Fakultas Informatika
School of Computing
Telkom University

Machine Learning

Ensamble Method

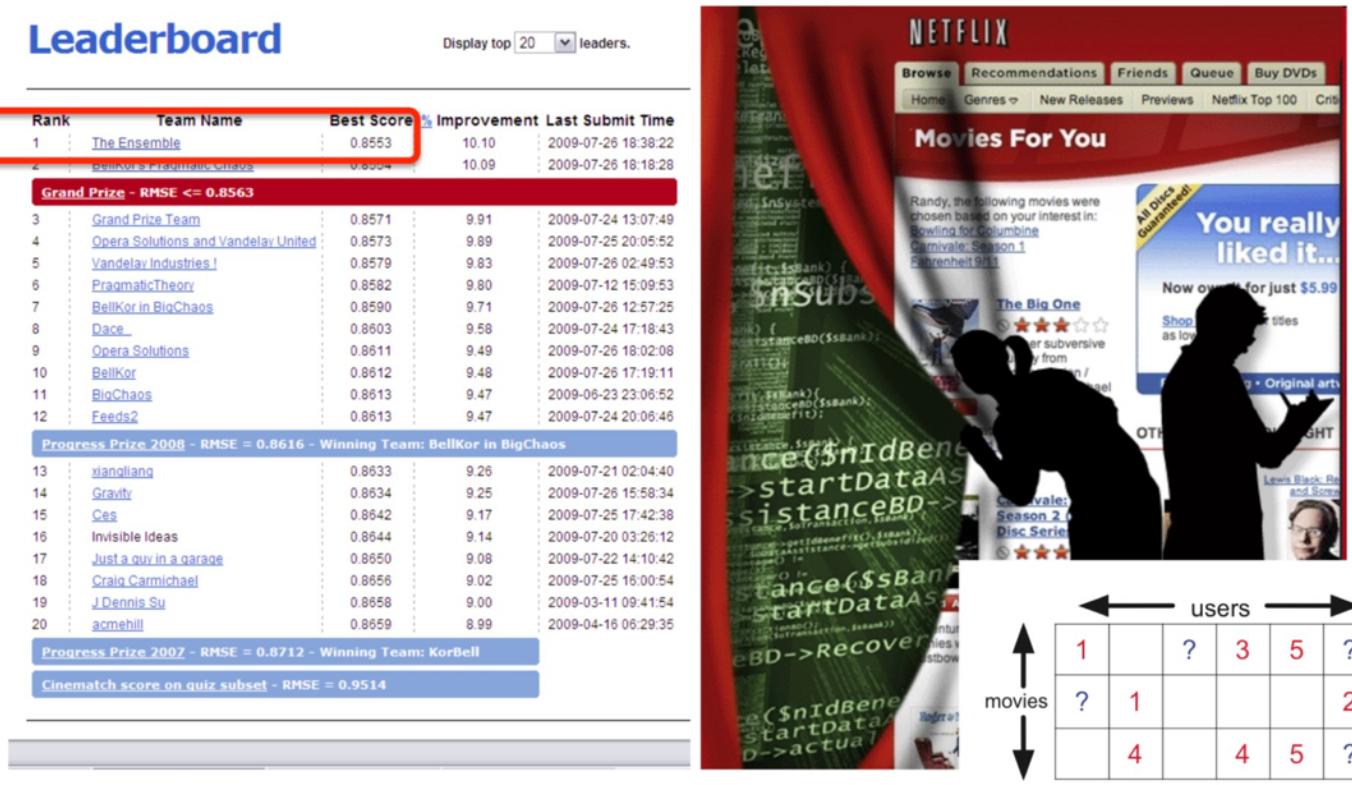
SFY

Outline

- Ensamble Methods
- Bias vs Variance Error
- Bagging
- Boosting
- Review Ensamble Methods

Ensamble Methods

Machine learning competition with a $1 million prize

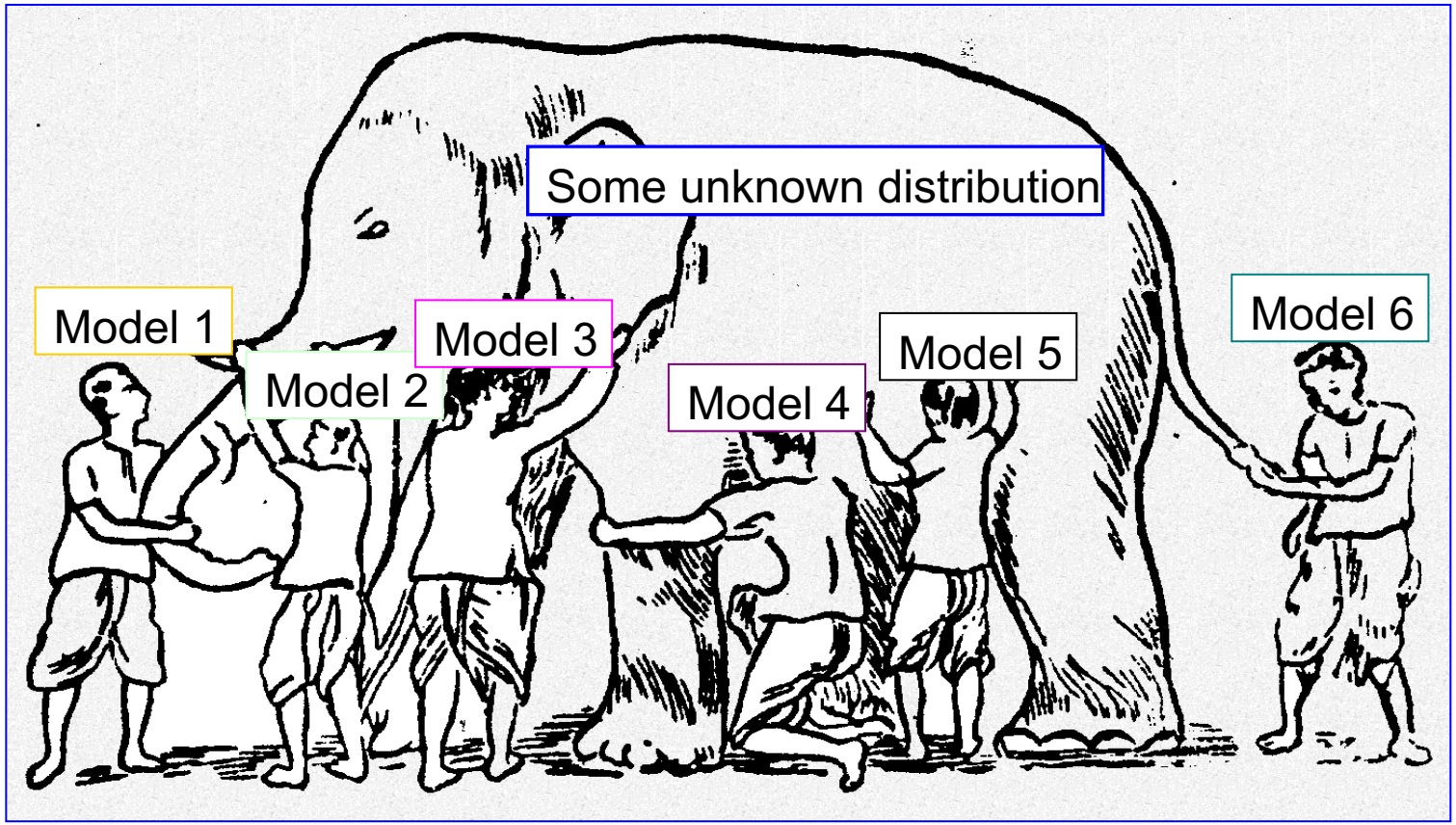


Leaderboard

Display top 20 leaders.

Rank	Team Name	Best Score	% Improvement	Last Submit Time
1	The Ensemble	0.8553	10.10	2009-07-26 18:38:22
2	BellKor's Pragmatic Chaos	0.8554	10.09	2009-07-26 18:18:28
	Grand Prize - RMSE <= 0.8563			
3	Grand Prize Team	0.8571	9.91	2009-07-24 13:07:49
4	Opera Solutions and Vandelay United	0.8573	9.89	2009-07-25 20:05:52
5	Vandelay Industries !	0.8579	9.83	2009-07-26 02:49:53
6	PragmaticTheory	0.8582	9.80	2009-07-12 15:09:53
7	BellKor in BigChaos	0.8590	9.71	2009-07-26 12:57:25
8	Dace	0.8603	9.58	2009-07-24 17:18:43
9	Opera Solutions	0.8611	9.49	2009-07-26 18:02:08
10	BellKor	0.8612	9.48	2009-07-26 17:19:11
11	BigChaos	0.8613	9.47	2009-06-23 23:06:52
12	Feeds2	0.8613	9.47	2009-07-24 20:06:46
	Progress Prize 2008 - RMSE = 0.8616 - Winning Team: BellKor in BigChaos			
13	xiangliang	0.8633	9.26	2009-07-21 02:04:40
14	Gravity	0.8634	9.25	2009-07-26 15:58:34
15	Ces	0.8642	9.17	2009-07-25 17:42:38
16	Invisible Ideas	0.8644	9.14	2009-07-20 03:26:12
17	Just a guy in a garage	0.8650	9.08	2009-07-22 14:10:42
18	Craig Carmichael	0.8656	9.02	2009-07-25 16:00:54
19	J Dennis Su	0.8658	9.00	2009-03-11 09:41:54
20	acmehill	0.8659	8.99	2009-04-16 06:29:35
	Progress Prize 2007 - RMSE = 0.8712 - Winning Team: KorBell			
	Cinematch score on quiz subset - RMSE = 0.9514			

Ensamble Methods

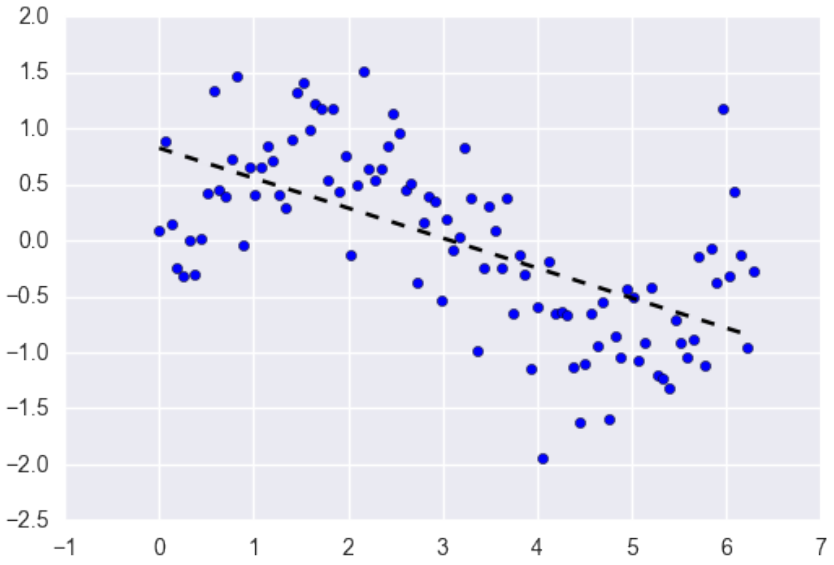


Ensamble Methods

- Ensemble methods are meta-algorithms that combine several machine learning techniques into one predictive model in order to :
  - **decrease variance**(bagging),
  - **decrease bias** (boosting)

Bias Error

- Seberapa jauh **perbedaan antara prediksi dan target**, terhadap data training.
- Biasanya karena model terlalu simple, sehingga tidak bisa menghasilkan prediksi yang baik.
- Ciri :
  - Error training : Tinggi
  - Error testing : Tinggi

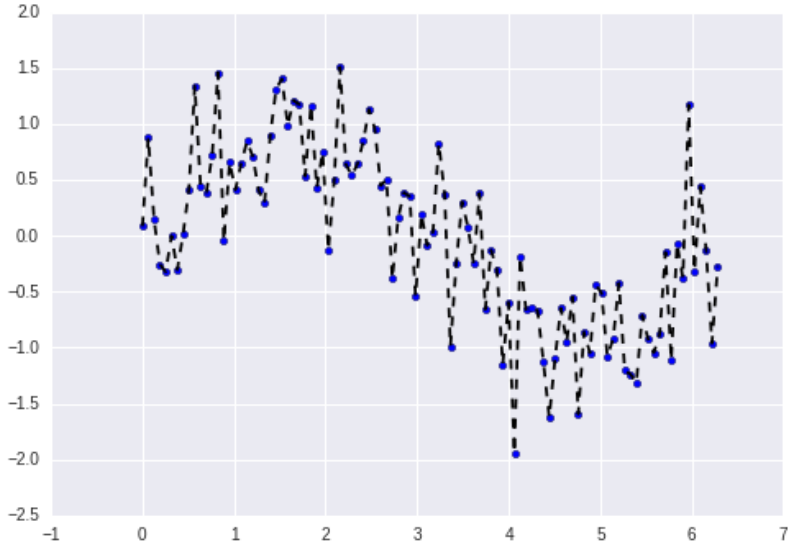


Bias Error

- Low bias
  - linear regression applied to linear data
  - 2nd degree polynomial applied to quadratic data
  - ANN with many hidden units trained to completion
- High bias
  - constant function
  - linear regression applied to non-linear data
  - ANN with few hidden units applied to non-linear data

Variance Error

- Seberapa jauh perbedaan antara prediksi dari model yang dibentuk dari **satu dataset training dengan dataset training lainnya**.
- Biasanya karena model terlalu kompleks, sehingga hanya cocok untuk data training saat ini. Jika data training diganti, maka model akan berubah drastis mengikuti data training yang baru.
- Ciri :
  - Error Training : Rendah
  - Error Testing : Tinggi



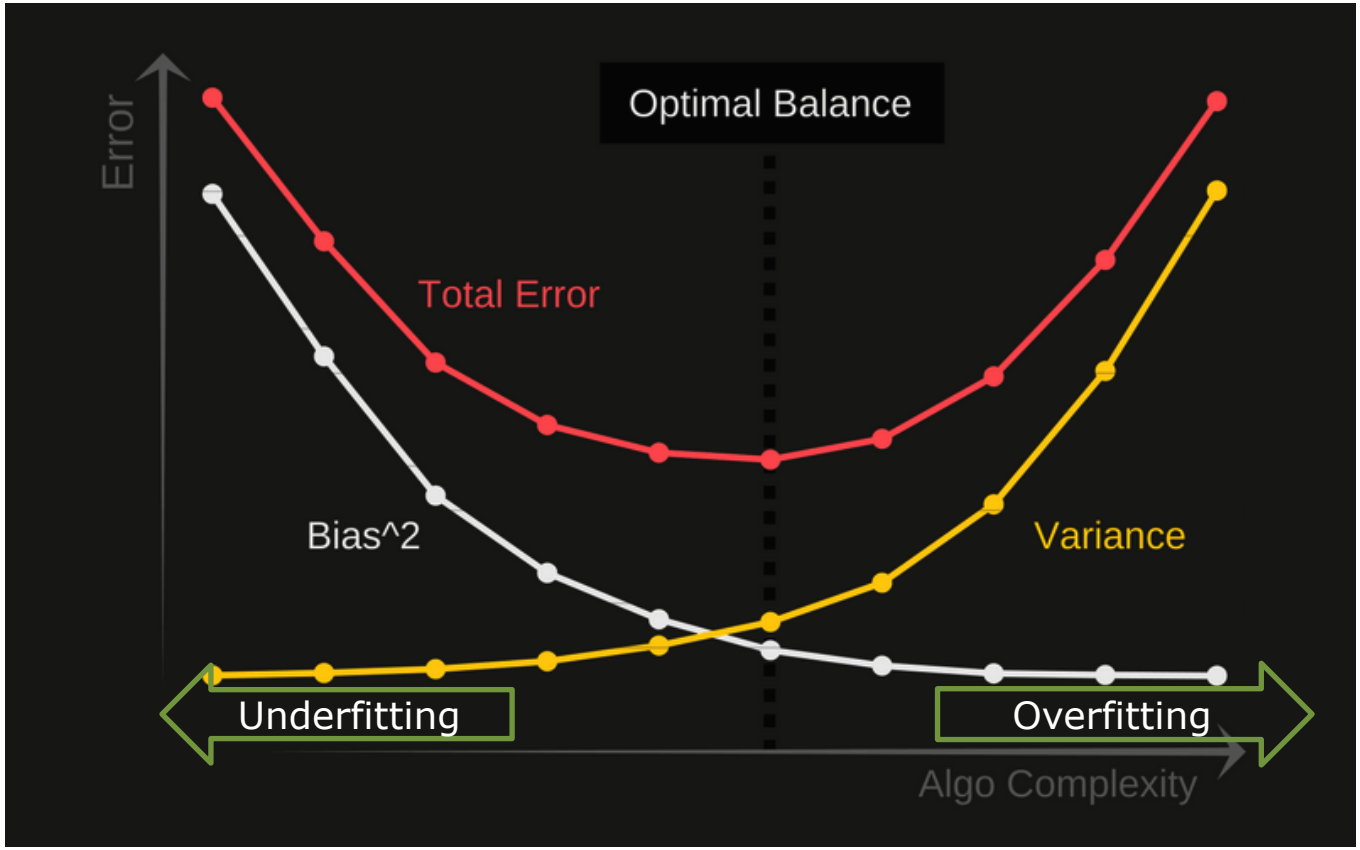
Variance Error

- Low variance
  - constant function
  - model independent of training data
  - model depends on stable measures of data
    - mean
    - median
- High variance
  - high degree polynomial
  - ANN with many hidden units trained to completion

Sources of Variance in Supervised Learning

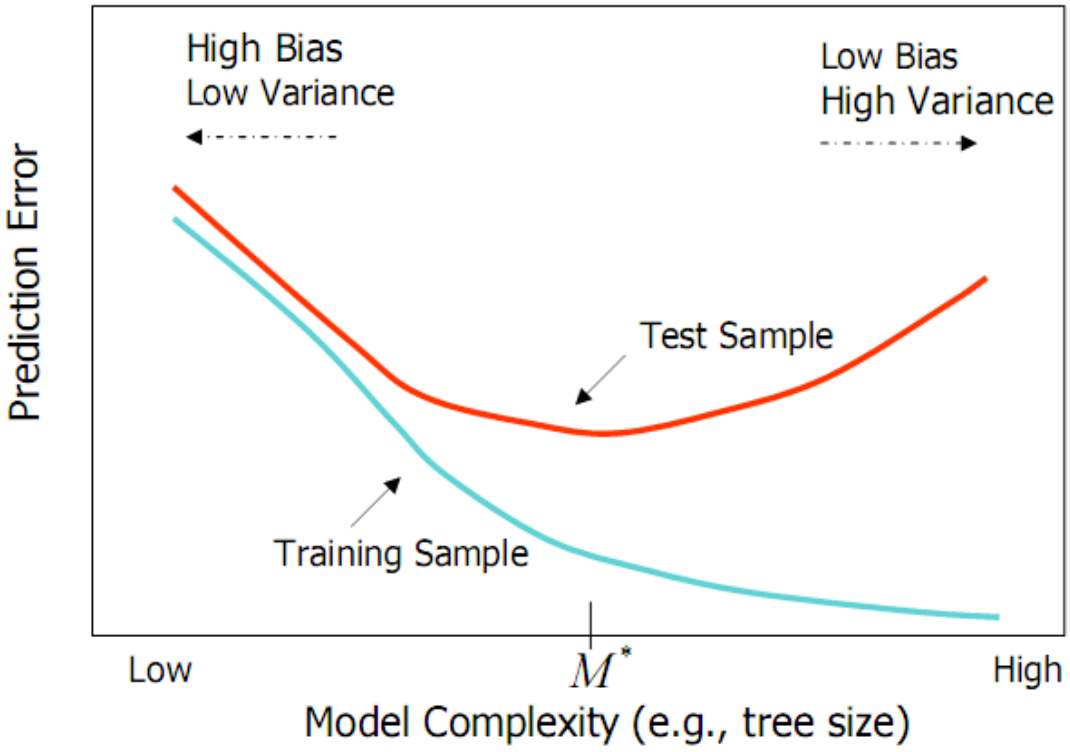
- noise in targets or input attributes
- bias (model mismatch)
- training sample
- randomness in learning algorithm
  - neural net weight initialization
- Randomized subsetting of trainset:
  - cross validation, train and early stopping set

Bias vs Variance Trade-off



Bias vs Variance Trade-off

- Menggunakan training error dan testing error untuk mengukur bias dan variance



Reduce Variance

- Cara termudah untuk mengurangi variance?
  - Constant predictions, hanya bisa mengoutputkan 1 jawaban apapun inputannya
  - Linear model walau datanya non-linear

- Model yang simple memang bisa **mengurangi *variance***, namun juga dapat **meningkatkan *bias***

Reduce Variance without Increasing Bias

- Averaging reduces variance
- So, Average models to reduce model variance !!
- Problem :
  - Only one training set
  - Where do multiple models come from ?

BAGGING

BAGGING (BootstrAp aGGregatING)

- Bootstrap
  - Create multiple sample sets/ datasets
  - Sampling **with replacement**
  - Contains around 63.2% original records in each sample

Training Data

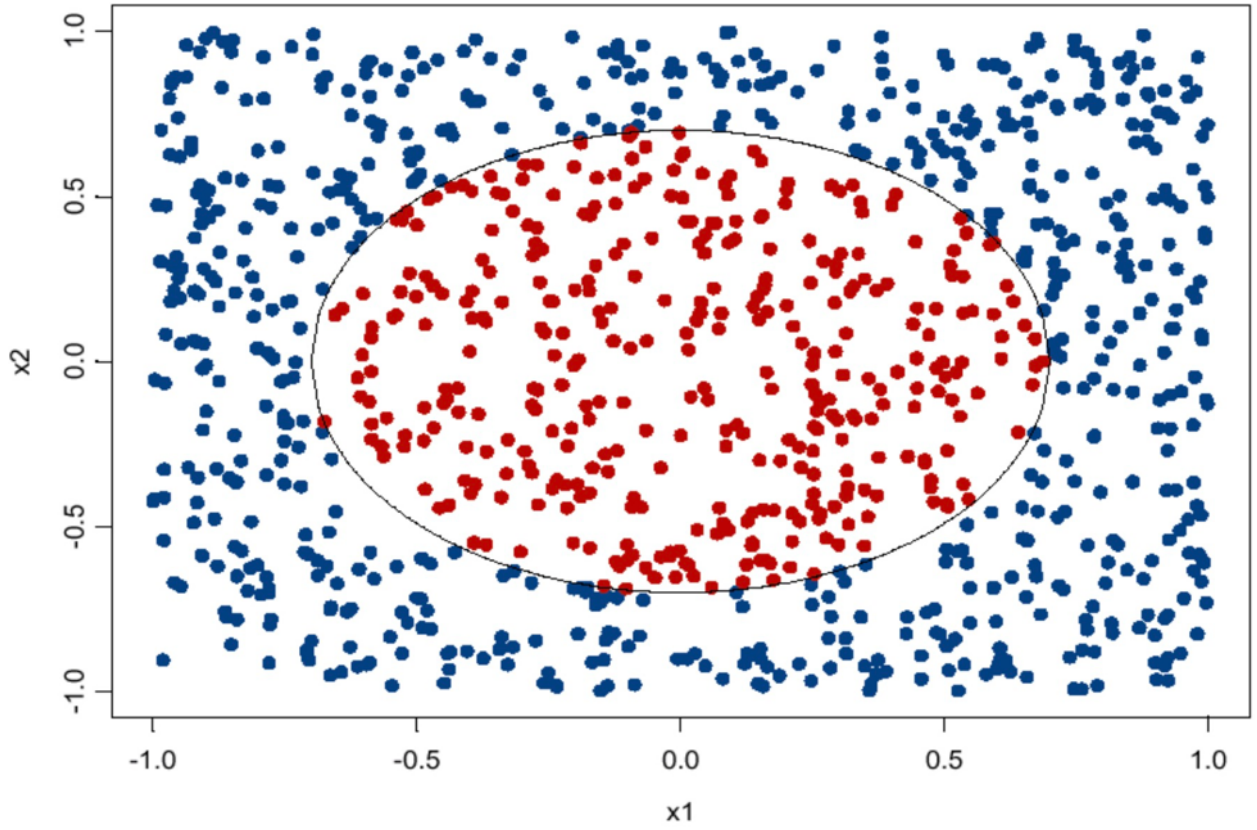
Data ID

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

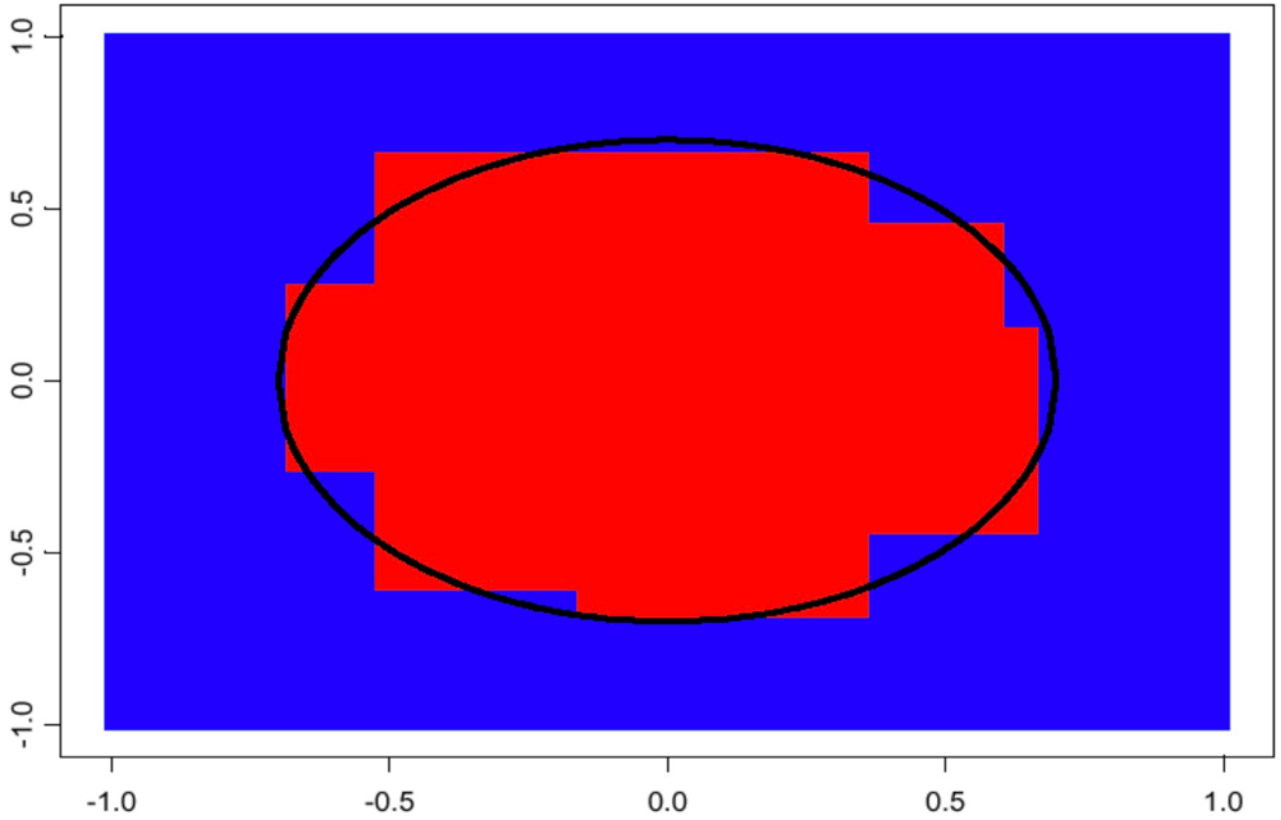
BAGGING

- Bootstrap Aggregating
  - Build classifier on each bootstrap sample
  - Use majority voting to determine the class label of ensemble classifier

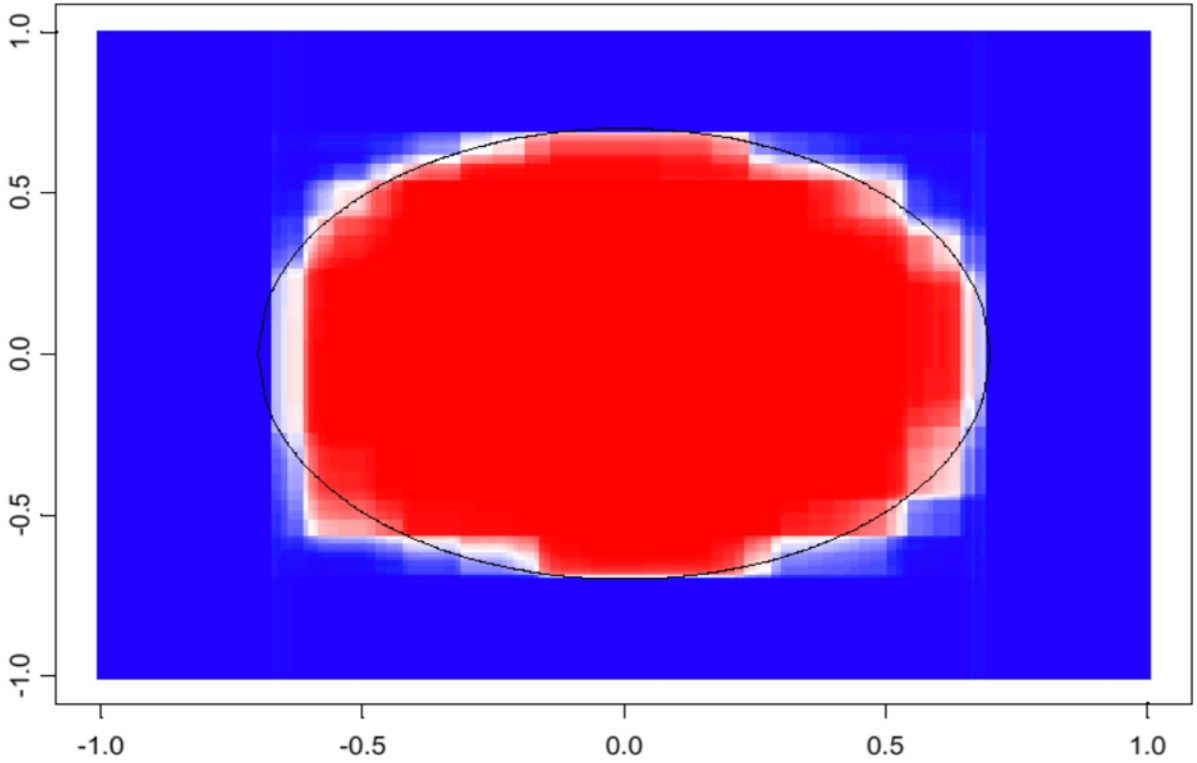
Bagging Example



Prediction by 1 Decision Tree



100 Bagged Tree



shades of blue/red indicate strength of vote for particular classification

Review Bagging

- Bagging reduces variance by averaging
- Bagging has little effect on bias
- Can we average *and* reduce bias?
- Yes:

BOOSTING

Boosting

- 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

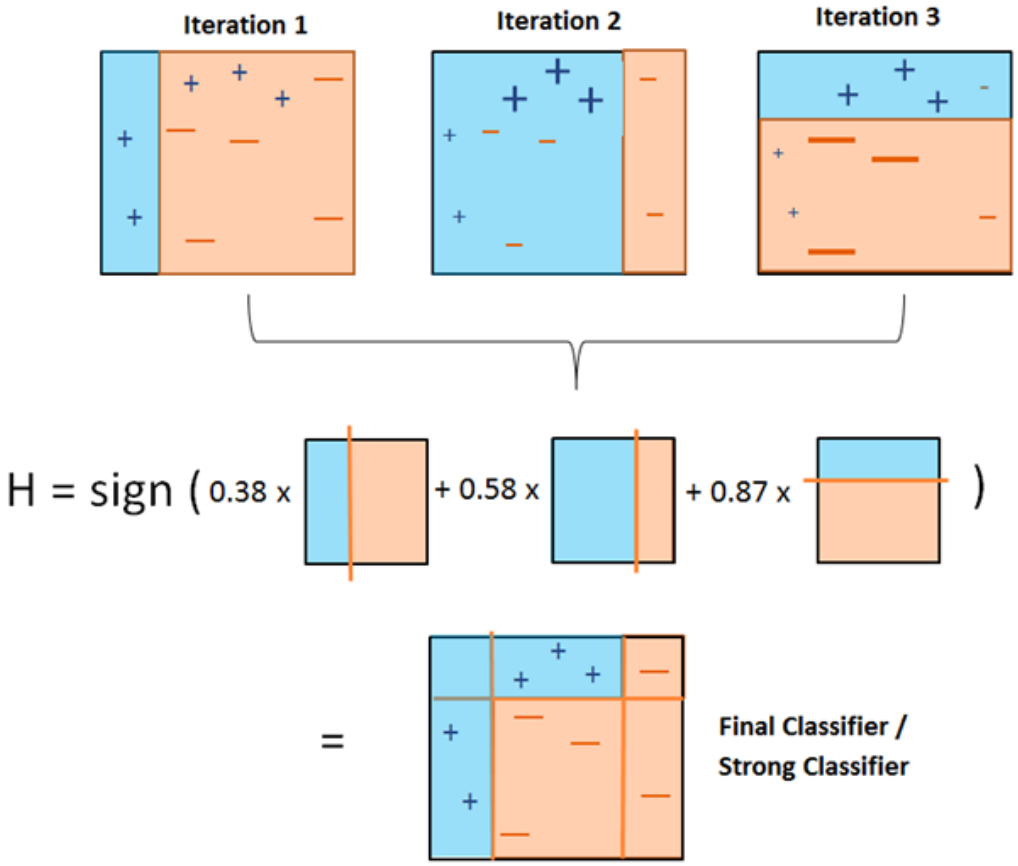
AdaBoost

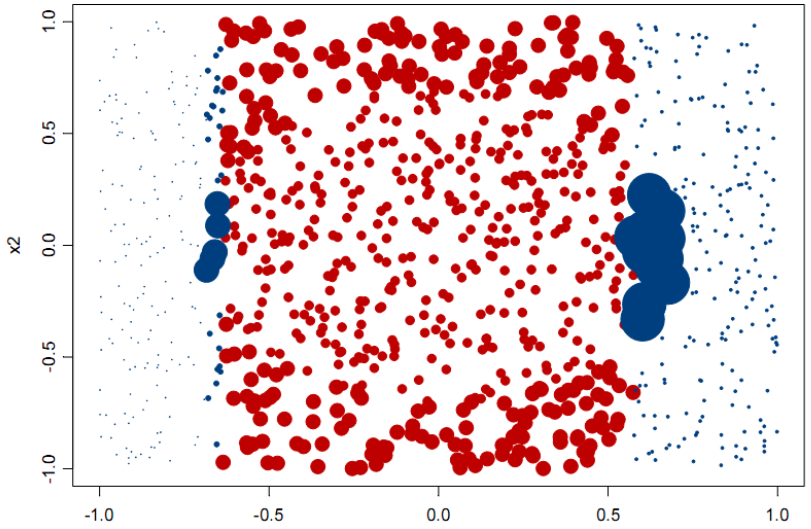
- Initially, set uniform weights on all the records
- At each round
  - Create a bootstrap sample based on the weights
  - Train a classifier on the sample and apply it on the original training set
  - Records that are wrongly classified will have their weights increased
  - Records that are classified 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

AdaBoost Algorithm

1. Initialize the observation weights
   $w_i = 1/N,\ i = 1, 2, \ldots, N.$
2. For $m = 1$ to M repeat steps (a)–(d):
   (a) Fit a classifier $G_m(x)$ to the training data using weights w_i.
   (b) Compute
   $$\text{err}_m = \frac{\sum_{i=1}^{N} w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^{N} w_i}.$$
   (c) Compute $\alpha_m = \log((1 - \text{err}_m)/\text{err}_m)$.
   (d) Update weights for $i = 1, \ldots, N$:
   $w_i \leftarrow w_i \cdot \exp[\alpha_m \cdot I(y_i \neq G_m(x_i))]$
   and renormalize to w_i to sum to 1.
3. Output $G(x) = \text{sign}\left[\sum_{m=1}^{M} \alpha_m G_m(x)\right]$.

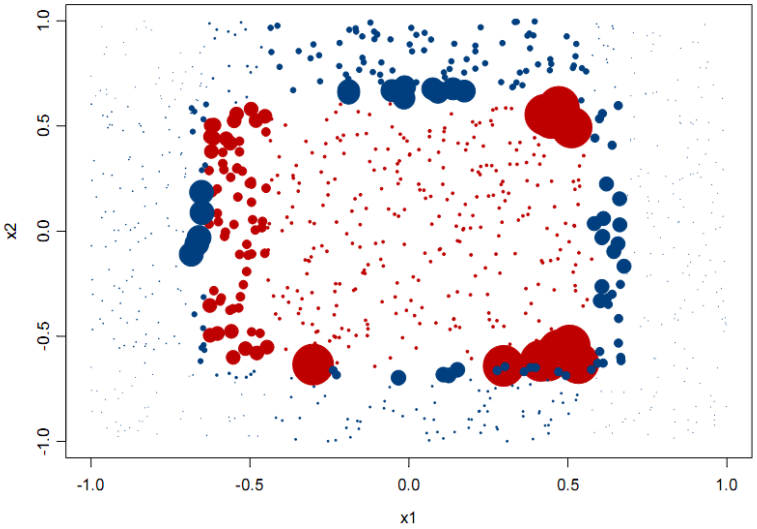
AdaBoost Classifier Working Principle with Decision Stump as a Base Classifier



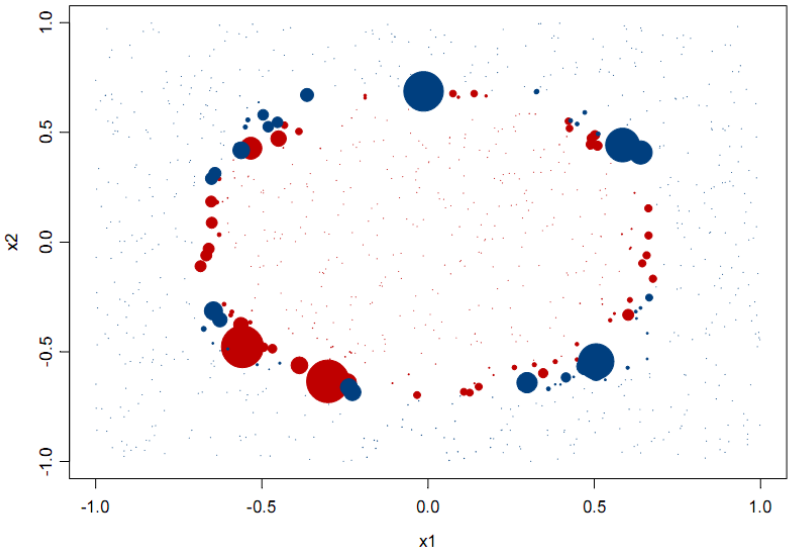


Classifications (colors) and Weights (size) after *1 iteration* Of AdaBoost

3 iterations



20 iterations

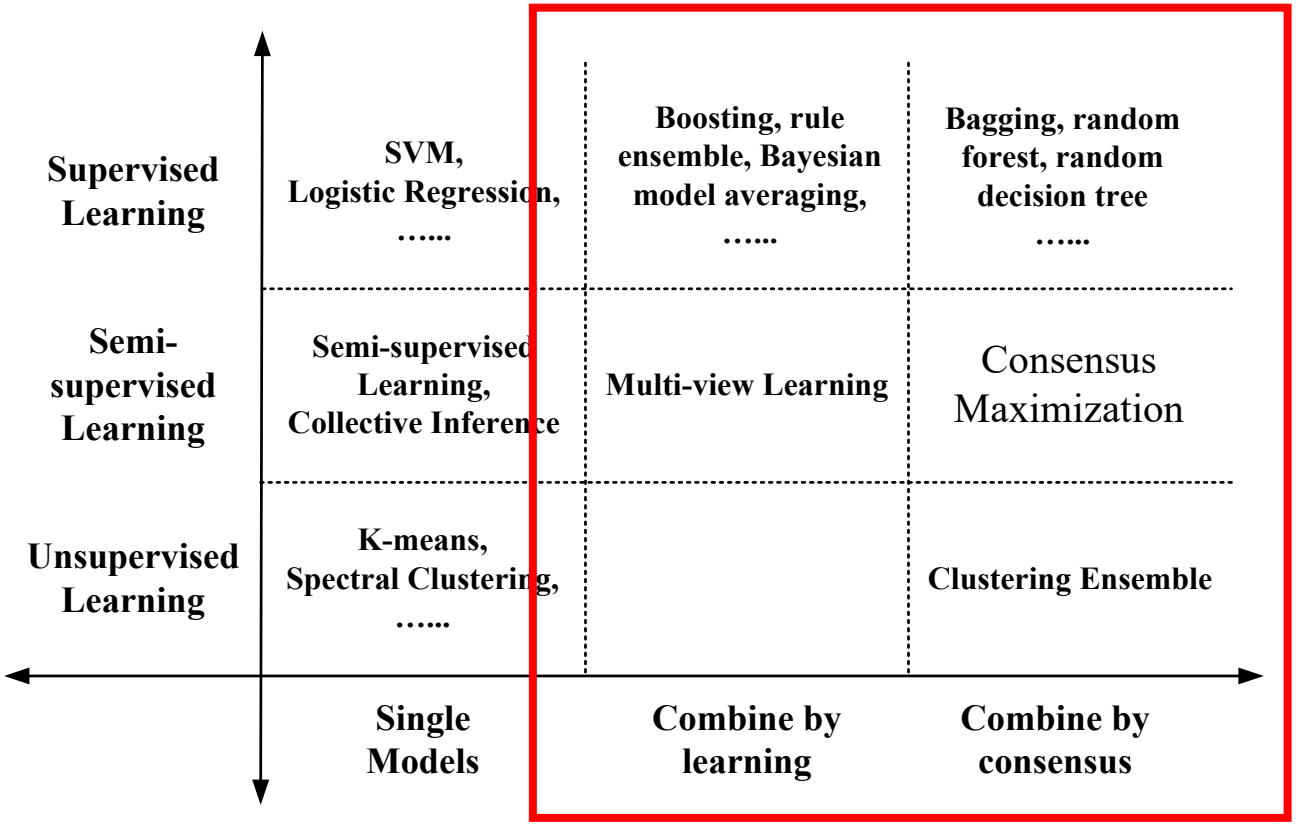


from Elder, John. From Trees to Forests and Rule Sets - A Unified Overview of Ensemble Methods. 2007.

Bagging vs Boosting

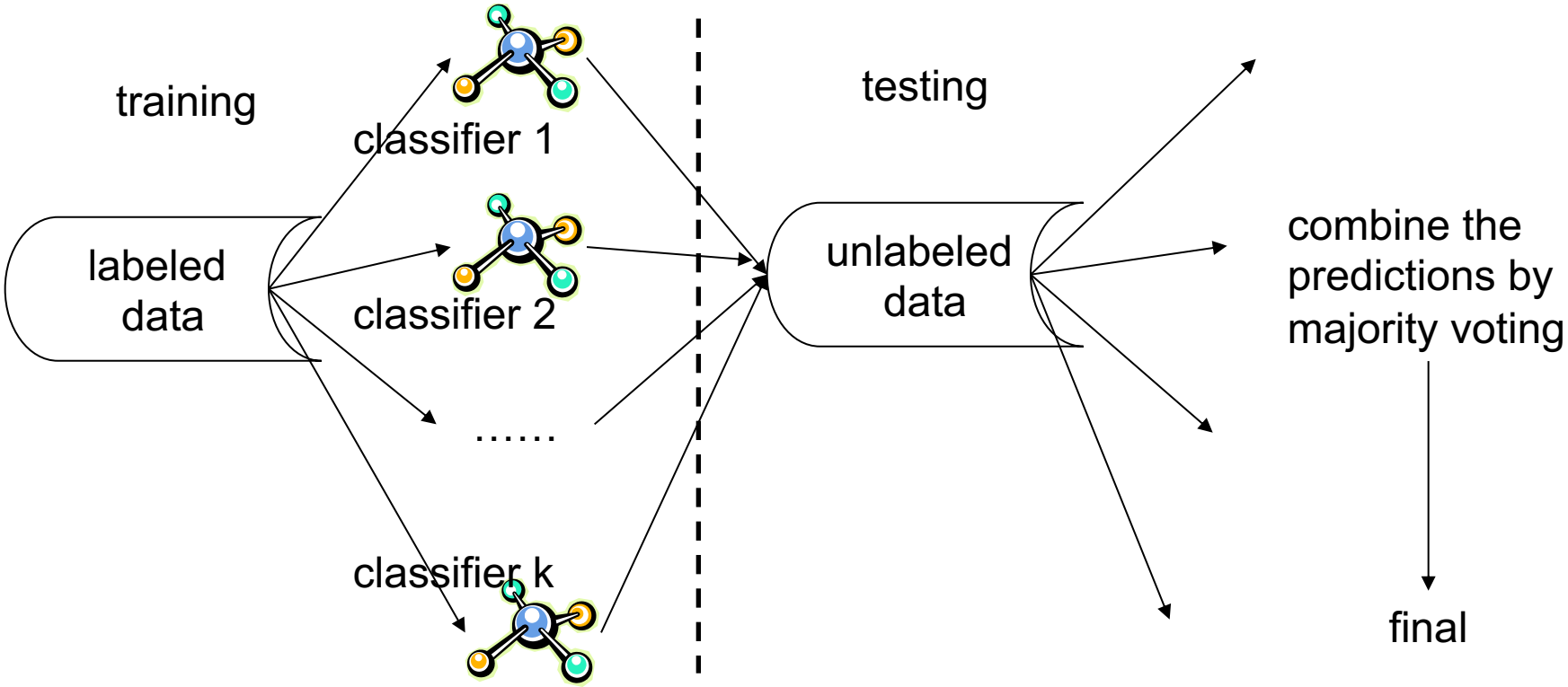
- ***Bagging***:
  - **parallel** ensemble: each model is built independently
  - aim to **decrease variance**, not bias
  - suitable for high variance low bias models (complex models)
  - an example of a tree based method is **random forest**, which develop fully grown trees (note that RF modifies the grown procedure to reduce the correlation between trees)
- ***Boosting***:
  - **sequential** ensemble: try to add new models that do well where previous models lack
  - aim to **decrease bias**
  - suitable for low variance high bias models
  - an example of a tree based method is **gradient boosting**

Review Ensamble Methods



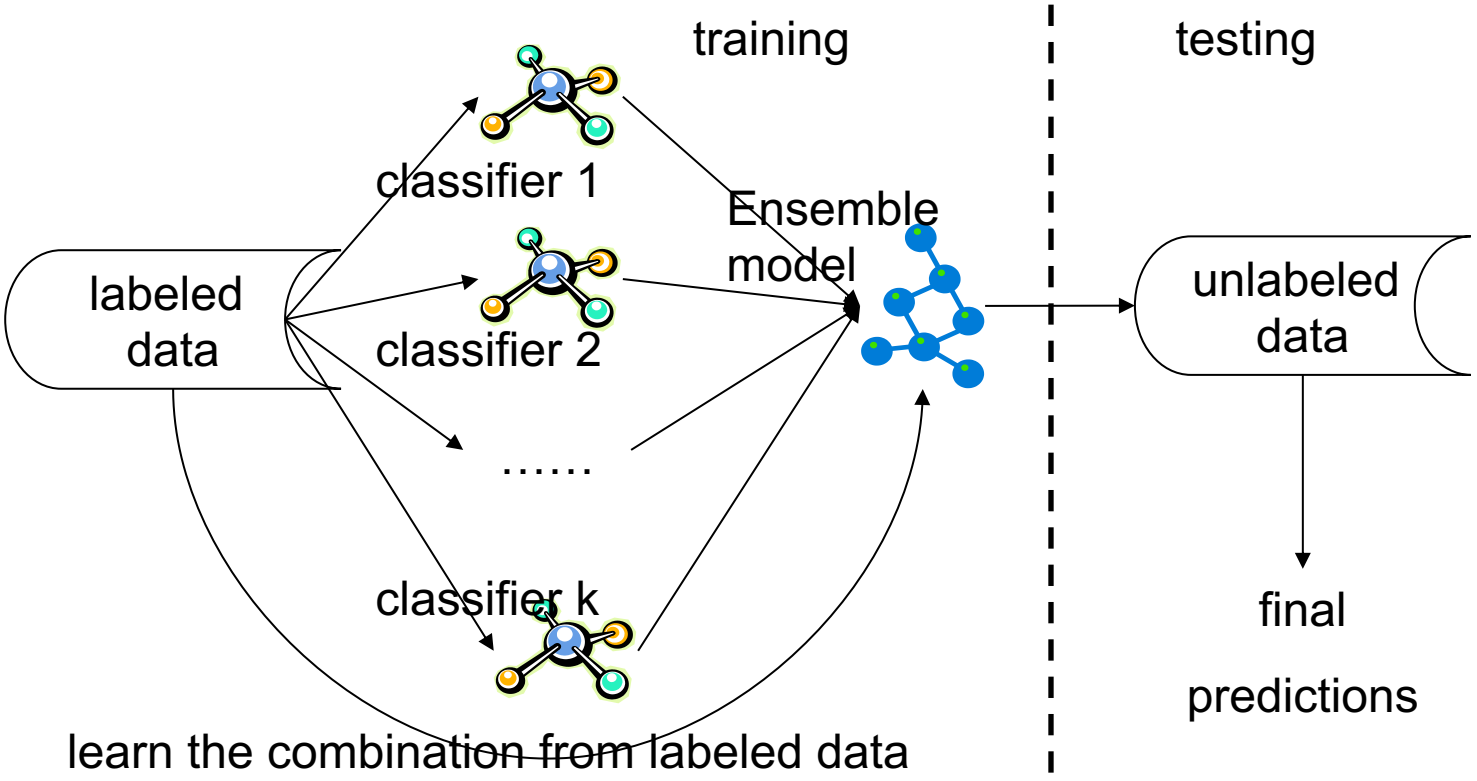
Ensamble Methods

Consensus



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

Learn to Combine



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

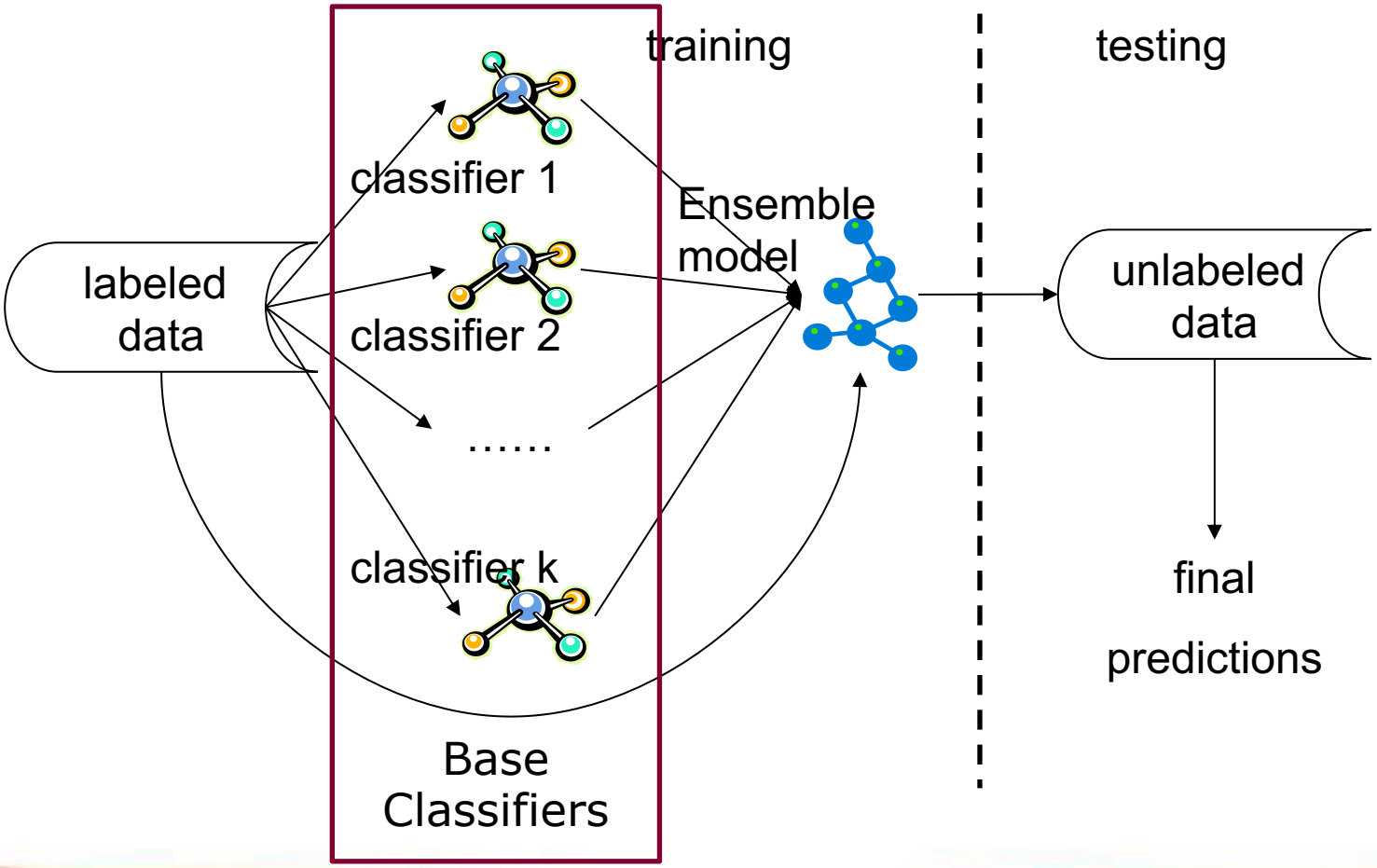
Pros and Cons

	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

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

Base Classifiers



Supervised Ensambled Methods

- Bagging (voting for classification, averaging for prediction)
- Boosting

Next Week :

- Random Forests
- Random Decision Tree

Question?

Fakultas Informatika
School of Computing
Telkom University

THANK YOU