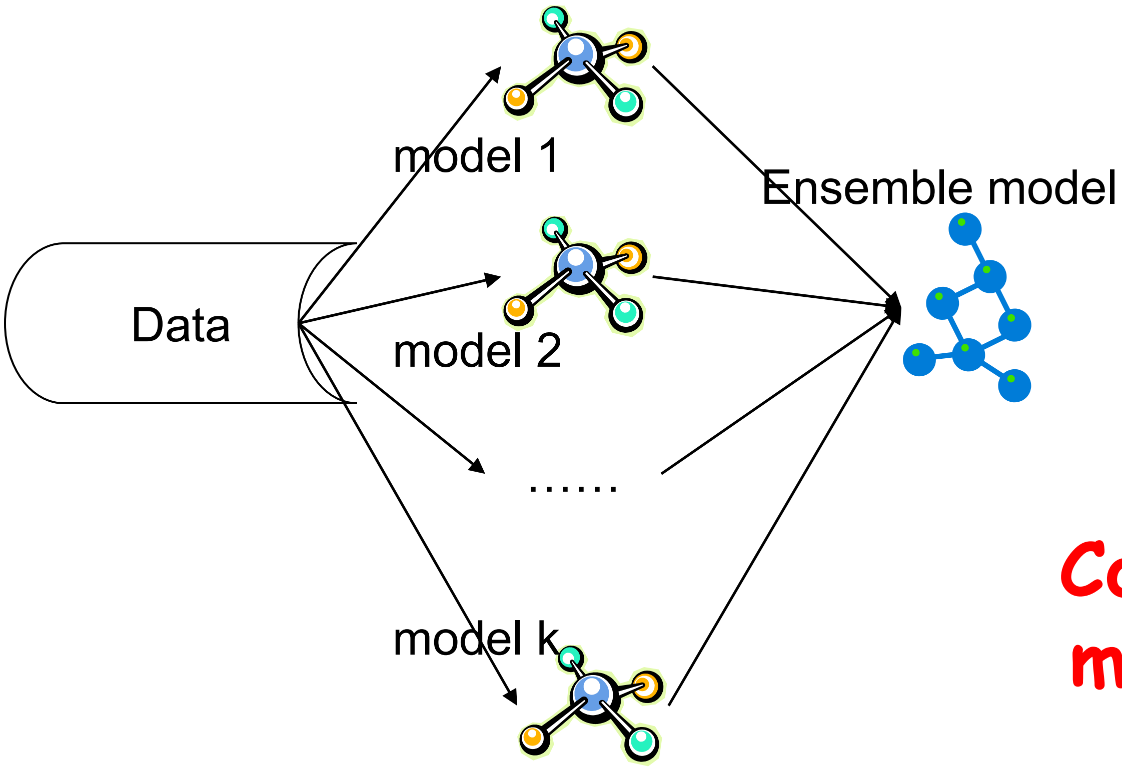
1 Department of Computer Science
University of Illinois
2 IBM TJ Watson Research Center

*Slides and references available at
http://ews.uiuc.edu/~jinggao3/sdm10ensemble.htm

Outline

- An overview of ensemble methods
  - Motivations
  - Tutorial overview
- Supervised ensemble
- Unsupervised ensemble
- Semi-supervised ensemble
  - Multi-view learning
  - Consensus maximization among supervised and unsupervised models
- Applications
  - Stream classification, transfer learning, anomaly detection

Ensemble

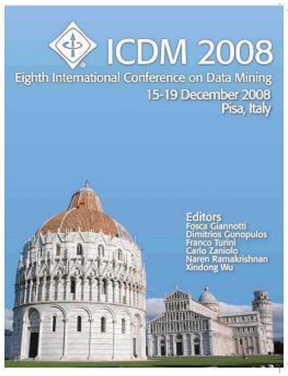


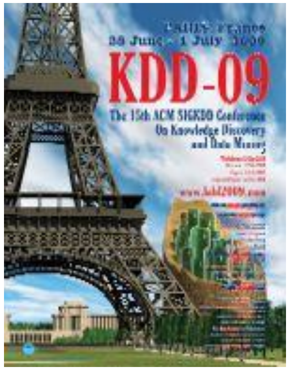
Combine multiple models into one!

Applications: classification, clustering, collaborative filtering, anomaly detection……

Stories of Success

- Million-dollar prize
  - Improve the baseline movie recommendation approach of Netflix by 10% in accuracy
  - The top submissions all combine several teams and algorithms as an ensemble





- Data mining competitions
  - Classification problems
  - Winning teams employ an ensemble of classifiers

Netflix Prize

- Supervised learning task
  - Training data is a set of users and ratings (1,2,3,4,5 stars) those users have given to movies.
  - Construct a classifier that given a user and an unrated movie, correctly classifies that movie as either 1, 2, 3, 4, or 5 stars
  - $1 million prize for a 10% improvement over Netflix's current movie recommender (MSE = 0.9514)
- Competition
  - At first, single-model methods are developed, and performances are improved
  - However, improvements slowed down
  - Later, individuals and teams merged their results, and significant improvements are observed

Leaderboard

Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
Grand Prize - RMSE = 0.8567 - Winning Team: BellKor's Pragmatic Chaos				
1	BellKor's Pragmatic Chaos	0.8567	10.06	2009-07-26 18:18:28
2	The Ensemble	0.8567	10.06	2009-07-26 18:38:22
3	Grand Prize Team	0.8582	9.90	2009-07-10 21:24:40
4	Opera Solutions and Vandelay United	0.8588	9.84	2009-07-10 01:12:31
5	Vandelay Industries !	0.8591	9.81	2009-07-10 00:32:20
6	PragmaticTheory	0.8594	9.77	2009-06-24 12:06:56
7	BellKor in BigChaos	0.8601	9.70	2009-05-13 08:14:09
8	Dace	0.8612	9.59	2009-07-24 17:18:43
9	Feeds2	0.8622	9.48	2009-07-12 13:11:51
10	BigChaos	0.8623	9.47	2009-04-07 12:33:59
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
12	BellKor	0.8624	9.46	2009-07-26 17:19:11
Progress Prize 2008 - RMSE = 0.8627 - Winning Team: BellKor in BigChaos				
13	xiangliang	0.8642	9.27	2009-07-15 14:53:22
14	Gravity	0.8643	9.26	2009-04-22 18:31:32
15	Ces	0.8651	9.18	2009-06-21 19:24:53
16	Invisible Ideas	0.8653	9.15	2009-07-15 15:53:04
17	Just a guy in a garage	0.8662	9.06	2009-05-24 10:02:54
18	J Dennis Su	0.8666	9.02	2009-03-07 17:16:17
19	Craig Carmichael	0.8666	9.02	2009-07-25 16:00:54
20	acmehill	0.8668	9.00	2009-03-21 16:20:50
Progress Prize 2007 - RMSE = 0.8723 - Winning Team: KorBell				
Cinematch score - RMSE = 0.9525				

"Our final solution (RMSE=0.8712) consists of blending 107 individual results. "

Motivations

- Motivations of ensemble methods
  - Ensemble model improves accuracy and robustness over single model methods
  - Applications:
    - distributed computing
    - privacy-preserving applications
    - large-scale data with reusable models
    - multiple sources of data
  - Efficiency: a complex problem can be decomposed into multiple sub-problems that are easier to understand and solve (divide-and-conquer approach)

Relationship with Related Studies (1)

- Multi-task learning
  - Learn **multiple** tasks simultaneously
  - Ensemble methods: use multiple models to learn **one** task
- Data integration
  - Integrate raw data
  - Ensemble methods: integrate information at the **model** level
- Mixture of models
  - Each model captures **part** of the global knowledge where the data have multi-modality
  - Ensemble methods: each model usually captures the **global** picture, but the models can complement each other

Relationship with Related Studies (2)

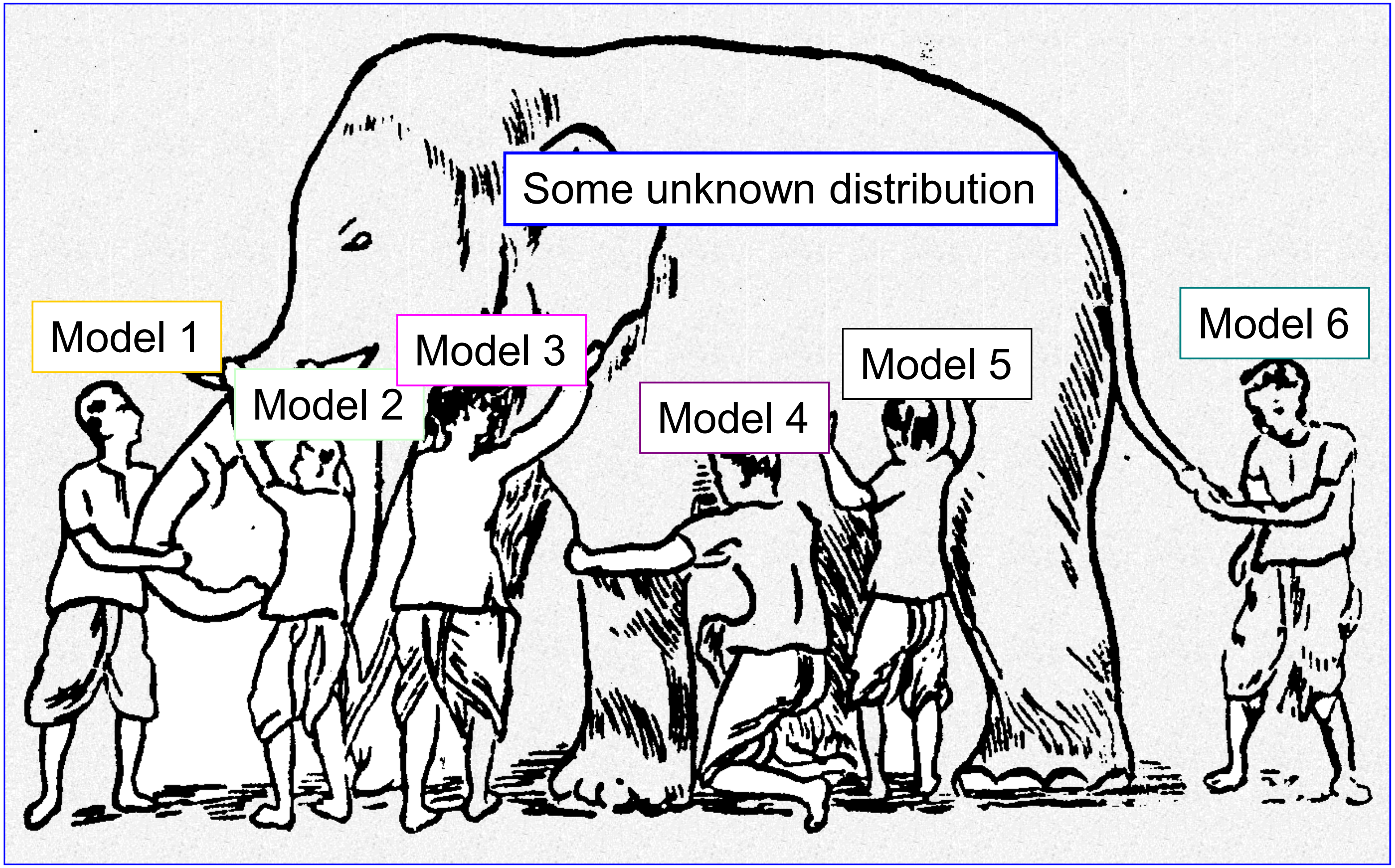
- Meta learning
  - **Learn** on meta-data (include base model output)
  - Ensemble methods: besides learn a joint model based on model output, we can also combine the output by **consensus**
- Non-redundant clustering
  - Give **multiple** non-redundant clustering solutions to users
  - Ensemble methods: give **one** solution to users which represents the consensus among all the base models

Why Ensemble Works? (1)

- Intuition
  - combining diverse, independent opinions in human decision-making as a protective mechanism (e.g. stock portfolio)
- Uncorrelated error reduction
  - Suppose we have 5 completely independent classifiers for majority voting
  - If accuracy is 70% for each
    - 10 (.7^3)(.3^2)+5(.7^4)(.3)+(.7^5)
    - **83.7% majority vote accuracy**
  - 101 such classifiers
    - **99.9% majority vote accuracy**

from T. Holloway, Introduction to Ensemble Learning, 2007.

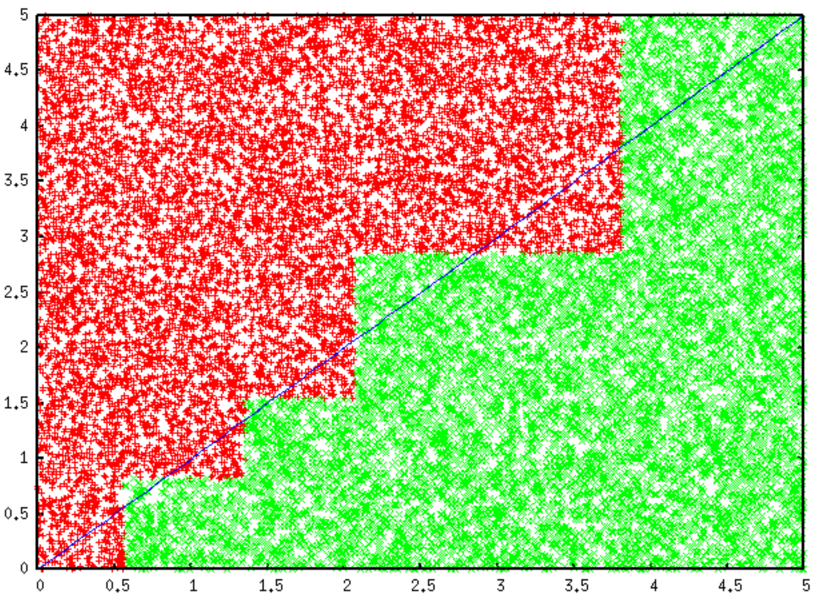
Why Ensemble Works? (2)



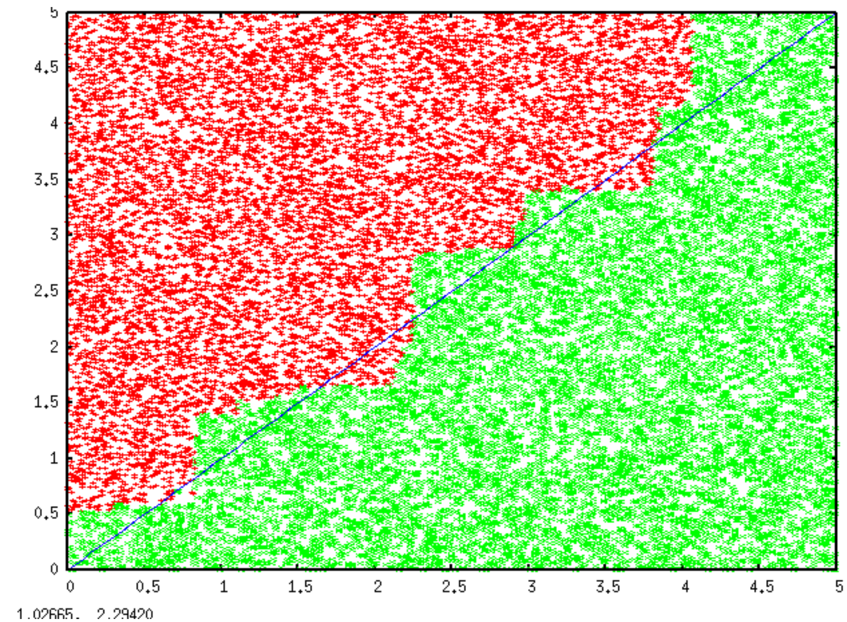
Ensemble gives the global picture!

Why Ensemble Works? (3)

- Overcome limitations of single hypothesis
  - The target function may not be implementable with individual classifiers, but may be approximated by model averaging



Decision Tree



Model Averaging

Research Focus

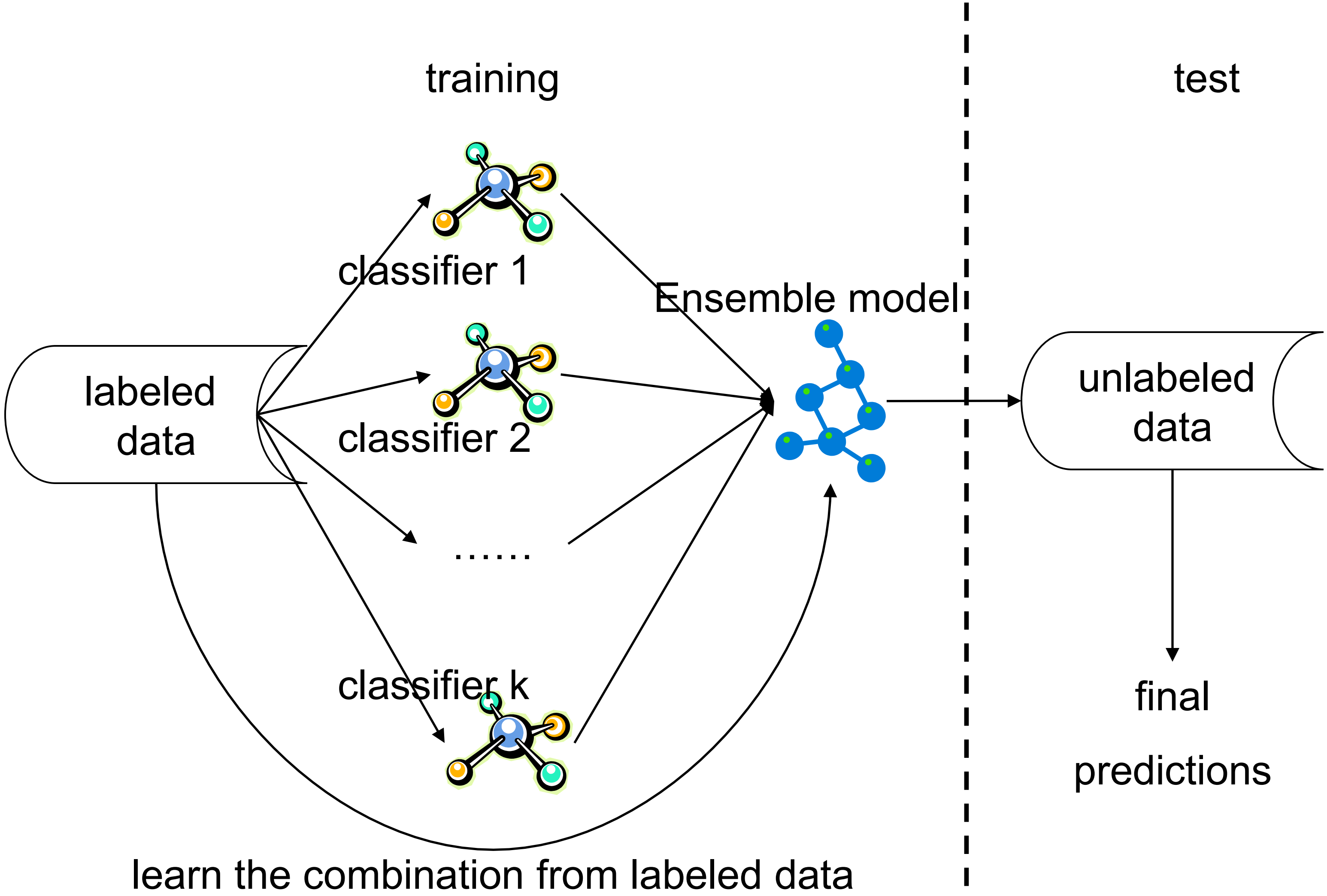
- Base models
  - Improve diversity!
- Combination scheme
  - Consensus (unsupervised)
  - Learn to combine (supervised)
- Tasks
  - Classification (supervised ensemble)
  - Clustering (unsupervised ensemble)

Summary

	Single Models	Combine by learning	Combine by consensus
Supervised Learning	SVM, Logistic Regression, ……	Boosting, rule ensemble, Bayesian model averaging, ……	Bagging, random forest, random decision tree ……
Semi-supervised Learning	Semi-supervised Learning, Collective Inference	Multi-view Learning	Consensus Maximization
Unsupervised Learning	K-means, Spectral Clustering, ……		Clustering Ensemble

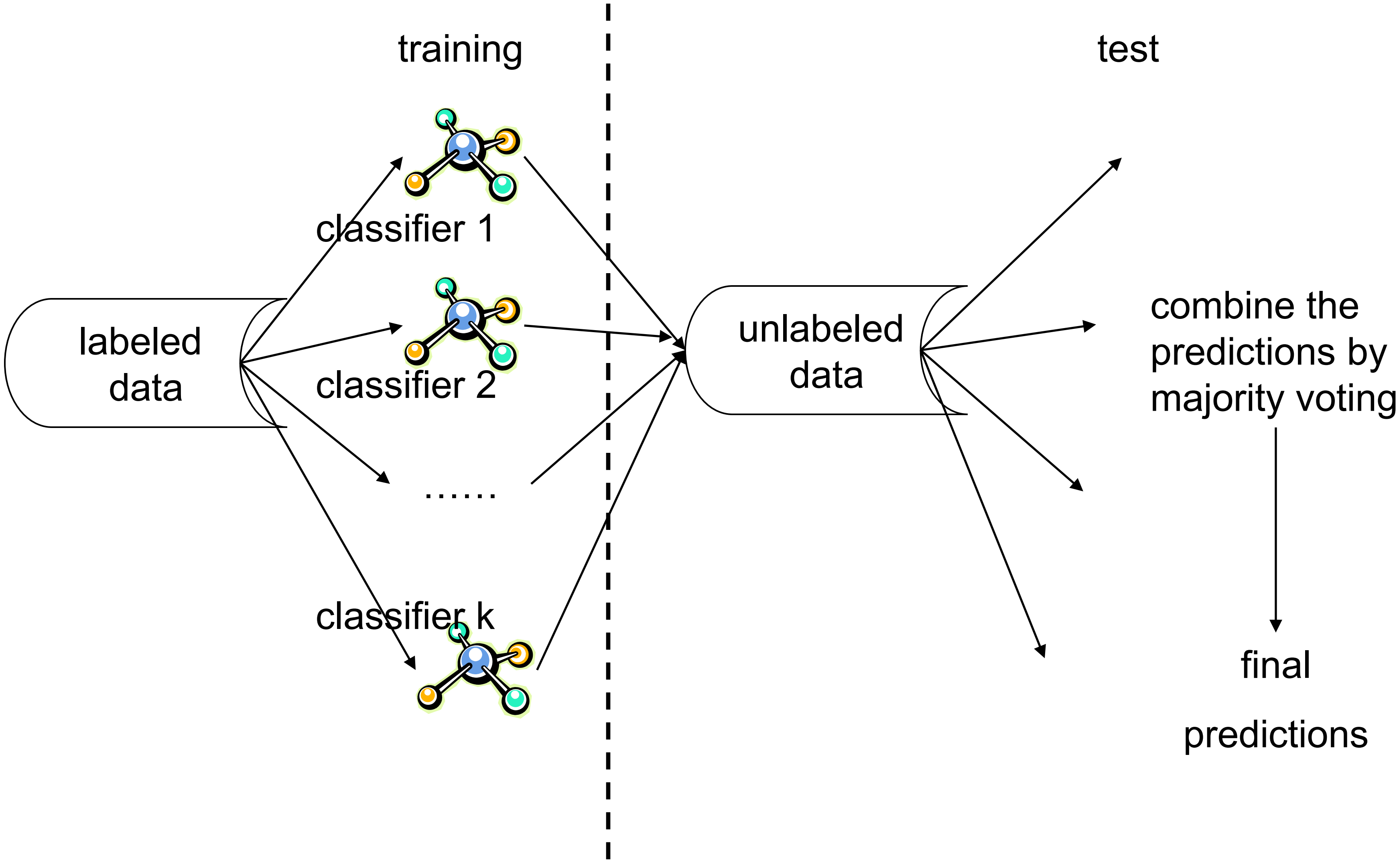
Review the ensemble methods in the tutorial

Ensemble of Classifiers—Learn to Combine



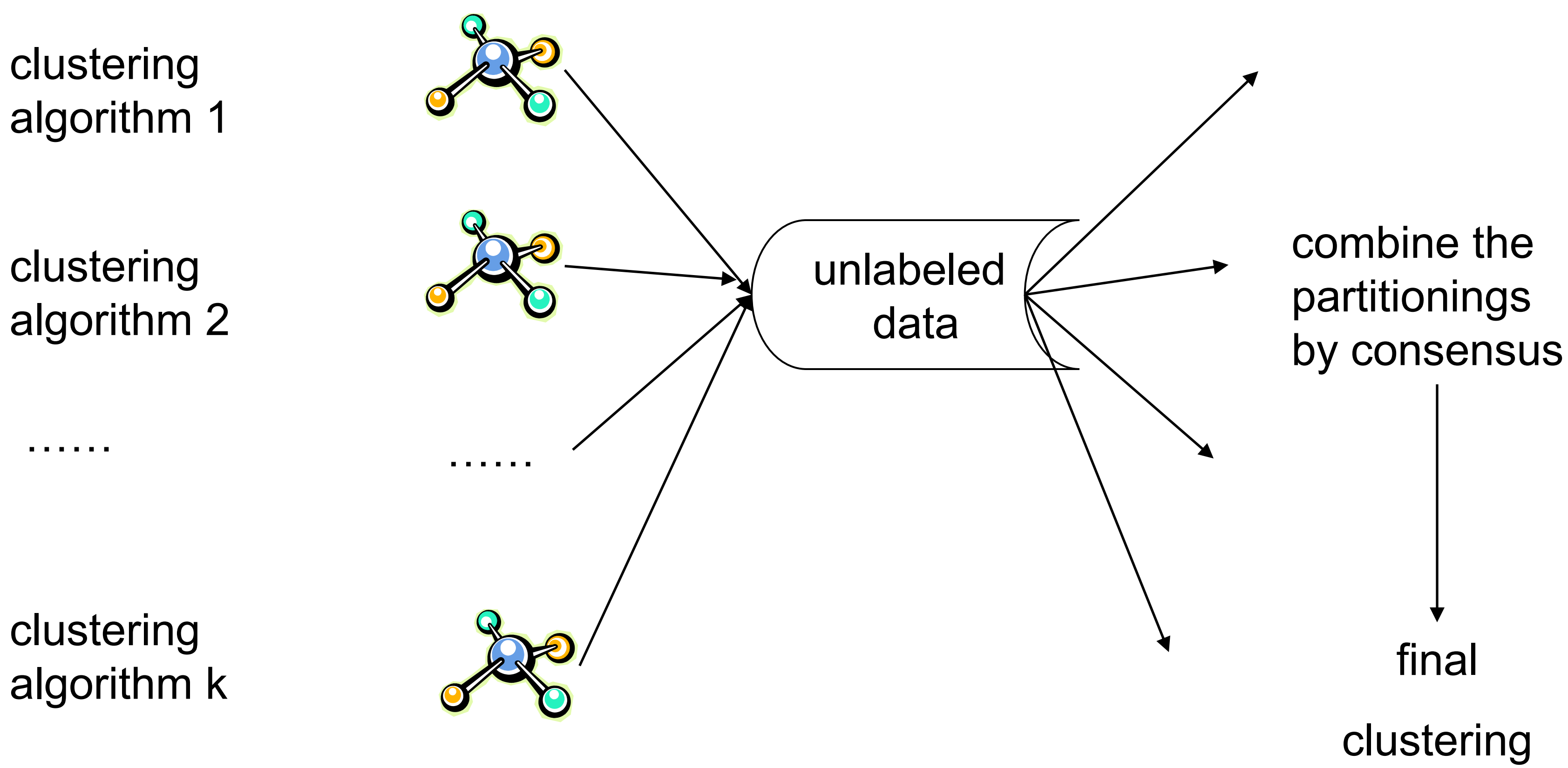
Algorithms: boosting, stacked generalization, rule ensemble, Bayesian model averaging……

Ensemble of Classifiers—Consensus



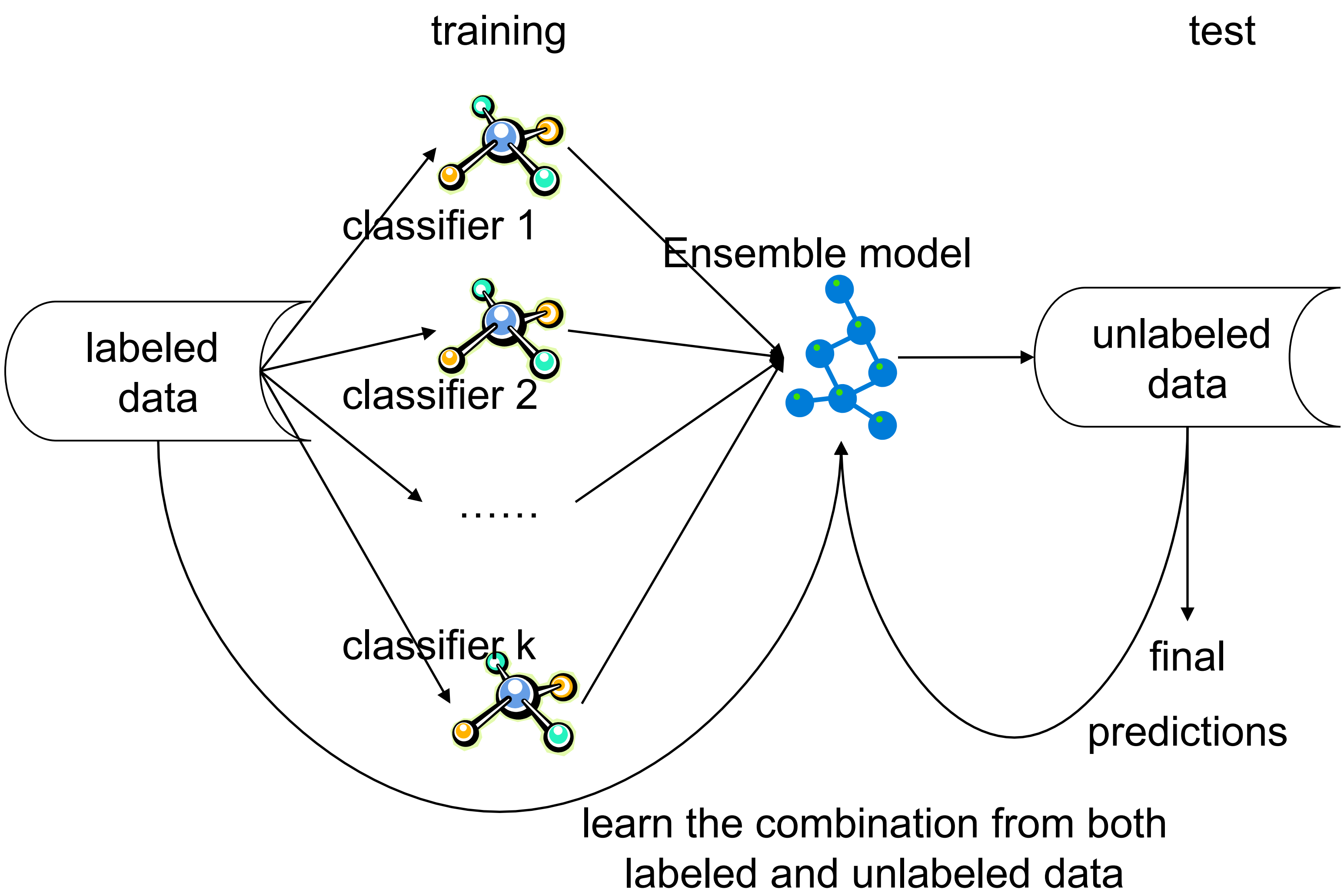
Algorithms: bagging, random forest, random decision tree, model averaging of probabilities……

Clustering Ensemble—Consensus



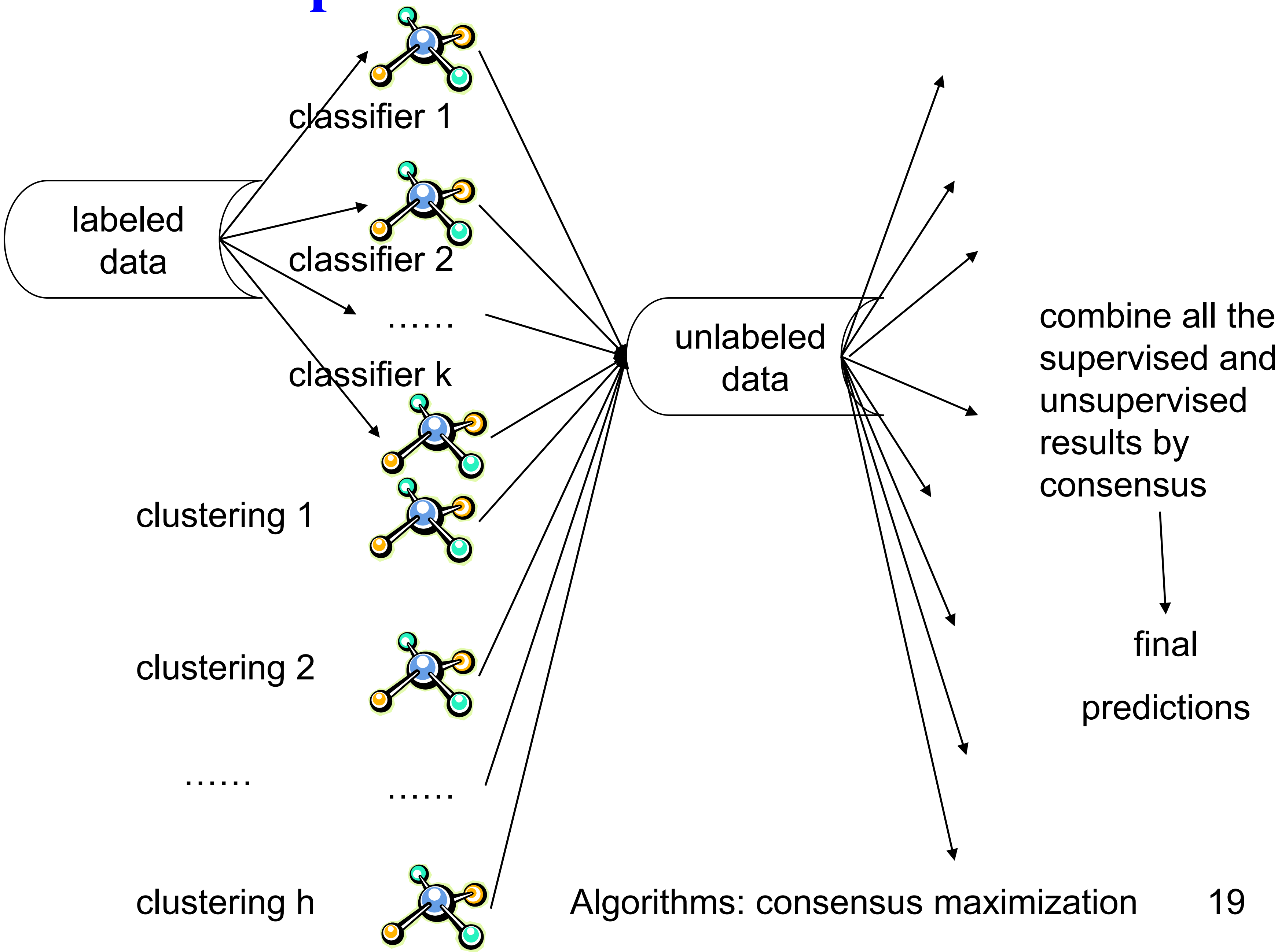
Algorithms: EM-based approach, instance-based, cluster-based approaches, correlation clustering, bipartite graph partitioning

Semi-Supervised Ensemble—Learn to Combine



Algorithms: multi-view learning

Semi-supervised Ensemble—Consensus



Algorithms: consensus maximization

Pros and Cons

	Combine by learning	Combine by consensus
Pros	Get useful feedbacks from labeled data Can potentially improve accuracy	Do not need labeled data Can improve the generalization performance
Cons	Need to keep the labeled data to train the ensemble May overfit the labeled data Cannot work when no labels are available	No feedbacks from the labeled data Require the assumption that consensus is better

Outline

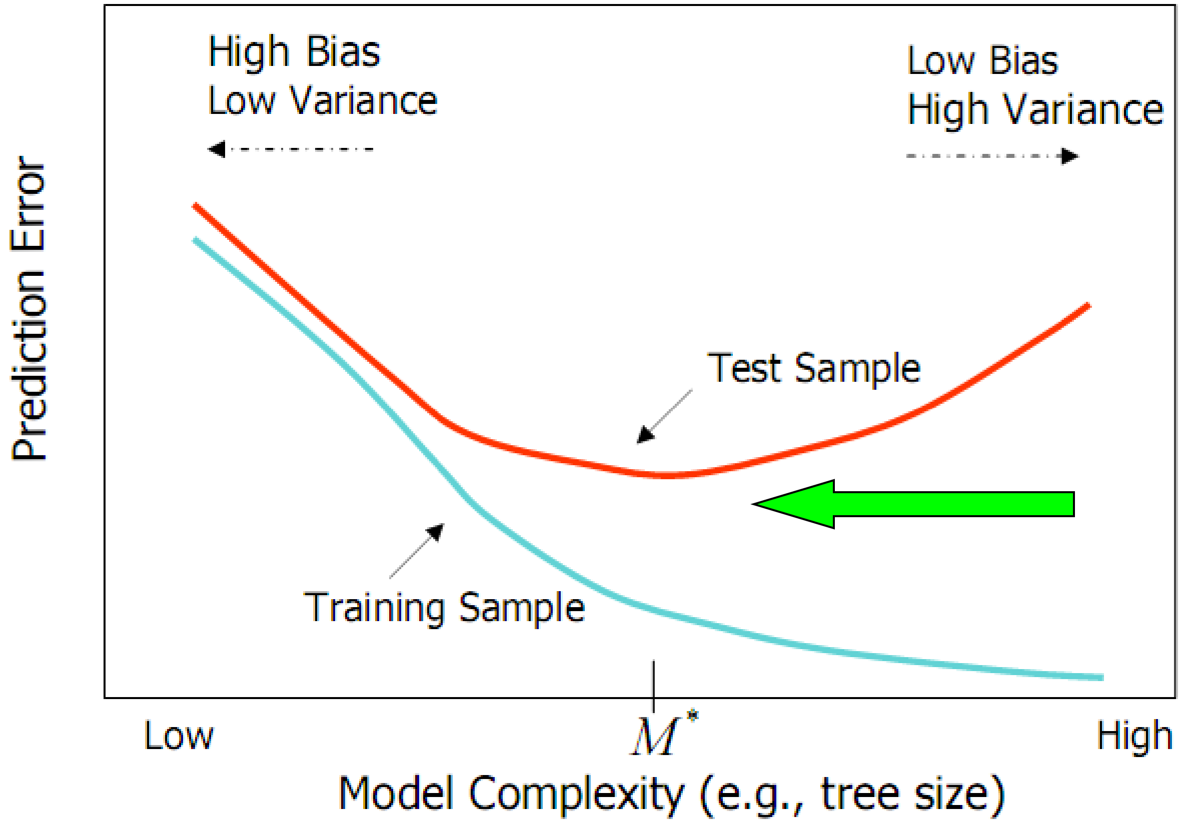
- An overview of ensemble methods
  - Motivations
  - Tutorial overview
- Supervised ensemble
- Unsupervised ensemble
- Semi-supervised ensemble
  - Multi-view learning
  - Consensus maximization among supervised and unsupervised models
- Applications
  - Stream classification, transfer learning, anomaly detection

Supervised Ensemble Methods

- Problem
  - Given a data set $D=\{x_1,x_2,\ldots,x_n\}$ and their corresponding labels $L=\{l_1,l_2,\ldots,l_n\}$
  - An ensemble approach computes:
    - A set of classifiers $\{f_1,f_2,\ldots,f_k\}$, each of which maps data to a class label: $f_j(x)=l$
    - A combination of classifiers f^* which minimizes generalization error: $f^*(x)= w_1 f_1(x)+ w_2 f_2(x)+\ldots+ w_k f_k(x)$

Bias and Variance

- Ensemble methods
  - Combine weak learners to reduce variance



from Elder, John. From Trees to Forests and Rule Sets - A Unified Overview of Ensemble Methods. 2007.

Generating Base Classifiers

- Sampling training examples
  - Train k classifiers on k subsets drawn from the training set
- Using different learning models
  - Use all the training examples, but apply different learning algorithms
- Sampling features
  - Train k classifiers on k subsets of features drawn from the feature space
- Learning —andomly”
  - Introduce randomness into learning procedures

Bagging* (1)

- Bootstrap
  - Sampling with replacement
  - Contains around 63.2% original records in each sample
- Bootstrap Aggregation
  - Train a classifier on each bootstrap sample
  - Use majority voting to determine the class label of ensemble classifier

*[Breiman96]

Bagging (2)

Original Data:

x	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
y	1	1	1	-1	-1	-1	-1	1	1	1

Bootstrap samples and classifiers:

x	0.1	0.2	0.2	0.3	0.4	0.4	0.5	0.6	0.9	0.9
y	1	1	1	1	-1	-1	-1	-1	1	1

x	0.1	0.2	0.3	0.4	0.5	0.5	0.9	1	1	1
y	1	1	1	-1	-1	-1	1	1	1	1

x	0.1	0.2	0.3	0.4	0.4	0.5	0.7	0.7	0.8	0.9
y	1	1	1	-1	-1	-1	-1	-1	1	1

x	0.1	0.2	0.5	0.5	0.5	0.7	0.7	0.8	0.9	1
y	1	1	-1	-1	-1	-1	-1	1	1	1

Combine predictions by majority voting

from P. Tan et al. Introduction to Data Mining.

Bagging (3)

- Error Reduction
  - Under mean squared error, bagging reduces variance and leaves bias unchanged
  - Consider idealized bagging estimator: $\bar{f}(x) = E(\hat{f}_z(x))$
  - The error is

$$E[Y - \hat{f}_z(x)]^2 = E[Y - \bar{f}(x) + \bar{f}(x) - \hat{f}_z(x)]^2$$

$$= E[Y - \bar{f}(x)]^2 + E[\bar{f}(x) - \hat{f}_z(x)]^2 \geq E[Y - \bar{f}(x)]^2$$

  - Bagging usually decreases MSE

from Elder, John. From Trees to Forests and Rule Sets - A Unified Overview of Ensemble Methods. 2007.

Boosting* (1)

- Principles
  - Boost a set of weak learners to a strong learner
  - Make records currently misclassified more important

- Example
  - Record 4 is hard to classify
  - Its weight is increased, therefore it is more likely to be chosen again in subsequent rounds

Original Data	1	2	3	4	5	6	7	8	9	10
Boosting (Round 1)	7	3	2	8	7	9	4	10	6	3
Boosting (Round 2)	5	4	9	4	2	5	1	7	4	2
Boosting (Round 3)	4	4	8	10	4	5	4	6	3	4

*[FrSc97]

from P. Tan et al. Introduction to Data Mining.

Boosting (2)

- AdaBoost
  - Initially, set uniform weights on all the records
  - At each round
    - Create a bootstrap sample based on the weights
    - Train a classifier on the sample and apply it on the original training set
    - Records that are wrongly classified will have their weights increased
    - Records that are classified correctly will have their weights decreased
    - If the error rate is higher than 50%, start over
  - Final prediction is weighted average of all the classifiers with weight representing the training accuracy

Boosting (3)

- Determine the weight
  - For classifier i, its error is

$$\varepsilon_i = \frac{\sum_{j=1}^{N} w_j \delta(C_i(x_j) \neq y_j)}{\sum_{j=1}^{N} w_j}$$

  - The classifier's importance is represented as:

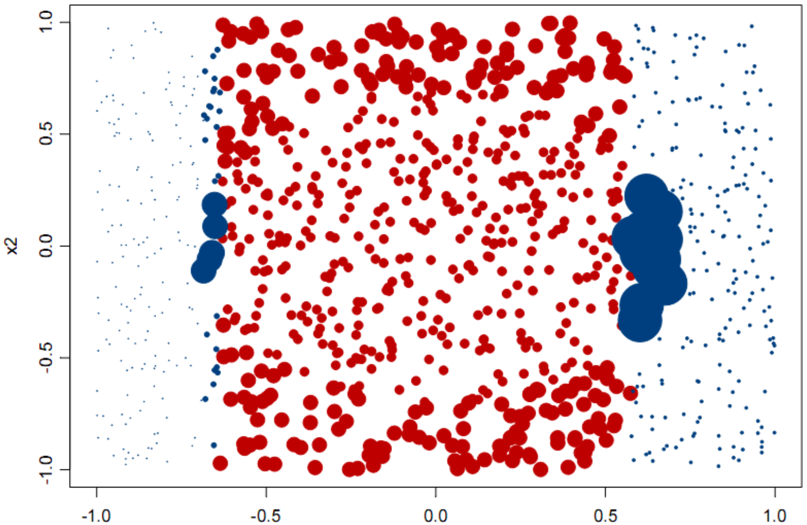
$$\alpha_i = \frac{1}{2} \ln\left(\frac{1-\varepsilon_i}{\varepsilon_i}\right)$$

  - The weight of each record is updated as:

$$w_j^{(i+1)} = \frac{w_j^{(i)} \exp\left(-\alpha_i y_j C_i(x_j)\right)}{Z^{(i)}}$$

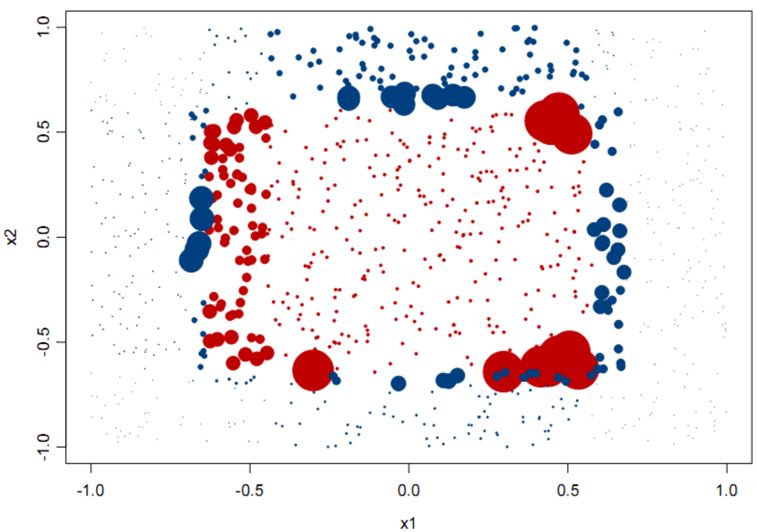
  - Final combination:

$$C^*(x) = \arg\max_y \sum_{i=1}^{K} \alpha_i \delta\left(C_i(x) = y\right)$$

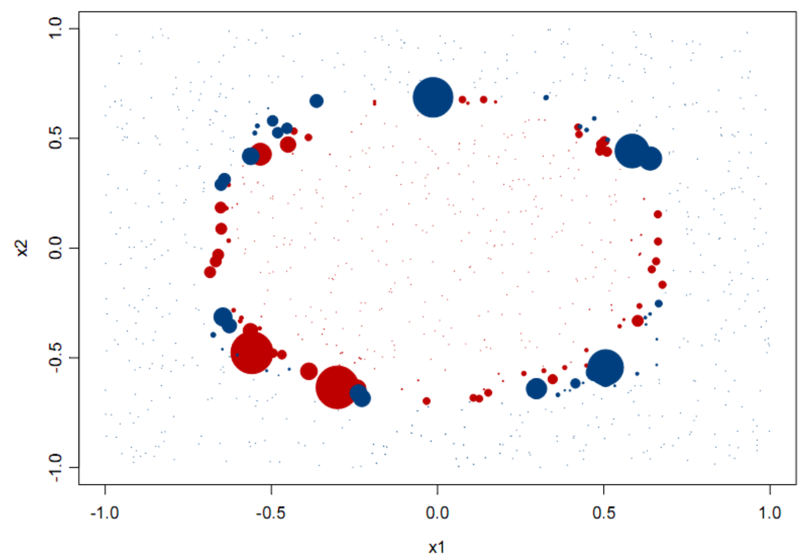


Classifications (colors) and Weights (size) after *1 iteration* Of AdaBoost

3 iterations



20 iterations



from Elder, John. From Trees to Forests and Rule Sets - A Unified Overview of Ensemble Methods. 2007.

Boosting (4)

- Explanation
  - Among the classifiers of the form:

$$f(x) = \sum_{i=1}^{K} \alpha_i C_i(x)$$

  - We seek to minimize the exponential loss function:

$$\sum_{j=1}^{N} \exp\left(-y_j f(x_j)\right)$$

  - Not robust in noisy settings

Random Forests* (1)

- Algorithm
  - Choose *T*—number of trees to grow
  - Choose $m << M$ (M is the number of total features) —number of features used to calculate the best split at each node
  - For each tree
    - Choose a training set by choosing *N* times (N is the number of training examples) with replacement from the training set
    - For each node, randomly choose *m* features and calculate the best split
    - Fully grown and not pruned
  - Use majority voting among all the trees

*[Breiman01]

Random Forests (2)

- Discussions
  - Bagging+random features
  - Improve accuracy
    - Incorporate more diversity and reduce variances
  - Improve efficiency
    - Searching among subsets of features is much faster than searching among the complete set

Data set	Adaboost	Selection	Forest-RI single input	One tree
Glass	22.0	20.6	21.2	36.9
Breast cancer	3.2	2.9	2.7	6.3
Diabetes	26.6	24.2	24.3	33.1
Sonar	15.6	15.9	18.0	31.7
Vowel	4.1	3.4	3.3	30.4
Ionosphere	6.4	7.1	7.5	12.7
Vehicle	23.2	25.8	26.4	33.1
German credit	23.5	24.4	26.2	33.3
Image	1.6	2.1	2.7	6.4
Ecoli	14.8	12.8	13.0	24.5
Votes	4.8	4.1	4.6	7.4
Liver	30.7	25.1	24.7	40.6
Letters	3.4	3.5	4.7	19.8
Sat-images	8.8	8.6	10.5	17.2
Zip-code	6.2	6.3	7.8	20.6
Waveform	17.8	17.2	17.3	34.0
Twonorm	4.9	3.9	3.9	24.7
Threenorm	18.8	17.5	17.5	38.4
Ringnorm	6.9	4.9	4.9	25.7

Random Decision Tree* (1)

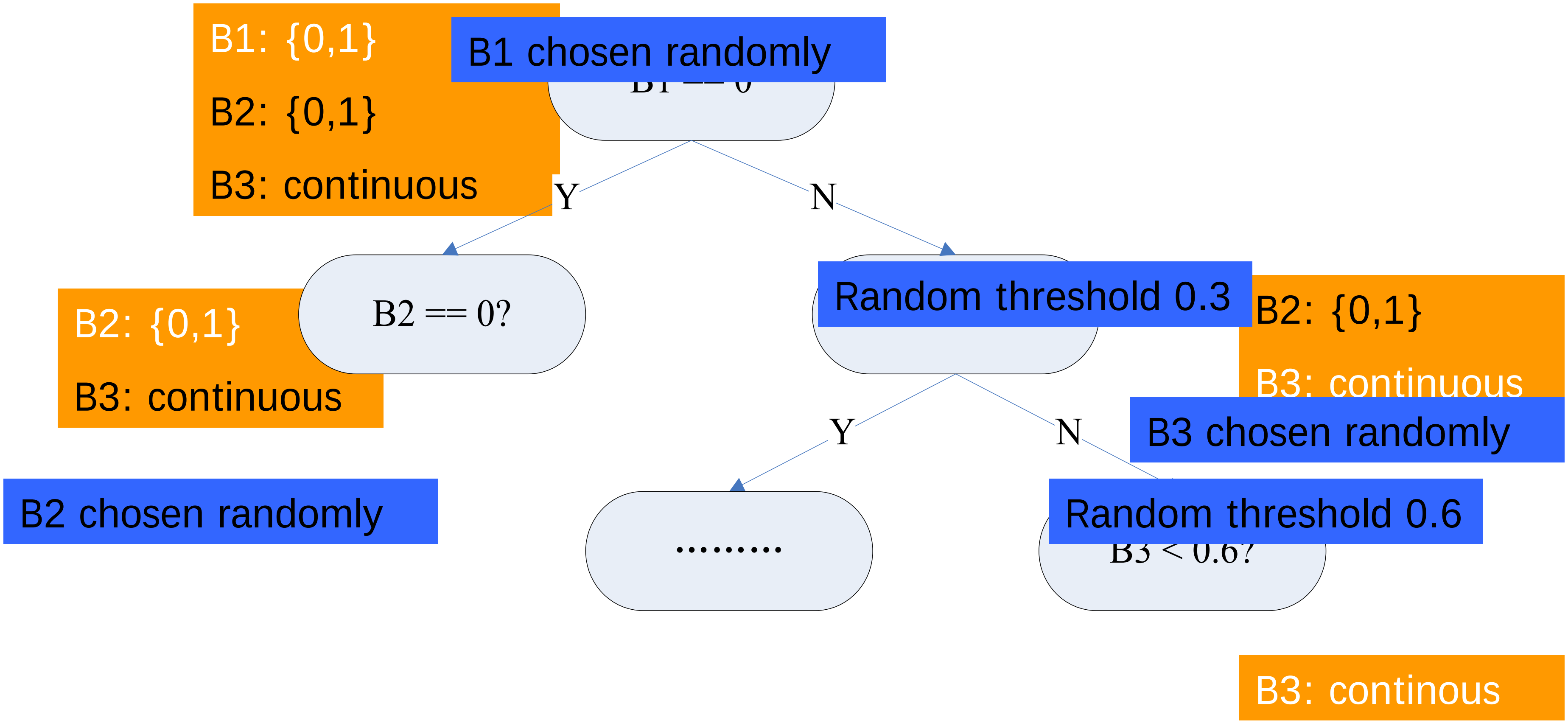
- Principle
  - Single-model learning algorithms
    - Fix structure of the model, minimize some form of errors, or maximize data likelihood (eg., Logistic regression, Naive Bayes, etc.)
    - Use some —free-form” functions to match the data given some —preference criteria” such as information gain, gini index and MDL. (eg., Decision Tree, Rule-based Classifiers, etc.)
  - Such methods will make mistakes if
    - Data is insufficient
    - Structure of the model or the preference criteria is inappropriate for the problem
  - Ensemble
    - Make no assumption about the true model, neither parametric form nor free form
    - Do not prefer one base model over the other, just average them

*[FGM+05]

Random Decision Tree (2)

- Algorithm
  - At each node, an un-used feature is chosen randomly
    - A discrete feature is un-used if it has never been chosen previously on a given decision path starting from the root to the current node.
    - A continuous feature can be chosen multiple times on the same decision path, but each time a different threshold value is chosen
  - We stop when one of the following happens:
    - A node becomes too small (<= 3 examples).
    - Or the total height of the tree exceeds some limits, such as the total number of features.
  - Prediction
    - Simple averaging over multiple trees

Random Decision Tree (3)

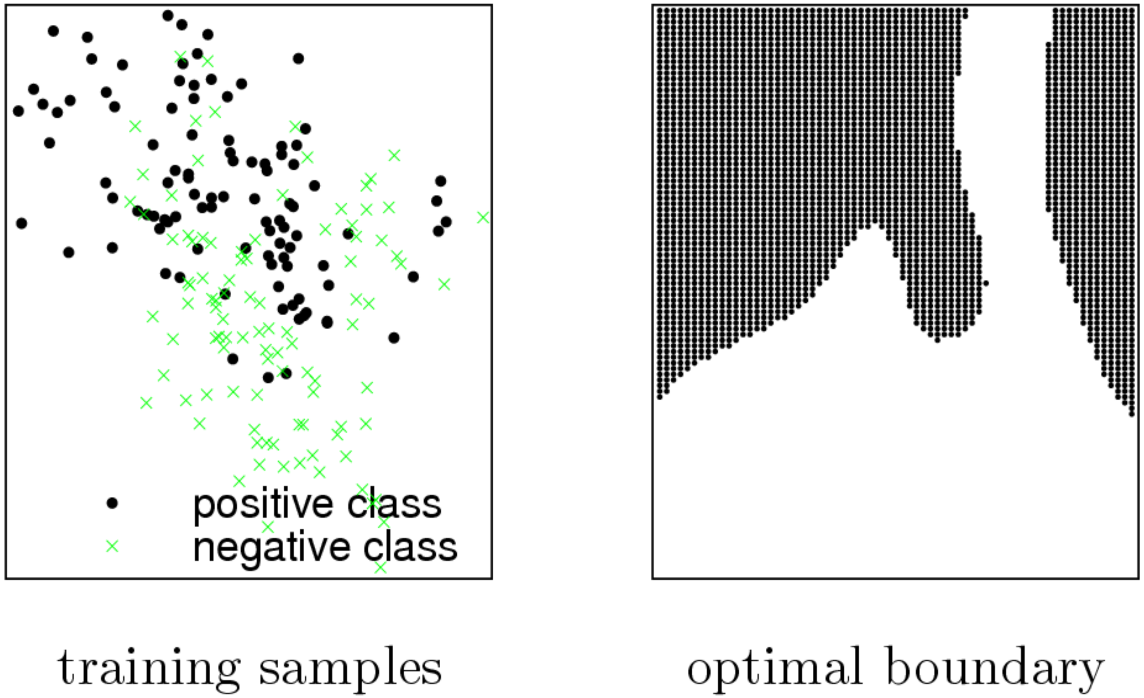


Random Decision Tree (4)

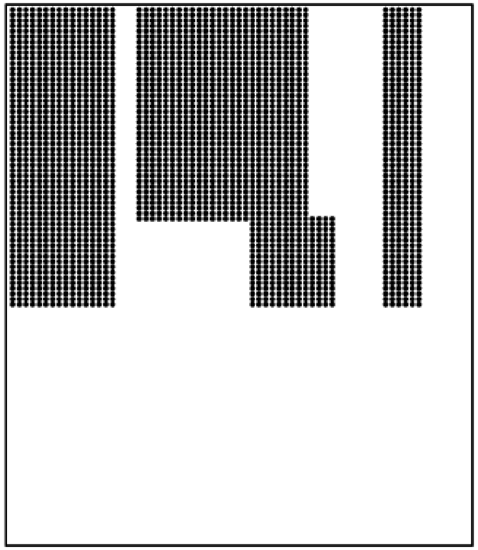
- Potential Advantages
  - Training can be very efficient. Particularly true for very large datasets.
    - No cross-validation based estimation of parameters for some parametric methods.
  - Natural multi-class probability.
  - Natural multi-label classification and probability estimation.
  - Imposes very little about the structures of the model.

Optimal Decision Boundary

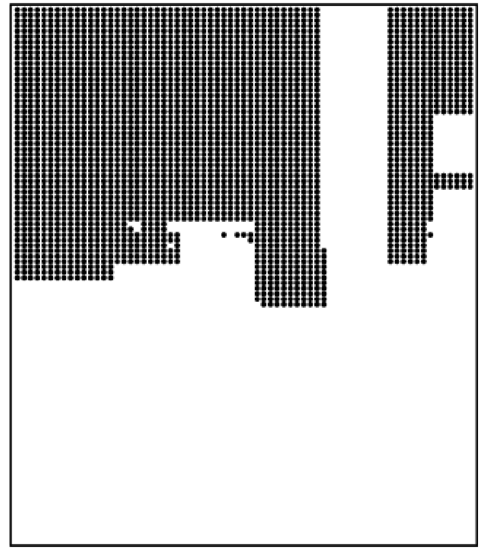
Figure 3.5: Gaussian mixture training samples and optimal boundary.



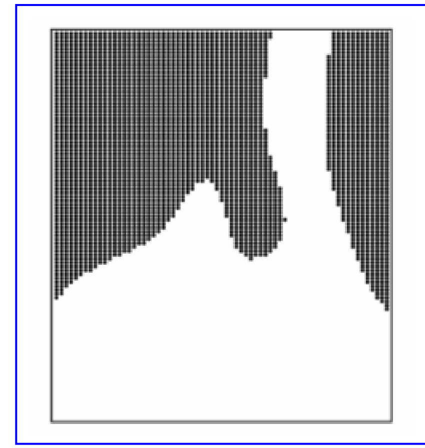
from Tony Liu’s thesis (supervised by Kai Ming Ting)

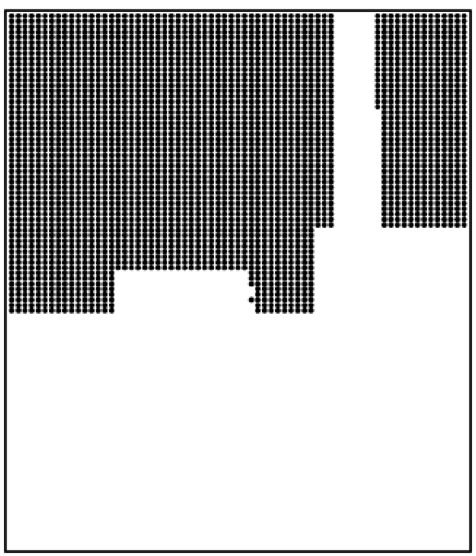


(a) unpruned C4.5

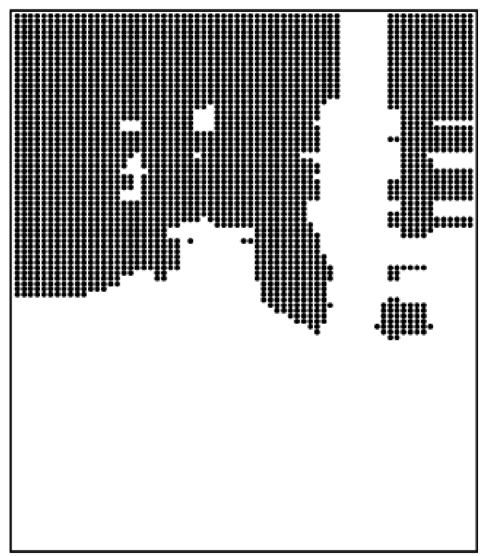


(b) Bagging





(c) Random Forests



(d) Complete-random tree ensemble

RDT looks like the optimal boundary

Outline

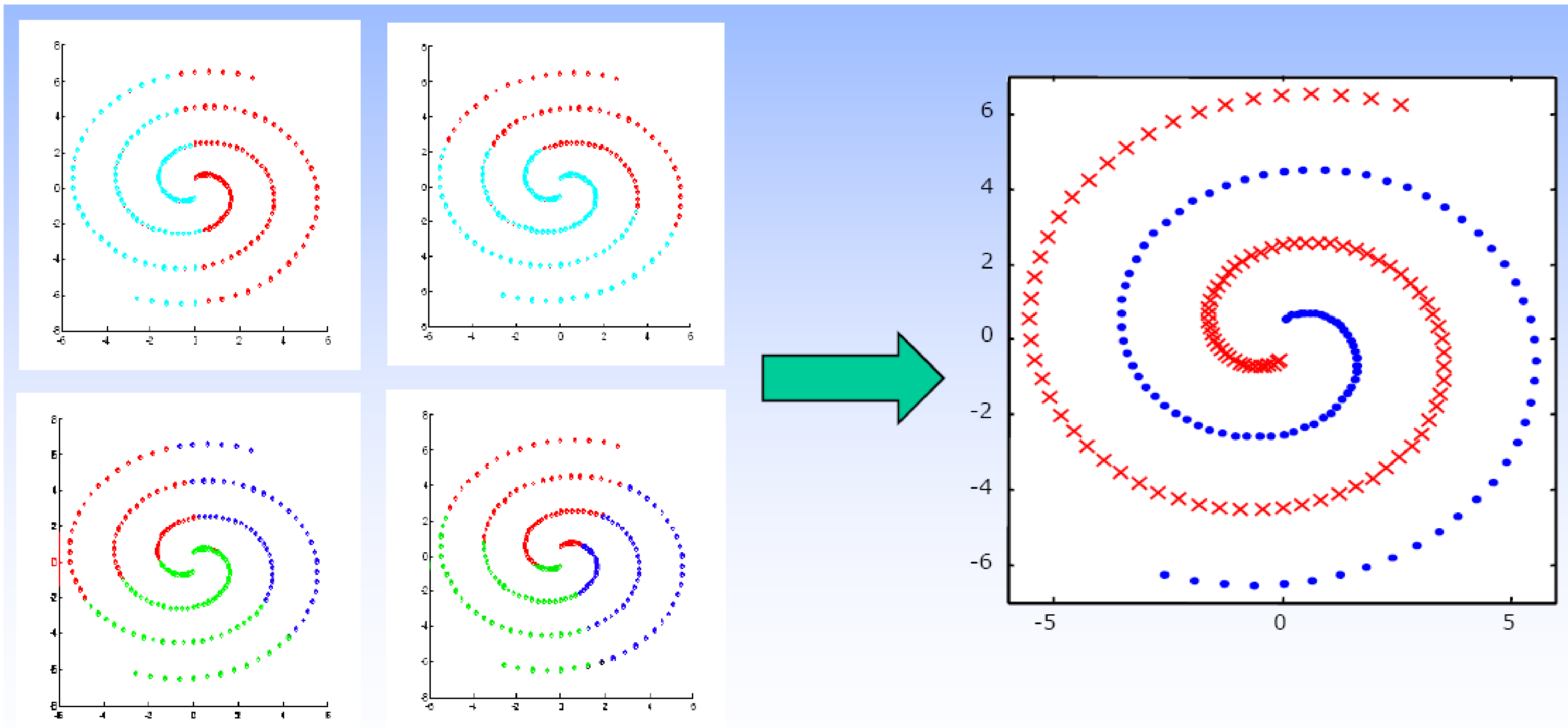
- An overview of ensemble methods
  - Motivations
  - Tutorial overview
- Supervised ensemble
- Unsupervised ensemble
- Semi-supervised ensemble
  - Multi-view learning
  - Consensus maximization among supervised and unsupervised models
- Applications
  - Stream classification, transfer learning, anomaly detection

Clustering Ensemble

- Problem
  - Given an unlabeled data set $D=\{x_1,x_2,\ldots,x_n\}$
  - An ensemble approach computes:
    - A set of clustering solutions $\{C_1,C_2,\ldots,C_k\}$, each of which maps data to a cluster: $f_j(x)=m$
    - A unified clustering solutions f^* which combines base clustering solutions by their consensus

Motivations

- Goal
  - Combine “weak” clusterings to a better one



from A. Topchy et. al. Clustering Ensembles: Models of Consensus and Weak Partitions. PAMI, 2005

Methods (1)

- How to get base models?
  - Bootstrap samples
  - Different subsets of features
  - Different clustering algorithms
  - Random number of clusters
  - Random initialization for K-means
  - Incorporating random noises into cluster labels
  - Varying the order of data in on-line methods such as BIRCH

Methods (2)

- How to combine the models?
  - Direct approach
    - Find the correspondence between the labels in the partitions and fuse the clusters with the same labels
  - Indirect approach (Meta clustering)
    - Treat each output as a categorical variable and cluster in the new feature space
    - Avoid relabeling problems
    - Algorithms differ in how they represent base model output and how consensus is defined
    - Focus on hard clustering methods in this tutorial

An Example

base clustering models

	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_3$	$\mathcal{C}$
v_1	1	1	1	1
v_2	1	2	2	2
v_3	2	1	1	1
v_4	2	2	2	2
v_5	3	3	3	3
v_6	3	4	3	3

objects

they may not represent
the same cluster!

The goal: get the consensus clustering

from A. Gionis et. al. Clustering Aggregation. TKDD, 2007

Cluster-based Similarity Partitioning Algorithm (CSPA)

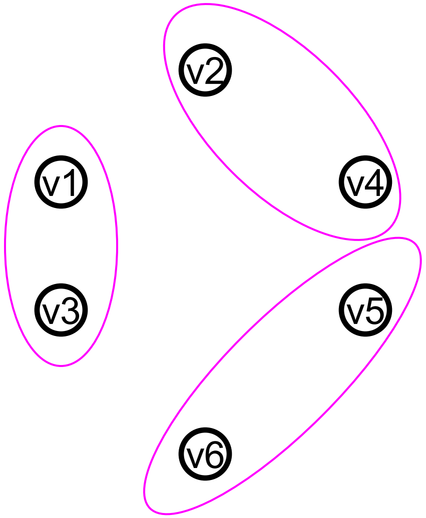
- Clustering objects
  - Similarity between two objects is defined as the percentage of common clusters they fall into
  - Conduct clustering on the new similarity matrix

Similarity between v_i and v_j is:

$$s(v_i, v_j) = \frac{\sum_{k=1}^{K} \delta(C_k(v_i) - C_k(v_j))}{K}$$

Cluster-based Similarity Partitioning Algorithm (CSPA)

	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_3$	$\mathcal{C}$
v_1	1	1	1	1
v_2	1	2	2	2
v_3	2	1	1	1
v_4	2	2	2	2
v_5	3	3	3	3
v_6	3	4	3	3

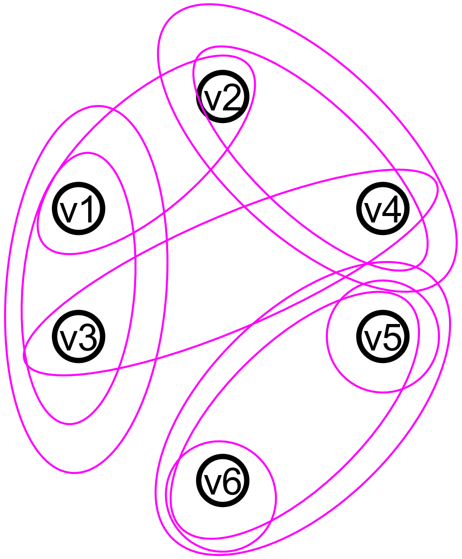


HyperGraph-Partitioning Algorithm (HGPA)

- Hypergraph representation and clustering
  - Each node denotes an object
  - A hyperedge is a generalization of an edge in that it can connect any number of nodes
  - For objects that are put into the same cluster by a clustering algorithm, draw a hyperedge connecting them
  - Partition the hypergraph by minimizing the number of cut hyperedges
  - Each component forms a meta cluster

HyperGraph-Partitioning Algorithm (HGPA)

	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_3$	$\mathcal{C}$
v_1	1	1	1	1
v_2	1	2	2	2
v_3	2	1	1	1
v_4	2	2	2	2
v_5	3	3	3	3
v_6	3	4	3	3



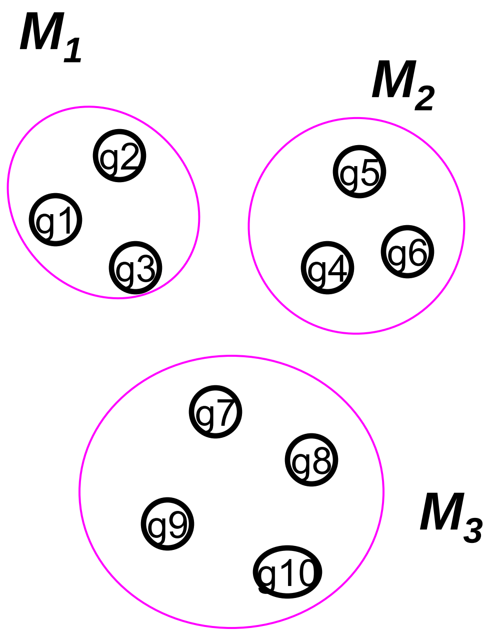
Hypergraph representation– a circle denotes a hyperedge

Meta-Clustering Algorithm (MCLA)

- Clustering clusters
  - Regard each cluster from a base model as a record
  - Similarity is defined as the percentage of shared common objects
    - eg. Jaccard measure
  - Conduct meta-clustering on these clusters
  - Assign an object to its most associated meta-cluster

Meta-Clustering Algorithm (MCLA)

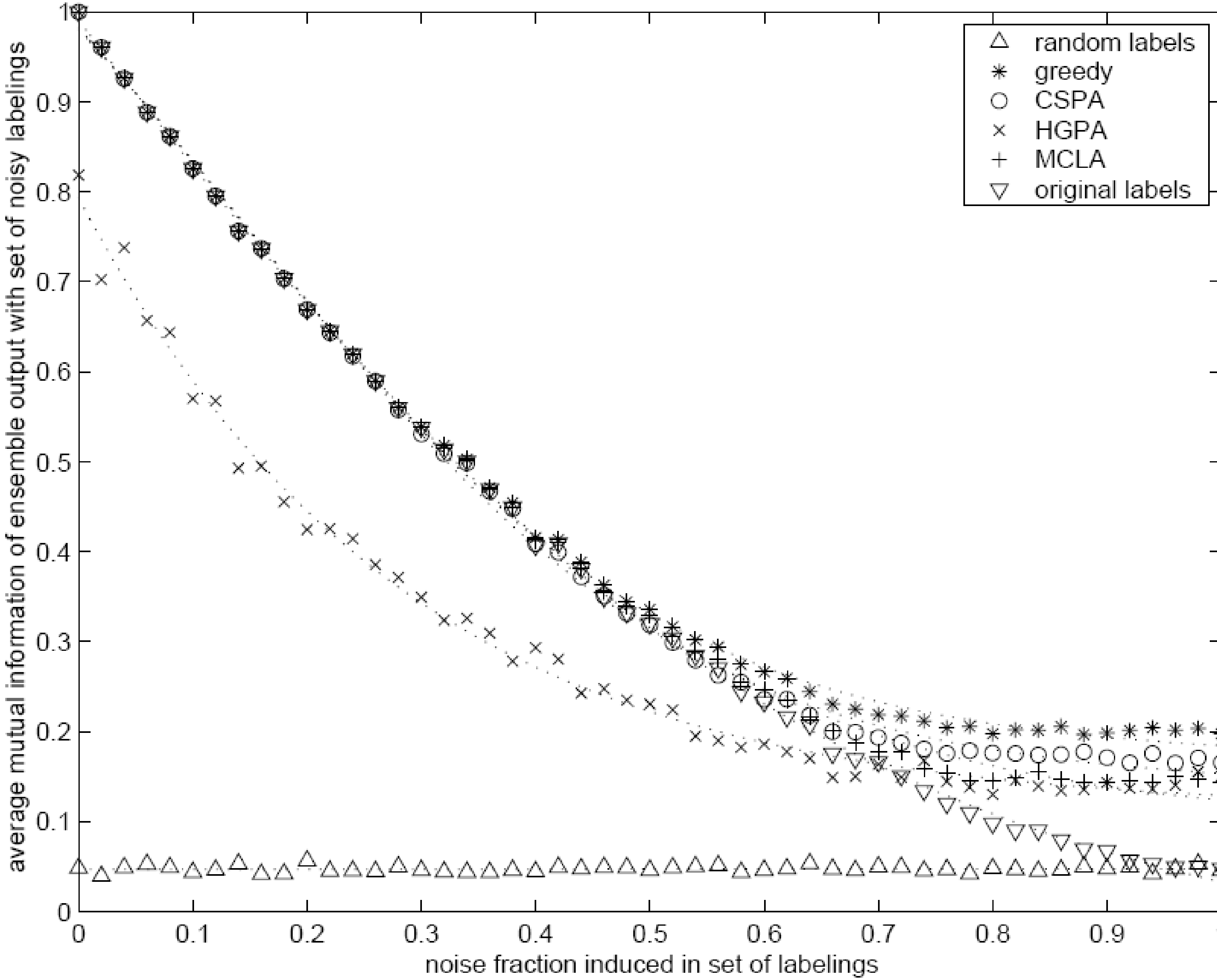
	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_3$	$\mathcal{C}$	M_1	M_2	M_3
v_1	1	1	1	1	3	0	0
v_2	1	2	2	2	1	2	0
v_3	2	1	1	1	2	1	0
v_4	2	2	2	2	0	3	0
v_5	3	3	3	3	0	0	3
v_6	3	4	3	3	0	0	3



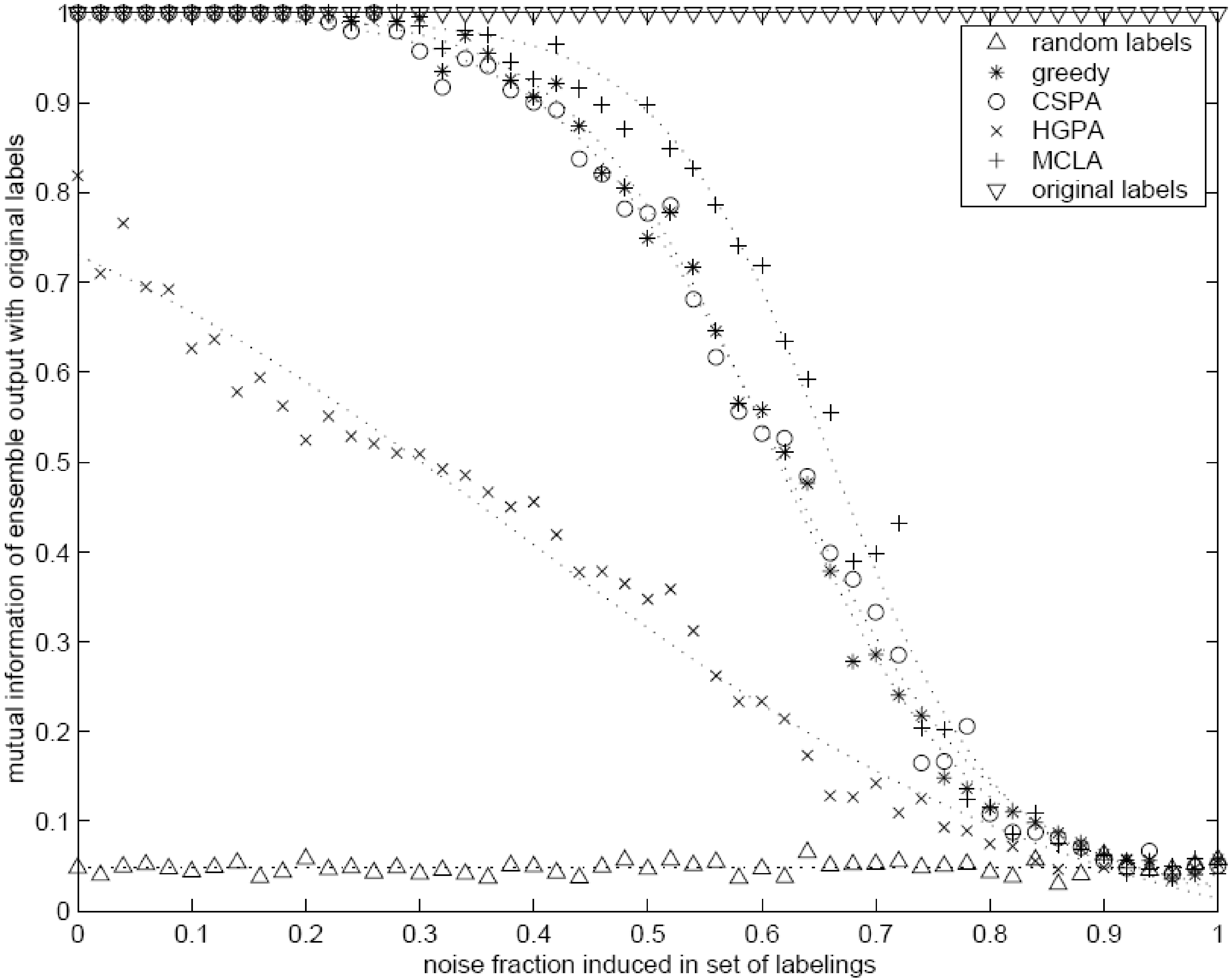
Comparisons*

- Time complexity
  - CSPA (clustering objects): $O(n^2kr)$
  - HGPA (hypergraph partitioning): $O(nkr)$
  - MCLA (clustering clusters): $O(nk^2r^2)$
  - n-number of objects, k-number of clusters, r-number of clustering solutions
- Clustering quality
  - MCLA tends to be best in low noise/diversity settings
  - HGPA/CSPA tend to be better in high noise/diversity settings

All three algorithms are from *[StGh03]



random labels
greedy
CSPA
HGPA
MCLA
original labels
average mutual information of ensemble output with set of noisy labelings
noise fraction induced in set of labelings
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


random labels
greedy
CSPA
HGPA
MCLA
original labels
mutual information of ensemble output with original labels
noise fraction induced in set of labelings
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

A Mixture Model of Consensus*

- Probability-based
  - Assume output comes from a mixture of models
  - Use EM algorithm to learn the model
- Generative model
  - The clustering solutions for each object are represented as nominal features--v_i
  - v_i is described by a mixture of k components, each component follows a multinomial distribution
  - Each component is characterized by distribution parameters θ_j

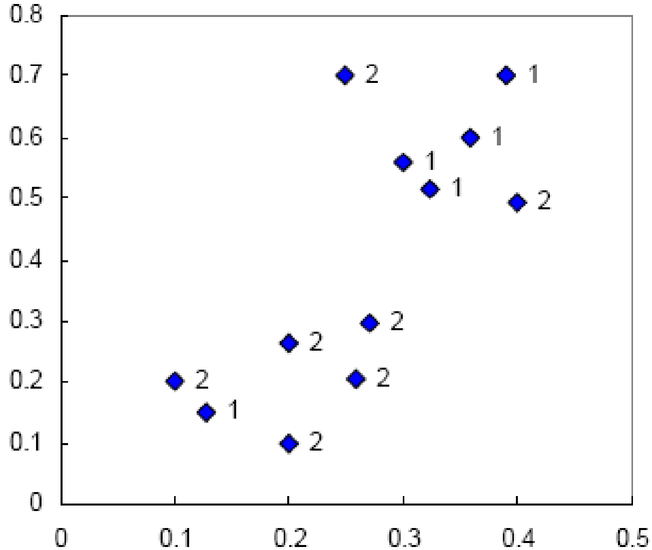
*[PTJ05]

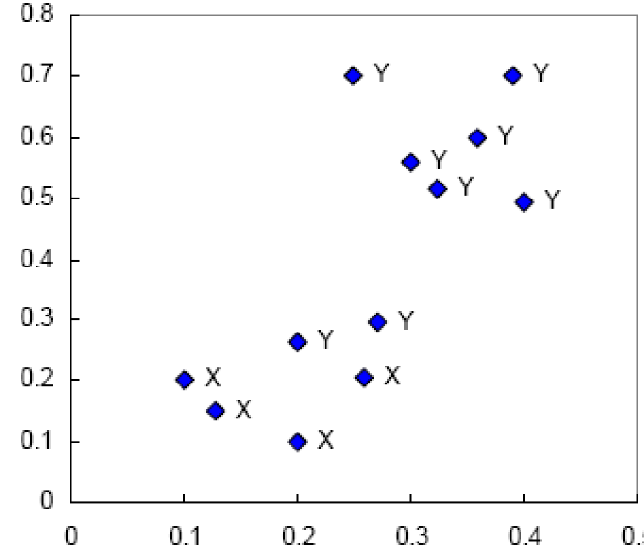
EM Method

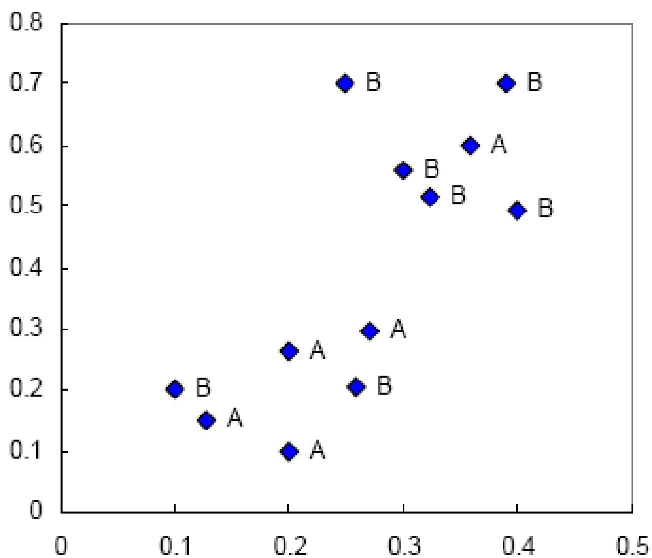
- Maximize log likelihood

$$\sum_{i=1}^{n} \log\left(\sum_{j=1}^{k} \alpha_j P(v_i | \theta_j)\right)$$

- Hidden variables
  - z_i denotes which consensus cluster the object belongs to
- EM procedure
  - E-step: compute expectation of z_i
  - M-step: update model parameters to maximize likelihood


0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
2
1
1
1
1
2
2
2
2
2
1
2


0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
Y
Y
Y
Y
Y
Y
Y
Y
X
X
X
X


0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
B
B
A
B
B
B
A
A
B
B
A
A

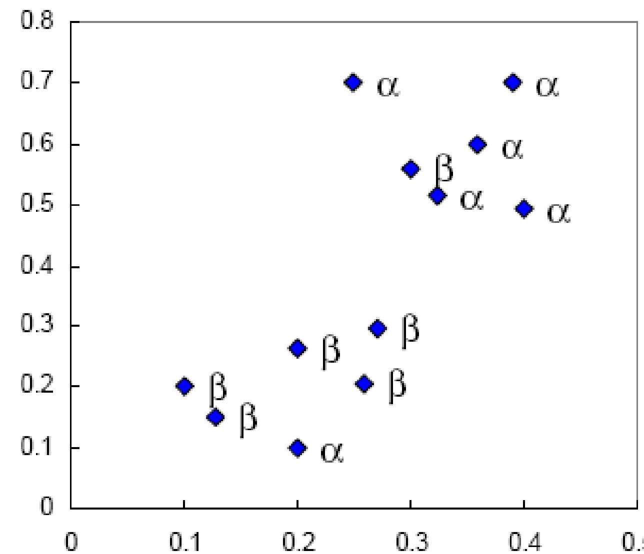

0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
α
α
α
β
α
α
β
β
β
β
β
α

Table 1: Clustering ensemble and consensus solution

	π_1	π_2	π_3	π_4	$E[z_{i1}]$	$E[z_{i2}]$	Consensus
$\mathbf{y}_1$	2	B	X	β	0.999	0.001	**1**
$\mathbf{y}_2$	2	A	X	α	0.997	0.003	**1**
$\mathbf{y}_3$	2	A	Y	β	0.943	0.057	**1**
$\mathbf{y}_4$	2	B	X	β	0.999	0.001	**1**
$\mathbf{y}_5$	1	A	X	β	0.999	0.001	**1**
$\mathbf{y}_6$	2	A	Y	β	0.943	0.057	**1**
$\mathbf{y}_7$	2	B	Y	α	0.124	0.876	**2**
$\mathbf{y}_8$	1	B	Y	α	0.019	0.981	**2**
$\mathbf{y}_9$	1	B	Y	β	0.260	0.740	**2**
$\mathbf{y}_{10}$	1	A	Y	α	0.115	0.885	**2**
$\mathbf{y}_{11}$	2	B	Y	α	0.124	0.876	**2**
$\mathbf{y}_{12}$	1	B	Y	α	0.019	0.981	**2**

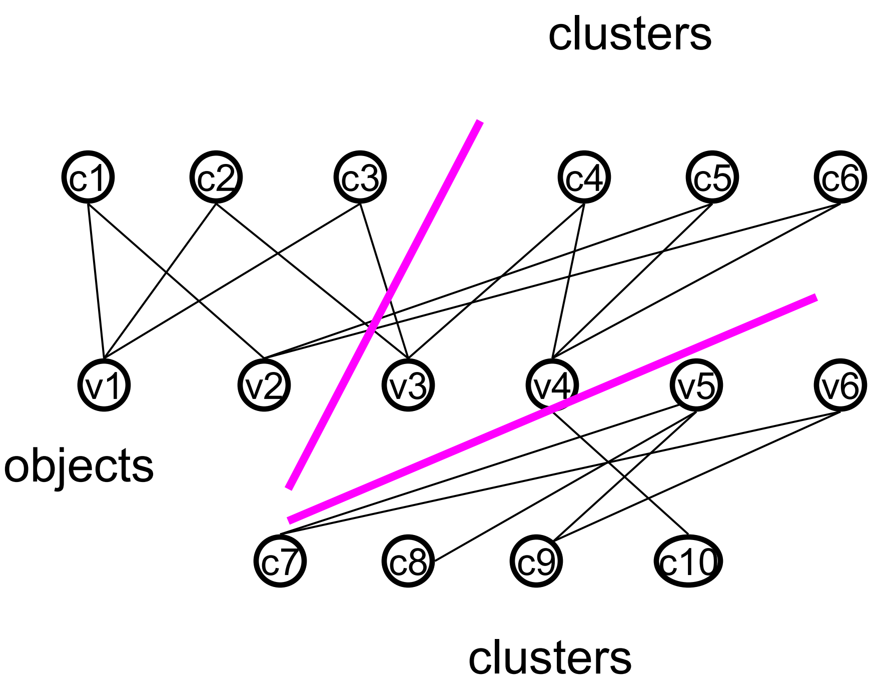
Bipartite Graph Partitioning*

- Hybrid Bipartite Graph Formulation
  - Summarize base model output in a bipartite graph
  - Lossless summarization—base model output can be reconstructed from the bipartite graph
  - Use spectral clustering algorithm to partition the bipartite graph
  - Time complexity $O(nkr)$—due to the special structure of the bipartite graph
  - Each component represents a consensus cluster

*[FeBr04]

Bipartite Graph Partitioning

	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_3$	$\mathcal{C}$
v_1	1	1	1	1
v_2	1	2	2	2
v_3	2	1	1	1
v_4	2	2	2	2
v_5	3	3	3	3
v_6	3	4	3	3



Evaluation criterion:

Normalized Mutual Information (NMI)

Baseline methods:

IBGF: clustering objects

CBGF: clustering clusters

	Random Subsampl. 20	Random Subsampl. 40	Random Subsampl. 60	Random Proj. 20	Random Proj. 40	Random Proj. 60
EOS						
IBGF	0.263	0.262	0.262	0.260	0.263	0.269
CBGF	0.262	0.264	0.263	0.246	0.247	0.247
HBGF	0.340	0.319	0.303	0.357	0.343	0.325
		(0.263)			(0.246)	
GLASS						
IBGF	0.400	0.405	0.388	0.376	0.373	0.368
CBGF	0.393	0.398	0.395	0.379	0.378	0.377
HBGF	0.405	0.398	0.399	0.401	0.386	0.390
		(0.378)			(0.334)	
HRCT						
IBGF	0.310	0.312	0.313	0.283	0.299	0.301
CBGF	0.279	0.277	0.280	0.256	0.267	0.274
HBGF	0.303	0.318	0.321	0.274	0.292	0.301
		(0.292)			(0.196)	
ISOLET6						
IBGF	0.804	0.799	0.812	0.761	0.802	0.811
CBGF	0.832	0.837	0.833	0.750	0.790	0.802
HBGF	0.844	0.823	0.823	0.765	0.801	0.813
		(0.790)			(0.447)	
MODIS						
IBGF	0.478	0.478	0.478	0.485	0.493	0.491
CBGF	0.476	0.478	0.478	0.482	0.490	0.491
HBGF	0.478	0.478	0.478	0.485	0.487	0.494
		(0.473)			(0.389)	

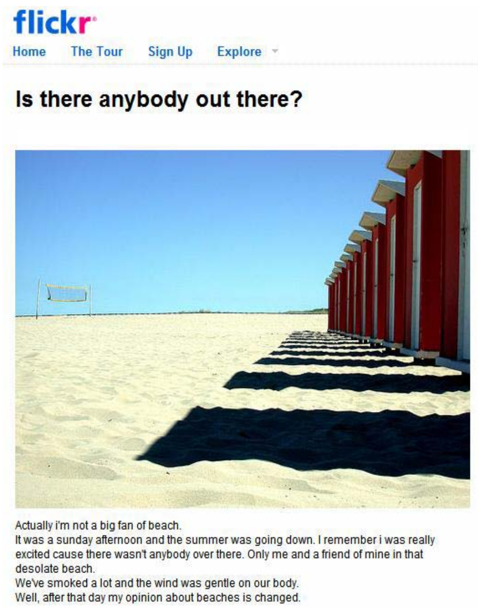
Summary of Unsupervised Ensemble

- Difference from supervised ensemble
  - No theories behind the success of clustering ensemble approaches
  - Moderate diversity is favored in the base models of clustering ensemble
  - There exist label correspondence problems
- Characteristics
  - Experimental results demonstrate that cluster ensembles are better than single models!
  - There is no single, universally successful, cluster ensemble method

Outline

- An overview of ensemble methods
  - Motivations
  - Tutorial overview
- Supervised ensemble
- Unsupervised ensemble
- Semi-supervised ensemble
  - Multi-view learning
  - Consensus maximization among supervised and unsupervised models
- Applications
  - Stream classification, transfer learning, anomaly detection

Multiple Source Classification



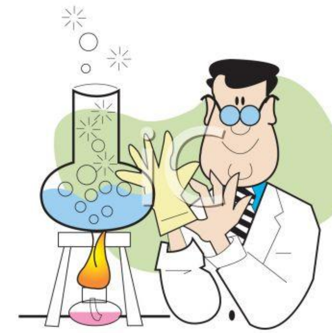


Image Categorization

images, descriptions, notes, comments, albums, tags…….

Like? Dislike?

movie genres, cast, director, plots…….

users viewing history, movie ratings…

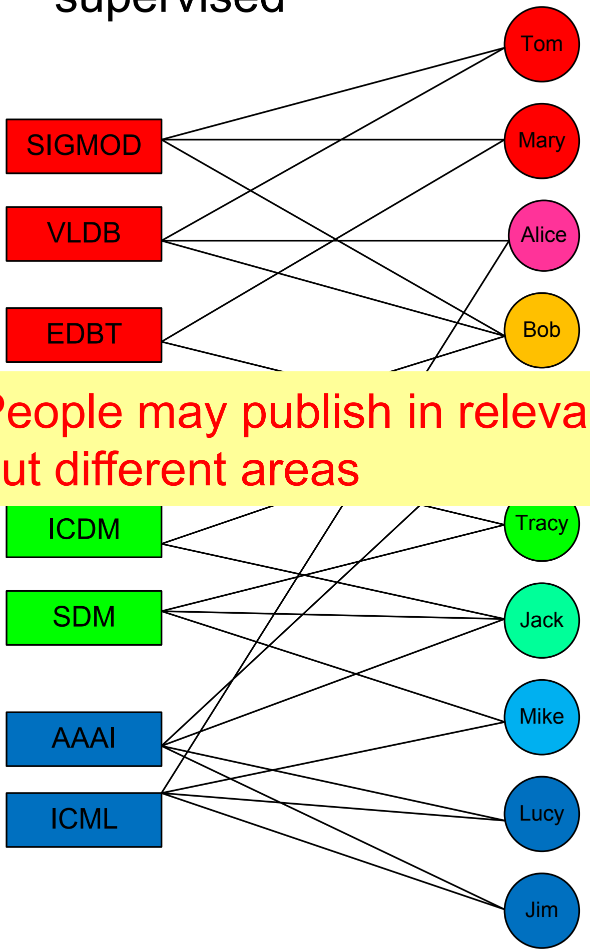
Research Area

publication and co-authorship network, published papers, …….

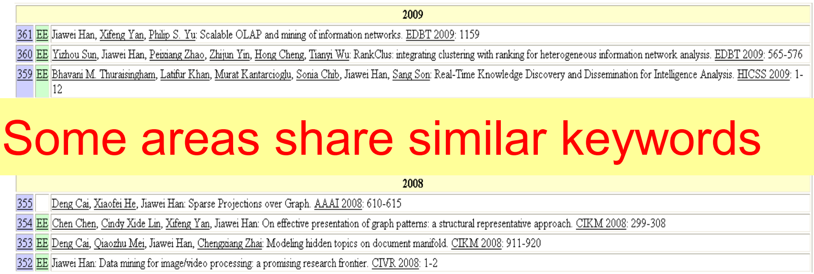
Model Combination helps!

Supervised or unsupervised

supervised



People may publish in relevant but different areas



		2009
361	EE	Jiawei Han, Xifeng Yan, Philip S. Yu: Scalable OLAP and mining of information networks. EDBT 2009: 1159
360	EE	Yizhou Sun, Jiawei Han, Peixiang Zhao, Zhijun Yin, Hong Cheng, Tianyi Wu: RankClus: integrating clustering with ranking for heterogeneous information network analysis. EDBT 2009: 565-576
359	EE	Bhavani M. Thuraisingham, Latifur Khan, Murat Kantarcioglu, Sonia Chib, Jiawei Han, Sang Son: Real-Time Knowledge Discovery and Dissemination for Intelligence Analysis. HICSS 2009: 1-12
		2008
355		Deng Cai, Xiaofei He, Jiawei Han: Sparse Projections over Graph. AAAI 2008: 610-615
354	EE	Chen Chen, Cindy Xide Lin, Xifeng Yan, Jiawei Han: On effective presentation of graph patterns: a structural representative approach. CIKM 2008: 299-308
353	EE	Deng Cai, Qiaozhu Mei, Jiawei Han, Chengxiang Zhai: Modeling hidden topics on document manifold. CIKM 2008: 911-920
352	EE	Jiawei Han: Data mining for image/video processing: a promising research frontier. CIVR 2008: 1-2

Some areas share similar keywords

There may be cross-discipline co-operations

unsupervised

Multi-view Learning (1)

- 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

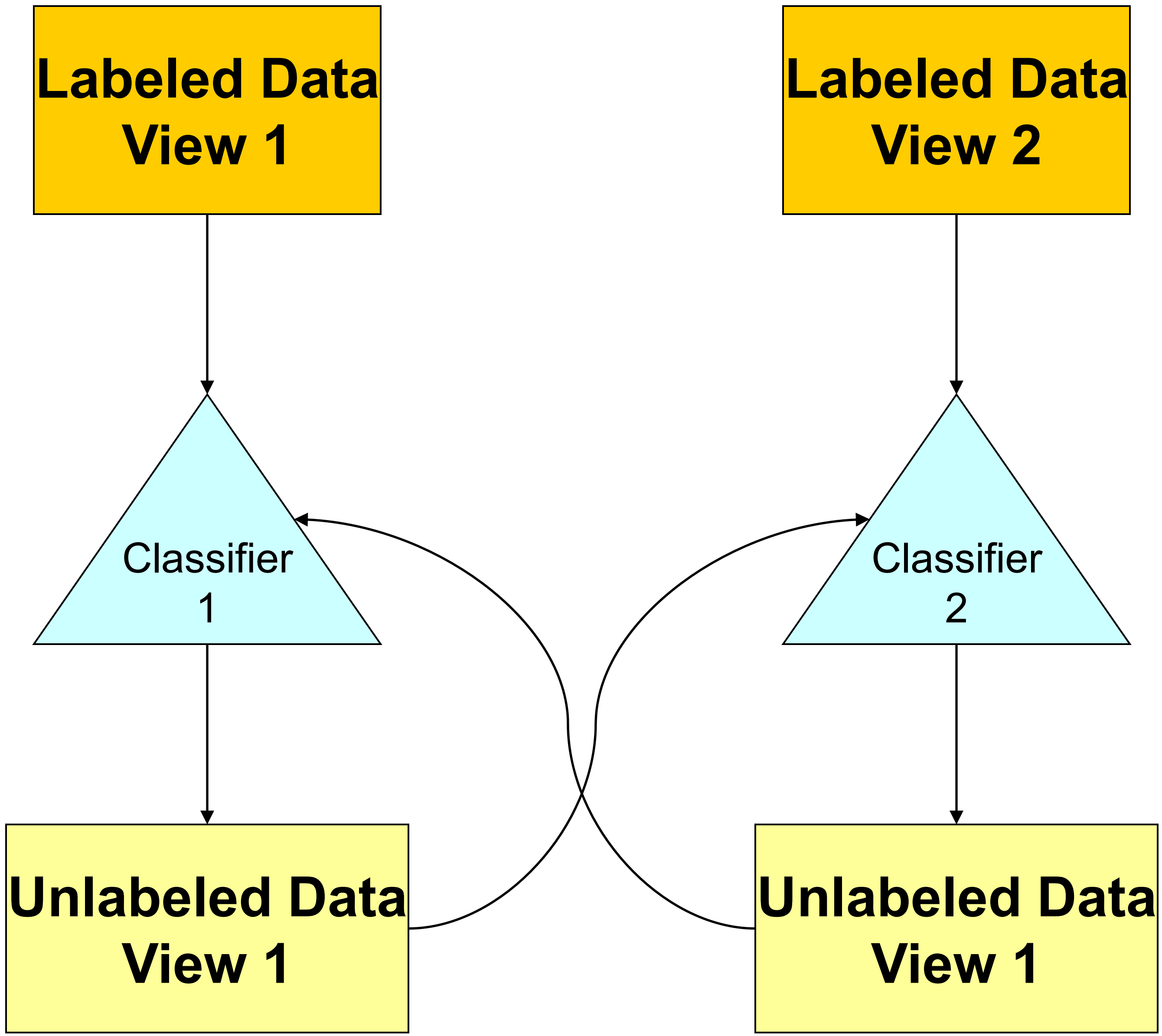
Multi-view Learning (2)

- 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.
- Problems
  - Require raw data to learn the models
  - Supervised and unsupervised information sources are symmetric
- Algorithms
  - Co-training

Co-Training*

- 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
- 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

*[BlMi98]


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

Given:

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

Create a pool U' of examples 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

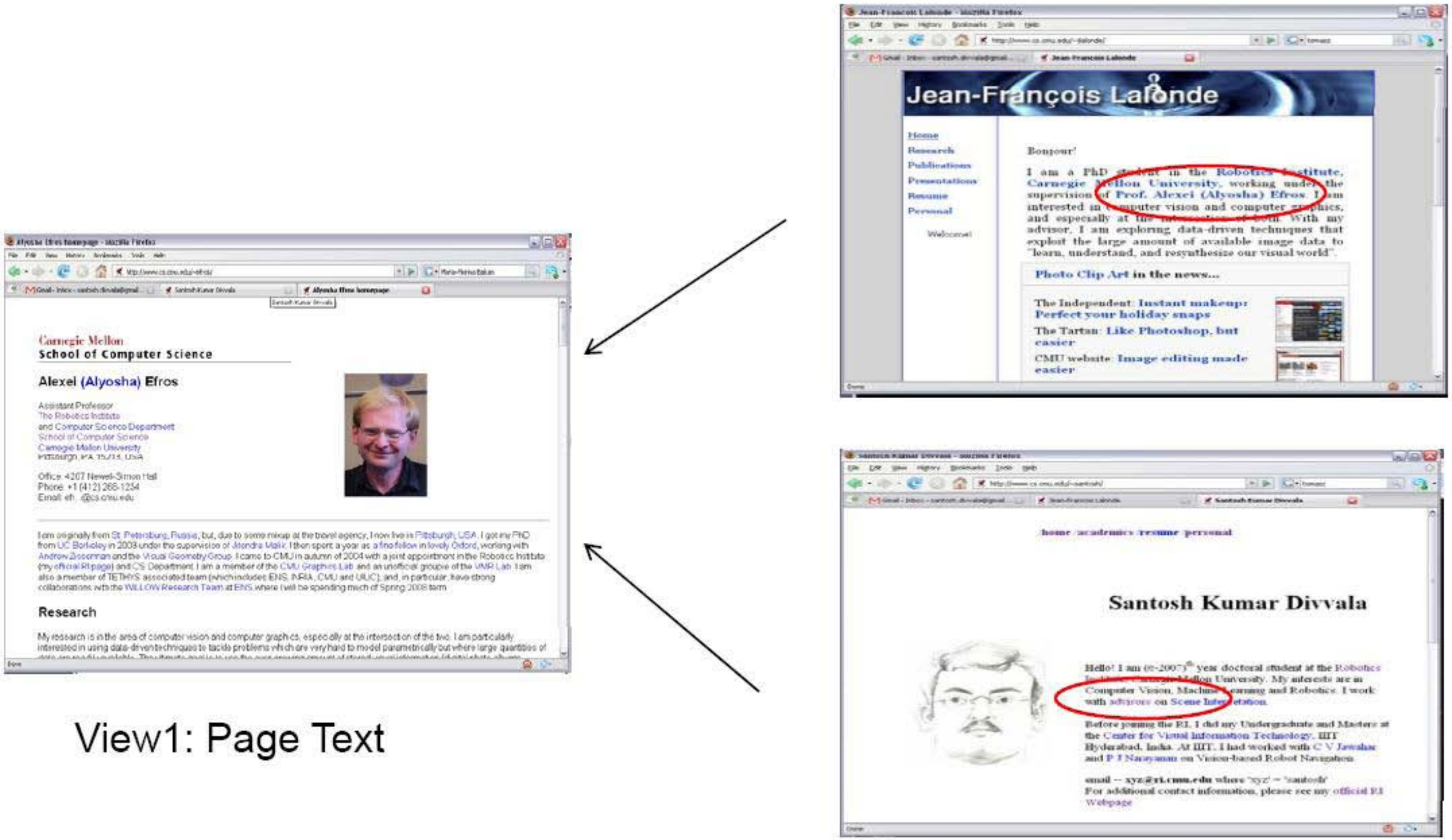
Allow h_1 to label p positive and n negative examples from U'

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'

Applications: Faculty Webpages Classification



View1: Page Text

View2: Hyperlink Text

from S. K. Divvala. Co-Training & Its Applications in Vision.

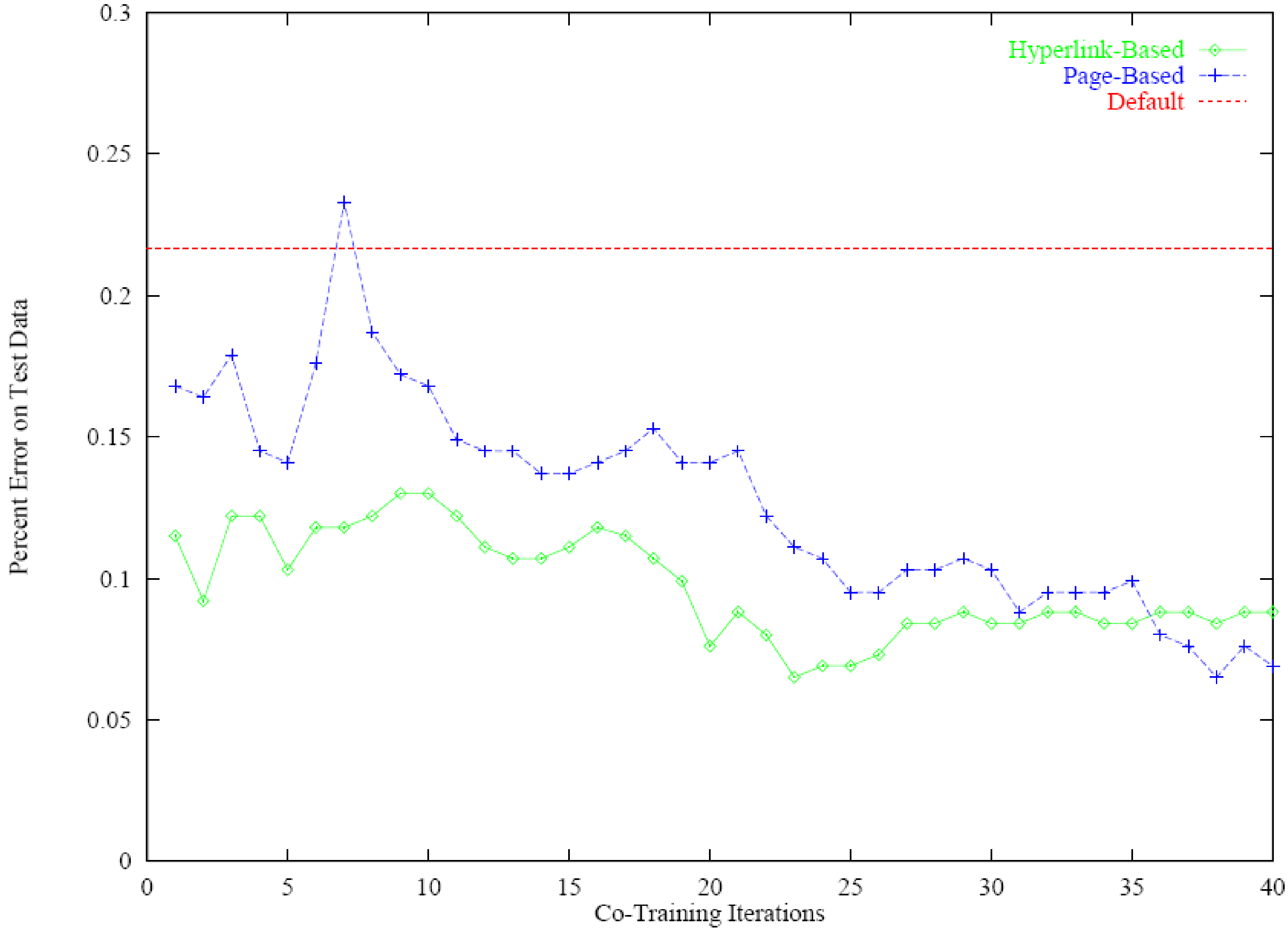


Figure 2: Error versus number of iterations for one run of co-training experiment.

Outline

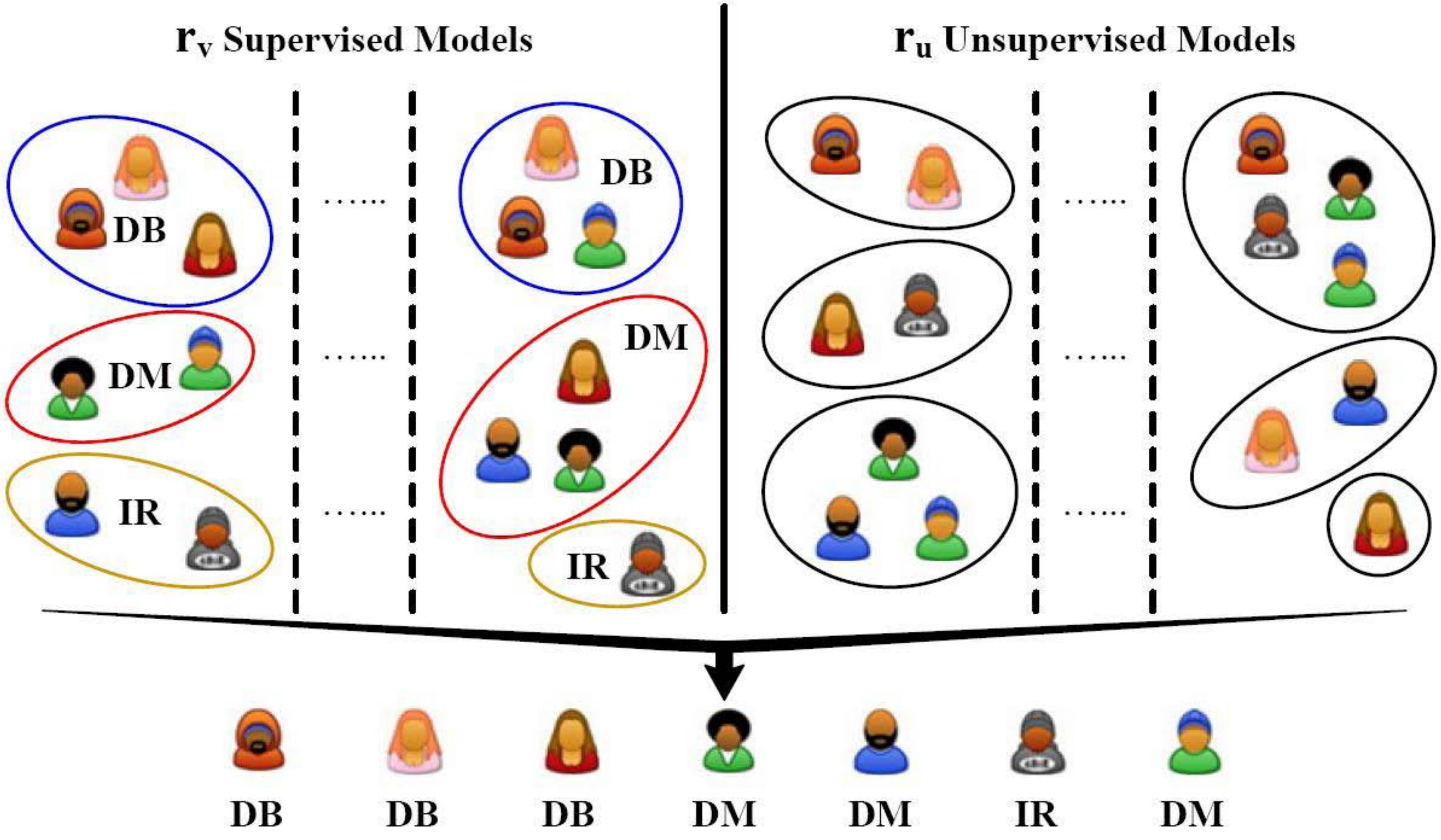
- An overview of ensemble methods
  - Motivations
  - Tutorial overview
- Supervised ensemble
- Unsupervised ensemble
- Semi-supervised ensemble
  - Multi-view learning
  - Consensus maximization among supervised and unsupervised models
- Applications
  - Stream classification, transfer learning, anomaly detection

Consensus Maximization*

- Goal
  - Combine output of multiple supervised and unsupervised models on a set of objects
  - The predicted labels should agree with the base models as much as possible
- Motivations
  - Unsupervised models provide useful constraints for classification tasks
  - Model diversity improves prediction accuracy and robustness
  - Model combination at output level is needed due to privacy-preserving or incompatible formats

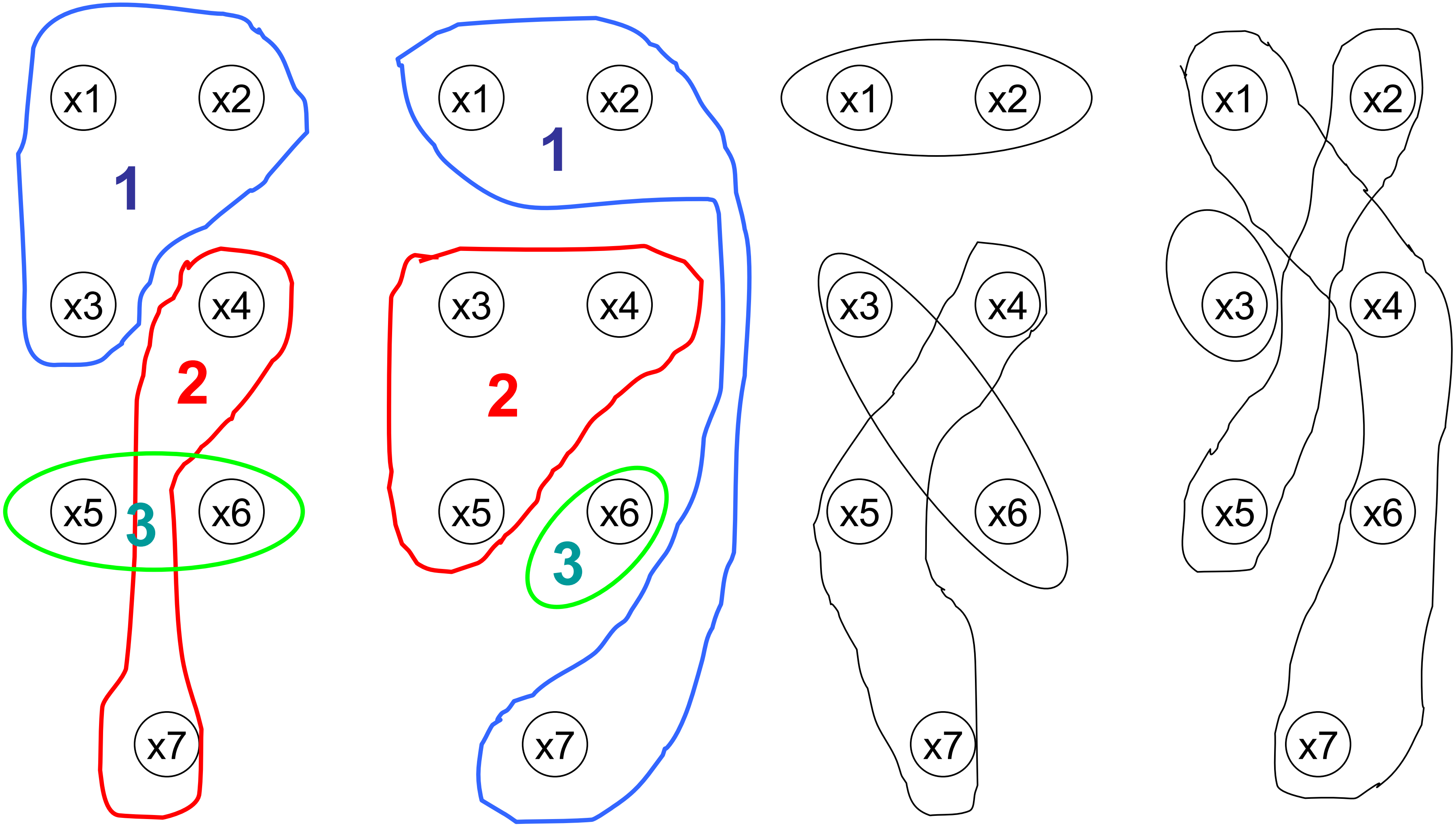
*[GLF+09]

Problem

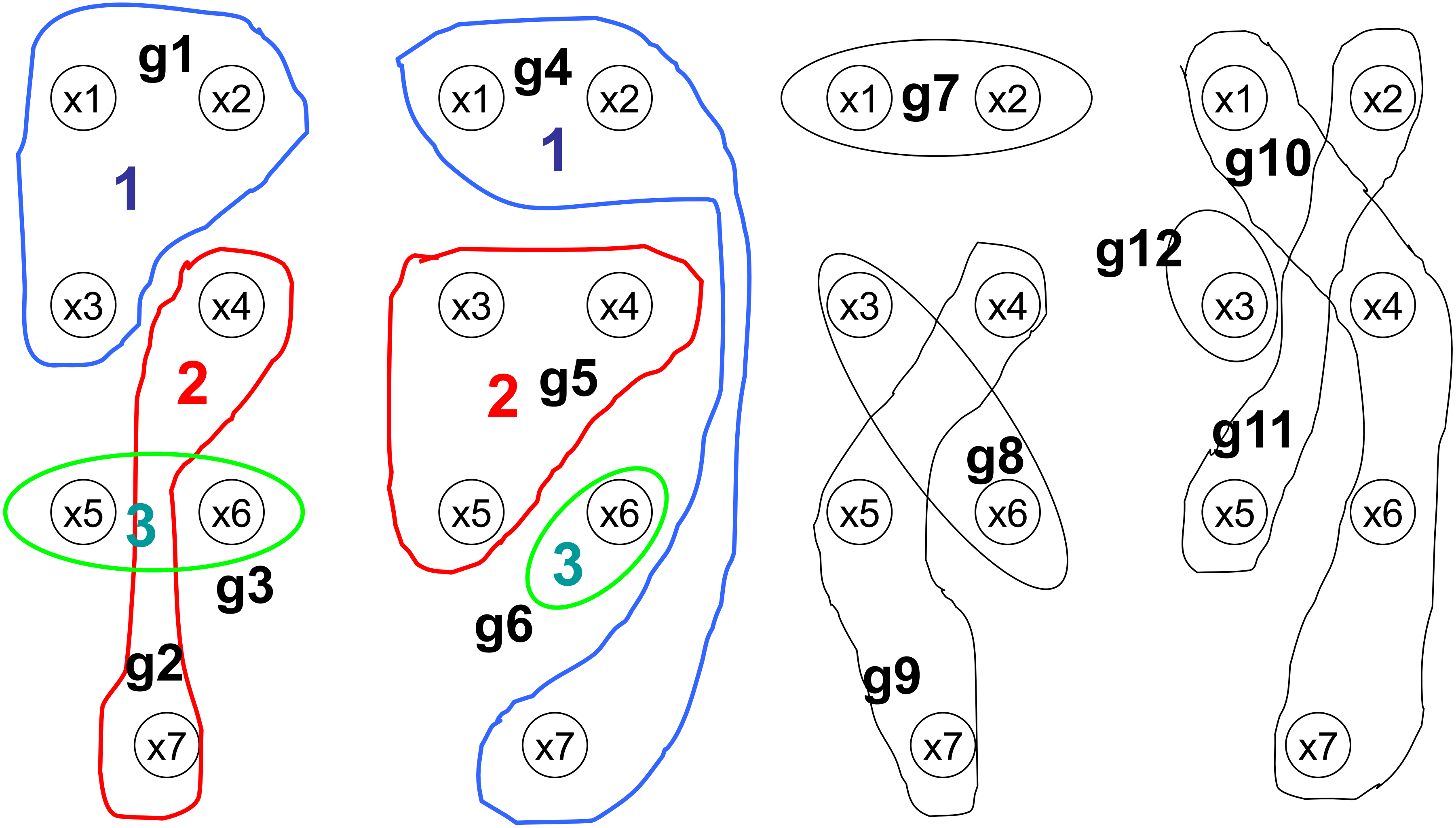


r_v Supervised Models
r_u Unsupervised Models
DB
DM
IR
……
DB
DM
IR
DB
DB
DB
DM
DM
IR
DM

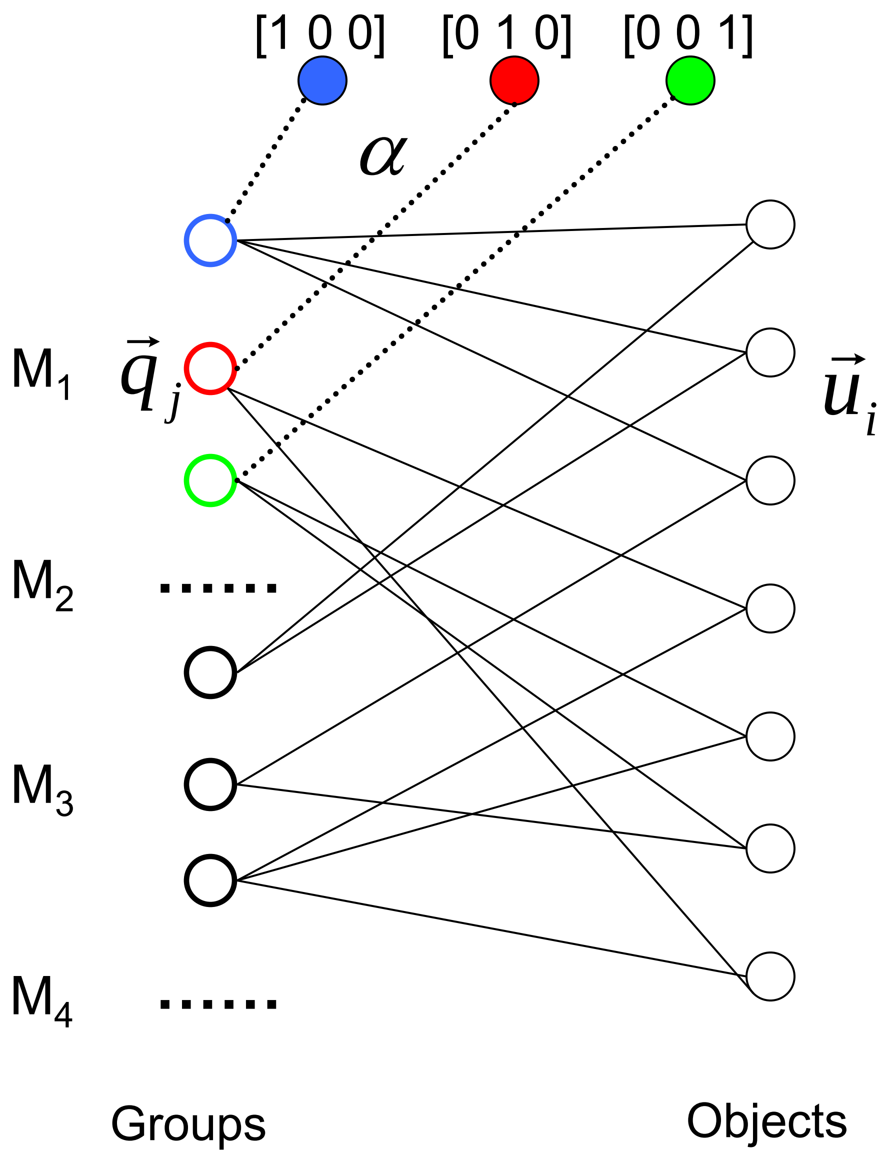
A Toy Example



Groups-Objects



Bipartite Graph



object i $\vec{u}_i = [u_{i1}, \ldots, u_{ic}]$

group j $\vec{q}_j = [q_{j1}, \ldots, q_{jc}]$

conditional prob vector

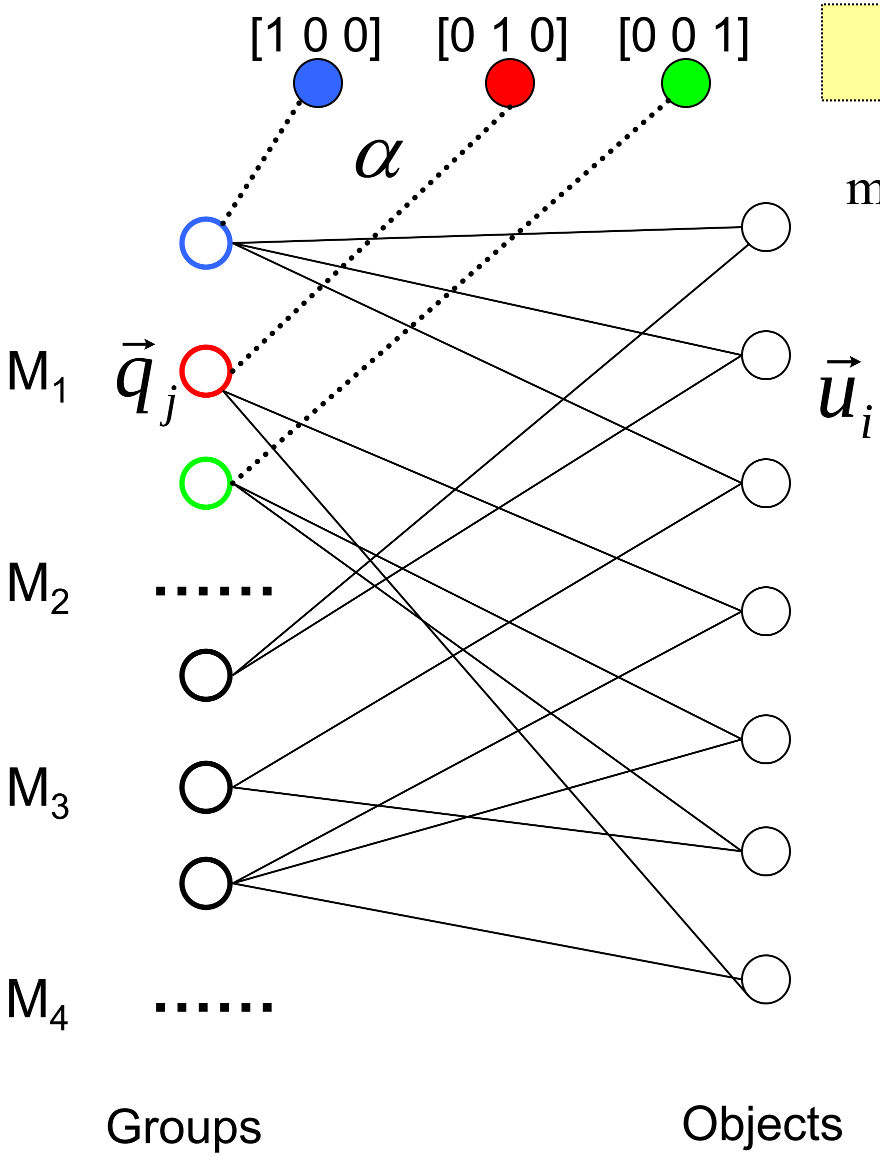
adjacency

$$a_{ij} = \begin{cases} 1 & u_i \leftrightarrow q_j \\ 0 & otherwise \end{cases}$$

initial probability

$$\vec{y}_j = \begin{cases} [1\,0 \ldots 0] & g_j \in 1 \\ \ldots\ldots & \ldots\ldots \\ [0 \ldots 0\,1] & g_j \in c \end{cases}$$

Objective



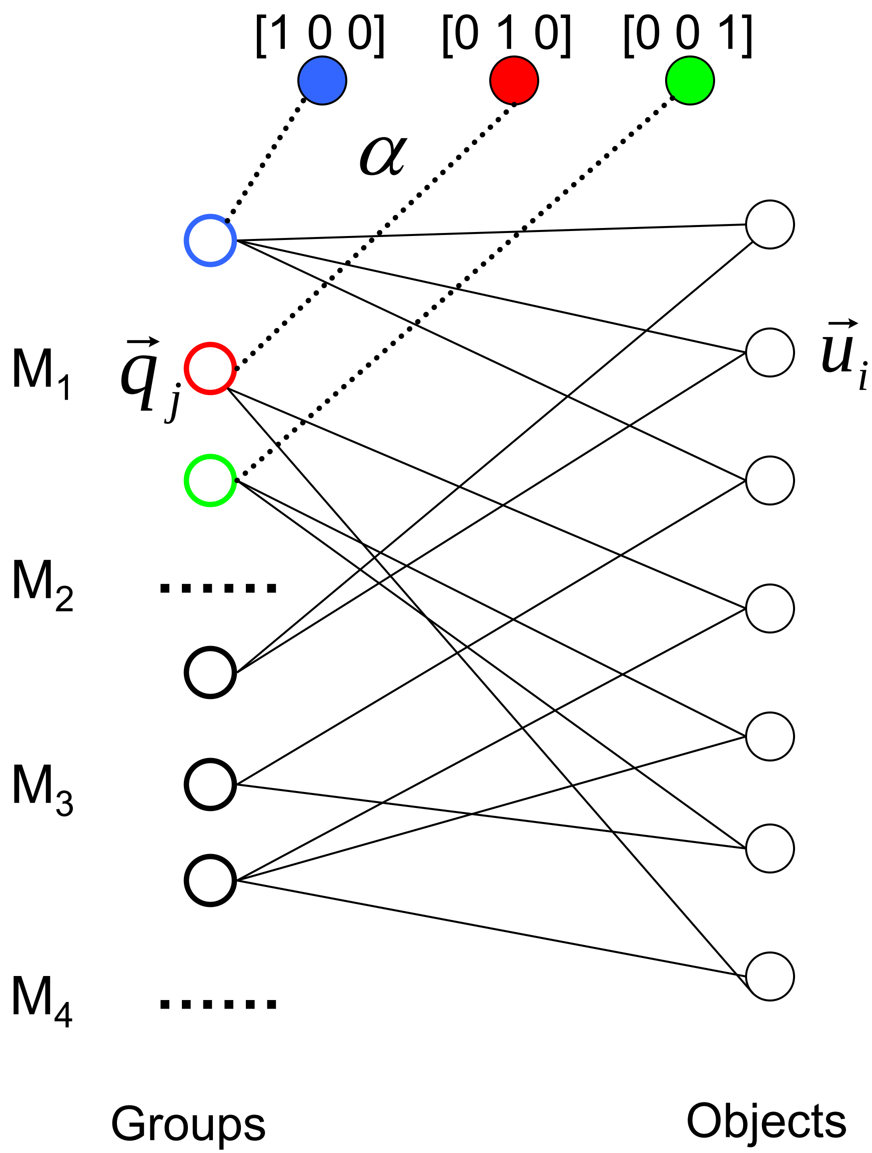
minimize disagreement

$$\min_{Q,U}\left(\sum_{i=1}^{n}\sum_{j=1}^{v} a_{ij}\,\|\vec{u}_i-\vec{q}_j\|^2+\alpha\sum_{j=1}^{s}\|\vec{q}_j-\vec{y}_j\|^2\right)$$

Similar conditional probability if the object is connected to the group

Do not deviate much from the initial probability

Methodology



Iterate until convergence

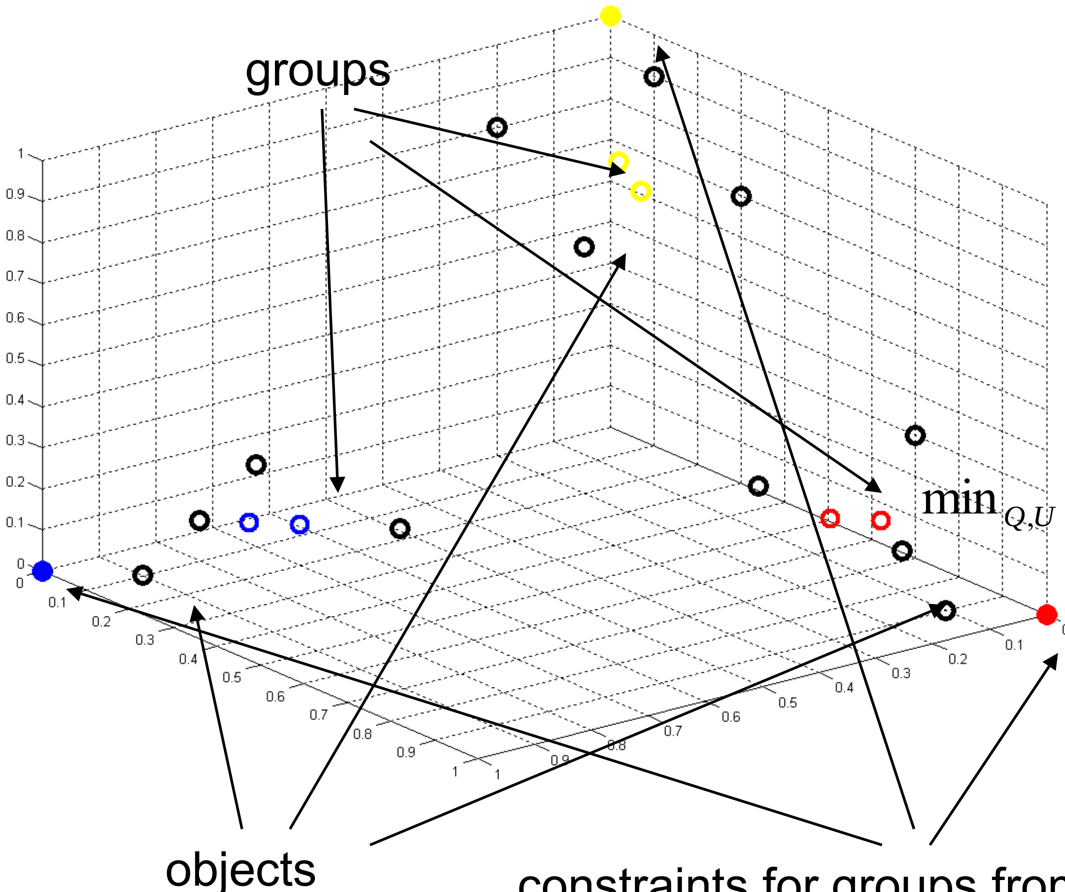
Update probability of a group

$$\vec{q}_j = \frac{\sum_{i=1}^{n} a_{ij}\vec{u}_i + \alpha\vec{y}_j}{\sum_{i=1}^{n} a_{ij} + \alpha} \qquad \vec{q}_j = \frac{\sum_{i=1}^{n} a_{ij}\vec{u}_i}{\sum_{i=1}^{n} a_{ij}}$$

Update probability of an object

$$\vec{u}_i = \frac{\sum_{j=1}^{v} a_{ij}\vec{q}_j}{\sum_{j=1}^{v} a_{ij}}$$

Constrained Embedding



$$\min_{Q,U}\sum_{j=1}^{v}\sum_{z=1}^{c}\left|q_{jz}-\frac{\sum_{i=1}^{n}a_{ij}u_{iz}}{\sum_{i=1}^{n}a_{ij}}\right|$$

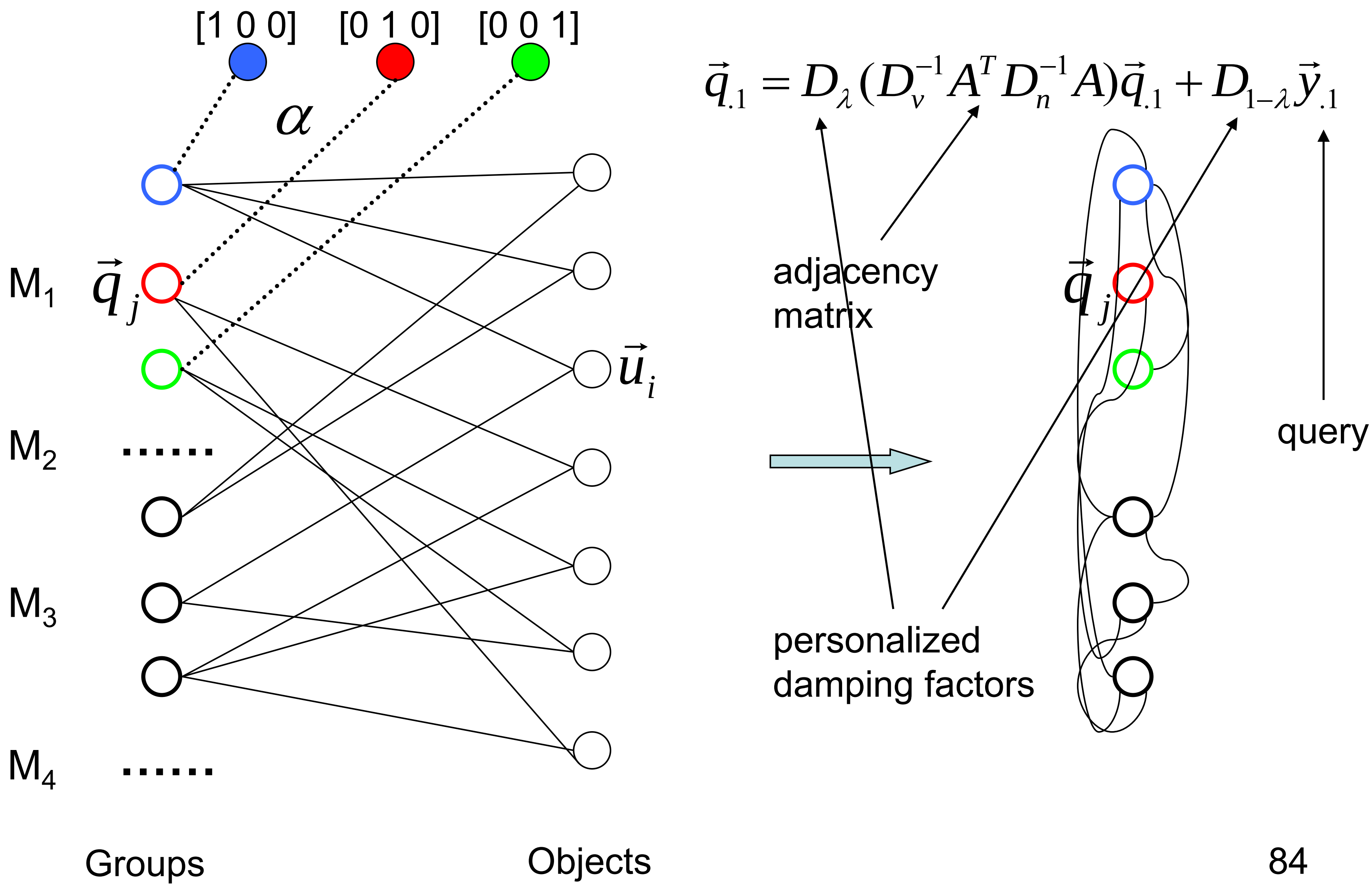
$$q_{jz}=1 \; if \; g_j\text{'s} \; label \; is \; z$$

$$\Updownarrow$$

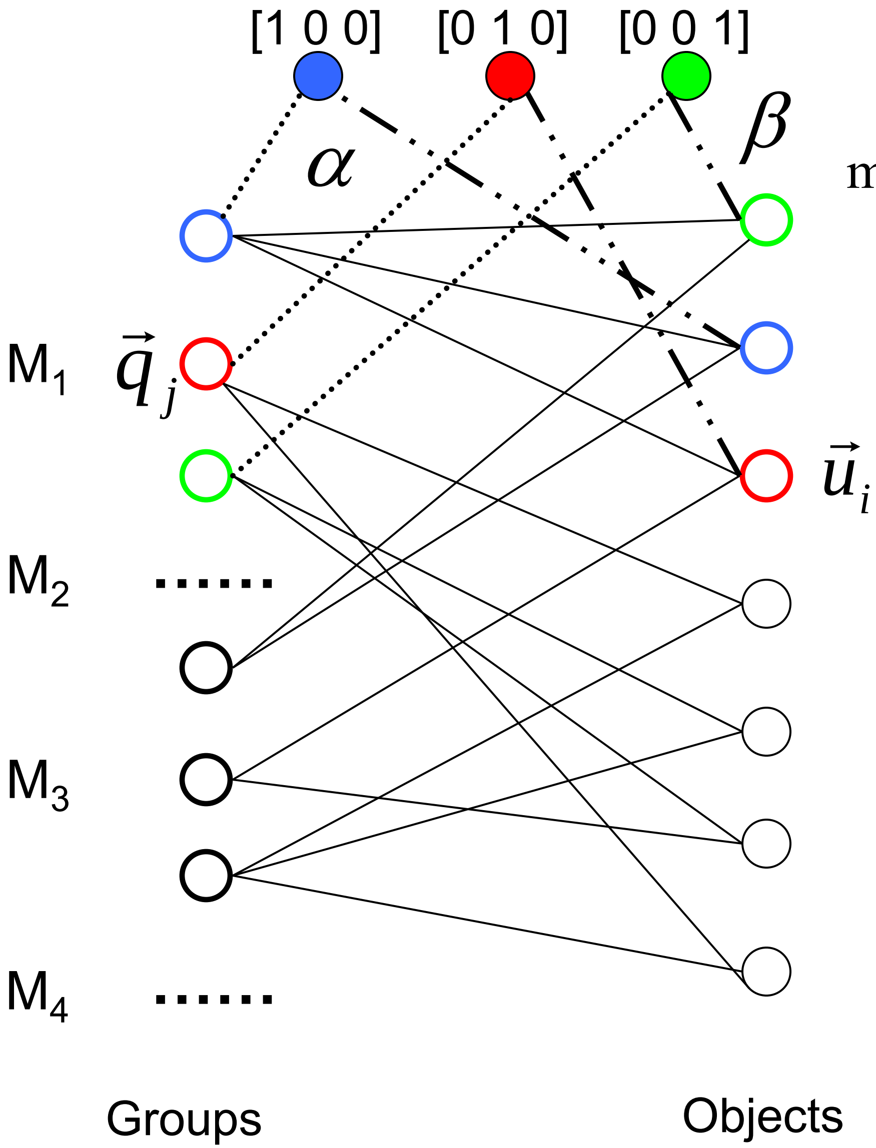
$$\min_{Q,U}\left(\sum_{i=1}^{n}\sum_{j=1}^{v}a_{ij}\|\vec{u}_i-\vec{q}_j\|^2+\alpha\sum_{j=1}^{s}\|\vec{q}_j-\vec{y}_j\|^2\right)$$

Ranking on Consensus Structure

$$\vec{q}_{.1} = D_{\lambda}(D_v^{-1}A^T D_n^{-1}A)\vec{q}_{.1} + D_{1-\lambda}\vec{y}_{.1}$$



Incorporating Labeled Information



Objective

$$\min_{Q,U}\left(\sum_{i=1}^{n}\sum_{j=1}^{v} a_{ij}\|\vec{u}_i-\vec{q}_j\|^2+\alpha\sum_{j=1}^{s}\|\vec{q}_j-\vec{y}_j\|^2\right)+\beta\sum_{i=1}^{l}\|\vec{u}_i-\vec{f}_i\|^2$$

Update probability of a group

$$\vec{q}_j=\frac{\sum_{i=1}^{n}a_{ij}\vec{u}_i+\alpha\vec{y}_j}{\sum_{i=1}^{n}a_{ij}+\alpha} \qquad \vec{q}_j=\frac{\sum_{i=1}^{n}a_{ij}\vec{u}_i}{\sum_{i=1}^{n}a_{ij}}$$

Update probability of an object

$$\vec{u}_i=\frac{\sum_{j=1}^{v}a_{ij}\vec{q}_j}{\sum_{j=1}^{v}a_{ij}} \qquad \vec{u}_i=\frac{\sum_{j=1}^{v}a_{ij}\vec{q}_j+\beta\vec{f}_i}{\sum_{j=1}^{v}a_{ij}+\beta}$$

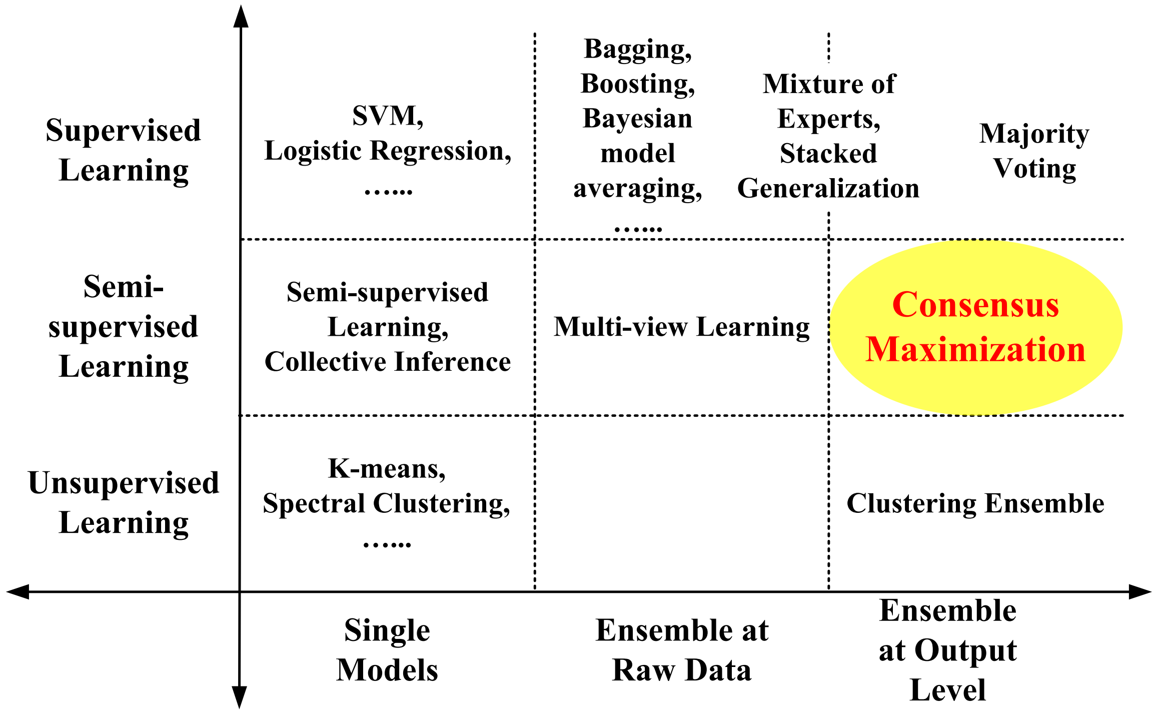
Experiments-Data Sets

- 20 Newsgroup
  - newsgroup messages categorization
  - only text information available
- Cora
  - research paper area categorization
  - paper abstracts and citation information available
- DBLP
  - researchers area prediction
  - publication and co-authorship network, and publication content
  - conferences' areas are known

Experiments-Baseline Methods (1)

- Single models
  - 20 Newsgroup:
    - logistic regression, SVM, K-means, min-cut
  - Cora
    - abstracts, citations (with or without a labeled set)
  - DBLP
    - publication titles, links (with or without labels from conferences)
- Proposed method
  - BGCM
  - BGCM-L: semi-supervised version combining four models
  - 2-L: two models
  - 3-L: three models

Experiments-Baseline Methods (2)



- Ensemble approaches
  - clustering ensemble on all of the four models-MCLA, HBGF

Accuracy (1)

Methods	20 Newsgroups					
	1	2	3	4	5	6
M_1	0.7967	0.8855	0.8557	0.8826	0.8765	0.8880
M_2	0.7721	0.8611	0.8134	0.8676	0.8358	0.8563
M_3	0.8056	0.8796	0.8658	0.8983	0.8716	0.9020
M_4	0.7770	0.8571	0.8149	0.8467	0.8543	0.8578
MCLA	0.7592	0.8173	0.8253	0.8686	0.8295	0.8546
HBGF	0.8199	**0.9244**	0.8811	0.9152	0.8991	0.9125
BGCM	0.8128	0.9101	0.8608	0.9125	0.8864	0.9088
2-L	0.7981	0.9040	0.8511	0.8728	0.8830	0.8977
3-L	0.8188	0.9206	0.8820	0.9158	0.8989	0.9121
BGCM-L	**0.8316**	0.9197	**0.8859**	**0.9240**	**0.9016**	**0.9177**
STD	0.0040	0.0038	0.0037	0.0040	0.0027	0.0030

Accuracy (2)

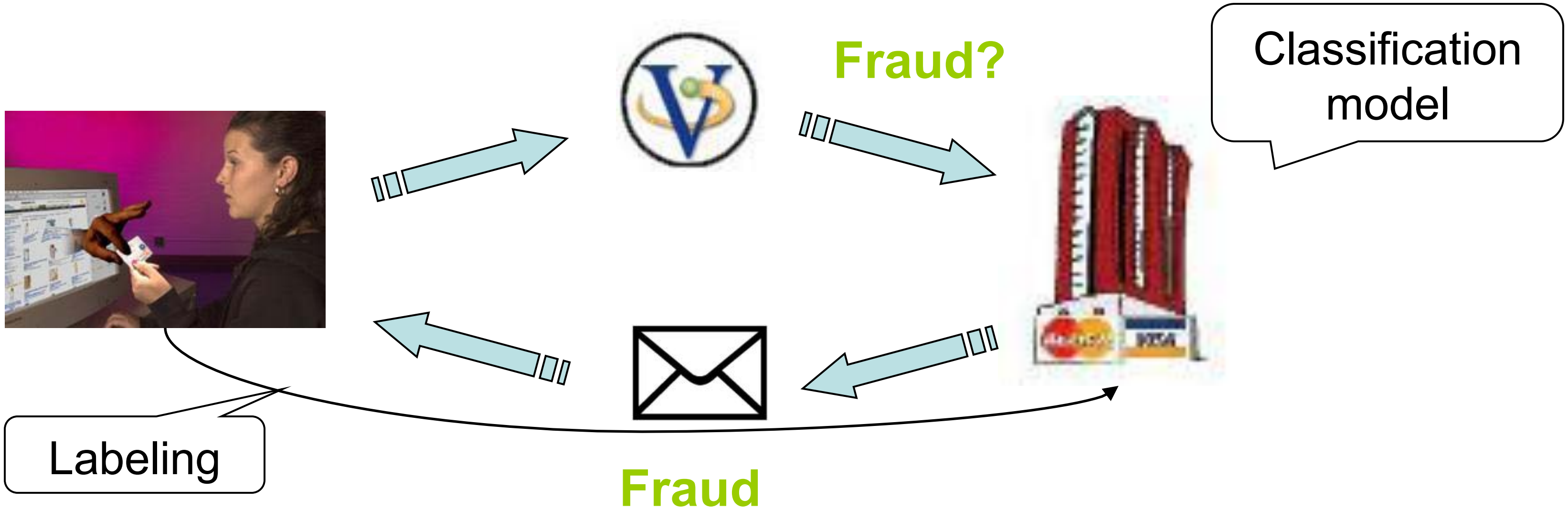
Methods	Cora				DBLP
	1	2	3	4	1
M_1	0.7745	0.8858	0.8671	0.8841	0.9337
M_2	0.7797	0.8594	0.8508	0.8879	0.8766
M_3	0.7779	0.8833	0.8646	0.8813	0.9382
M_4	0.7476	0.8594	0.7810	0.9016	0.7949
MCLA	0.8703	0.8388	0.8892	0.8716	0.8953
HBGF	0.7834	0.9111	0.8481	0.8943	0.9357
BGCM	0.8687	0.9155	0.8965	0.9090	0.9417
2-L	0.8066	0.8798	0.8932	0.8951	0.9054
3-L	0.8557	0.9086	0.9202	0.9141	0.9332
BGCM-L	**0.8891**	**0.9181**	**0.9246**	**0.9206**	**0.9480**
STD	0.0096	0.0027	0.0052	0.0044	0.0020

Outline

- An overview of ensemble methods
  - Motivations
  - Tutorial overview
- Supervised ensemble
- Unsupervised ensemble
- Semi-supervised ensemble
  - Multi-view learning
  - Consensus maximization among supervised and unsupervised models
- Applications
  - Stream classification, transfer learning, anomaly detection

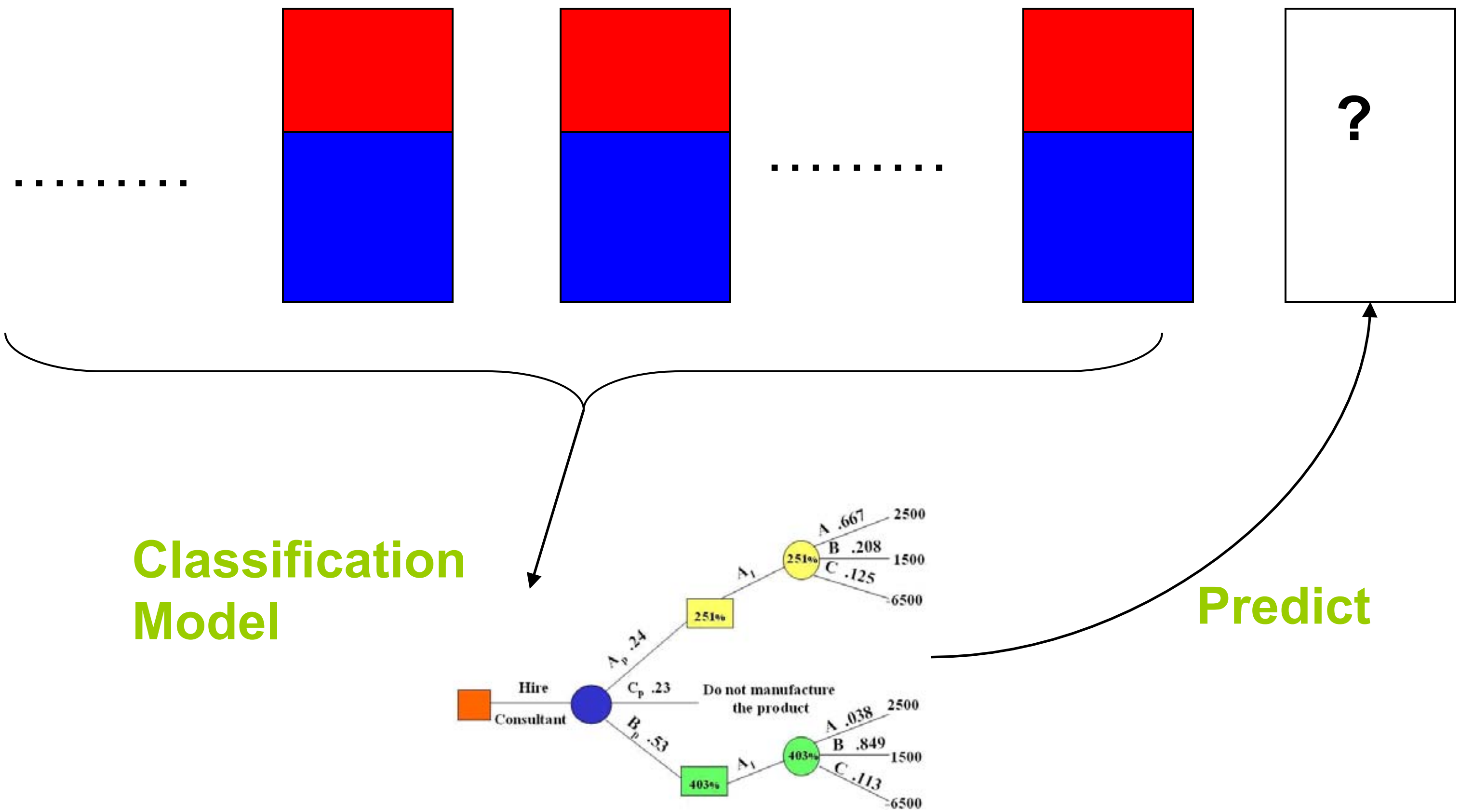
Stream Classification*

- Process
  - Construct a classification model based on past records
  - Use the model to predict labels for new data
  - Help decision making



*[GFH07]

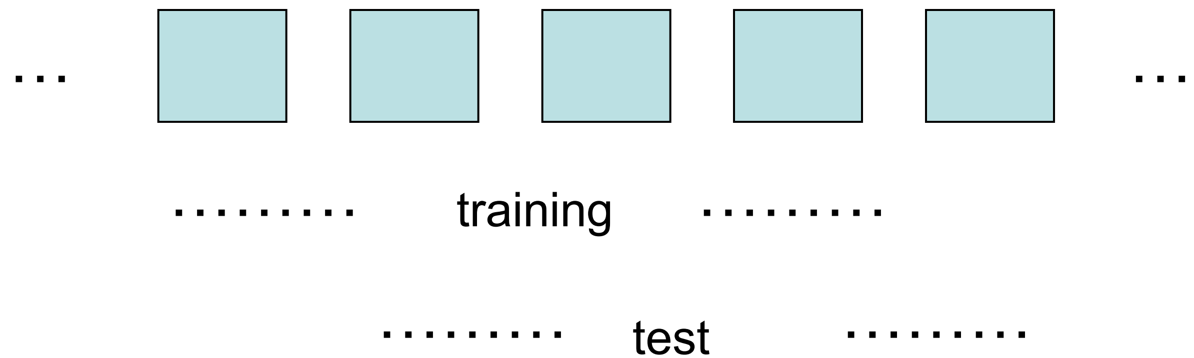
Framework



?
Classification
Model
Predict
Hire
Consultant
Do not manufacture
the product
2500
1500
6500
2500
1500
6500

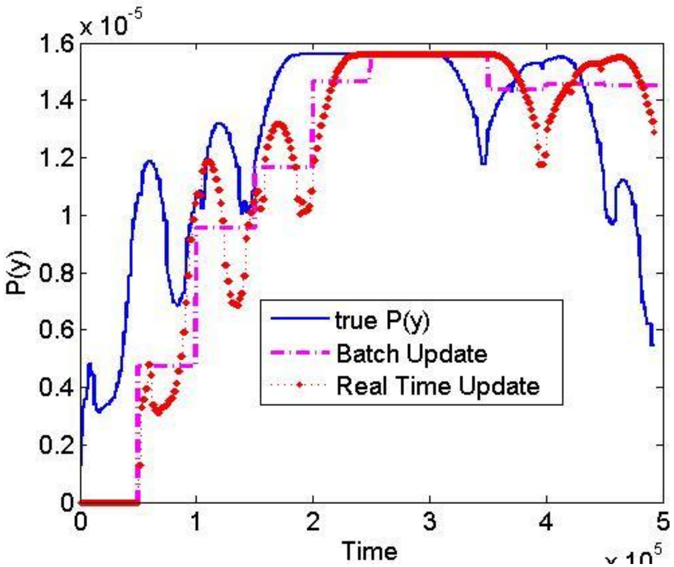
Existing Stream Mining Methods

- Shared distribution assumption
  - Training and test data are from the same distribution P(x,y) x-feature vector, y-class label
  - Validity of existing work relies on the shared distribution assumption
- Difference from traditional learning
  - Both distributions evolve



Evolving Distributions (1)

- An example of stream data
  - KDDCUP'99 Intrusion Detection Data
  - P(y) evolves

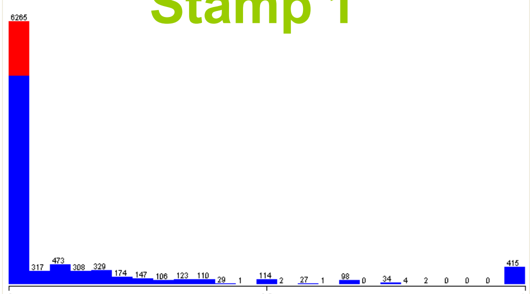


- Shift or delay inevitable
  - The future data could be different from current data
  - Matching the current distribution to fit the future one is a wrong way
  - The shared distribution assumption is inappropriate

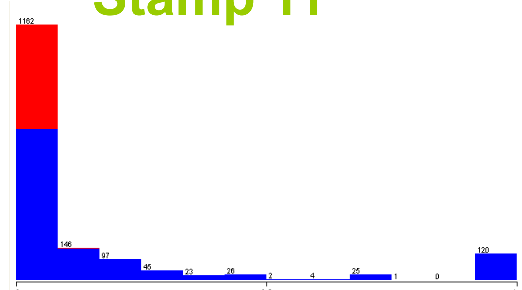
Evolving Distributions (2)

- Changes in P(y)
  - $P(y) \propto P(x,y) = P(y|x)P(x)$
  - The change in P(y) is attributed to changes in P(y|x) and P(x)

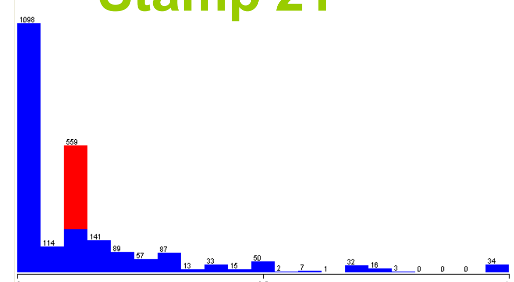
Time Stamp 1



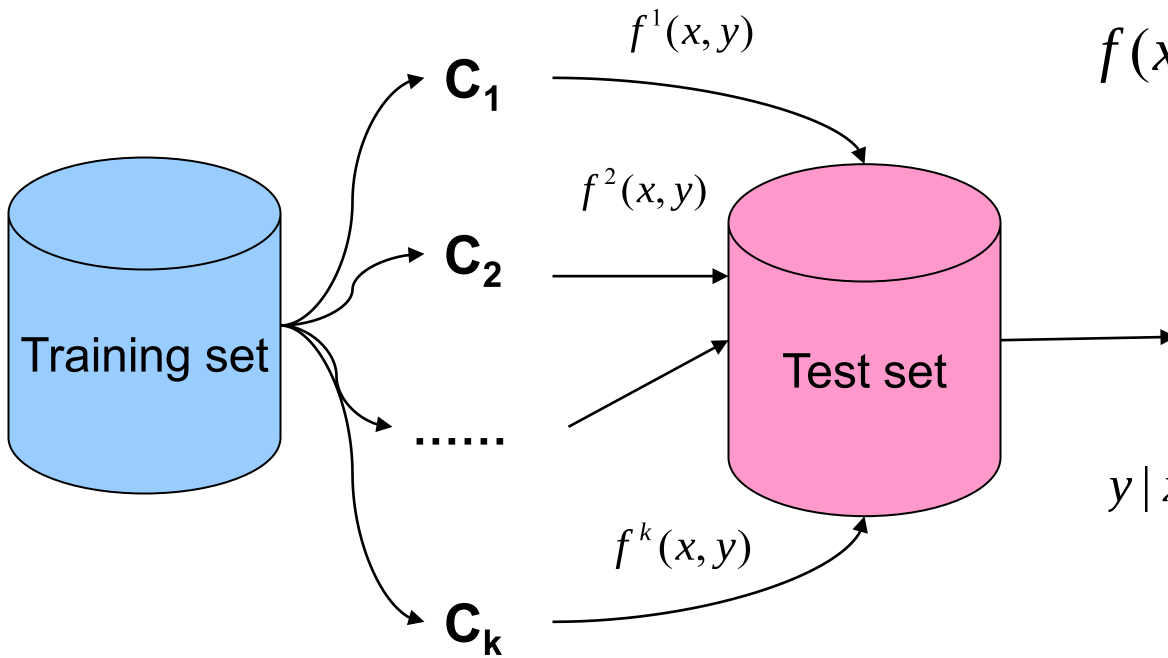
Time Stamp 11



Time Stamp 21



Ensemble Method



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

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

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

Simple Voting(SV)

$$f^i(x,y) = \begin{cases} 1 & \text{model } i \text{ predicts } y \\ 0 & \text{otherwise} \end{cases}$$

Averaging Probability(AP)

$$f^i(x,y) = \text{probability of predicting } y \text{ for model } i$$

Why it works?

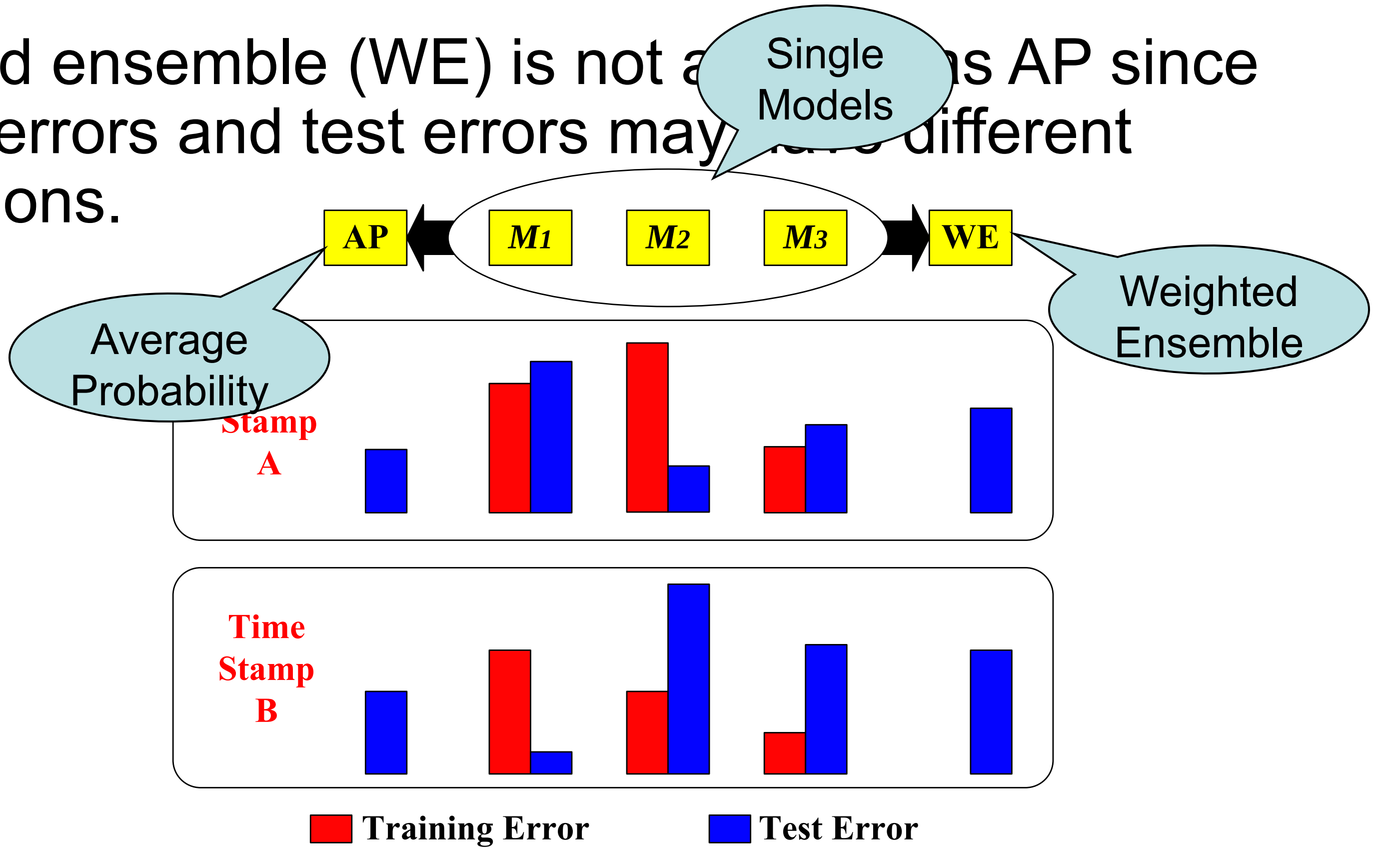
- Ensemble
  - Reduce variance caused by single models
  - Is more robust than single models when the distribution is evolving
- Simple averaging
  - Simple averaging: uniform weights w_i=1/k

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

  - Weighted ensemble: non-uniform weights
    - w_i is inversely proportional to the training errors
  - w_i should reflect P(M), the probability of model M after observing the data
  - P(M) is changing and we could never estimate the true P(M) and when and how it changes
  - Uniform weights could minimize the expected distance between P(M) and weight vector

An illustration

- Single models (M1, M2, M3) have huge variance.
- Simple averaging ensemble (AP) is more stable and accurate.
- Weighted ensemble (WE) is not a[...]s AP since training errors and test errors may [...] different distributions.



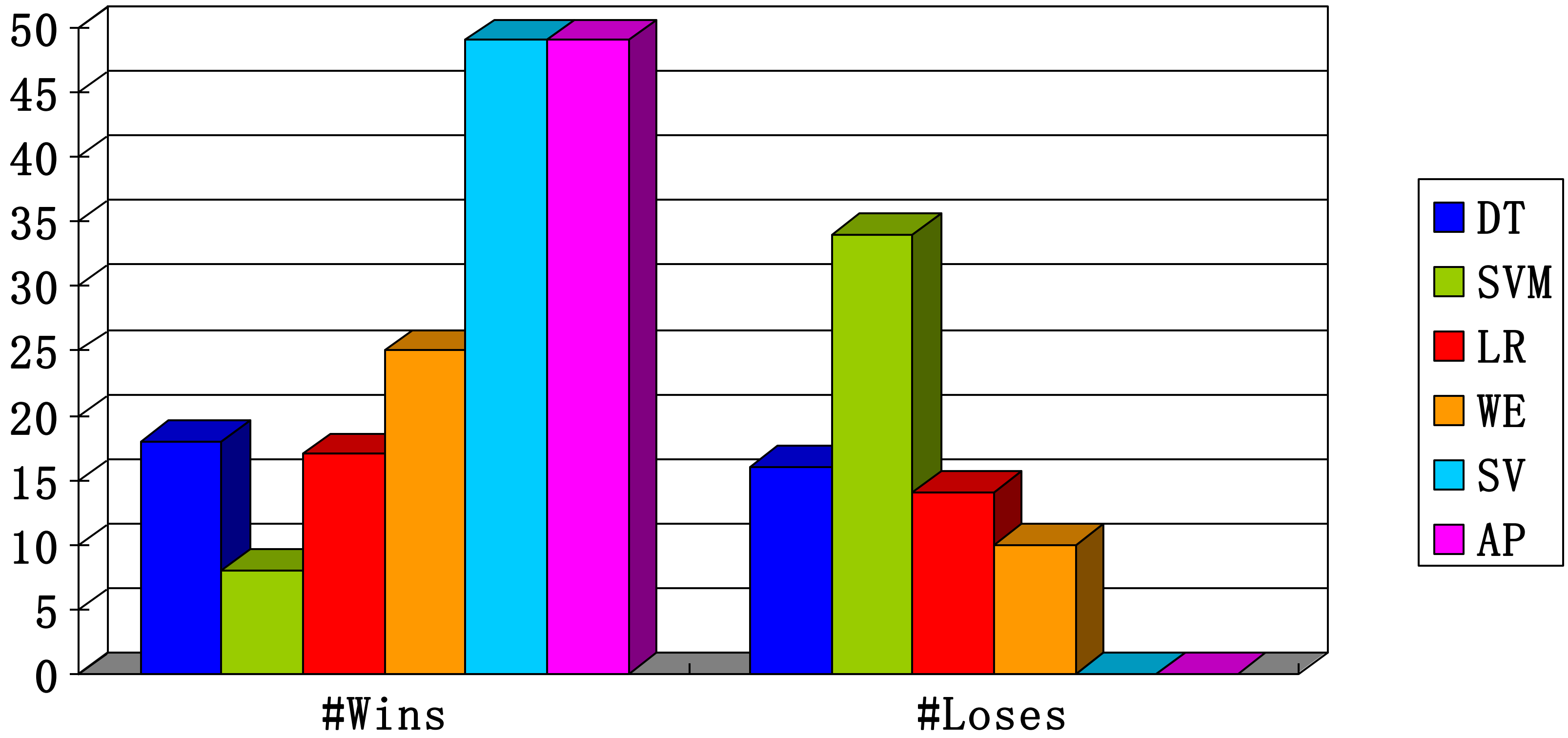
Experiments

- Set up
  - Data streams with chunks T_1, T_2, …, T_N
  - Use T_i as the training set to classify T_{i+1}
- Measures
  - Mean Squared Error, Accuracy
  - Number of Wins, Number of Loses
  - Normalized Accuracy, MSE

$$h(A,T) = h(A,T) / \max_A (h(A,T))$$

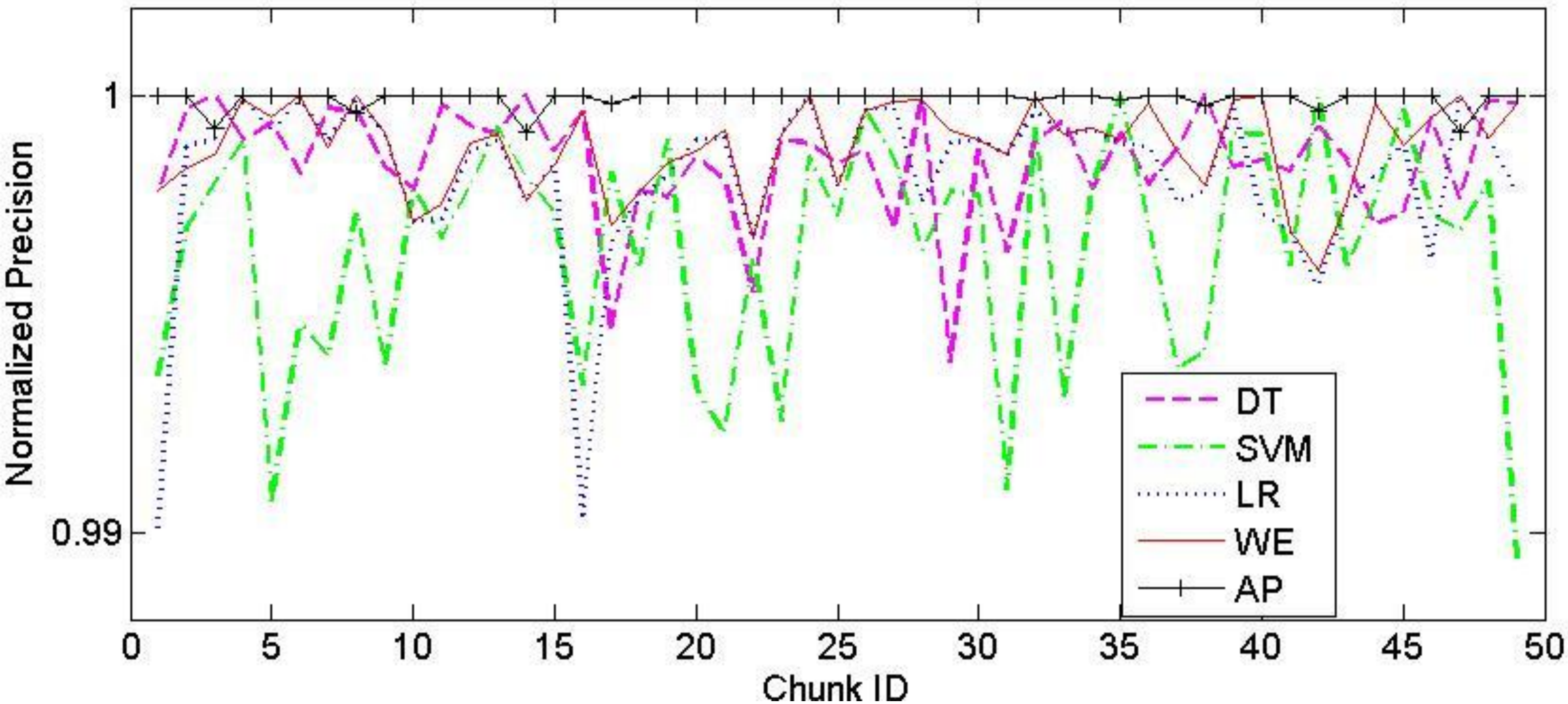
- Methods
  - Single models: Decision tree (DT), SVM, Logistic Regression (LR)
  - Weighted ensemble: weights reflect the accuracy on training set (WE)
  - **Simple ensemble: voting (SV) or probability averaging (AP)**

Experimental Results (2)



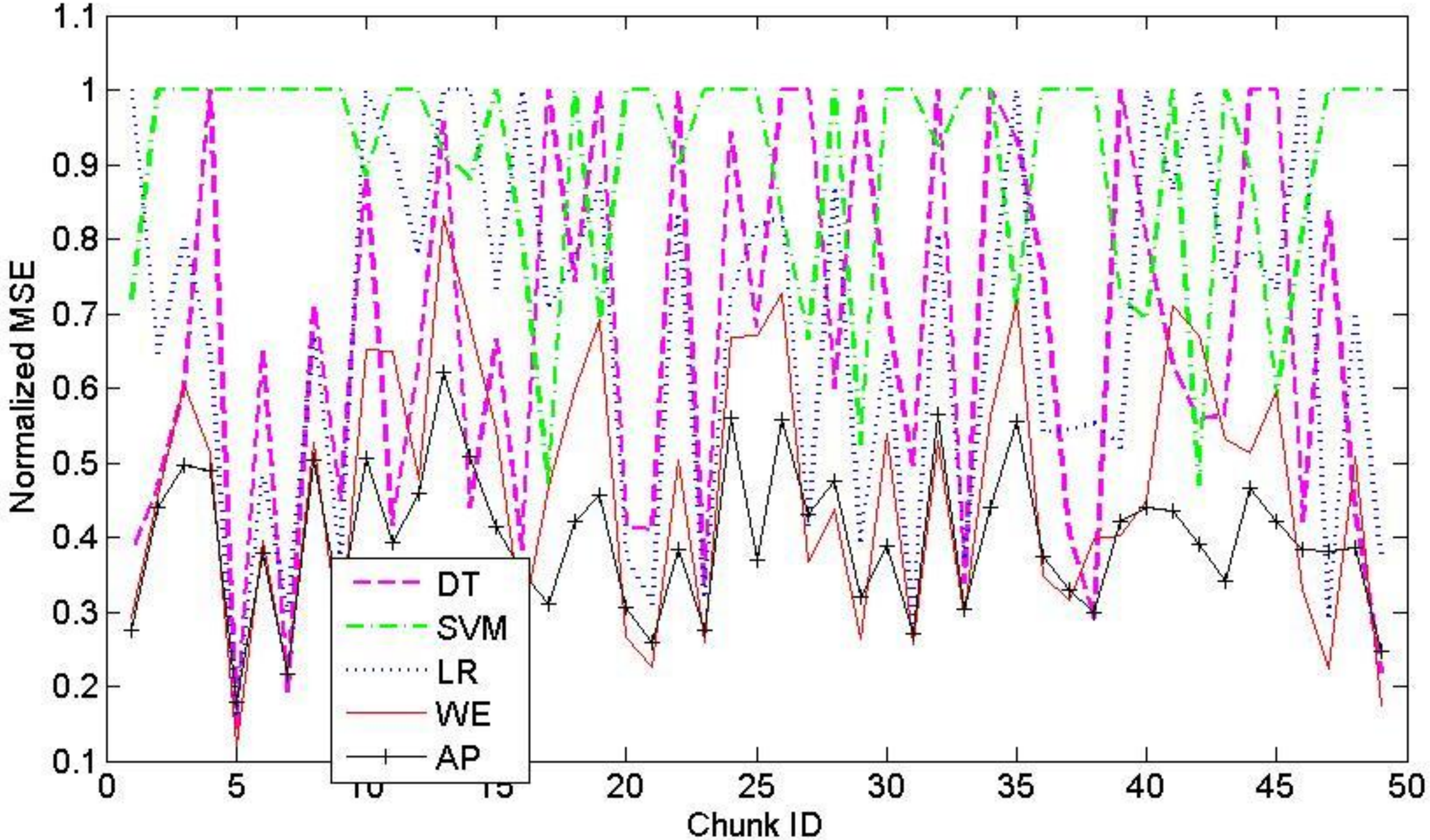
Comparison on Intrusion Data Set

Experimental Results (3)



Classification Accuracy Comparison

Experimental Results (4)

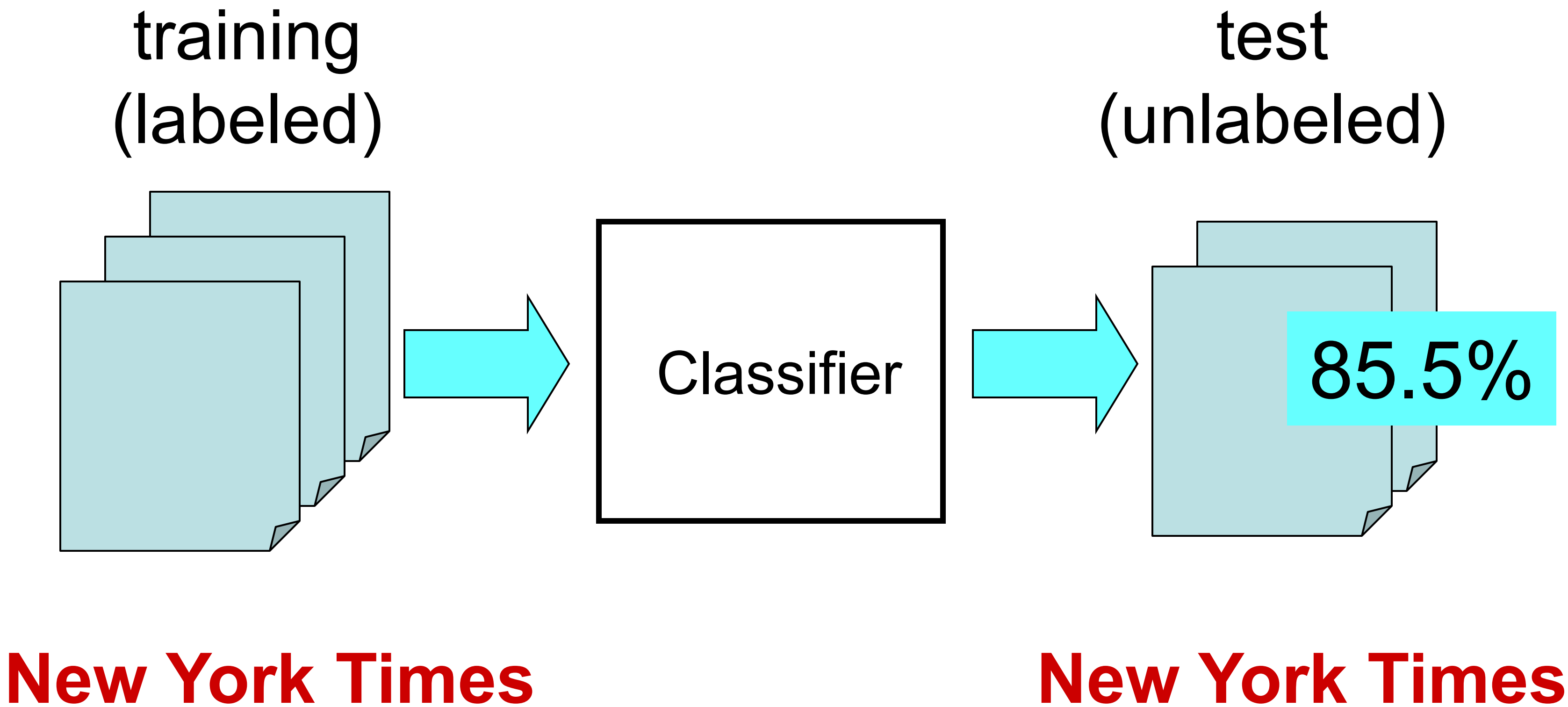


Mean Squared Error Comparison

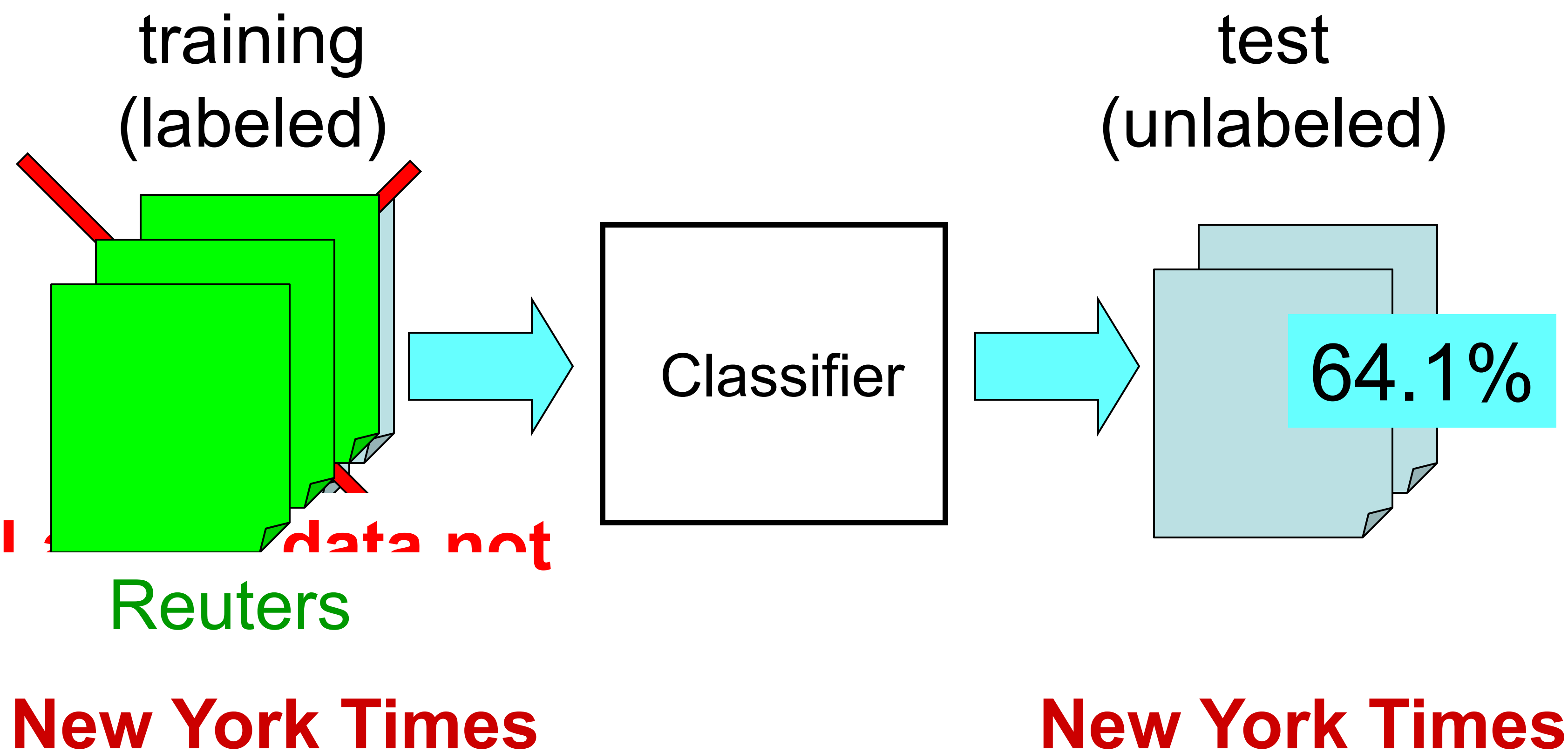
Outline

- An overview of ensemble methods
  - Motivations
  - Tutorial overview
- Supervised ensemble
- Unsupervised ensemble
- Semi-supervised ensemble
  - Multi-view learning
  - Consensus maximization among supervised and unsupervised models
- Applications
  - Stream classification, transfer learning, anomaly detection

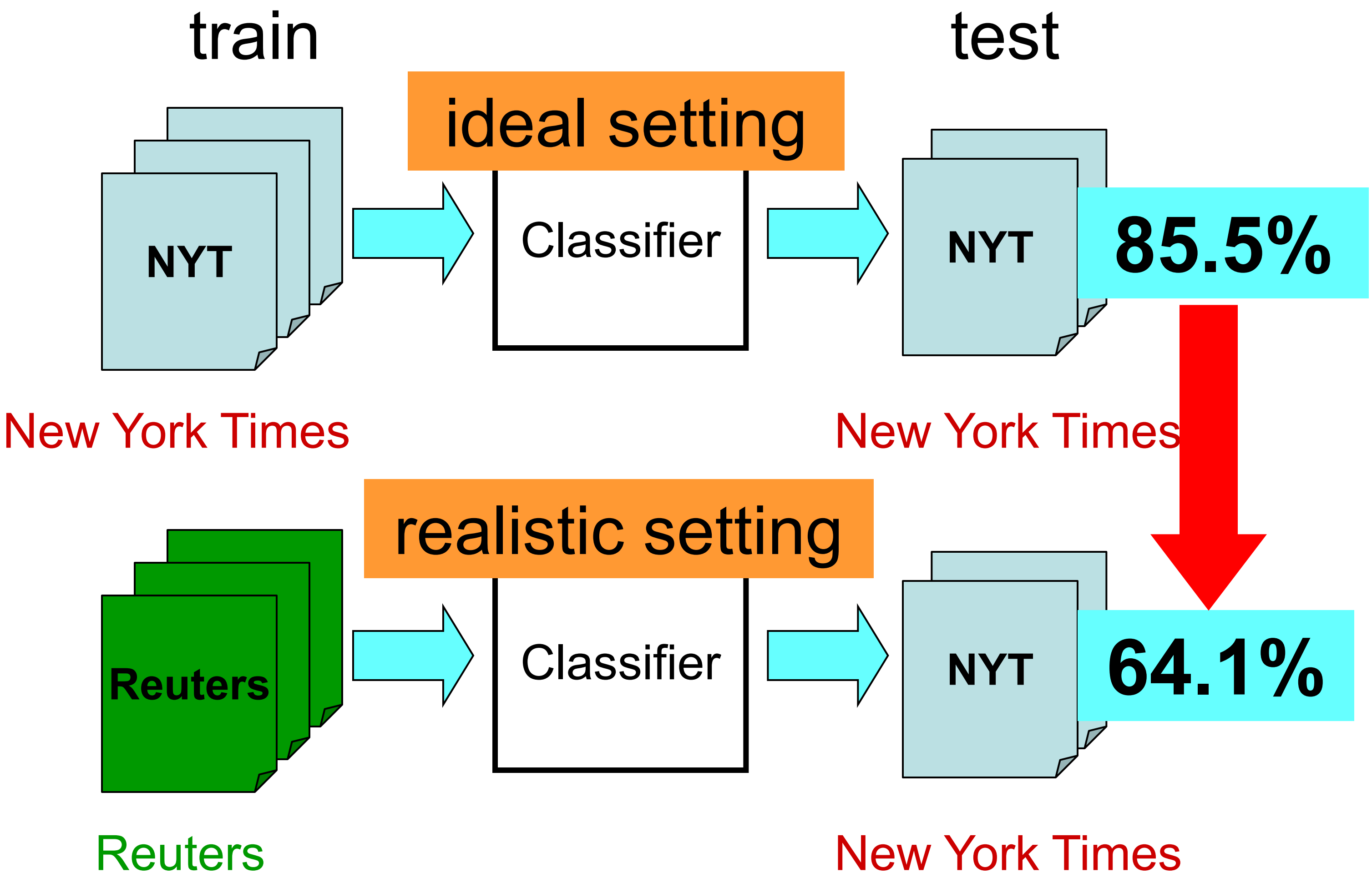
Standard Supervised Learning



In Reality……


training
(labeled)
test
(unlabeled)
Classifier
64.1%
Reuters
New York Times
New York Times

Domain Difference → Performance Drop

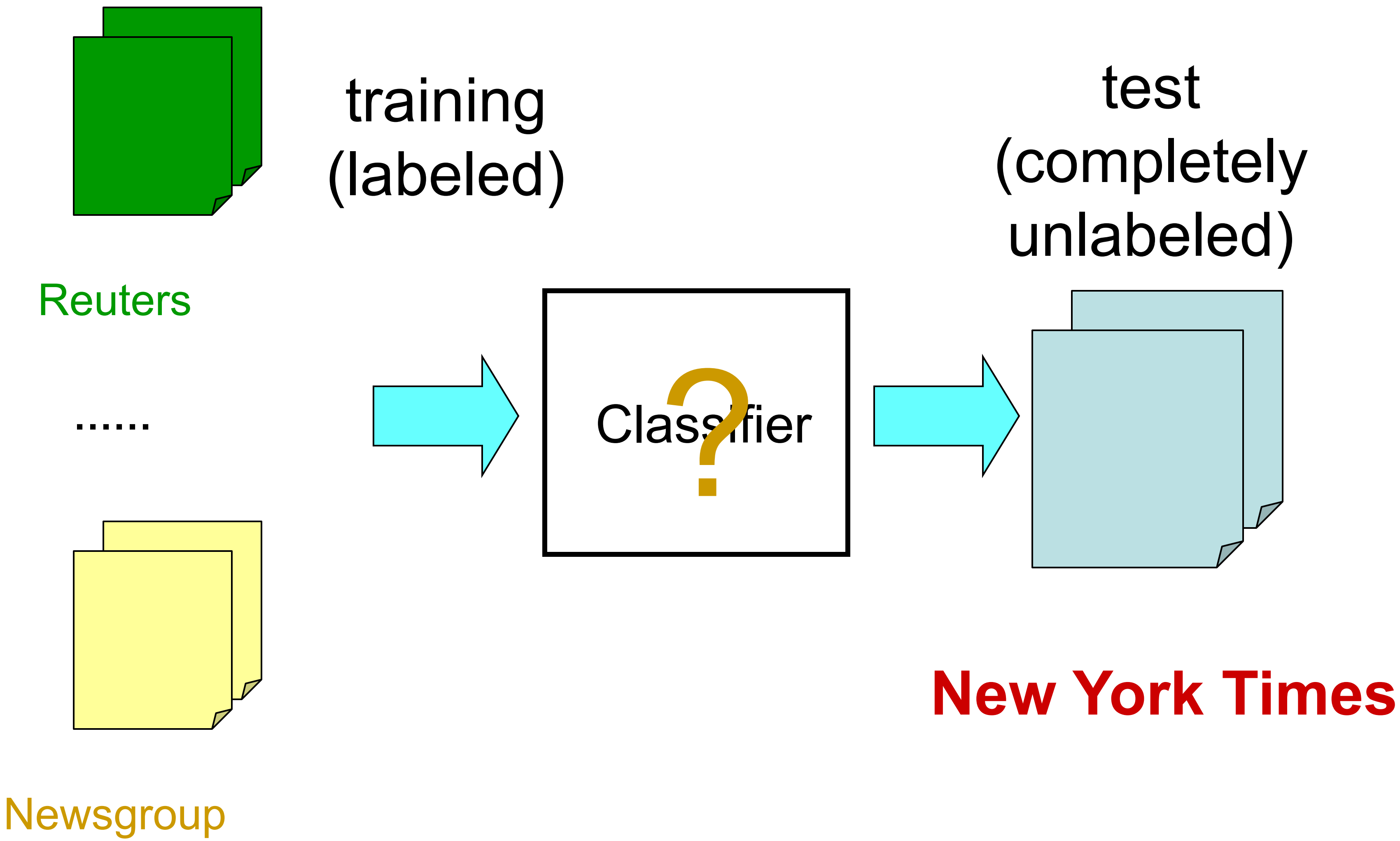


From Jing Jiang's slides

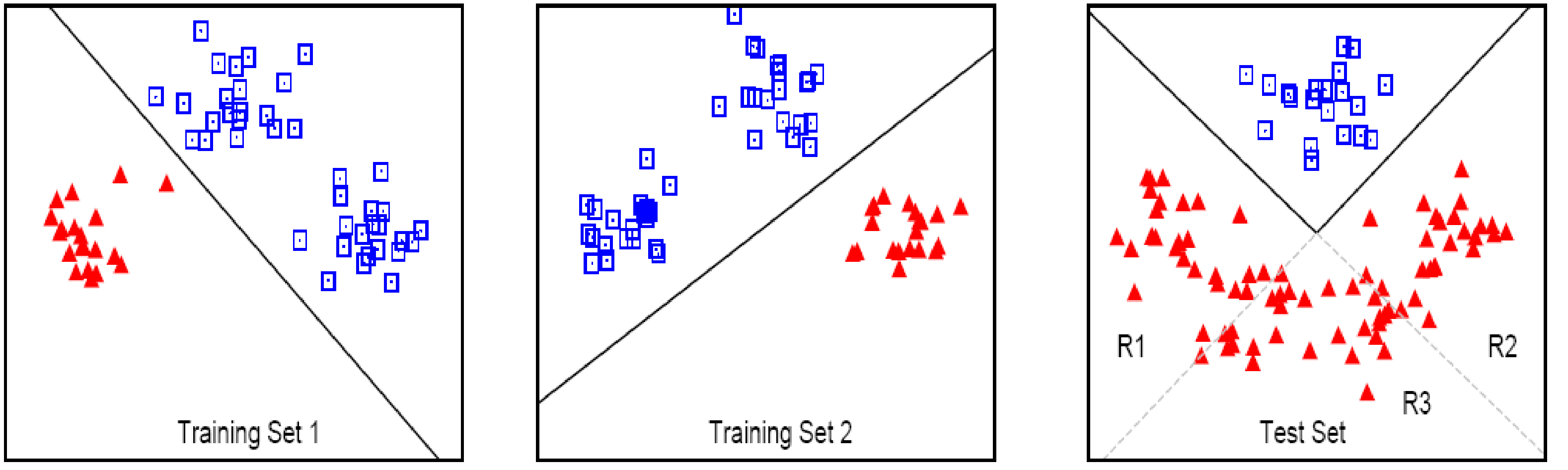
Other Examples

- 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

All Sources of Labeled Information



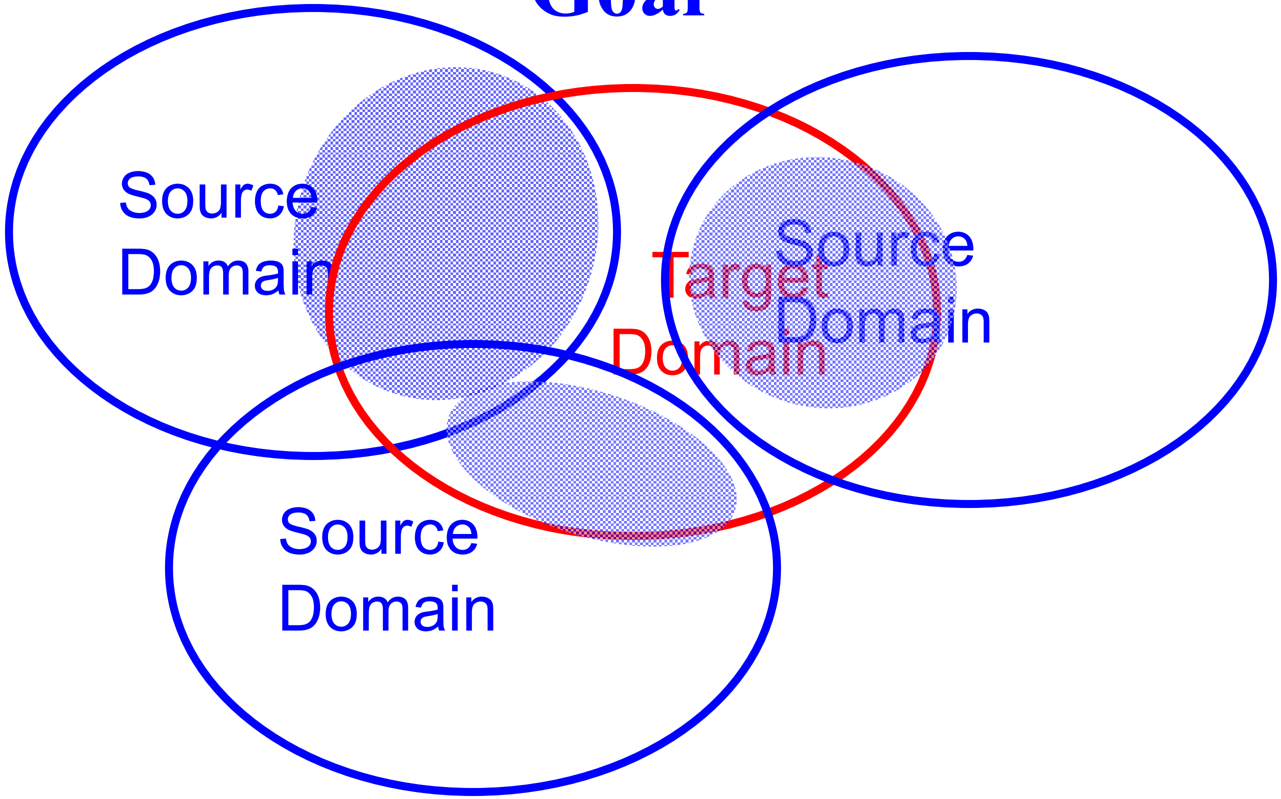
A Synthetic Example



Training **Test**

(have conflicting concepts) **Partially overlapping**

Goal



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

Modified Bayesian Model Averaging

Bayesian Model Averaging

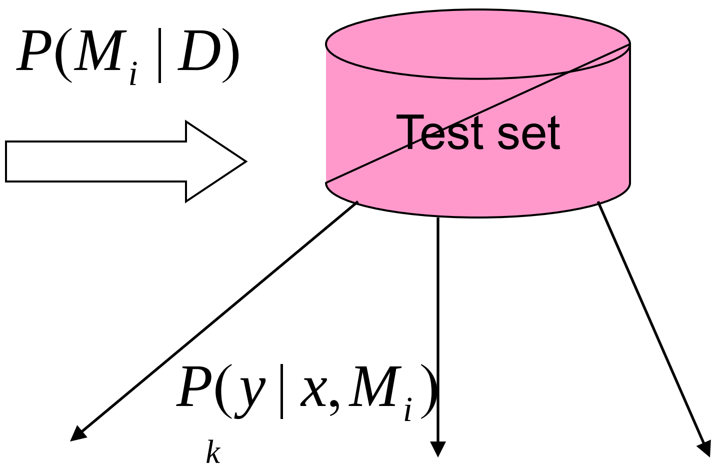
M_1

M_2

......

M_k

$P(M_i \mid D)$



$P(y \mid x, M_i)$

$$P(y \mid x) = \sum_{i=1}^{k} P(M_i \mid D) P(y \mid x, M_i)$$

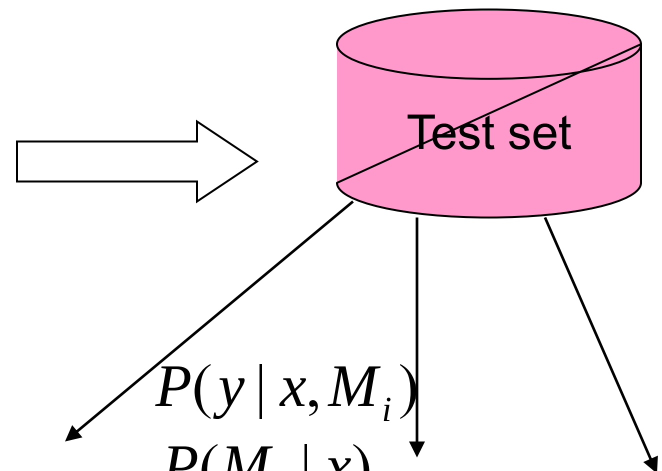
Modified for Transfer Learning

M_1

M_2

......

M_k



$P(y \mid x, M_i)$

$P(M_i \mid x)$

$$P(y \mid x) = \sum_{i=1}^{k} P(M_i \mid x) P(y \mid x, M_i)$$

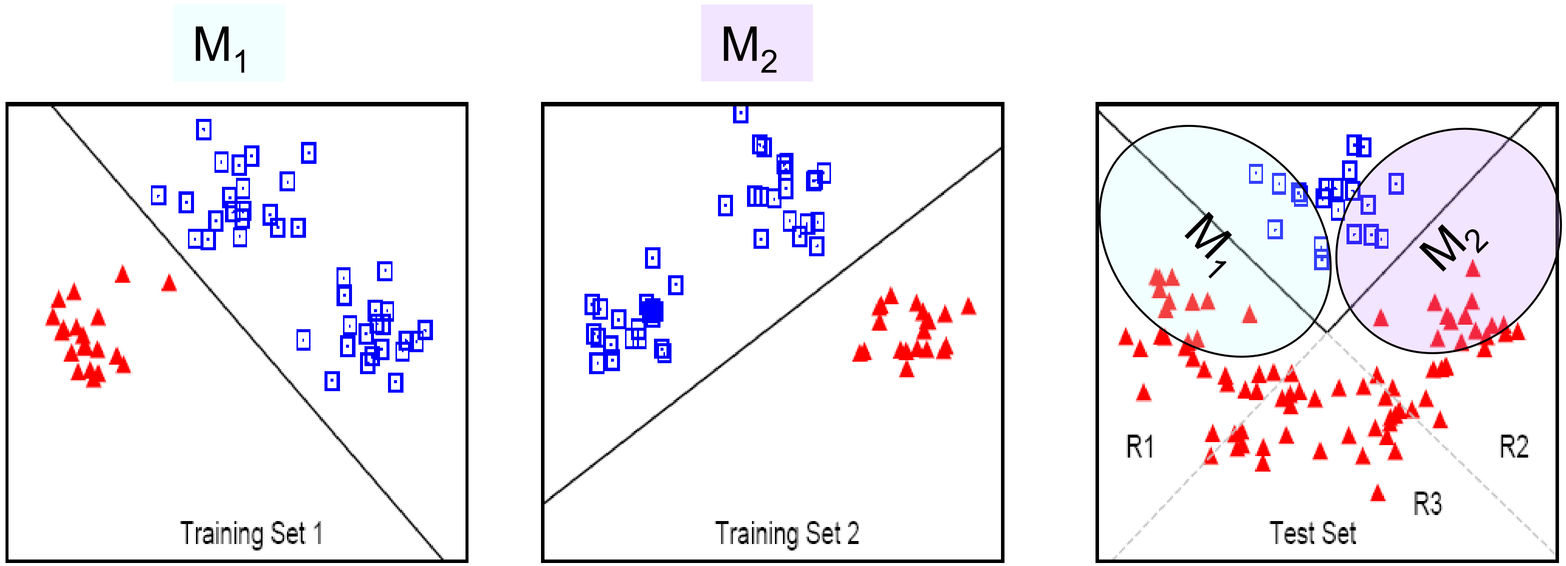
Global versus Local Weights

x		y	M_1	w_g	w_l	M_2	w_g	w_l
2.40	5.23	1	0.6	0.3	0.2	0.9	0.7	0.8
-2.69	0.55	0	0.4	0.3	0.6	0.6	0.7	0.4
-3.97	-3.62	0	0.2	0.3	0.7	0.4	0.7	0.3
2.08	-3.73	0	0.1	0.3	0.5	0.1	0.7	0.5
5.08	2.15	0	0.6	0.3	0.3	0.3	0.7	0.7
1.43	4.48	1	1	0.3	1	0.2	0.7	0
……		…	…	…	…	…	…	…

Training

- Locally weighting scheme
  - Weight of each model is computed per example
  - Weights are determined according to models' performance on the test set, not training set

Synthetic Example Revisited



Training → **Test**

(have conflicting concepts) **Partially overlapping**

Optimal Local Weights

Higher Weight

C_1 | 0.9 0.1

Test example x | 0.8 0.2

C_2 | 0.4 0.6

$$\underbrace{\begin{pmatrix} 0.9 & 0.4 \\ 0.1 & 0.6 \end{pmatrix}}_{\mathbf{H}} \underbrace{\begin{pmatrix} w^1 \\ w^2 \end{pmatrix}}_{\mathbf{w}} = \underbrace{\begin{pmatrix} 0.8 \\ 0.2 \end{pmatrix}}_{\mathbf{f}}$$

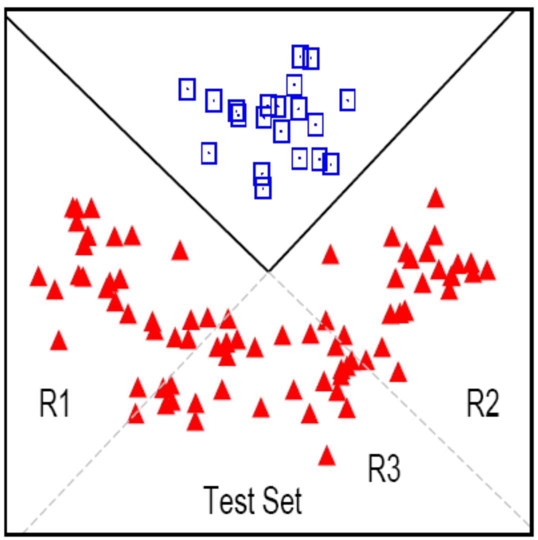
$$\sum_{i=1}^{k} w^i(x) = 1$$

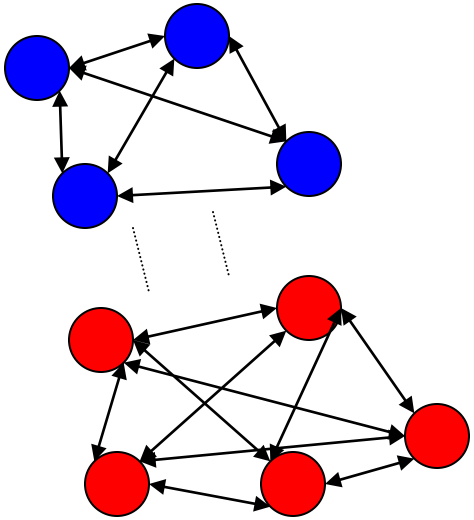
- Optimal weights
  - Solution to a regression problem
  - Impossible to get since f is unknown!

$$\mathbf{w}^* = (\mathbf{H}^T\mathbf{H})^{-1}(\mathbf{H}^T\mathbf{f} + \frac{1}{2}\lambda\mathbf{I}).$$

Clustering-Manifold Assumption

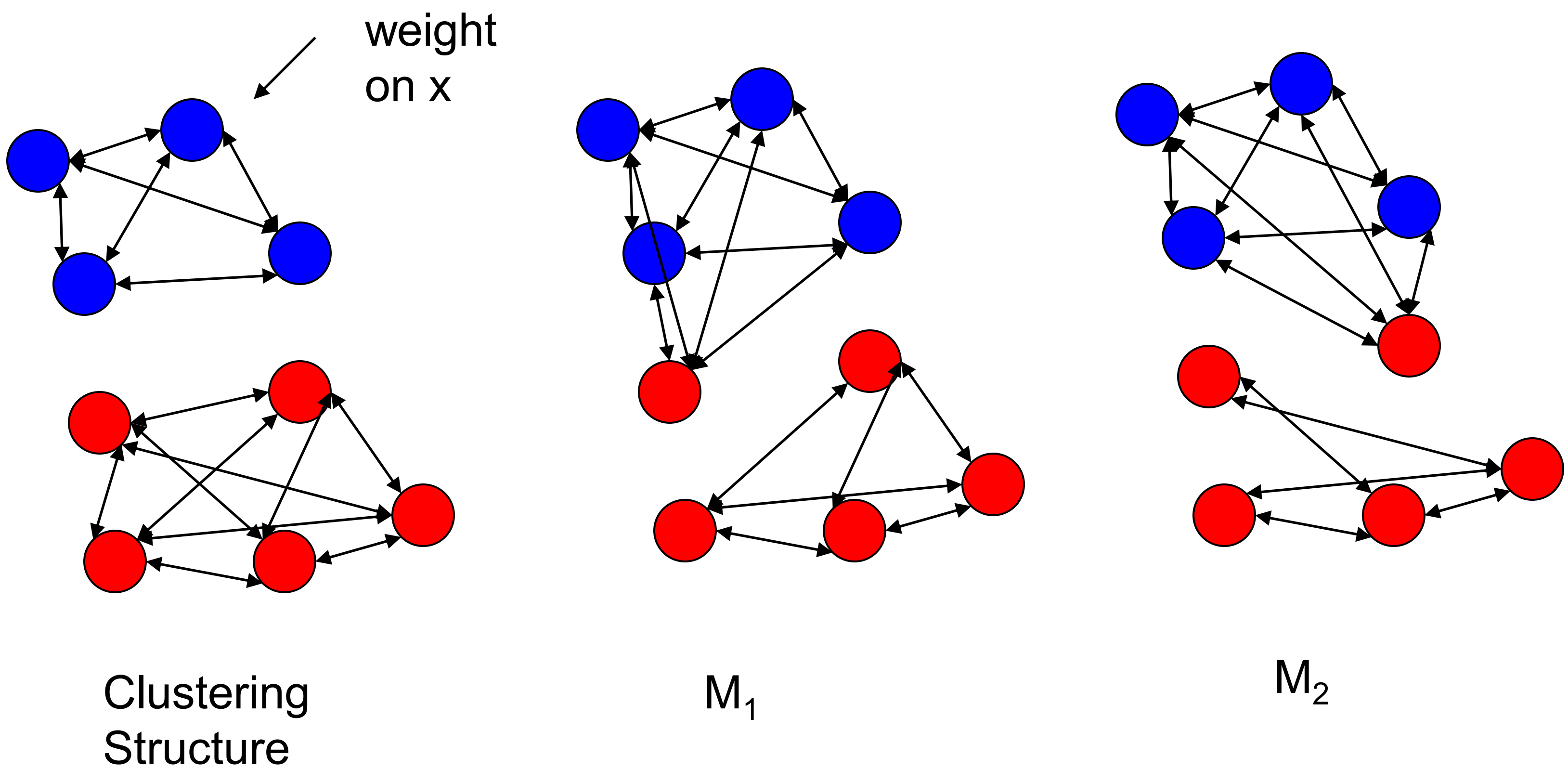
Test examples that are closer in feature space are more likely to share the same class label.



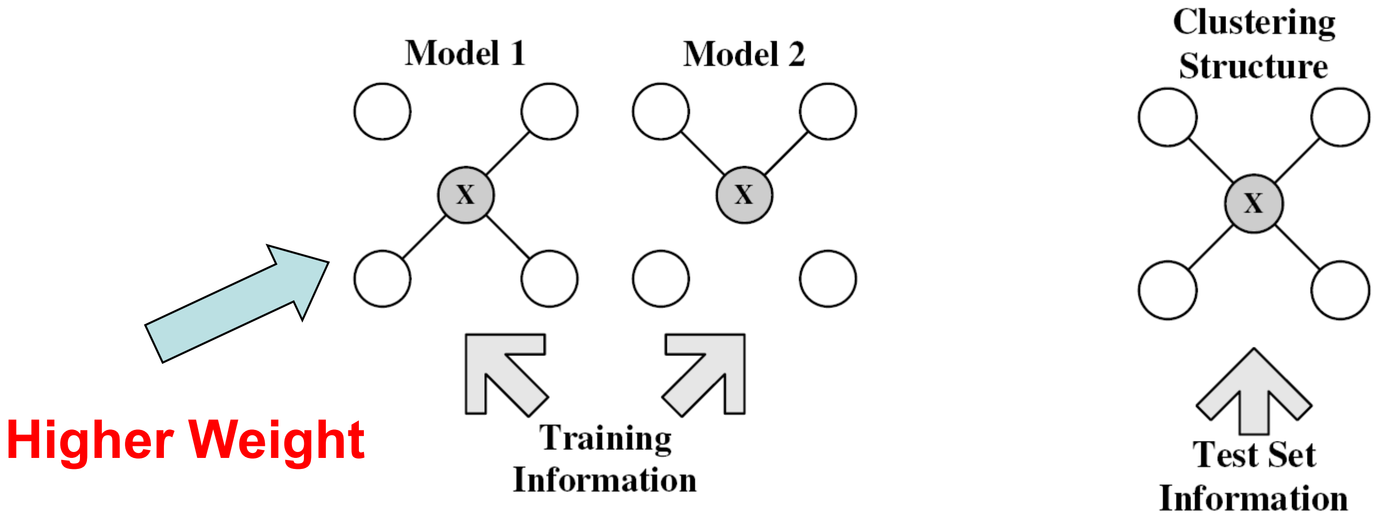


Graph-based Heuristics

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



Graph-based Heuristics



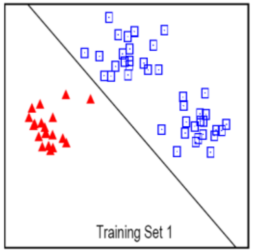
- Local weights calculation
  - Weight of a model is proportional to the similarity between its neighborhood graph and the clustering structure around x.

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

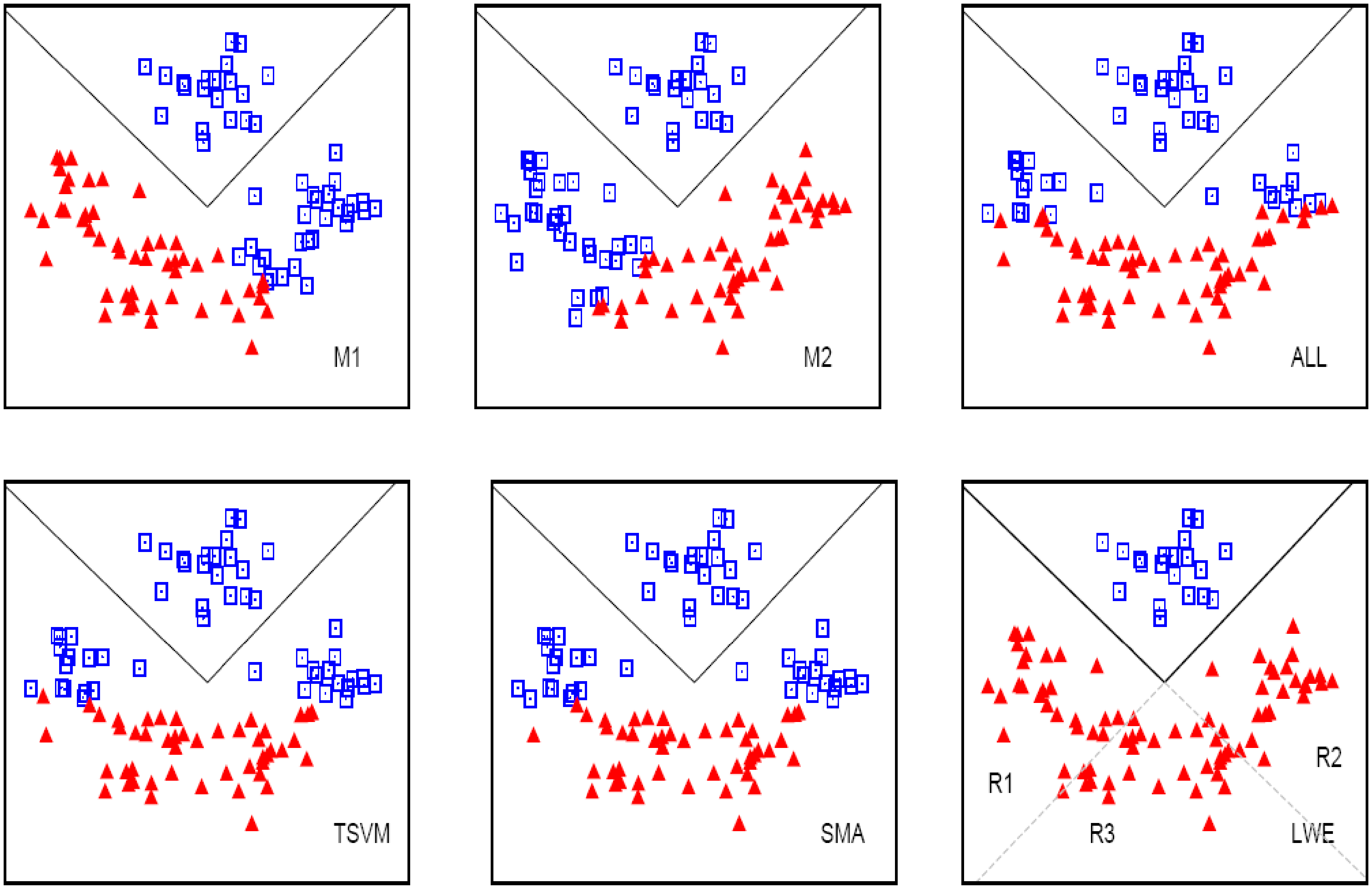
Experiments Setup*

- Data Sets
  - Synthetic data sets
  - Spam filtering: public email collection → personal inboxes (u01, u02, u03) (ECML/PKDD 2006)
  - Text classification: same top-level classification problems with different sub-fields in the training and test sets (Newsgroup, Reuters)
  - Intrusion detection data: different types of intrusions in training and test sets.
- Baseline Methods
  - One source domain: single models (WNN, LR, SVM)
  - Multiple source domains: SVM on each of the domains
  - Merge all source domains into one: ALL
  - Simple averaging ensemble: SMA
  - Locally weighted ensemble: LWE

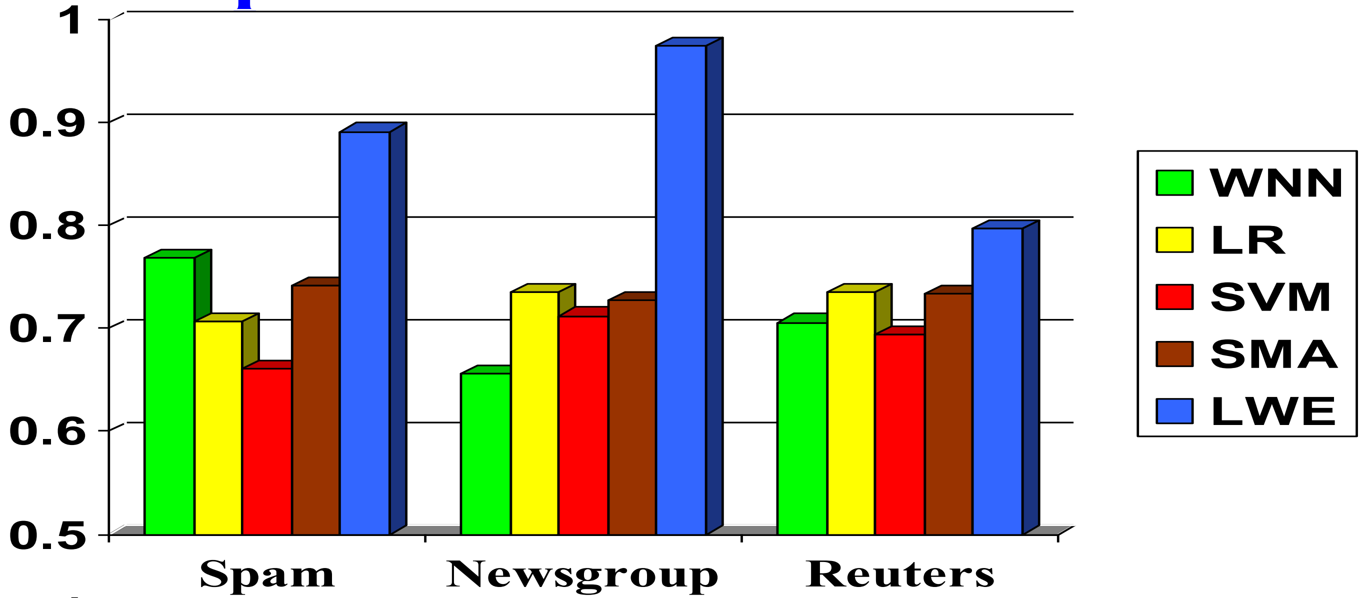
*[GFJ+08]

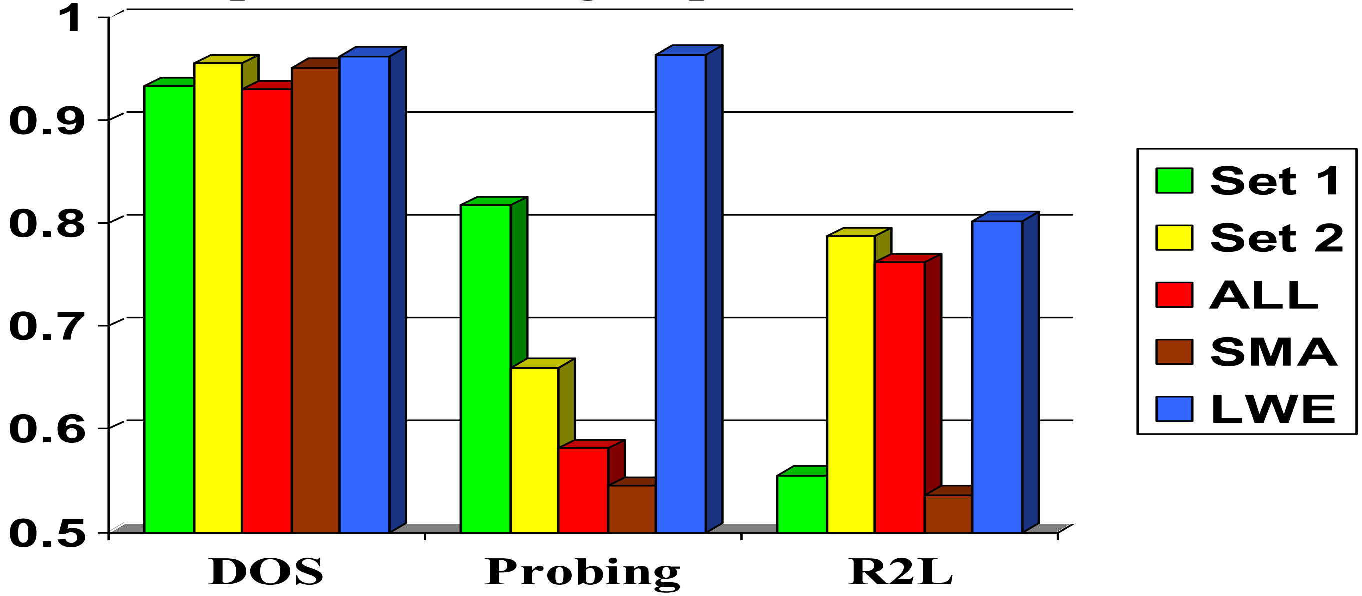


Experiments on Synthetic Data



Experiments on Real Data





Outline

- An overview of ensemble methods
  - Motivations
  - Tutorial overview
- Supervised ensemble
- Unsupervised ensemble
- Semi-supervised ensemble
  - Multi-view learning
  - Consensus maximization among supervised and unsupervised models
- Applications
  - Stream classification, transfer learning, anomaly detection

Combination of Anomaly Detectors

- Simple rules (or atomic rules) are relatively easy to craft.
- Problem:
  - there can be way too many simple rules
  - each rule can have high false alarm or FP rate
- Challenge: can we find their non-trivial combination that significantly improve accuracy?

Atomic Anomaly Detectors

	A_1	A_2	……	A_{k-1}	A_k
				Anomaly?	
Record 1	Y	N	……	N	N
Record 2	N	Y	……	Y	N
Record 3	Y	N	……	N	N
Record 4	Y	Y	……	N	Y
Record 5	N	N	……	Y	Y
Record 6	N	N	……	N	N
Record 7	N	N	……	N	N

……

Why We Need Combine Detectors?

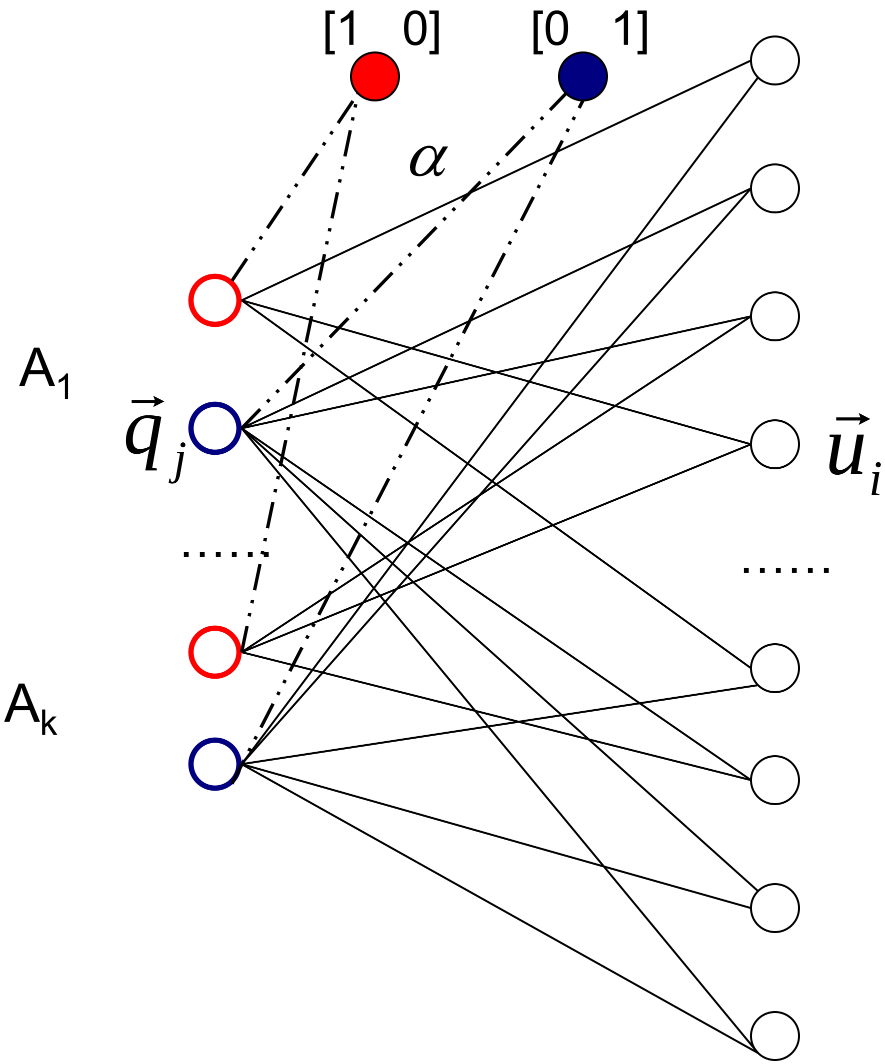
Combining Detectors

- is non-trivial
  - We aim at finding a consolidated solution without any knowledge of the true anomalies (**unsupervised**)
  - We don't know which atomic rules are better and which are worse
  - There could be bad base detectors so that majority voting cannot work

How to Combine Atomic Detectors?

- Basic Assumption:
  - Base detectors are better than random guessing and systemic flip.
- Principles
  - Consensus represents the best we can get from the atomic rules
    - Solution most consistent with atomic detectors
  - Atomic rules should be weighted according to their detection performance
  - We should rank the records according to their probability of being an anomaly
- Algorithm
  - Reach consensus among multiple atomic anomaly detectors in an unsupervised way
    - or semi-supervised if we have limited supervision (known botnet site)
    - and incremental in a streaming environment
  - Automatically derive weights of atomic rules and records

Framework



Detectors

Records

record i $\vec{u}_i = [u_{i0}, u_{i1}]$

detector j $\vec{q}_j = [q_{j0}, q_{j1}]$

probability of anomaly, normal

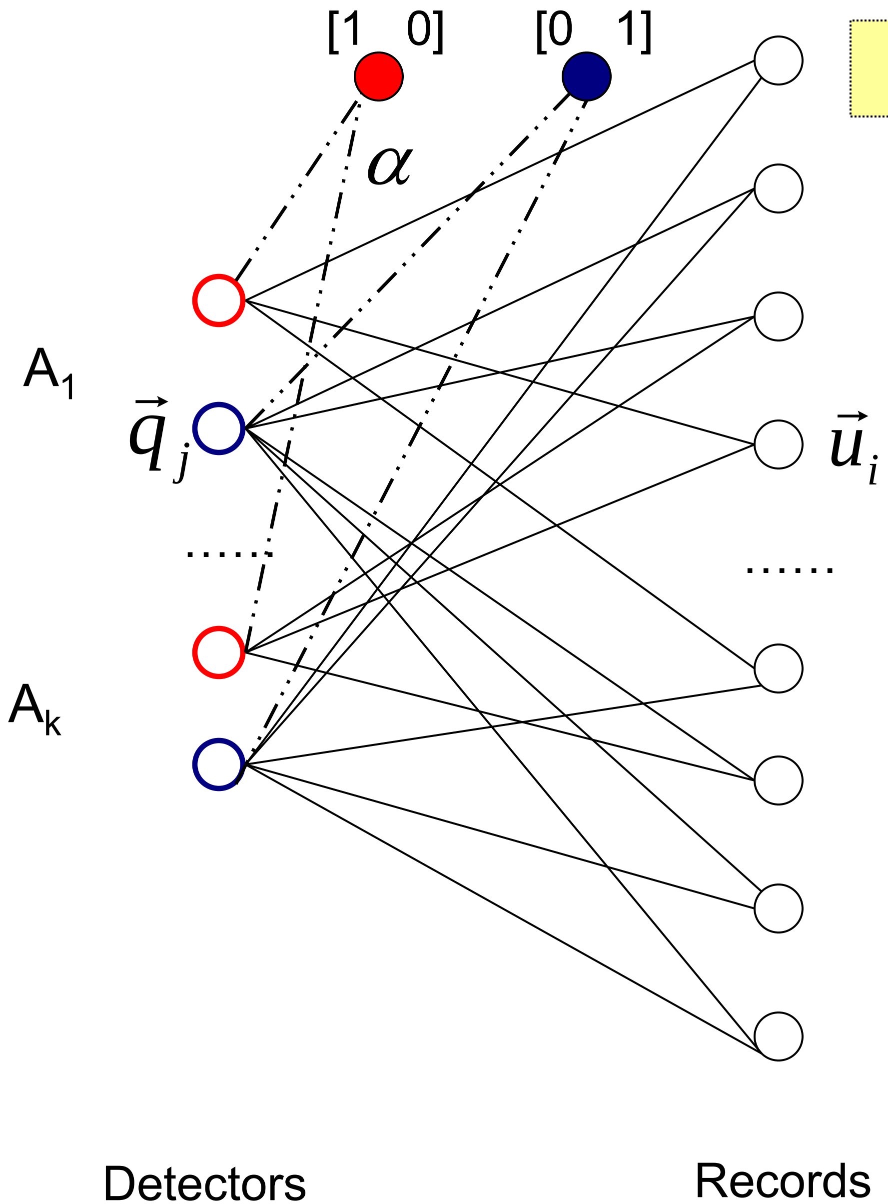
adjacency

$$a_{ij} = \begin{cases} 1 & u_i \leftrightarrow q_j \\ 0 & otherwise \end{cases}$$

initial probability

$$\vec{y}_j = \begin{cases} [1 \quad 0] & anomalous \\ [0 \quad 1] & normal \end{cases}$$

Methodology



Iterate until convergence (proven)

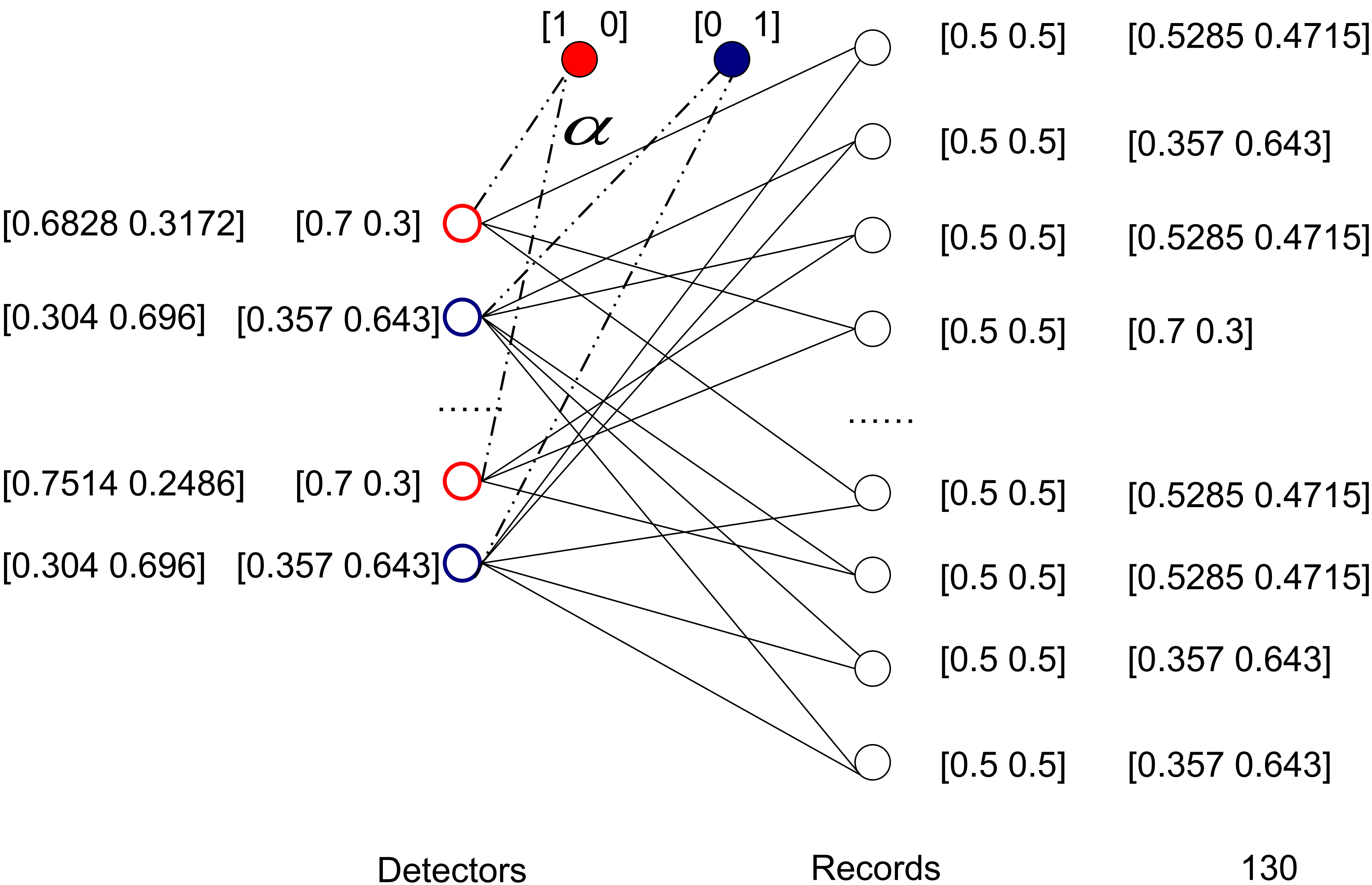
Update detector probability

$$\vec{q}_j = \frac{\sum_{i=1}^{n} a_{ij}\vec{u}_i + \alpha\vec{y}_j}{\sum_{i=1}^{n} a_{ij} + \alpha}$$

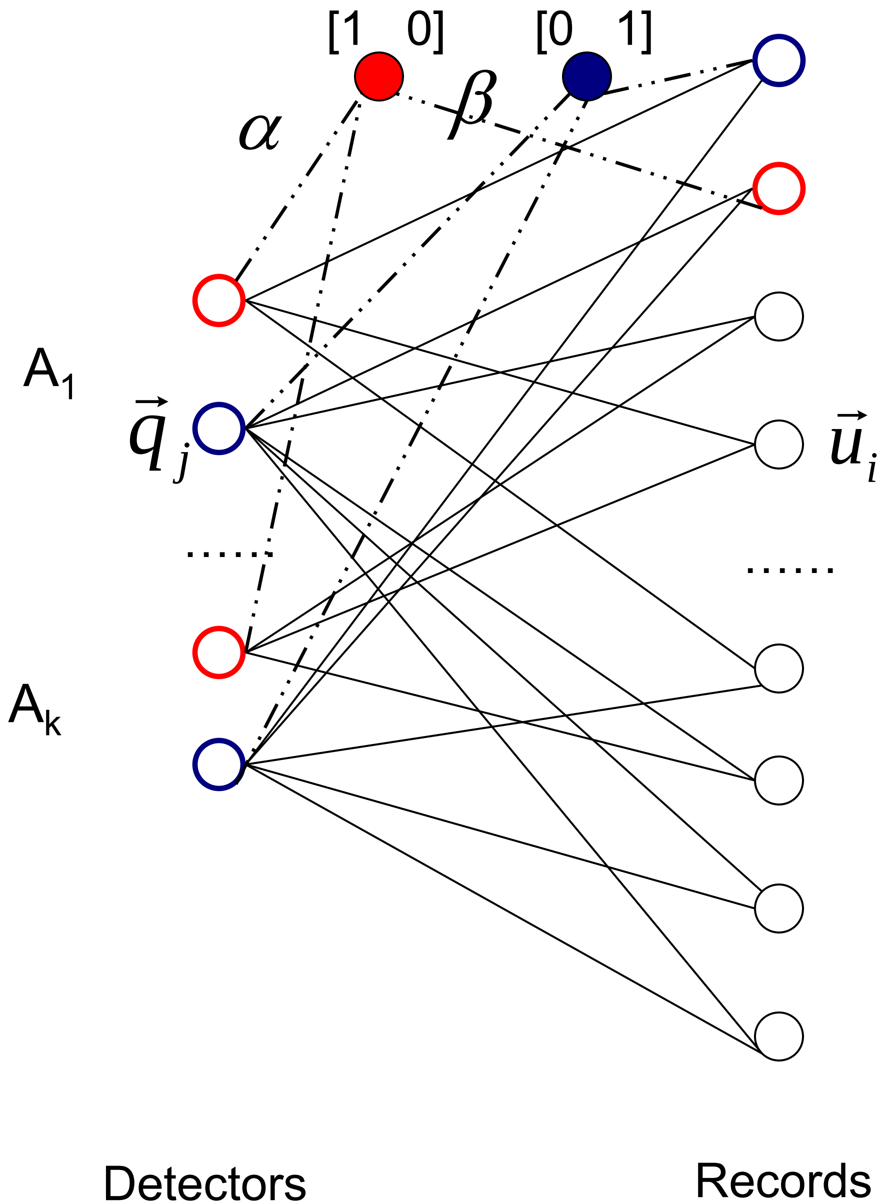
Update record probability

$$\vec{u}_i = \frac{\sum_{j=1}^{v} a_{ij}\vec{q}_j}{\sum_{j=1}^{v} a_{ij}}$$

Propagation Process



Semi-supervised



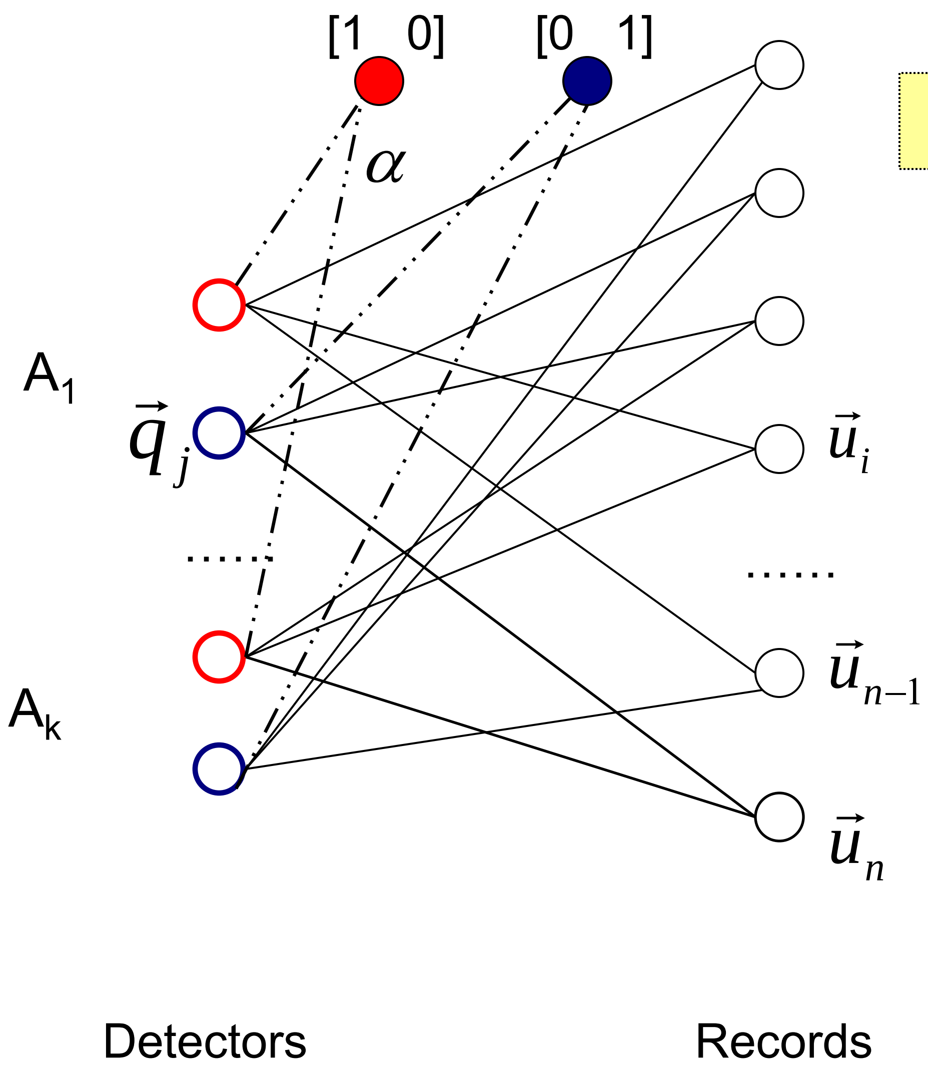
Iterate until convergence

$$\vec{q}_j = \frac{\sum_{i=1}^{n} a_{ij}\vec{u}_i + \alpha\vec{y}_j}{\sum_{i=1}^{n} a_{ij} + \alpha}$$

$$\vec{u}_i = \frac{\sum_{j=1}^{v} a_{ij}\vec{q}_j}{\sum_{j=1}^{v} a_{ij}}$$ unlabeled

$$\vec{u}_i = \frac{\sum_{j=1}^{v} a_{ij}\vec{q}_j + \beta\vec{f}_i}{\sum_{j=1}^{v} a_{ij} + \beta}$$ labeled

Incremental



When a new record arrives

Update detector probability

$$\vec{q}_j = \frac{\sum_{i=1}^{n-1} a_{ij}\vec{u}_i + a_{nj}\vec{u}_n + \alpha\vec{y}_j}{\sum_{i=1}^{n-1} a_{ij} + a_{nj} + \alpha}$$

Update record probability

$$\vec{u}_i = \frac{\sum_{j=1}^{v} a_{ij}\vec{q}_j}{\sum_{j=1}^{v} a_{ij}}$$

Experiments Setup

- Baseline methods
  - base detectors
  - majority voting
  - consensus maximization
  - semi-supervised (2% labeled)
  - stream (30% batch, 70% incremental)
- Evaluation measure
  - area under ROC curve (0-1, 1 is the best)
  - ROC curve: tradeoff between detection rate and false alarm rate

Case study-IDN data

- Data
  - A sequence of events: dos flood, syn flood, port scanning, etc.
  - 3 random subsets, each with size 1000
- Detector
  - Count of events at each time stamp with different thresholds
  - Entropy of events at each time stamp with different thresholds
  - 0.1-0.5, 0.3-0.7, 0.5-0.9

AUC on IDN data

	worst	best	average	Majority voting	Conse nsus	Semi-supervised	Increm ental
1	0.5269	0.6671	0.5904	0.7089	0.7255	0.7204	0.7270
2	0.2832	0.8059	0.5731	0.6854	0.7711	0.8048	0.7552
3	0.3745	0.8266	0.6654	0.8871	0.9076	0.9089	0.9090

- Summary
  - Large variance in detector performance
  - Consensus method improves over the base detector and majority voting
  - Semi-supervised method achieves the best

Case study-KDD cup'99 data

- Data
  - A series of TCP connection records, collected by MIT Lincoln labs
  - We use the 34 continuous derived features, including duration, number of bytes, error rate, etc.
  - 3 random subsets, each with size 1832
- Detector
  - Randomly select a subset of features, and apply unsupervised distance-based anomaly detection algorithm
  - Get 20 detectors

AUC on KDD cup data

	worst	best	average	Majority voting	Consensus	Semi-supervised	incremental
1	0.5804	0.6068	0.5981	0.7765	0.7812	0.8005	0.7730
2	0.5930	0.6137	0.6021	0.7865	0.7938	0.8173	0.7836
3	0.5851	0.6150	0.6022	0.7739	0.7796	0.7985	0.7727

- Summary
  - Small variance in detector performance
  - Consensus method improves over the base detector and majority voting
  - Semi-supervised method achieves the best

Conclusions

- Ensemble
  - Combining independent, diversified models improves accuracy
  - No matter in supervised, unsupervised, or semi-supervised scenarios, ensemble methods have demonstrated their strengths
  - Base models are combined by learning from labeled data or by their consensus
- Beyond accuracy improvements
  - Information explosion motivates multiple source learning
  - Various learning packages available
  - Combine the complementary predictive powers of multiple models
  - Distributed computing, privacy-preserving applications

References

- [BlMi98] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. *Proceedings of the Workshop on Computational Learning Theory*, pages 92-100, 1998.
- [Breiman96] L. Breiman. Bagging predictors. *Machine Learning*, 26:123-140, 1996.
- [Breiman01] L. Breiman. Random forests. *Machine Learning*, 45(1):5-32, 2001.
- [FGM+05] W. Fan, E. Greengrass, J. McCloskey, P. S. Yu, and K. Drummey. Effective estimation of posterior probabilities: Explaining the accuracy of randomized decision tree approaches. In *Proc. 2005 Int. Conf. Data Mining (ICDM'05)*, pages 154-161, 2005.
- [FeBr04] X. Z. Fern and C. E. Brodley. Solving cluster ensemble problems by bipartite graph partitioning. In *Proc. 2004 Int. Conf. Machine Learning (ICML'04)*, pages 281-288, 2004.
- [FrSc97] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119-139, 1997.
- [GFH07] J. Gao, W. Fan, and J. Han. On appropriate assumptions to mine data streams: Analysis and practice. In *Proc. 2007 Int. Conf. Data Mining (ICDM'07)*, pages 143-152, 2007.
- [GFJ+08] J. Gao, W. Fan, J. Jiang, and J. Han. Knowledge transfer via multiple model local structure mapping. In *Proc. 2008 ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD'08)*, pages 283-291, 2008.
- [GLF+09] J. Gao, F. Liang, W. Fan, Y. Sun, and J. Han. Graph-based consensus maximization among multiple supervised and unsupervised models. In *Advances in Neural Information Processing Systems 22 (to appear)*, 2009.
- [PTJ05] W. Punch, A. Topchy, and A. K. Jain. Clustering ensembles: Models of consensus and weak partitions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1866-1881, 2005.
- [StGh03] A. Strehl and J. Ghosh. Cluster ensembles --a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583-617, 2003.

References

- [AUL08] M. Amini, N. Usunier, and F. Laviolette. A transductive bound for the voted classifier with an application to semi-supervised learning. In *Advances in Neural Information Processing Systems 21*, 2008.
- [BBM07] A. Banerjee, S. Basu, and S. Merugu. Multi-way clustering on relation graphs. In *Proc. 2007 SIAM Int. Conf. Data Mining (SDM'07)*, 2007.
- [BaKo04] E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36:105-139, 2004.
- [BEM05] R. Bekkerman, R. El-Yaniv, and A. McCallum. Multi-way distributional clustering via pairwise interactions. In *Proc. 2005 Int. Conf. Machine Learning (ICML'05)*, pages 41-48, 2005.
- [BDH05] P. N. Bennett, S. T. Dumais, and E. Horvitz. The combination of text classifiers using reliability indicators. *Information Retrieval*, 8(1):67-100, 2005.
- [BiSc04] S. Bickel and T. Scheffer. Multi-view clustering. In *Proc. 2004 Int. Conf. Data Mining (ICDM'04)*, pages 19-26, 2004.
- [BGS+08] P. Brazdil, C. Giraud-Carrier, C. Soares, and R. Vilalta. *Metalearning: Applications to Data Mining*. Springer, 2008.
- [Caruana97] R. Caruana. Multitask learning. *Machine Learning*, 28(1):41-75, 1997.
- [CKW08] K. Crammer, M. Kearns, and J. Wortman. Learning from multiple sources. *Journal of Machine Learning Research*, 9:1757-1774, 2008.
- [DYX+07] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu. Boosting for transfer learning. In Proc. 2007 Int. Conf. Machine Learning (ICML'07), pages 193-200, 2007.
- [DaFa06] I. Davidson and W. Fan. When efficient model averaging out-performs boosting and bagging. In Proc. 2006 European Conf. Principles and Practice of Knowledge Discovery in Databases (PKDD'06), pages 478-486, 2006.
- [DMM03] I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In Proc. 2003 ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD'03), pages 89-98, 2003.
- [Dietterich00] T. Dietterich. Ensemble methods in machine learning. In Proc. 2000 Int. Workshop Multiple Classifier Systems, pages 1-15, 2000.

References

- [DoAl09] C. Domeniconi and M. Al-Razgan. Weighted cluster ensembles: Methods and analysis. ACM Transactions on Knowledge Discovery from Data (TKDD), 2(4):1-40, 2009.
- [Domingos00] P. Domingos. Bayesian averaging of classifiers and the overfitting problem. In Proc. 2000 Int. Conf. Machine Learning (ICML'00), pages 223-230, 2000.
- [DHS01] R. O. Duda, P. E. Hart, and D. G. Stork. Pattern Classification. John Wiley & Sons, second edition, 2001.
- [DzZe02] S. Dzeroski and B. Zenko. Is combining classifiers better than selecting the best one. In Proc. 2002 Int. Conf. Machine Learning (ICML'02), pages 123-130, 2002.
- [FaDa07] W. Fan and I. Davidson. On sample selection bias and its efficient correction via model averaging and unlabeled examples. In Proc. 2007 SIAM Int. Conf. Data Mining (SDM'07), 2007.
- [FeLi08] X. Z. Fern and W. Lin. Cluster ensemble selection. In Proc. 2008 SIAM Int. Conf. Data Mining (SDM'08), 2008.
- [FiSk03] V. Filkov and S. Skiena. Integrating microarray data by consensus clustering. In Proc. 2003 Int. Conf. Tools with Artificial Intelligence, pages 418-426, 2003.
- [FrPo08] J. H. Friedman and B. E. Popescu. Predictive learning via rule ensembles. Annals of Applied Statistics, 3(2):916-954, 2008.
- [GGB+08] K. Ganchev, J. Graca, J. Blitzer, and B. Taskar. Multi-view learning over structured and non-identical outputs. In Proc. 2008 Conf. Uncertainty in Artificial Intelligence (UAI'08), pages 204-211, 2008.
- [GFS+09] J. Gao, W. Fan, Y. Sun, and J. Han. Heterogeneous source consensus learning via decision propagation and negotiation. In Proc. 2009 ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD'09), pages 339-347, 2009.
- [GeTa07] L. Getoor and B. Taskar. Introduction to statistical relational learning. MIT Press, 2007.
- [GMT07] A. Gionis, H. Mannila, and P. Tsaparas. Clustering aggregation. ACM Transactions on Knowledge Discovery from Data (TKDD), 1(1), 2007.
- [GVB04] C. Giraud-Carrier, R. Vilalta, and P. Brazdil. Introduction to the special issue on meta-learning. Machine Learning, 54(3):187-193, 2004.
- [HKT06] S. T. Hadjitodorov, L. I. Kuncheva, and L. P. Todorova. Moderate diversity for better cluster ensembles. Information Fusion, 7(3):264-275, 2006.

References

- [HaKa06] J. Han and M. Kamber. Data mining: concepts and techniques. Morgan Kaufmann, second edition, 2006.
- [HTF09] T. Hastie, R. Tibshirani, and J. Friedman. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, second edition, 2009.
- [HMR+99] J. Hoeting, D. Madigan, A. Raftery, and C. Volinsky. Bayesian model averaging: a tutorial. Statistical Science, 14:382-417, 1999.
- [JJN+91] R. Jacobs, M. Jordan, S. Nowlan, and G. Hinton. Adaptive mixtures of local experts. Neural Computation, 3(1):79-87, 1991.
- [KoMa] J. Kolter and M. Maloof. Using additive expert ensembles to cope with concept drift. In Proc. 2005 Int. Conf. Machine Learning (ICML'05), pages 449-456, 2005.
- [KuWh03] L. I. Kuncheva and C. J. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. Machine Learning, 51(2):181-207, 2003.
- [LiDi08] T. Li and C. Ding. Weighted consensus clustering. In Proc. 2008 SIAM Int. Conf. Data Mining (SDM'08), 2008.
- [LiOg05] T. Li and M. Ogihara. Semisupervised learning from different information sources. Knowledge and Information Systems, 7(3):289-309, 2005.
- [LiYa06] C. X. Ling and Q. Yang. Discovering classification from data of multiple sources. Data Mining and Knowledge Discovery, 12(2-3):181-201, 2006.
- [LZY05] B. Long, Z. Zhang, and P. S. Yu. Combining multiple clusterings by soft correspondence. In Proc. 2005 Int. Conf. Data Mining (ICDM'05), pages 282-289, 2005.
- [LZX+08] P. Luo, F. Zhuang, H. Xiong, Y. Xiong, and Q. He. Transfer learning from multiple source domains via consensus regularization. In Proc. 2008 Int. Conf. Information and Knowledge Management (CIKM'08), pages 103-112, 2008.
- [OkVa08] O. Okun and G. Valentini. Supervised and Unsupervised Ensemble Methods and their Applications. Springer, 2008.
- [Polikar06] R. Polikar. Ensemble based systems in decision making. IEEE Circuits and Systems Magazine, 6(3):21-45, 2006.
- [PrSc08] C. Preisach and L. Schmidt-Thieme. Ensembles of relational classifiers. Knowledge and Information Systems, 14(3):249-272, 2008.

References

- [RoKa07] D. M. Roy and L. P. Kaelbling. Efficient bayesian task-level transfer learning. In Proc. 2007 Int. Joint Conf. Artificial Intelligence, pages 2599-2604, 2007.
- [SMP+07] V. Singh, L. Mukherjee, J. Peng, and J. Xu. Ensemble clustering using semidefinite programming. In Advances in Neural Information Processing Systems 20, 2007.
- [TuGh96] K. Tumer and J. Ghosh. Analysis of decision boundaries in linearly combined neural classifiers. Pattern Recognition, 29, 1996.
- [ViDr02] R. Vilalta and Y. Drissi. A perspective view and survey of meta-learning. Artificial Intelligence Review, 18(2):77-95, 2002.
- [WFY+03] H. Wang, W. Fan, P. Yu, and J. Han. Mining concept-drifting data streams using ensemble classifiers. In Proc. 2003 ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD'03), pages 226-235, 2003.
- [Wolpert92] D. H. Wolpert. Stacked generalization. Neural Networks, 5:241-259, 1992.
- [ZGY05] J. Zhang, Z. Ghahramani, and Y. Yang. Learning multiple related tasks using latent independent component. In Advances in Neural Information Processing Systems 18, 2005.
- [ZFY+06] K. Zhang, W. Fan, X. Yuan, I. Davidson, and X. Li. Forecasting skewed biased stochastic ozone days: Analyses and solutions. In Proc. 2006 Int. Conf. Data Mining (ICDM'06), pages 753-764, 2006.

Thanks!

- Any questions?

Slides and more references available at
http://ews.uiuc.edu/~jinggao3/sdm10ensemble.htm