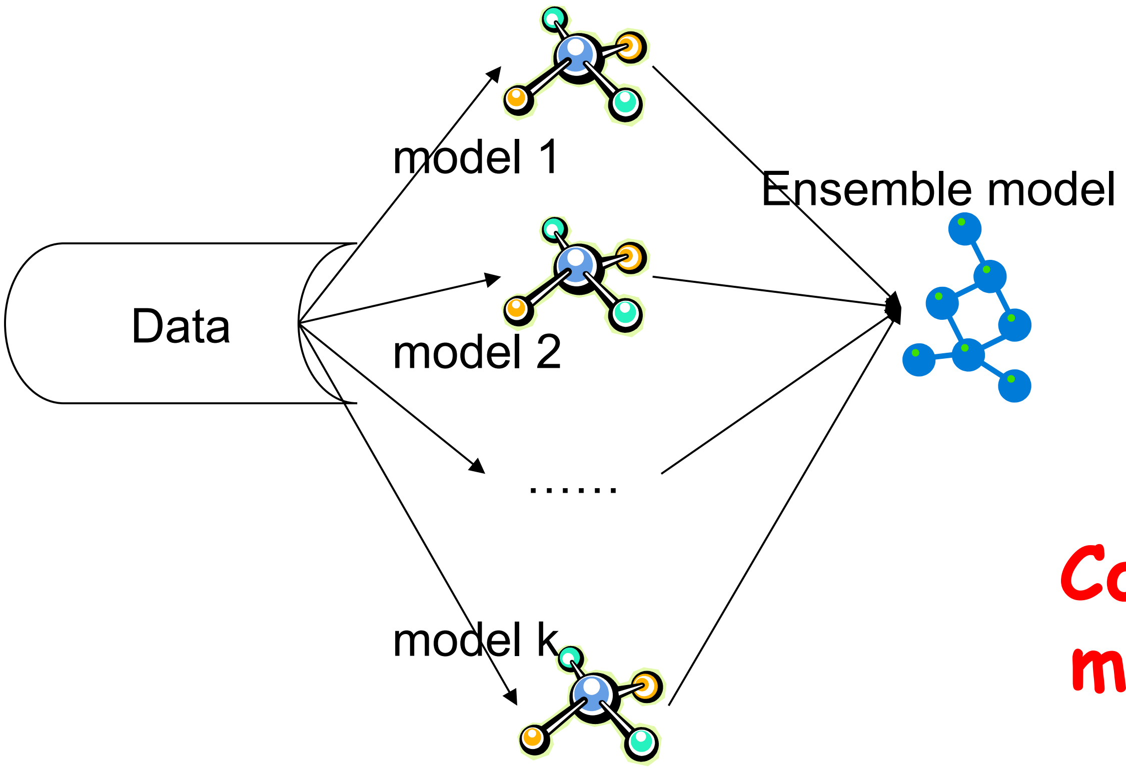
Ensemble Learning

Lectured by Shangsong Liang
Sun Yat-sen University

Outline

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

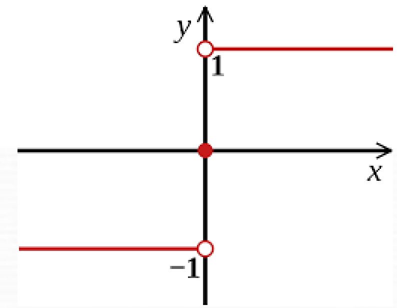
Ensemble



Combine multiple models into one!

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

Example: Ensemble for Classification



$$\left.\begin{array}{l} h_1(x) \in \{-1,+1\} \\ h_2(x) \in \{-1,+1\} \\ \quad\vdots \\ h_T(x) \in \{-1,+1\} \end{array}\right\} \quad H_T(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

Weak classifiers strong classifier

slightly better than random

How to get weak classifiers?

1. Different weak classifiers as base classifiers.

2. The same weak classifier, but different parameters.

3. Using different subset of features/dimensions of the training data.

4. Different subset of training data: **bagging** (such as boostrap aggregating), and **boosting**

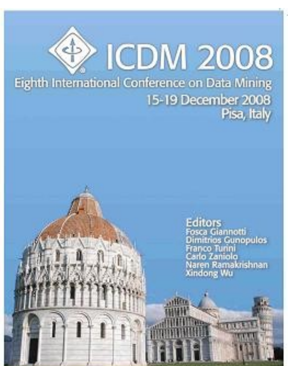
How to combine weak classifiers?

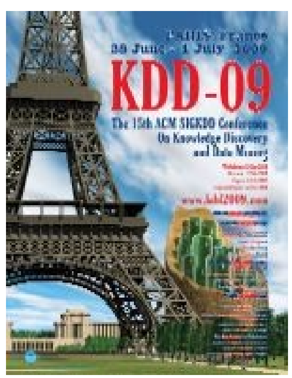
1. Multiple experts: Parallel architecture. Vote for the final decisions.

2. Cascade connection: The next base classifier can only make decision based on the output of the previous base classifier. E.g., cascading ensemble learning.

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
- 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
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
Progress Prize 2007 - RMSE = 0.8723 - Winning Team: KorBell				
Cinematch score - RMSE = 0.9525				

“A good solution (RMSE=0.8712) consists of blending 107 individual results. “

“Predictive accuracy is substantially improved when blending multiple predictors. Our experience is that most efforts should be concentrated in deriving substantially different approaches, rather than refining a single technique. ”

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

Relationship with Related Studies (2)

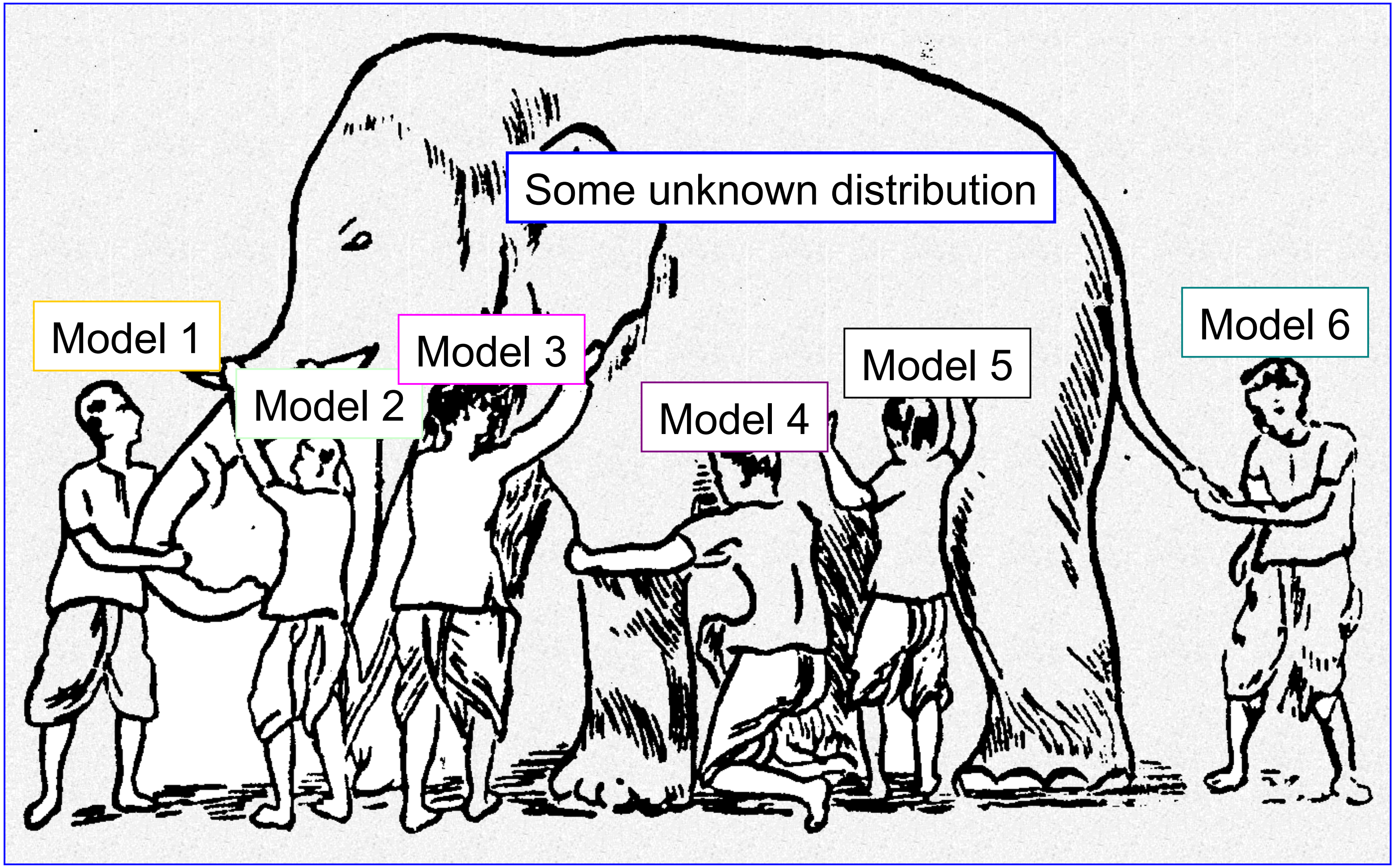
- Meta learning
  - **Learn** on meta-data (include base model output)
  - Ensemble methods: besides learn a joint model based on model output, we can also combine the output by **consensus**
- Non-redundant clustering
  - Give **multiple** non-redundant clustering solutions to users
  - Ensemble methods: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

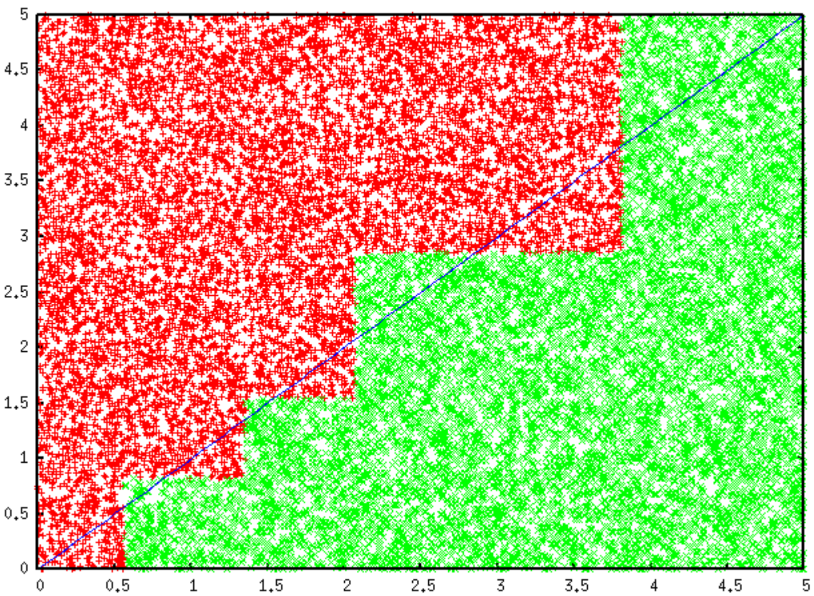
Why Ensemble Works? (2)



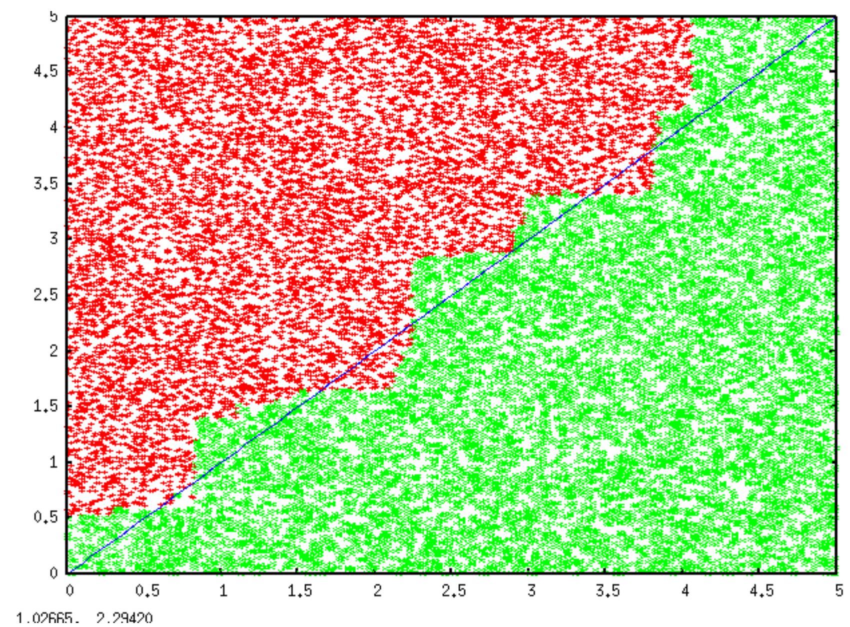
Ensemble gives the global picture!

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

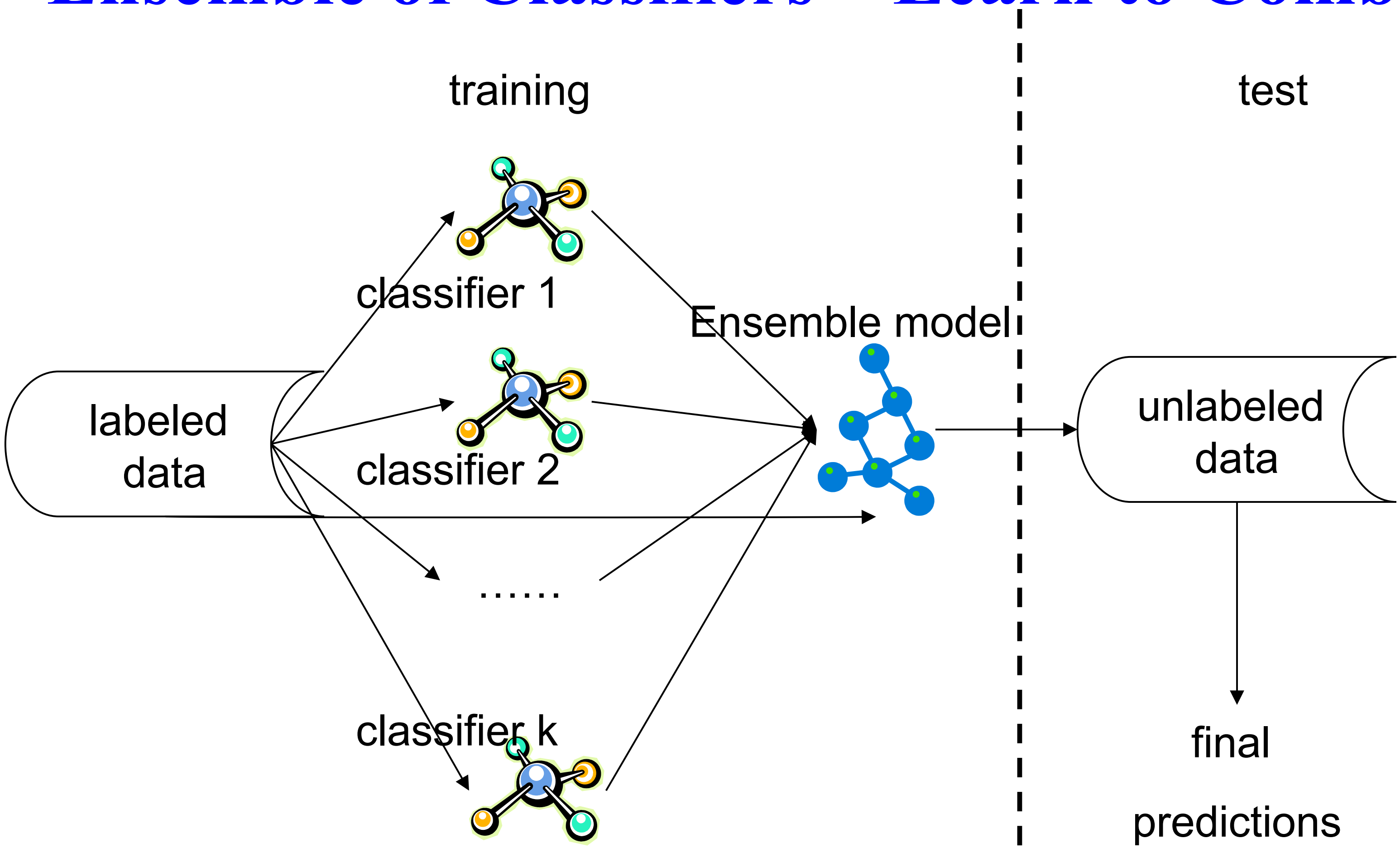
- Base models
  - Improve diversity!
- Combination scheme
  - Consensus (unsupervised)
  - Learn to combine (supervised)
- Tasks
  - Classification (supervised or semi-supervised ensemble)
  - Clustering (unsupervised ensemble)

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

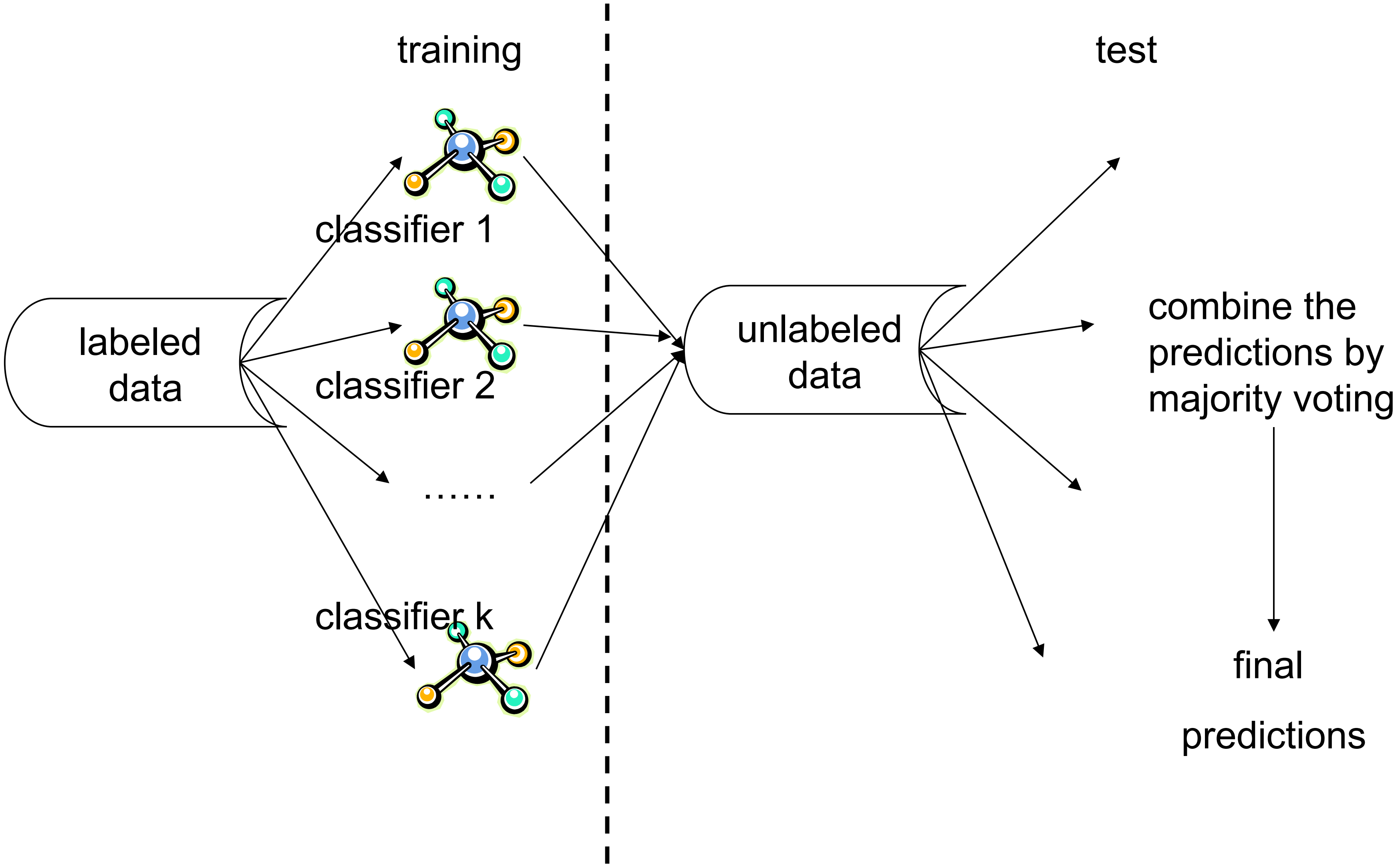
Ensemble of Classifiers—Learn to Combine



learn the combination from labeled data

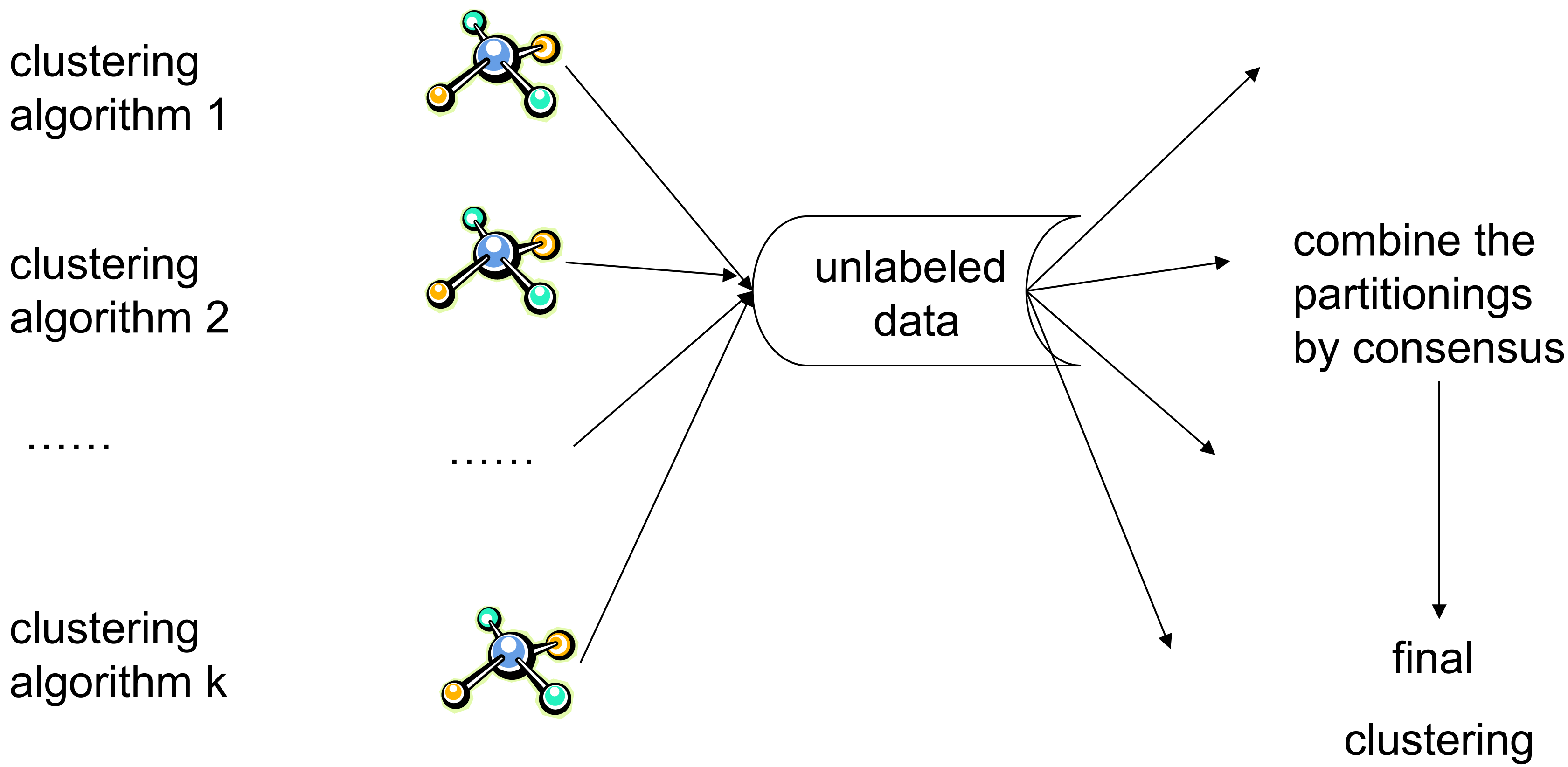
Algorithms: boosting, stacked generalization, rule ensemble, Bayesian model averaging……

Ensemble of Classifiers—Consensus



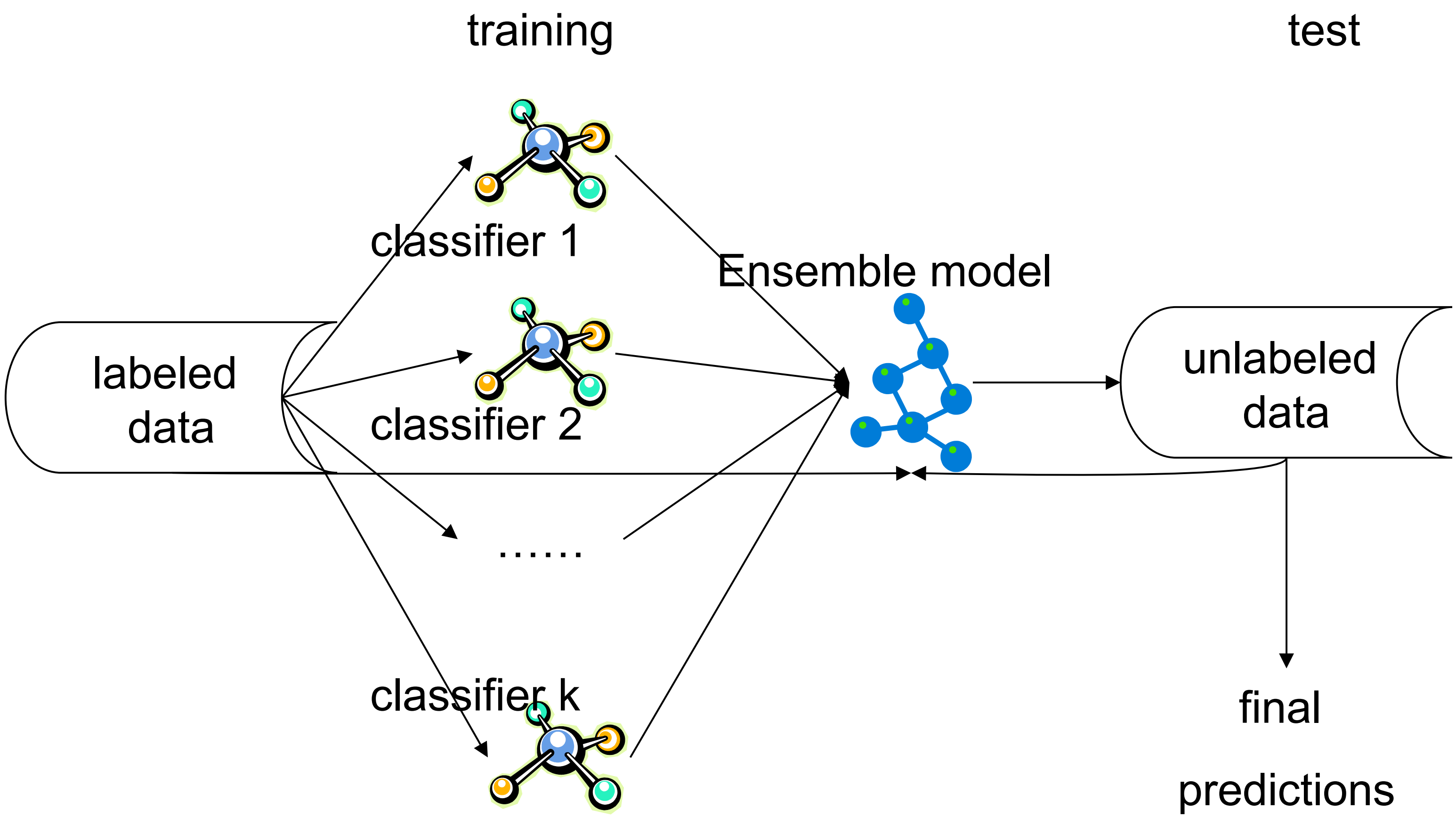
Algorithms: bagging, random forest, random decision tree, model averaging of probabilities……

Clustering Ensemble—Consensus



Algorithms: direct approach, object-based, cluster-based, object-cluster-based approaches, generative models

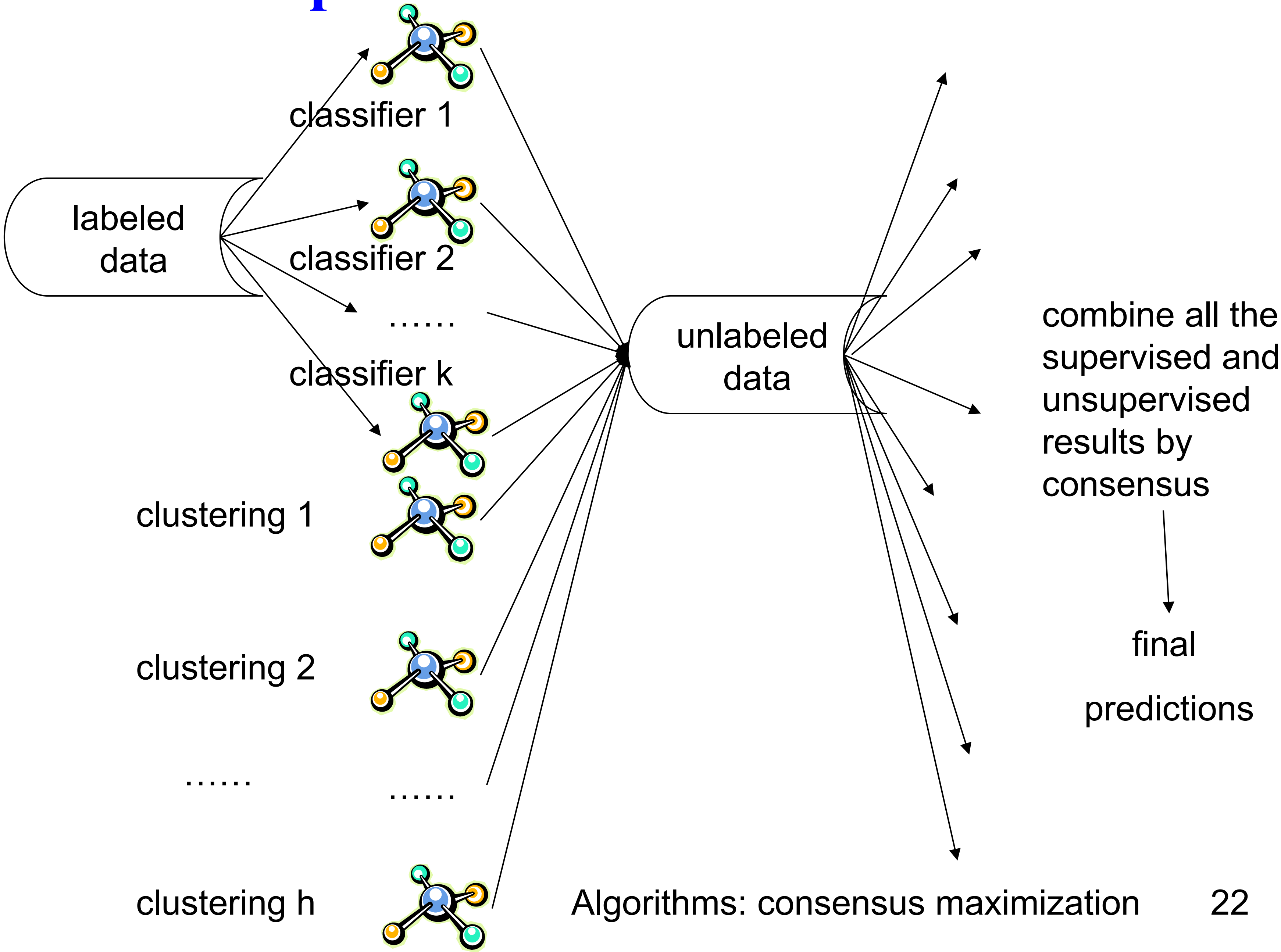
Semi-Supervised Ensemble—Learn to Combine



learn the combination from both
labeled and unlabeled data

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

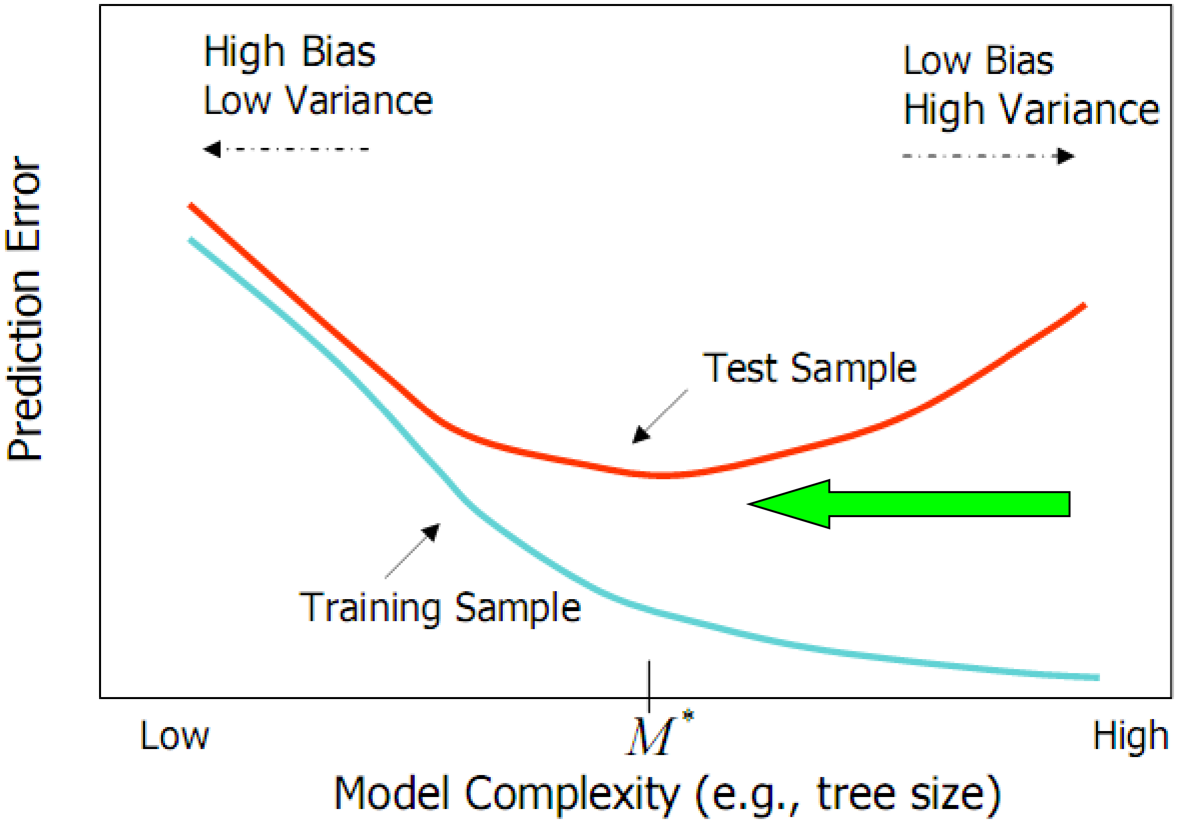
- An overview of ensemble methods
  - Motivations
  - Tutorial overview
- Supervised ensemble
- Unsupervised ensemble
- Semi-supervised ensemble
  - Multi-view learning
  - Consensus maximization among supervised and unsupervised models
- Applications
  - Transfer learning, stream classification, anomaly 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: $f^*(x)= w_1f_1(x)+ w_2f_2(x)+\ldots+ w_kf_k(x)$

Bias and Variance

- Ensemble methods
  - Combine 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 “randomly”
  - 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

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

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

  - 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]

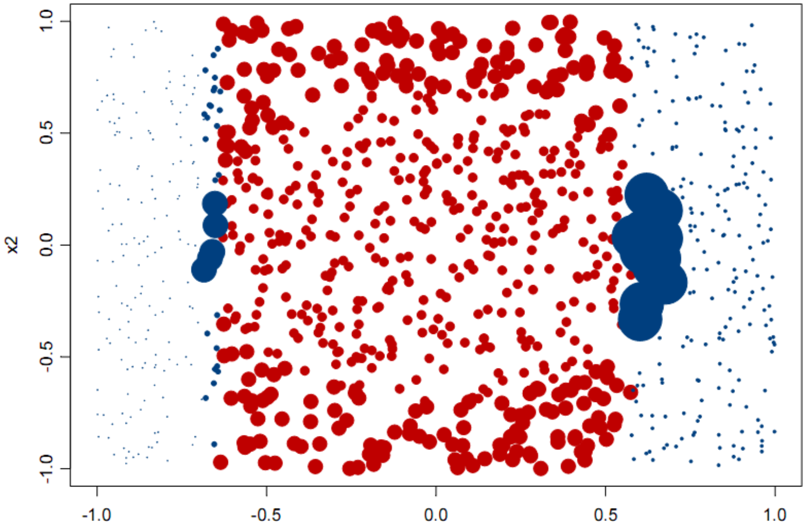
from P. Tan et al. Introduction to Data Mining.

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

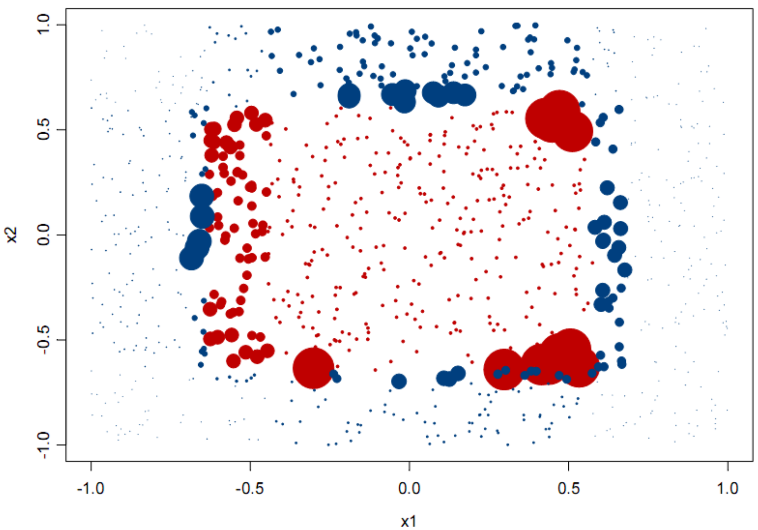
Boosting (3)

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

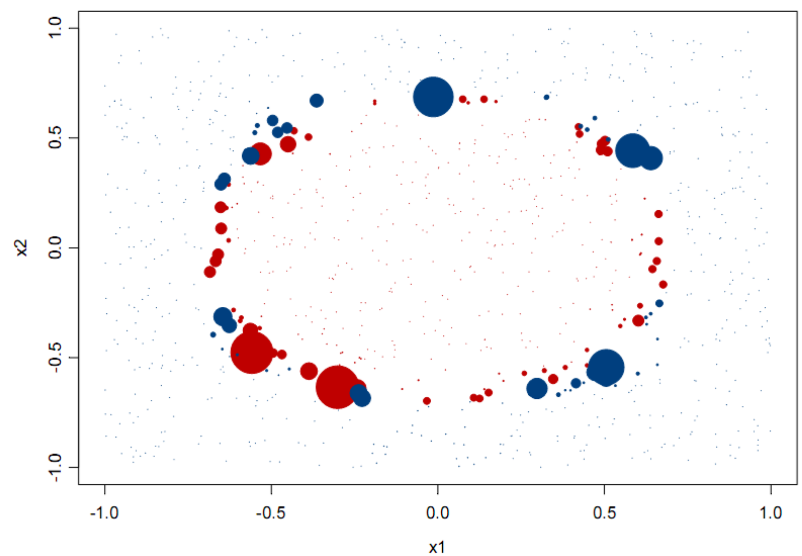


Classifications (colors) and Weights (size) after *1 iteration* Of AdaBoost

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 (typically 20%)
  - 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

Random Decision Tree* (1)

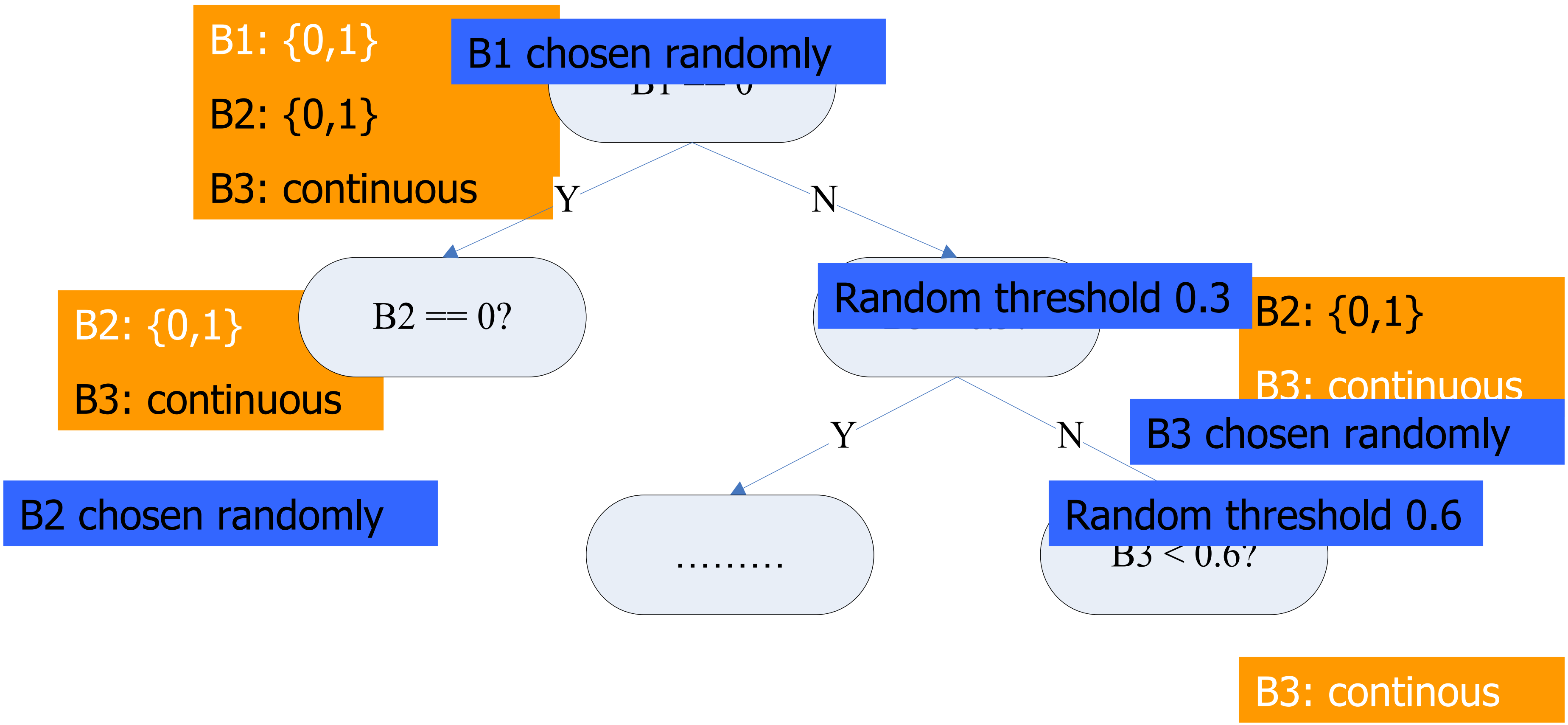
- Single-model learning algorithms
  - Fix structure of the model, minimize some form of errors, or maximize data likelihood (eg., Logistic regression, Naive Bayes, etc.)
  - Use some "free-form" functions to match the data given some "preference criteria" such as information gain, gini index and MDL. (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
- Learning as Encoding
  - 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

*[FWM+03]

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)

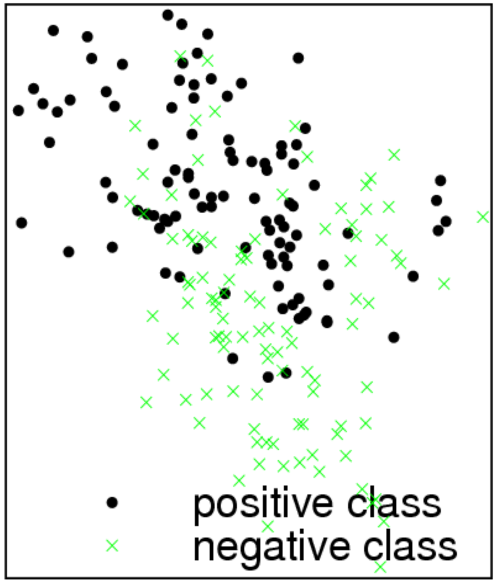


Random Decision Tree (4)

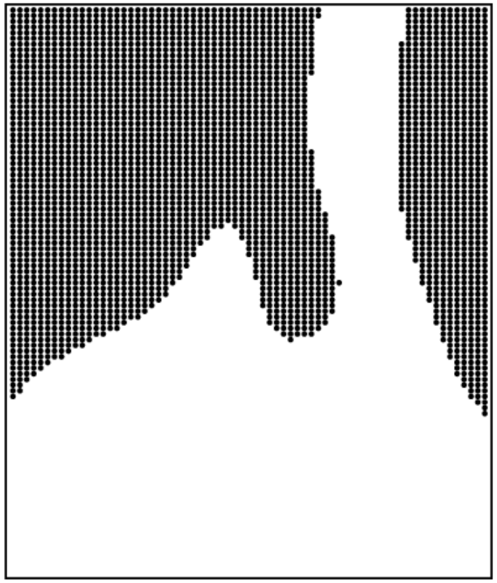
- Potential Advantages
  - Training can be very efficient. Particularly true for very large datasets.
    - No cross-validation based estimation of parameters for some parametric methods.
  - Natural multi-class probability.
  - Imposes very little about the structures of the model.

Optimal Decision Boundary

Figure 3.5: Gaussian mixture training samples and optimal boundary.

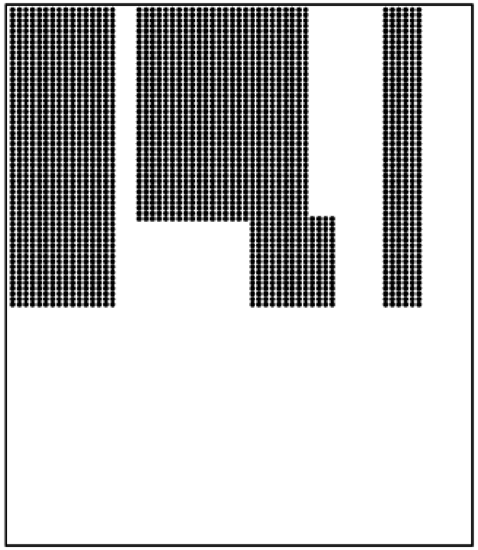


training samples

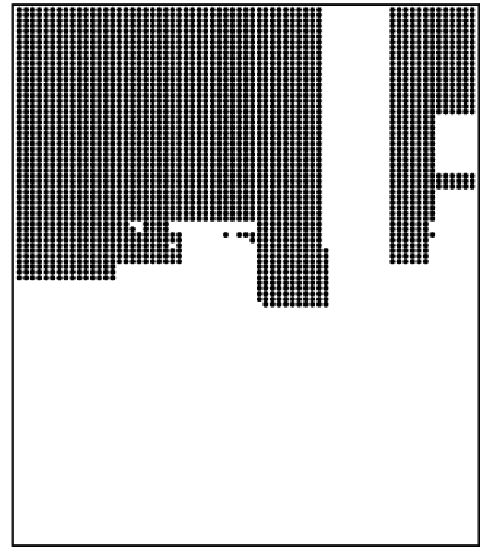


optimal boundary

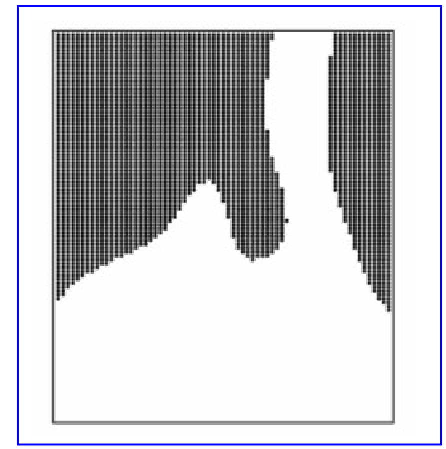
from Tony Liu's thesis (supervised by Kai Ming Ting)



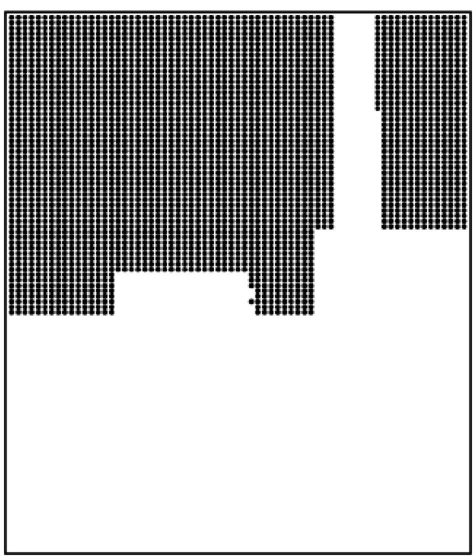
(a) unpruned C4.5



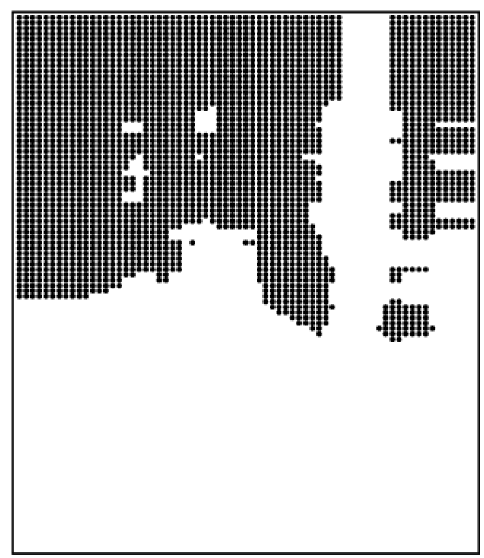
(b) Bagging



RDT looks like the optimal boundary



(c) Random Forests



(d) Complete-random tree ensemble

Outline

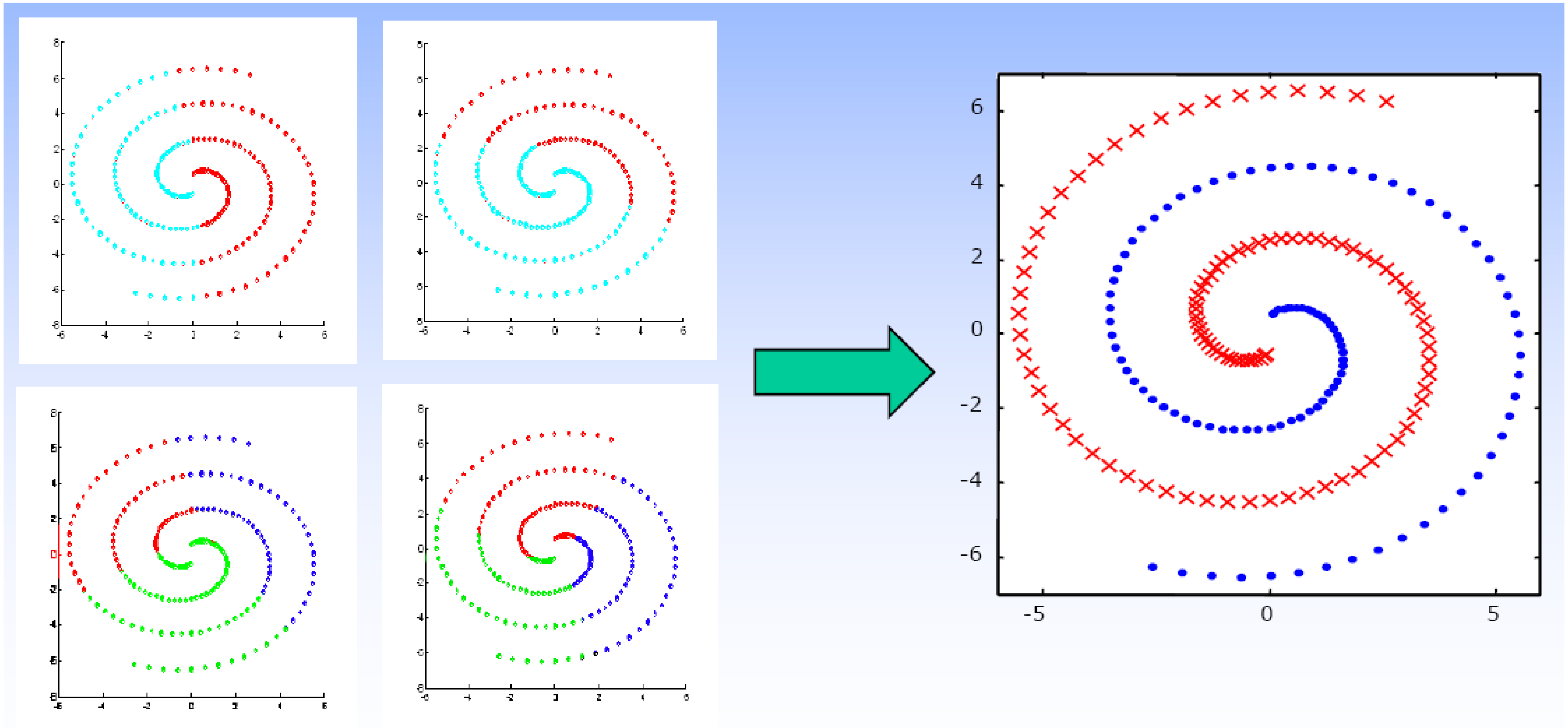
- An overview of ensemble methods
  - Motivations
  - Tutorial overview
- Supervised ensemble
- Unsupervised ensemble
- Semi-supervised ensemble
  - Multi-view learning
  - Consensus maximization among supervised and unsupervised models
- Applications
  - Transfer learning, stream classification, anomaly 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
- Challenges
  - The correspondence between the clusters in different clustering solutions is unknown
  - Unsupervised
  - Combinatorial optimization problem-NP-complete

Motivations

- Goal
  - Combine "weak" clusterings to a better one



[PTJ05]

An Example

base clustering models

objects

	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_3$	$\mathcal{C}$
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

they may not represent the same cluster!

The goal: get the consensus clustering

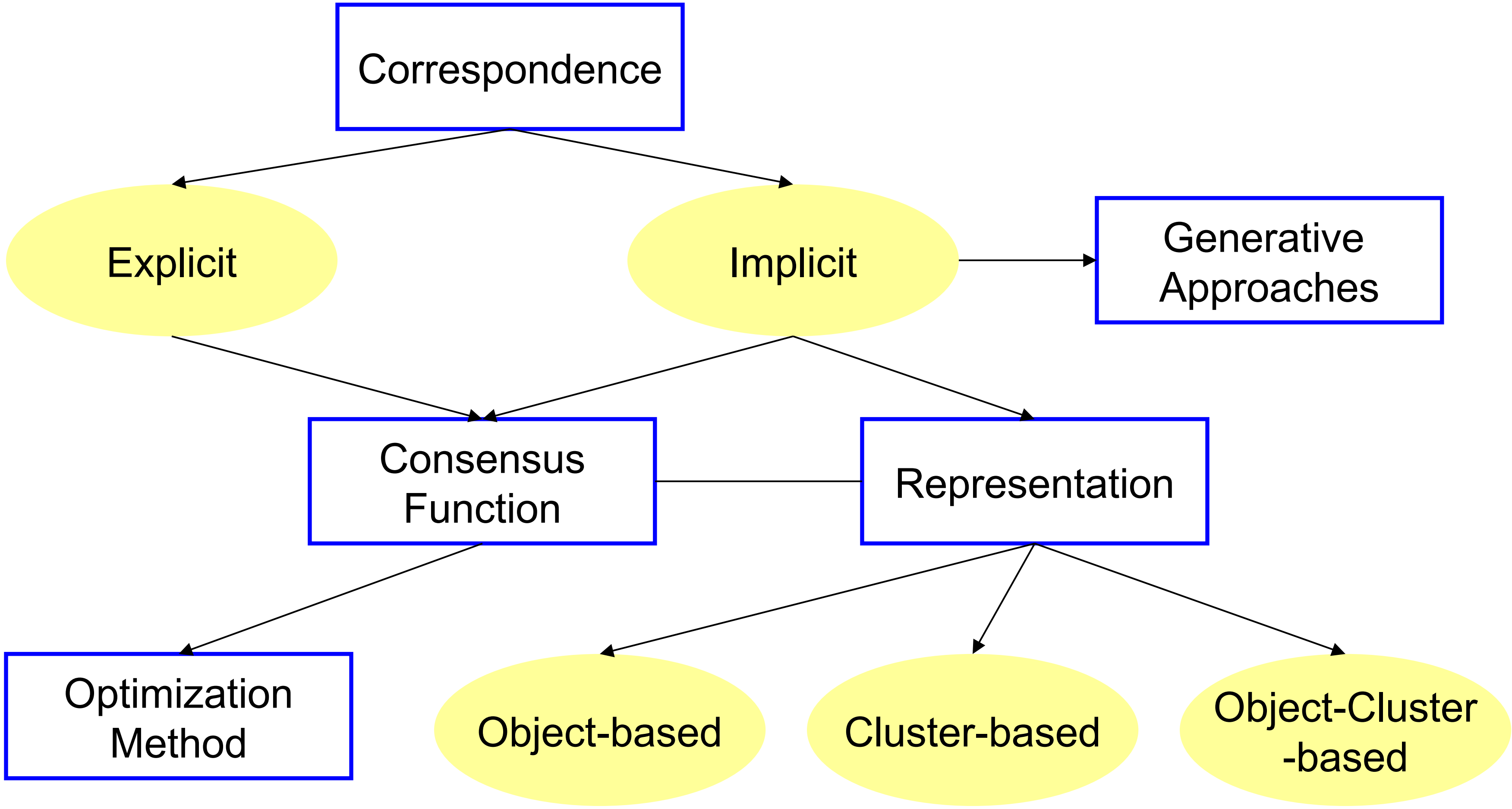
[GMT07]

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?



Hard Correspondence (1)

- Re-labeling+voting
  - Find the correspondence between the labels in the partitions and fuse the clusters with the same labels by voting [DuFr03,DWH01]

Re-labeling

	C_1	C_2	C_3
v_1	1	3	2
v_2	1	3	2
v_3	2	1	2
v_4	2	1	3
v_5	3	2	1
v_6	3	2	1

Voting

	C_1	C_2	C_3
v_1	1	1	1
v_2	1	1	1
v_3	2	2	1
v_4	2	2	2
v_5	3	3	3
v_6	3	3	3

C^*
1
1
2
2
3
3

Hard Correspondence (2)

- Details
  - Hungarian method to match clusters in two different clustering solutions
  - Match to a reference clustering or match in a pairwise manner
- Problems
  - In most cases, clusters do not have one-to-one correspondence

Soft Correspondence* (1)

- Notations
  - Membership matrix $M_1, M_2, \ldots, M_k$
  - Membership matrix of consensus clustering M
  - Correspondence matrix $S_1, S_2, \ldots, S_k$
  - $M_i S_i = M$

	C_1	C_2	C_3
v_1	1	3	2
v_2	1	3	2
v_3	2	1	2
v_4	2	1	3
v_5	3	2	1
v_6	3	2	1

$$\underset{M_2}{\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}} \times \underset{S_2}{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}} = \underset{M}{\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}}$$

*[LZY05]

Soft Correspondence (2)

- Consensus function
  - Minimize disagreement $\min \sum_{j=1}^{k} \| M - M_j S_j \|^2$
  - Constraint 1: column-sparseness
  - Constraint 2: each row sums up to 1
  - Variables: M, S_1, S_2, … ,S_k
- Optimization
  - EM-based approach
  - Iterate until convergence
    - Update S using gradient descent
    - Update M as $M = \frac{1}{k} \sum_{j=1}^{k} M_j S_j$

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

Thank You!

Shangsong Liang
Sun Yat-sen University