Week 9-1

Machine Learning 3: Ensembles, Evaluations

Big Data

Prof. Hwanjo Yu
POSTECH

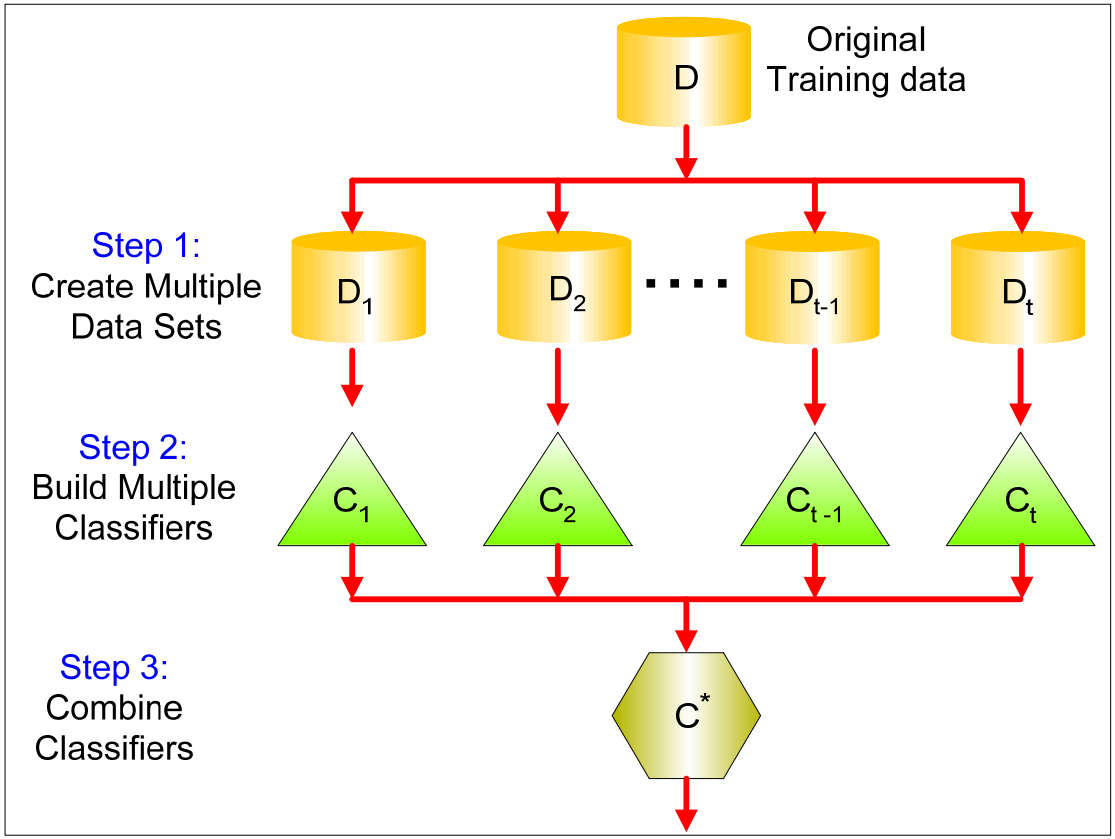
POSTECH
POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY

Ensemble Methods

Big Data

Ensemble methods

- Construct a set of base classifiers from the training data
- Predict class label of a testing instance by taking a majority vote on the predictions made by the base classifiers



Why does it work?

- Suppose there are 25 base classifiers
  - Each classifier has error rate, $\varepsilon = 0.35$
  - If the base classifiers are identical, then the ensemble will misclassify the same examples predicted incorrectly by the base classifiers.
  - Assume classifiers are independent, i.e. their errors are uncorrelated. Then the ensemble makes a wrong prediction only if more than half of the base classifiers predict incorrectly.
  - Probability that the ensemble classifier makes a wrong prediction:

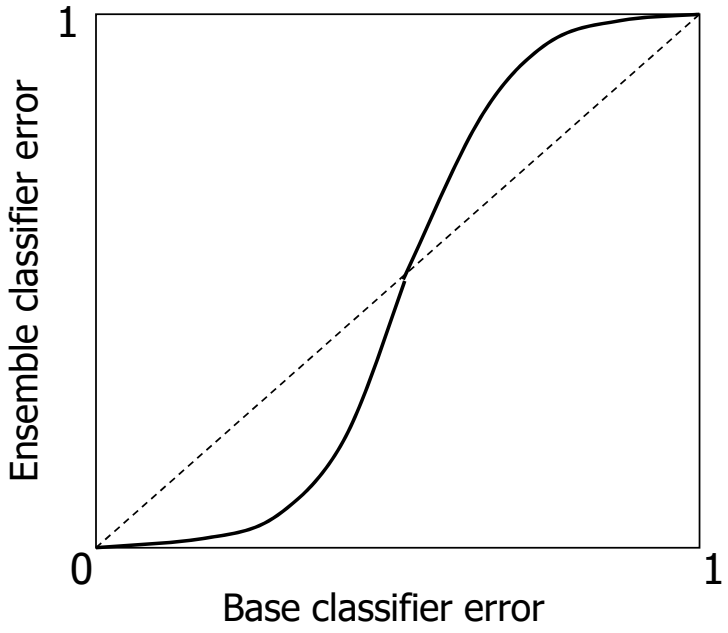
$$\sum_{i=13}^{25} \binom{25}{i} \varepsilon^i (1-\varepsilon)^{25-i} = 0.06$$

Ensemble methods

- The diagonal line is when the base classifiers are identical.
- The solid line is when they are independent.

Conditions for Ensemble Methods

- The base classifiers should be independent.
- The base classifiers should do better than a classifier that performs random guessing. (error < 0.5)
- In practice, it is hard to have base classifiers perfectly independent. Nevertheless, improvements have been observed in ensemble methods when they are slightly correlated.



Ensemble methods

- How to generate an ensemble of classifiers?
  - Bagging
  - Boosting
  - Random Forests

Bagging

- Sampling (with replacement) according to a uniform probability distribution
  - Each *bootstrap sample D* has the same size as the original data.
  - Some instances could appear several times in the same training set, while others may be omitted.

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

- Build classifier on each bootstrap sample D
- D will contain approximately 63% of the original data.
- Each data object has probability $1- (1 – 1/n)^n$ of being selected in D

Bagging – Example

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

- Decision stump: one-level binary decision tree, with a test condition $x \leq k$.
- Without bagging, the split will be either $x \leq 0.35$ or $x \leq 0.75$, with accuracy of 70% at most.

Bagging – Example

10 bootstrap samples

X	0.1	0.2	0.2	0.3	0.4	0.4	0.5	0.6	0.9	0.9	x <= 0.35 → y = 1
Y	1	1	1	1	-1	-1	-1	-1	1	1	x > 0.35 → y = -1
X	0.1	0.2	0.3	0.4	0.5	0.8	0.9	1	1	1	x <= 0.65 → y = 1
Y	1	1	1	-1	-1	1	1	1	1	1	x > 0.65 → y = -1
X	0.1	0.2	0.3	0.4	0.4	0.5	0.7	0.7	0.8	0.9	x <= 0.35 → y = 1
Y	1	1	1	-1	-1	-1	-1	-1	1	1	x > 0.35 → y = -1
X	0.1	0.1	0.2	0.4	0.4	0.5	0.5	0.7	0.8	0.9	x <= 0.3 → y = 1
Y	1	1	1	-1	-1	-1	-1	-1	1	1	x > 0.3 → y = -1
X	0.1	0.1	0.2	0.5	0.6	0.6	0.6	1	1	1	x <= 0.35 → y = 1
Y	1	1	1	-1	-1	-1	-1	1	1	1	x > 0.35 → y = -1
X	0.2	0.4	0.5	0.6	0.7	0.7	0.7	0.8	0.9	1	x <= 0.75 → y = 1
Y	1	-1	-1	-1	-1	-1	-1	1	1	1	x > 0.75 → y = -1
X	0.1	0.4	0.4	0.6	0.7	0.8	0.9	0.9	0.9	1	x <= 0.75 → y = 1
Y	1	-1	-1	-1	-1	1	1	1	1	1	x > 0.75 → y = -1
X	0.1	0.2	0.5	0.5	0.5	0.7	0.7	0.8	0.9	1	x <= 0.75 → y = 1
Y	1	1	-1	-1	-1	-1	-1	1	1	1	x > 0.75 → y = -1
X	0.1	0.3	0.4	0.4	0.6	0.7	0.7	0.8	1	1	x <= 0.75 → y = 1
Y	1	1	-1	-1	-1	-1	-1	1	1	1	x > 0.75 → y = -1
X	0.1	0.1	0.1	0.1	0.3	0.3	0.8	0.8	0.9	0.9	x <= 0.05 → y = 1
Y	1	1	1	1	1	1	1	1	1	1	x > 0.05 → y = -1

Bagging – Example

Voting result

Round	x=0.1	x=0.2	x=0.3	x=0.4	x=0.5	x=0.6	x=0.7	x=0.8	x=0.9	x=1
1	1	1	1	-1	-1	-1	-1	-1	-1	-1
2	1	1	1	1	1	1	1	1	1	1
3	1	1	1	-1	-1	-1	-1	-1	-1	-1
4	1	1	1	-1	-1	-1	-1	-1	-1	-1
5	1	1	1	-1	-1	-1	-1	-1	-1	-1
6	-1	-1	-1	-1	-1	-1	-1	1	1	1
7	-1	-1	-1	-1	-1	-1	-1	1	1	1
8	-1	-1	-1	-1	-1	-1	-1	1	1	1
9	-1	-1	-1	-1	-1	-1	-1	1	1	1
10	1	1	1	1	1	1	1	1	1	1
Sum	2	2	2	-6	-6	-6	-6	2	2	2
Y	**1**	**1**	**1**	**-1**	**-1**	**-1**	**-1**	**1**	**1**	**1**
Correct	1	1	1	1	1	1	1	1	1	1

Bagging

- The performance of bagging depends on the stability of the base classifier.
  - If a base classifier is unstable, bagging helps to reduce the errors.
  - If a base classifier is stable, bagging may not be able to improve, rather it could degrade the performance.
- Bagging is less susceptible to model overfitting when applied to noisy data.

Boosting

- An iterative procedure to adaptively change distribution of training data by focusing more on previously misclassified records
  - Initially, all N records are assigned equal weights
  - Unlike bagging, weights may change at the end of boosting round
- Weights can be used
  1. As a sampling distribution to draw a set of bootstrap samples from the original data
  2. By the base classifier to learn a model that is biased toward higher-weight examples

Boosting

- Records that are wrongly classified will have their weights increased
- Records that are classified correctly will have their weights decreased

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

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

- Boosting algorithms differ in terms of (1) how the weights of the training examples are updated at the end of each round, and (2) how the predictions made by each classifier are combined.

Example: AdaBoost

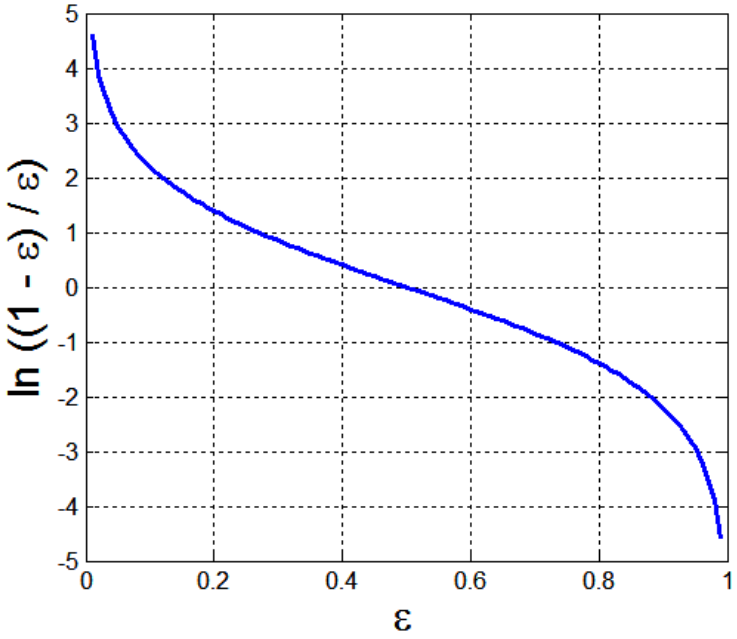
- Base classifiers: C1, C2, …, C.
- $\delta(p) = 1$ if p is true, $\delta(p) = 0$ if p is false
- Error rate of a classifier:

$$\varepsilon_j = \frac{1}{N}\sum_{i=1}^{N} w_i \delta(C_j(x_i) \neq y_i)$$

- Importance of a classifier:

$$\alpha_j = \frac{1}{2}\ln\left(\frac{1-\varepsilon_j}{\varepsilon_j}\right)$$

- Classifier of lower error is more important



Example: AdaBoost

- Weight update for each example:

$$w_i^{(j+1)} = \frac{w_i^{(j)}}{Z_j} \times \begin{cases} \exp^{-\alpha_j} & \text{if } C_j(x_i) = y_i \\ \exp^{\alpha_j} & \text{if } C_j(x_i) \neq y_i \end{cases}$$

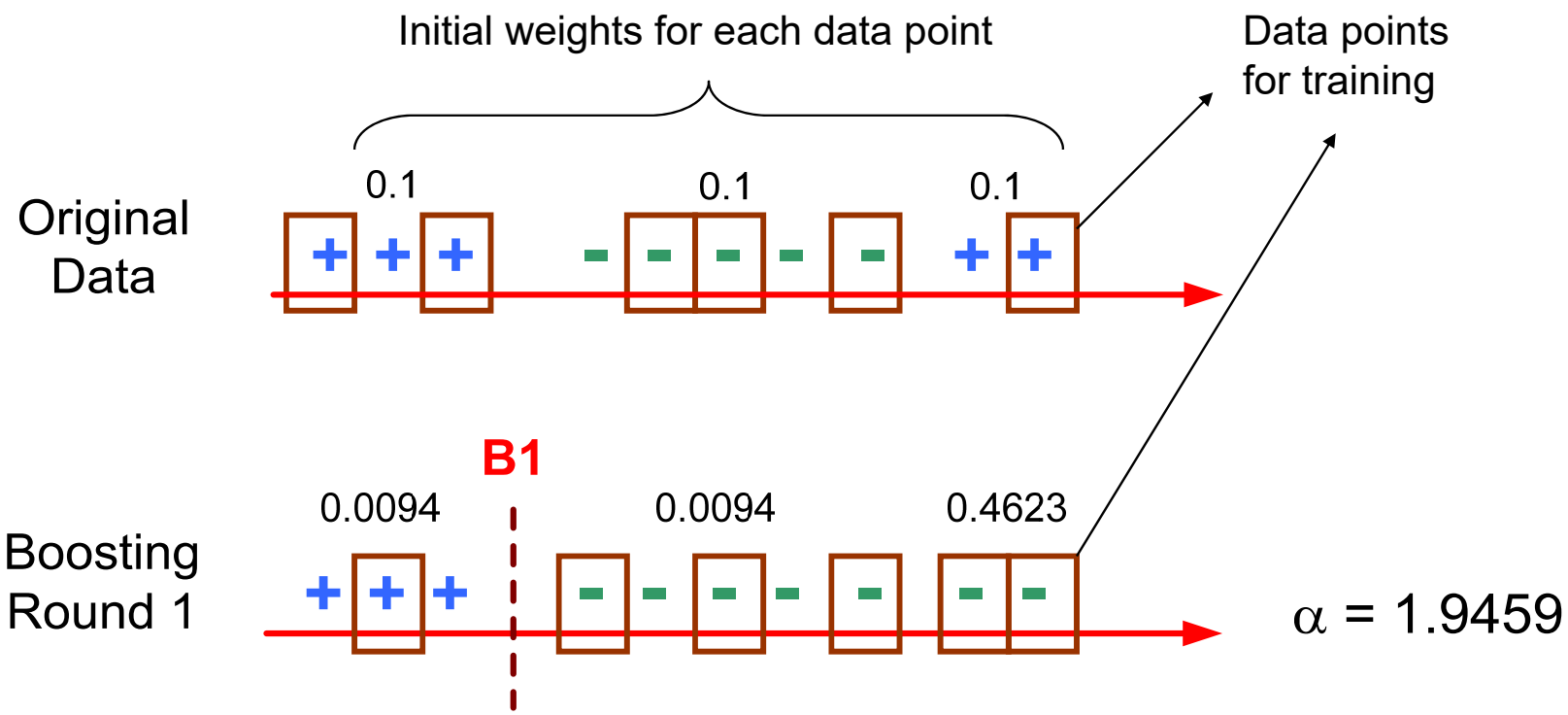
where Z_j is the normalization factor to ensure $\sum_i w_i^{(j+1)} = 1$

  - Misclassified objects will have weight increased

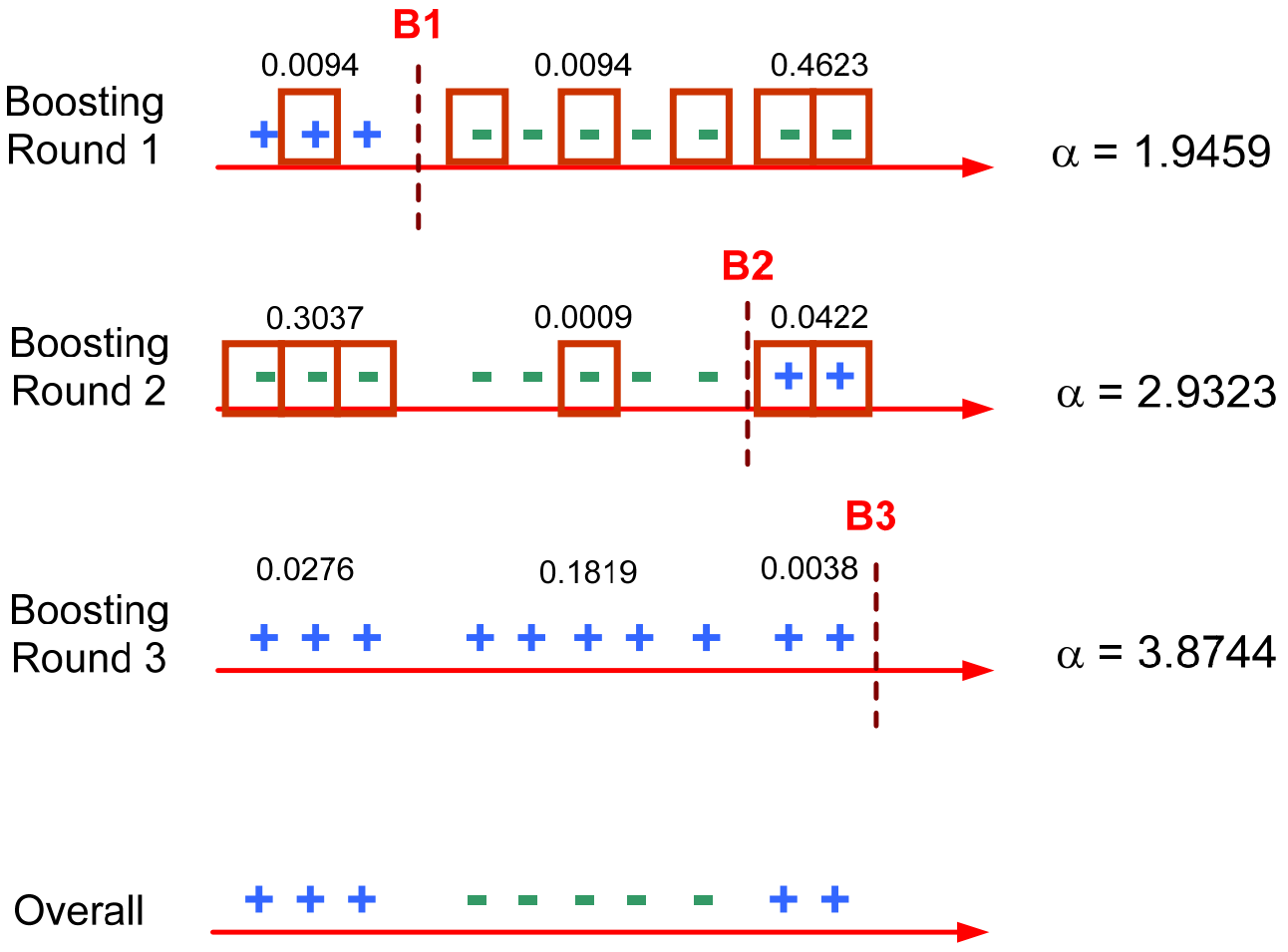
- Classification: $C^*(\mathrm{x}) = \arg\max_y \sum_{j=1}^{T} \alpha_j \delta(C_j(x) = y)$

- It penalizes models that have poor accuracy
- If any intermediate rounds produce error rate higher than 50%, the weights are reverted back to 1/n and the resampling procedure is repeated

Illustrating AdaBoost



Illustrating AdaBoost



Random forests

- Random forest is a class of ensemble methods specifically designed for decision tree classifiers.
- Random forest is a kind of bagging with decision trees, but it is not the same as the bagging with decision trees
- Random forest randomly determines splits without computing the information gain
  - More efficient by not computing the IG
  - More accurate by base classifiers being more independent

Example: Ensemble methods

Data Set	Number of (Attributes, Classes, Records)	Decision Tree (%)	Bagging (%)	Boosting (%)	RF (%)
Anneal	(39, 6, 898)	92.09	94.43	95.43	95.43
Australia	(15, 2, 690)	85.51	87.10	85.22	85.80
Auto	(26, 7, 205)	81.95	85.37	85.37	84.39
Breast	(11, 2, 699)	95.14	96.42	97.28	96.14
Cleve	(14, 2, 303)	76.24	81.52	82.18	82.18
Credit	(16, 2, 690)	85.8	86.23	86.09	85.8
Diabetes	(9, 2, 768)	72.40	76.30	73.18	75.13
German	(21, 2, 1000)	70.90	73.40	73.00	74.5
Glass	(10, 7, 214)	67.29	76.17	77.57	78.04
Heart	(14, 2, 270)	80.00	81.48	80.74	83.33
Hepatitis	(20, 2, 155)	81.94	81.29	83.87	83.23
Horse	(23, 2, 368)	85.33	85.87	81.25	85.33
Ionosphere	(35, 2, 351)	89.17	92.02	93.73	93.45
Iris	(5, 3, 150)	94.67	94.67	94.00	93.33
Labor	(17, 2, 57)	78.95	84.21	89.47	84.21
Led7	(8, 10, 3200)	73.34	73.66	73.34	73.06
Lymphography	(19, 4, 148)	77.03	79.05	85.14	82.43
Pima	(9, 2, 768)	74.35	76.69	73.44	77.60
Sonar	(61, 2, 208)	78.85	78.85	84.62	85.58
Tic-tac-toe	(10, 2, 958)	83.72	93.84	98.54	95.82
Vehicle	(19, 4, 846)	71.04	74.11	78.25	74.94
Waveform	(22, 3, 5000)	76.44	83.30	83.90	84.04
Wine	(14, 3, 178)	94.38	96.07	97.75	97.75
Zoo	(17, 7, 101)	93.07	93.07	95.05	97.03

From Data Mining: Practical Machine Learning Tools and Techniques, Third Edition

Performance Evaluation

Big Data

Confusion matrix

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	TP	FN
	Class=No	FP	TN

TP (true positive)

FN (false negative)

FP (false positive)

TN (true negative)

Accuracy

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	TP	FN
	Class=No	FP	TN

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- E.g. 2-class problem where |P| = 10, |N| = 9990
  - Model predicting everything to be the majority class
  - Its accuracy is 9990/10000 = 99.9 %

Cost matrix

	PREDICTED CLASS		
ACTUAL CLASS	C(i|j)	Class=Yes	Class=No
	Class=Yes	C(Yes|Yes) TP	C(No|Yes) FN
	Class=No	C(Yes|No) FP	C(No|No) TN

C(i|j): Cost of misclassifying class j example as class i

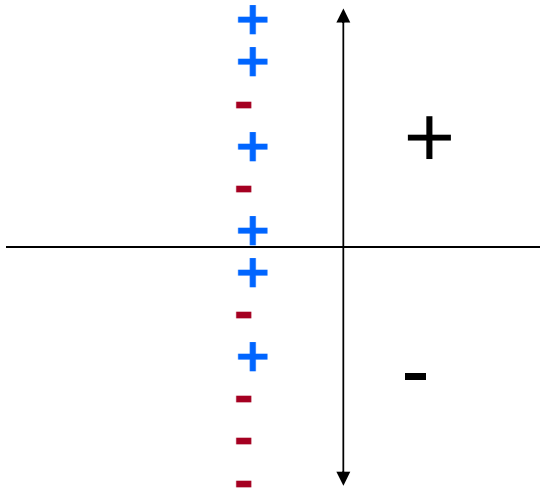
F-measures

- $\text{Precision}(p) = \frac{TP}{TP+FP}$
- $\text{Recall}(r) = \frac{TP}{TP+FN}$
- $\text{F} - \text{measure}(F) = \frac{2rp}{r+p} = \frac{2TP}{2TP+FP+FN}$

- Precision is biased towards C(Yes|Yes) & C(Yes|No)
- Recall is biased towards C(Yes|Yes) & C(No|Yes)
- F-measure is biased towards all except C(No|No)

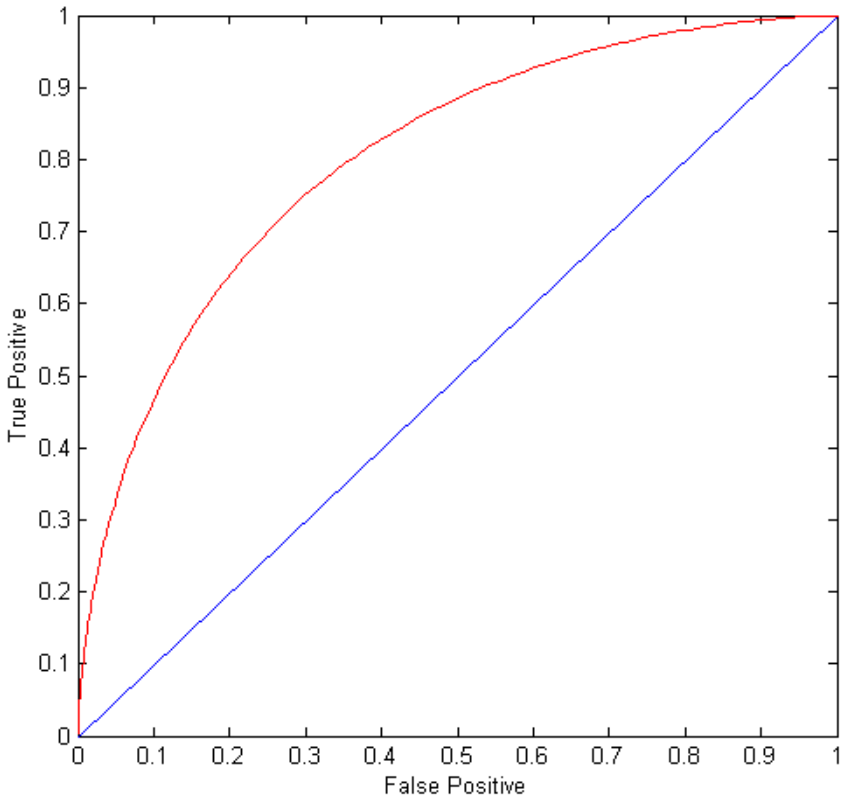
F-measures

- Evaluating a multiclass classification
  - Compute F-measure for each class as positive and average them
- What is Precision and Recall of =>
- Evaluating an IR (Information retrieval) system (e.g. Google)
  - Use average precision. Why?

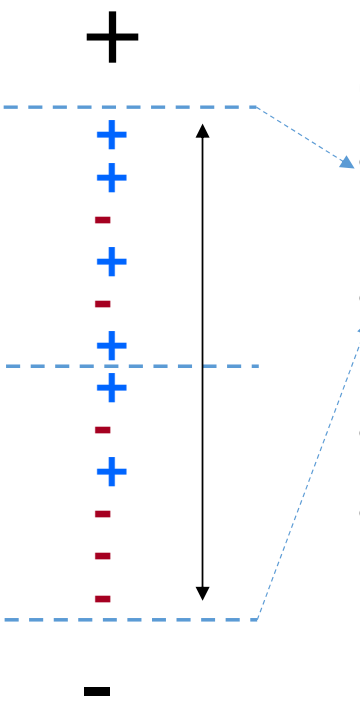


ROC (Receiver Operating Characteristic)

- ROC curve plots TP rate (on the y-axis) against FP rate (on the x-axis)
  - TP rate = TP/P (= sensitivity)
  - FP rate = FP/N (= 1 – specificity)
- Most classification methods provide a threshold that can control the tradeoff between TP and FP
- Performance of a classifier represented as a point on the ROC curve
  - changing the threshold of algorithm changes the location of the point

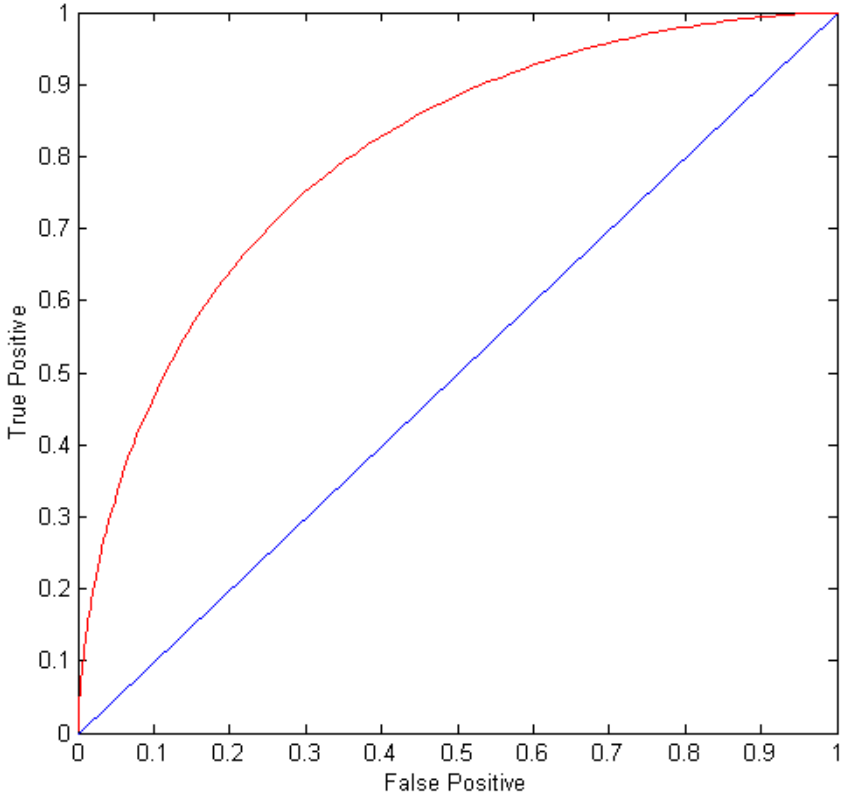


ROC (Receiver Operating Characteristic)



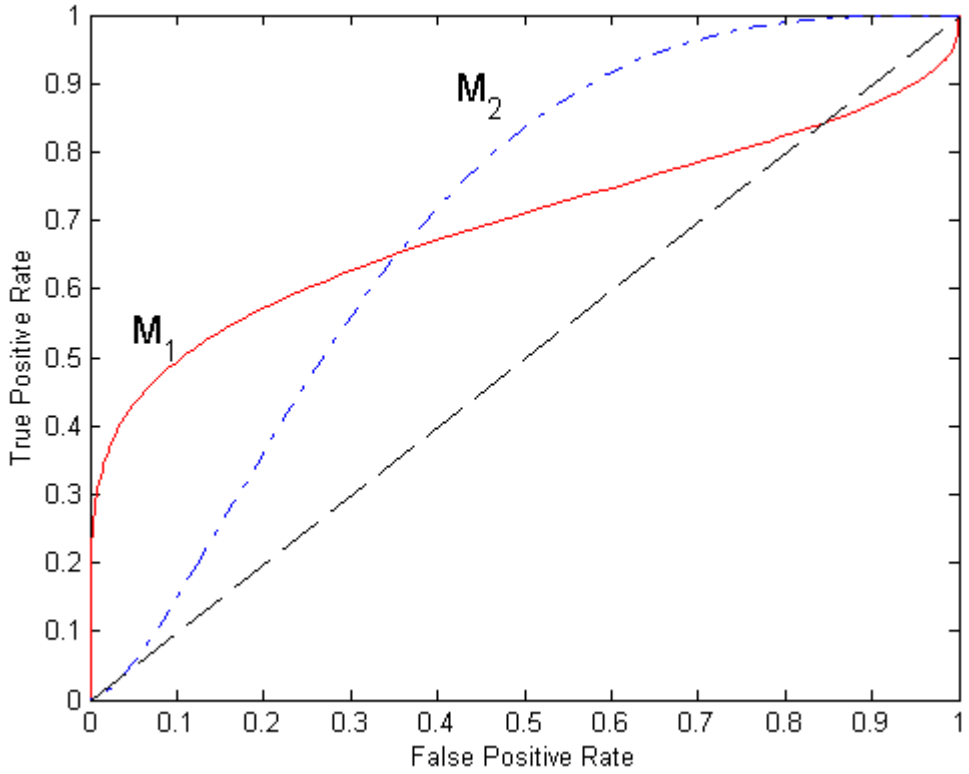
(TP,FP):

- (0,0): declare everything to be negative class
- (1,1): declare everything to be positive class
- (1,0): ideal
- Diagonal line: random guessing



ROC and AUC (Area Under the ROC Curve)

- M_1 is better for small FPR
- M_2 is better for large FPR
- Area Under the ROC Curve (AUC)
  - Another metric for evaluating classification performance
  - Ideal: Area = 1
  - Random guess: Area = 0.5



Classification performance measures

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	TP	FN
	Class=No	FP	TN

- precision = t-pos/(t-pos + f-pos)
- recall = t-pos/(t-pos + f-neg)
- F1 = 2*precision*recall / (precision + recall)
- AUC = the Area Under ROC Curve

Alternative accuracy measures (e.g. for cancer diagnosis)

- sensitivity = t-pos / (t-pos + f-neg) /* = recall */
- specificity = t-neg / (t-neg + f-pos) /* FP rate = 1 - specificity */