

Machine Learning and Data Mining

Ensemble Methods

UCSCextension
Silicon Valley



Patricia Hoffman, PhD

Slides are from

- Tan, Steinbach, and Kumar
- Ethem Alpaydin

Ensemble Methods

Main Sources

- Slides from Tan, Steinbach, Kumar
 - http://www-users.cs.umn.edu/~kumar/dmbook/dmslides/chap5_alternative_classification.pdf
- Slides from Ethem Alpaydin
 - http://www.realtechsupport.org/UB/MRIII/papers/MachineLearning/Alpaydin_MachineLearning_2010.pdf
- Seni, Giovanni and Elder, John, Ensemble Methods in Data Mining, Morgan & Claypool, 2010
 - http://www.amazon.com/Ensemble-Methods-Data-Mining-Predictions/dp/1608452840/ref=sr_1_1?ie=UTF8&s=books&qid=1287690908&sr=8-1
- Berkeley Professor Leo Breiman
 - <http://www.stat.berkeley.edu/~breiman/RandomForests/>

Ensemble Methods

Projects - Controversy

- Leo Breiman Bagging Predictors
 - Machine Learning, 24, 123–140 (1996)
 - 1996 Kluwer Academic Publishers, Boston. Manufactured in The Netherlands.
 - <http://www.scribd.com/doc/54567585/10-1-1-121>
- David Mease
 - Evidence Contrary to Statistical View of Boosting
 - <http://dl.acm.org/citation.cfm?id=1390687&dl=ACM&coll=DL&CFID=47172095&CFTOKEN=13891889>
- J. Friedman, T. Hastie, and R. Tibshirani
 - [Additive logistic regression: A statistical view of boosting. Annals of Statistics, 28:337–374, 2000a](#)

Ensemble Methods

R Packages

- R Package randomForest
 - <http://cran.r-project.org/web/packages/randomForest/randomForest.pdf>
- R Package Ada Boost - gbm
 - <http://cran.r-project.org/web/packages/gbm/gbm.pdf>

Ensemble Methods

Example Code

- `AdaBoostRandomForest.R`
 - Comparison using the Sonar Data Set
- `BabyEnsemble.R`
 - Uses ridge regression to combine models
- `MulticlassRegressionIrisData.R`
 - Multiclass Example
- `ROC.R` and `ROCSonar.R`
 - Developing a ROC curve for scoring models

Balanced Class

- Assumed the same loss for both types of misclassification
 - 50% is cutoff and is assigned to label the majority class
- This is appropriate if
 - 1) We suffer the same cost for both types of errors
 - 2) We are interested in the probability of 0.5 only
 - 3) The ratio of the two classes in our training data will match that in the population to which we will apply the model

Class Imbalance

- It is desirable to tune the number of observations being classified as positive
 - Various ways to approach this are available
 - Methods are model dependent
- Choose a probability different from 0.5, using a threshold on some continuous confidence output or under/over-sampling



Recall and Precision (page 297 of text)

ACTUAL CLASS	PREDICTED CLASS	
	Class=Yes	Class=No
Class=Yes	a (TP)	b (FN)
	c (FP)	d (TN)

$$\text{Recall} = \frac{a}{a+b} = \frac{TP}{TP+FN} = \frac{\text{Correctly Predicted Positives}}{\text{Actual Positives}} \quad (\text{Sensitivity})$$

$$\text{Precision} = \frac{a}{a+c} = \frac{TP}{TP+FP} = \frac{\text{Correctly Predicted Positives}}{\text{All Predicted Positives}} \quad (\text{Precision})$$

Perfect When: Sensitivity = Precision = 1

$$\text{Before we just used accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

F Measure (page 207)

- F combines recall and precision into one number

$$F = \frac{2rp}{r + p} = \frac{2TP}{2TP + FP + FN}$$

- F is the harmonic mean of recall and precision

$$F = \frac{2rp}{r + p} = \frac{2}{1/r + 1/p}$$

Receiver Operating Curve (ROC)

- Used to tune the number of observations classified as positive.

$$\text{Recall TPR} = \frac{TP}{TP + FN}$$

$$\text{FPR} = \frac{FP}{FP + TN}$$

- ROC curve is plot
 - TPR on the y-axis
 - FPR on the x

ROC Curve Plot

Two Models

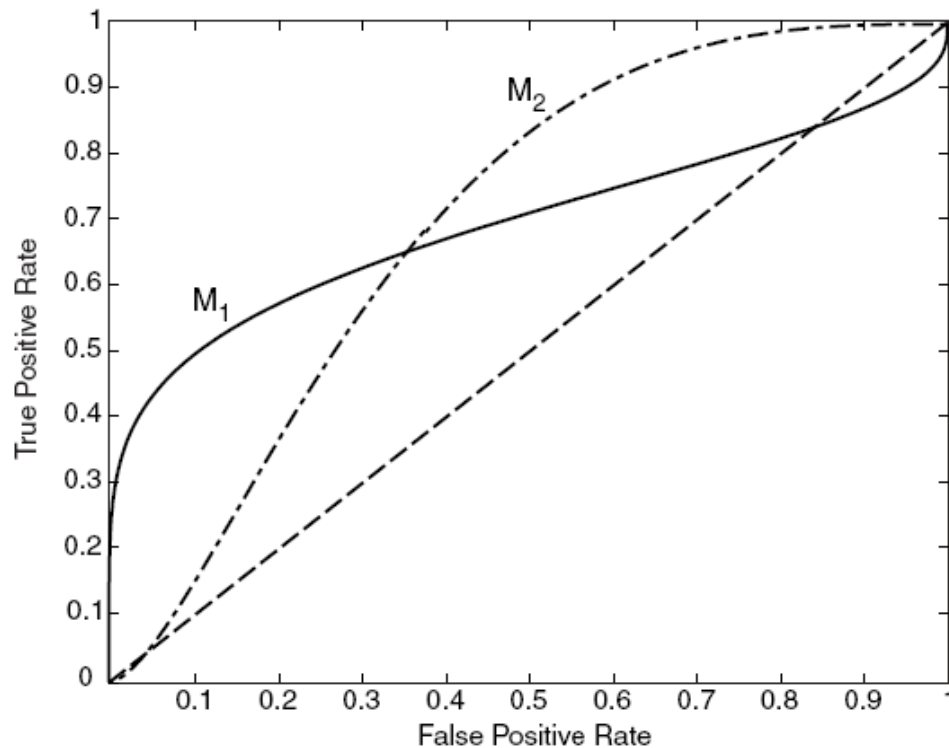


Figure 5.41. ROC curves for two different classifiers.

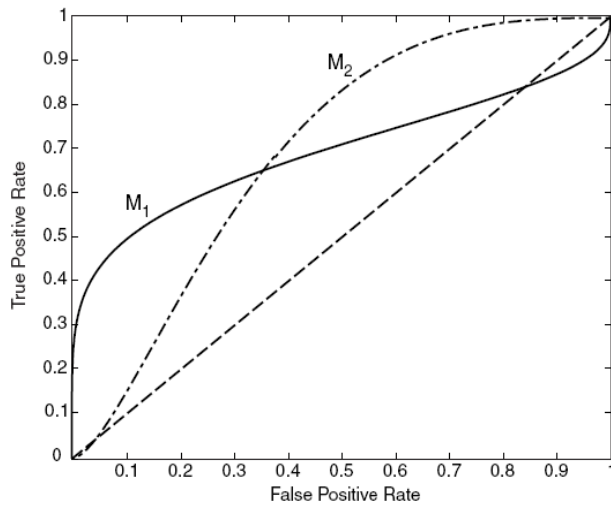


Figure 5.41. ROC curves for two different classifiers.

$$\text{TPR} = \text{TP}/(\text{TP}+\text{FN})$$

$$\text{FPR} = \text{FP}/(\text{TN}+\text{FP})$$

(TPR = 1, FPR = 0) The ideal model

(TPR = 0, FPR = 0)

Model Predicts every instance to be a negative class

(TPR = 1, FPR = 1)

Model predicts every instance to be a positive class

Dotted Line = Random Guesses

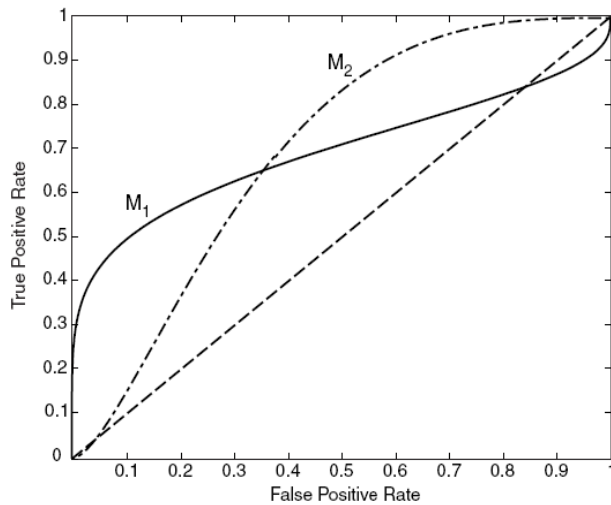


Figure 5.41. ROC curves for two different classifiers.

$$\text{TPR} = \text{TP}/(\text{TP}+\text{FN})$$

$$\text{FPR} = \text{FP}/(\text{TN}+\text{FP})$$

M1 is better than M2

when FPR is less than 0.36

M2 is better than M1

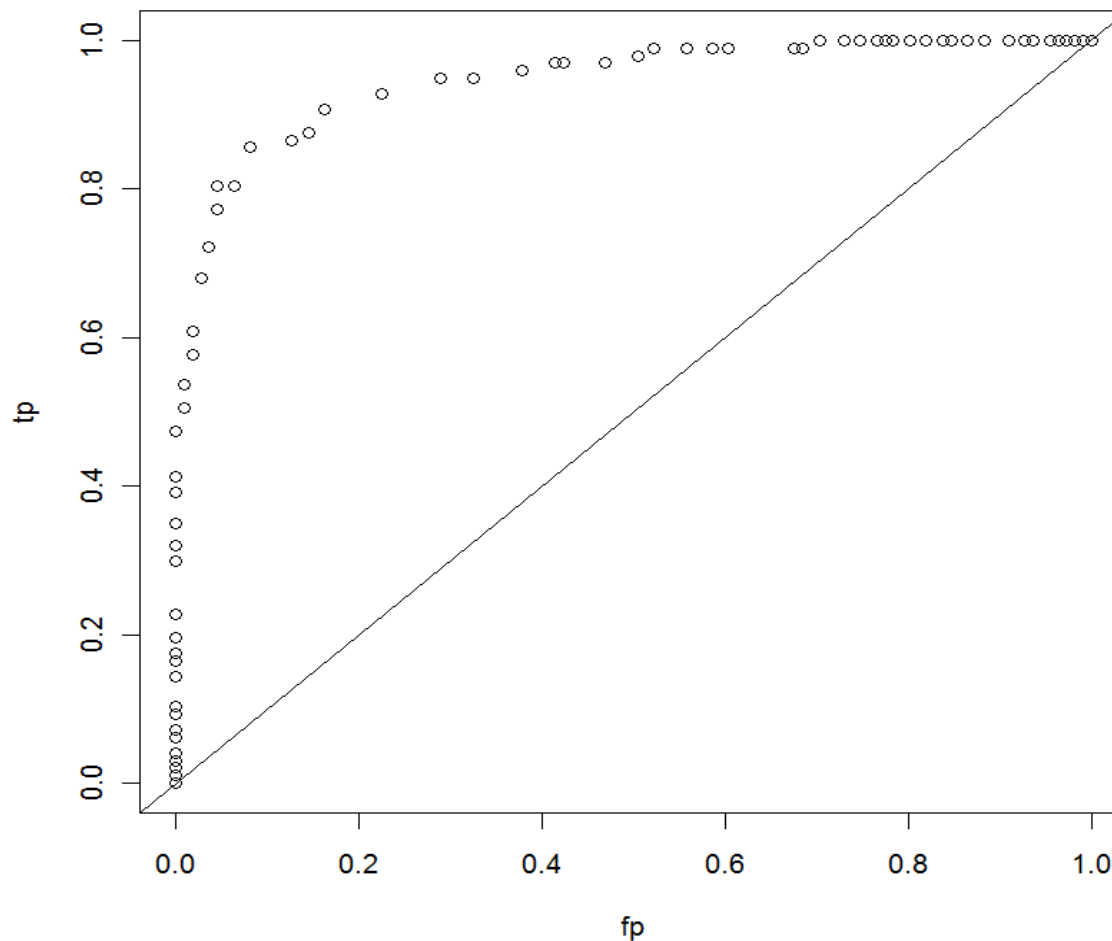
when FPR is greater than 0.36

ROC Curve

- Compares 2 Classifiers
- Better Classifier lies on top more often
- The Area Under the Curve (AUC) is often used as metric in evaluating models

ROCSonar.R

(ridge regression model)



Multiclass Problem

The Target $Y = \{y_1, y_2, \dots, y_k\}$ has multiple values – Iris Data is Example

- One-Against-Rest (1-r)
 - K binary classifiers, one for each y_i in Y
 - y_i is the positive example – rest are negative examples
- One-Against-One (1-1)
 - $K(K-1)/2$ binary classifiers
 - each classifier distinguishes between a pair of classes (y_i, y_j)
 - instances that do not belong to either y_i or y_j are ignored
- Test Instances Classified
 - combine predictions
 - from binary classifiers
 - voting scheme or probability estimate

Error-Correcting Output Codes

- K classes; L problems (Dietterich and Bakiri, 1995)

- Code matrix **W** codes classes in terms of learners

- One per class

$$L=K$$

$$\mathbf{W} = \begin{bmatrix} +1 & -1 & -1 & -1 \\ -1 & +1 & -1 & -1 \\ -1 & -1 & +1 & -1 \\ -1 & -1 & -1 & +1 \end{bmatrix}$$

- Pairwise

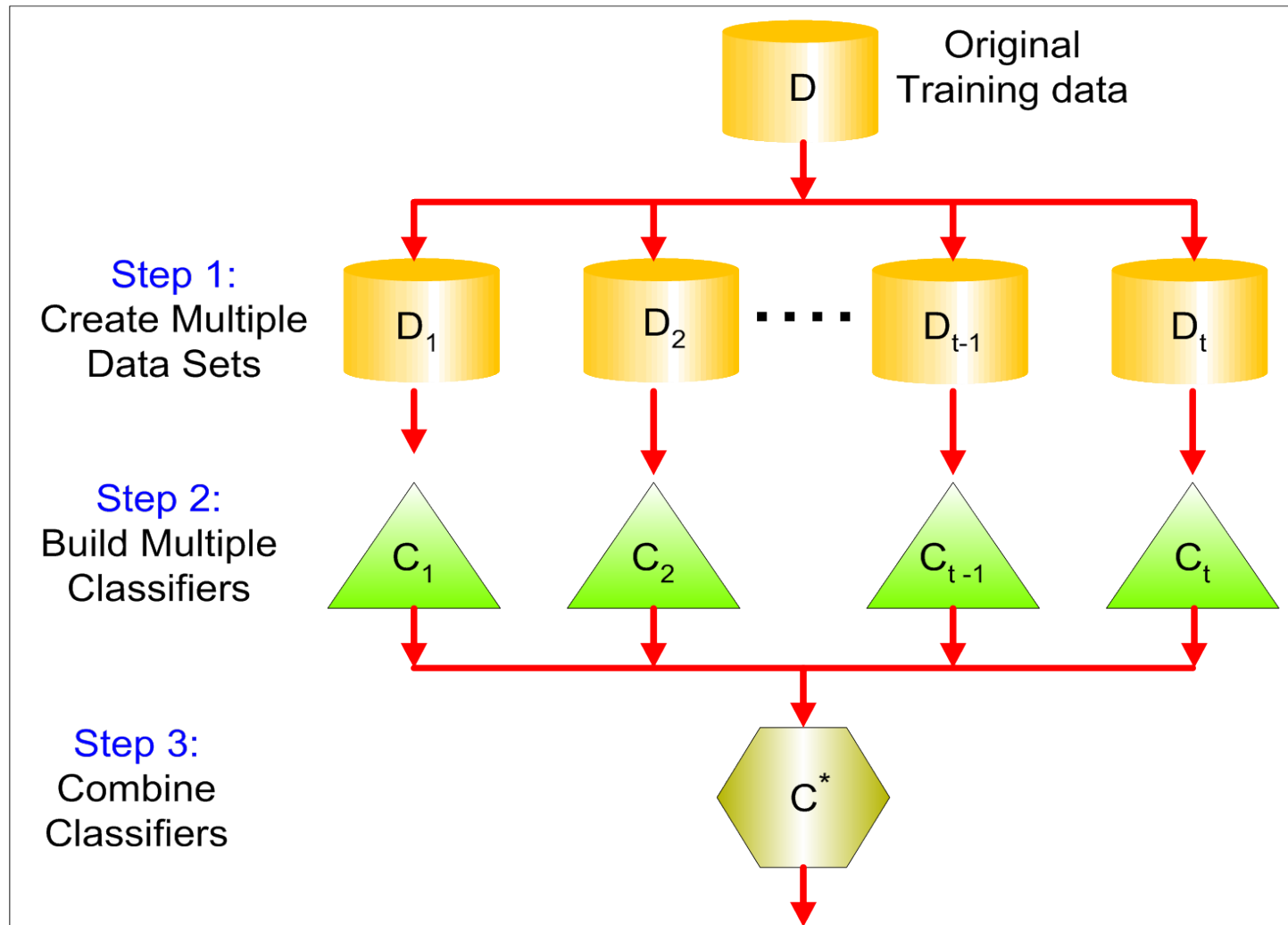
$$L=K(K-1)/2$$

$$\mathbf{W} = \begin{bmatrix} +1 & +1 & +1 & 0 & 0 & 0 \\ -1 & 0 & 0 & +1 & +1 & 0 \\ 0 & -1 & 0 & -1 & 0 & +1 \\ 0 & 0 & -1 & 0 & -1 & -1 \end{bmatrix}$$

Ensemble Methods

- Construct a set of classifiers from the training data
- Predict class label of previously unseen records by aggregating predictions made by multiple classifiers
 - Easiest Method: Average Classifiers
 - Take Median for Regression

General Idea





Bias Stays the Same

Variance Decreased by Number of Models

- Given L Models to combine
- Let d_j be the iid predictions for y (one for each model)

$$E[y] = E\left[\sum_j \frac{1}{L} d_j\right] = \frac{1}{L} L \cdot E[d_j] = E[d_j]$$

$$\text{Var}(y) = \text{Var}\left(\sum_j \frac{1}{L} d_j\right) = \frac{1}{L^2} \text{Var}\left(\sum_j d_j\right) = \frac{1}{L^2} L \cdot \text{Var}(d_j) = \frac{1}{L} \text{Var}(d_j)$$

Bias does not change, Variance decreases by L
Averaging over Randomness

Why does it work? Intuition

- Suppose there are 25 iid base classifiers
 - Each classifier has error rate, $\varepsilon = 0.35$
 - Assume classifiers are independent
 - Ensemble incorrectly predicts only if more than half of the base classifiers predict incorrectly.
 - Probability that the ensemble classifier makes a wrong prediction (Binomial Distribution):

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

Examples of Ensemble Methods

- How to generate an ensemble of classifiers?
 - Bagging – Bootstrap Aggregation (train on repeated samples with replacement from the data)
 - Boosting (Combine simple base classifiers by upweighting data points which are misclassified)

Bagging

- Sampling with replacement

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
- Each sample has probability $(1 - 1/n)^n$ of being selected

Random Forest

- Ensemble Method using Decision Trees
 - Create Random Set of Vectors
 - Use random vectors to build multiple decision trees
 - Combine decision trees
- Bagging – Random Vector
 - Used to select (with replacement) N samples from the original data set
- Feature Selection – Random Vector
 - Only the randomly selected features are used to make split decision for each node

Random Forest Characteristics

- Examples: bagging (Breiman, 1996), random split selection (Dietterich, 1998), random subspace (Ho, 1998), written character recognition (Amit and Geman, 1997)
- Easily Parallelized
- Gives Useful Internal estimates of error, strength, correlation
- Generalization error of Random Forest depends on the **strength** of the individual trees in the forest and the **correlation** between them
- To improve accuracy, the randomness injected has to minimize the correlation $\bar{\rho}$ while maintaining strength.

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
- Boosting algorithms differ in
 - How the weights of the training examples are updated
 - How the predictions are combined

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

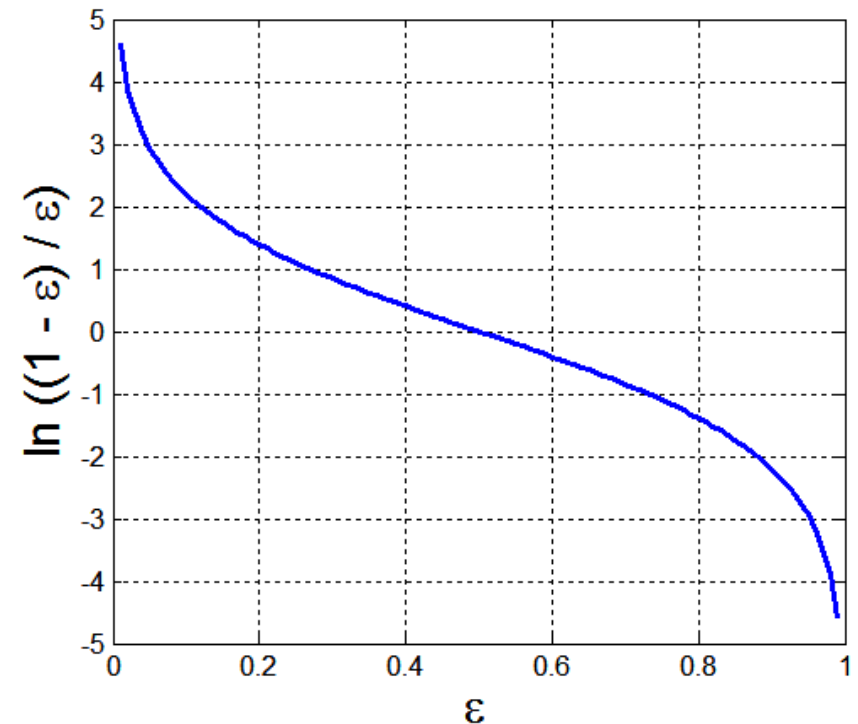
Example: AdaBoost


- Base classifiers: C_1, C_2, \dots, C_T
- Error rate:

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

- Importance of a classifier:

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





Example: AdaBoost

- Weight update:

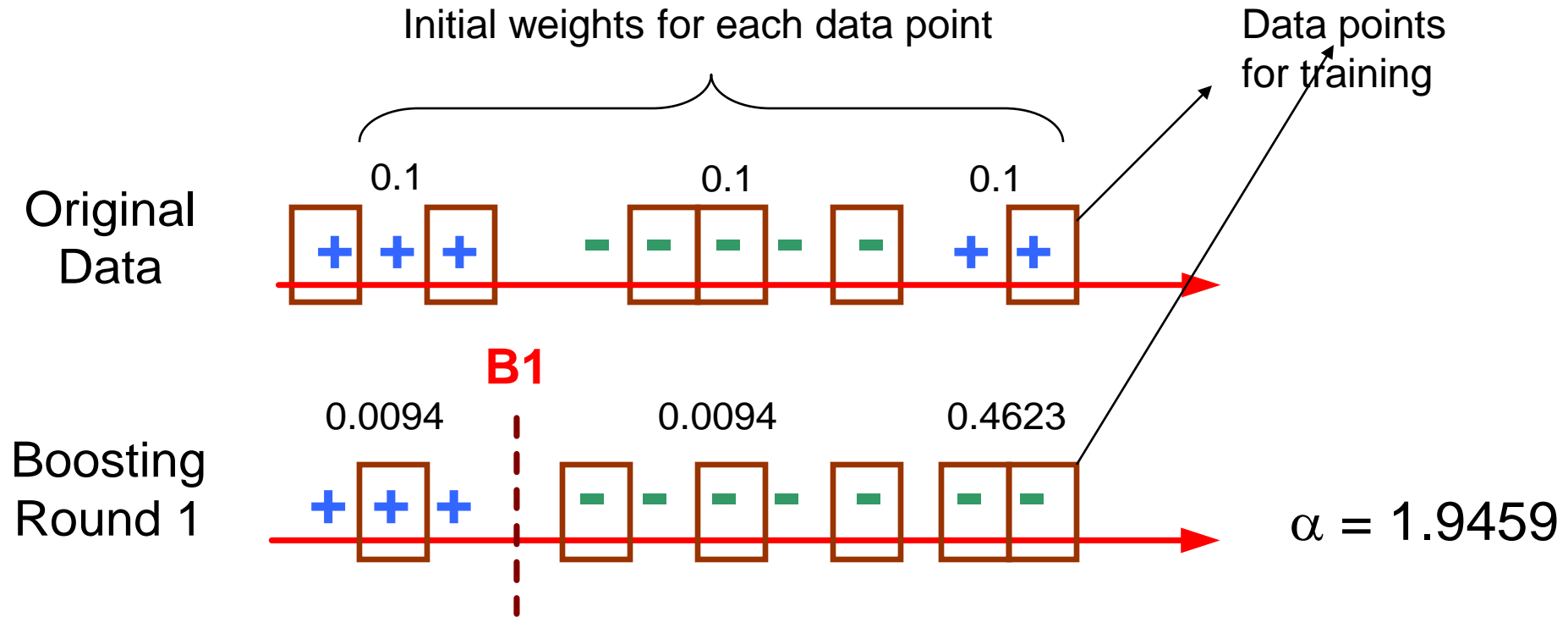
$$w_i^{(j+1)} = \frac{w_i^{(j)}}{Z_j} \begin{cases} \exp^{-\alpha_j} & \text{if } C_j(x_i) = y_i \\ \exp^{\alpha_j} & \text{if } C_j(x_i) \neq y_i \end{cases}$$

where Z_j is the normalization factor

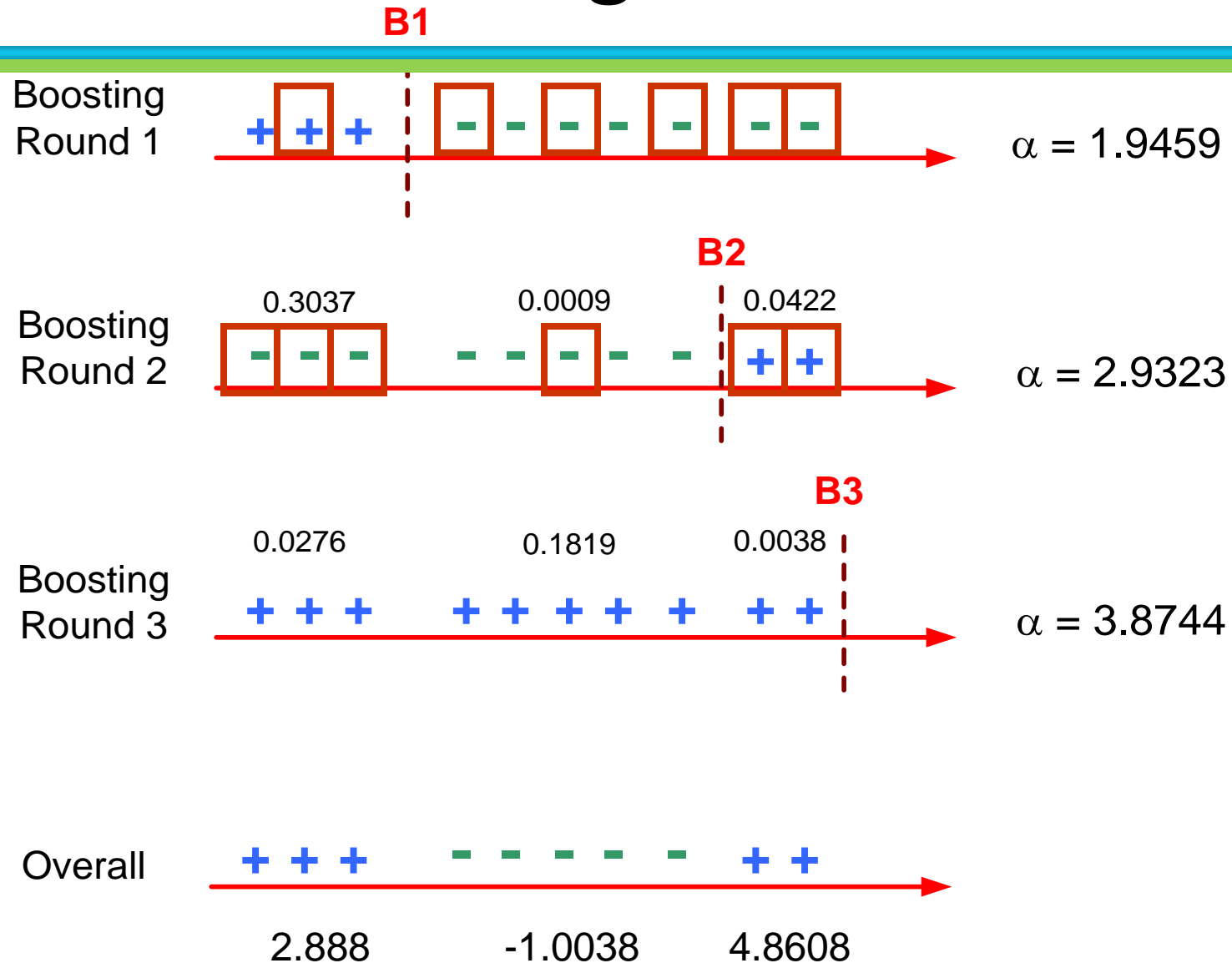
- If any intermediate rounds produce error rate higher than 50%, the weights are reverted back to $1/n$ and the resampling procedure is repeated

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

Illustrating AdaBoost



Illustrating AdaBoost



Ada Boost Algorithm

First let $F_0(x_i) = 0$ for all x_i and initialize weights $w_i = 1/n$ for $i = 1, \dots, n$. Then repeat the following for m from 1 to M :

- Fit the classifier g_m to the training data using weights w_i where g_m maps each x_i to -1 or 1.
- Compute the weighted error rate $\epsilon_m \equiv \sum_{i=1}^n w_i \mathbb{I}[y_i \neq g_m(x_i)]$ and half its log-odds, $\alpha_m \equiv \frac{1}{2} \log \frac{1-\epsilon_m}{\epsilon_m}$.
- Let $F_m = F_{m-1} + \alpha_m g_m$.
- Replace the weights w_i with $w_i \equiv w_i e^{-\alpha_m g_m(x_i) y_i}$ and then renormalize by replacing each w_i by $w_i / (\sum w_i)$.

The final classifier is 1 if $F_M > 0$ and -1 otherwise.

Algorithm repeats until chosen stopping time.

Final classifier is based on sign of F_M

Practical Advice for AdaBoost

- AdaBoost is one of the most successful boosting algorithms
- Do not assume that newer, regularized and modified versions of boosting are necessarily better
- Try standard AdaBoost along with these newer algorithms
- If classification is the goal, monitor the misclassification error on hold out (or cross-validation) samples
- Much of the evidence presented is counter-intuitive
 - keep an open mind when experimenting with AdaBoost
 - If stumps are causing overfitting, be willing to try larger trees
 - Intuition may suggest the larger trees will overfit, but we have seen that is not necessarily true

Ada Boost and Random Forest Comparison

- Accuracies are comparable
- Ada Boost
 - Susceptible to Overfitting
- Random Forest
 - More robust to noise
 - Generally runs faster

Table 5.5. Comparing the accuracy of a decision tree classifier against three 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