# Statistics 202: Statistical Aspects of Data Mining

Professor David Mease

Tuesday, Thursday 9:00-10:15 AM Terman 156

Lecture 11 = Finish ch. 4 and start ch. 5

Agenda:

1) Reminder for 4th Homework (due Tues Aug 7)
2) Finish lecturing over Ch. 4 (Sections 4.1-4.5)
3) Start lecturing over Ch. 5 (Section 5.7)

# Homework Assignment:

**Chapter 4 Homework and Chapter 5 Homework Part 1 is due Tuesday 8/7**

**Either email to me (dmease@stanford.edu), bring it to class, or put it under my office door.**

**SCPD students may use email or fax or mail.**

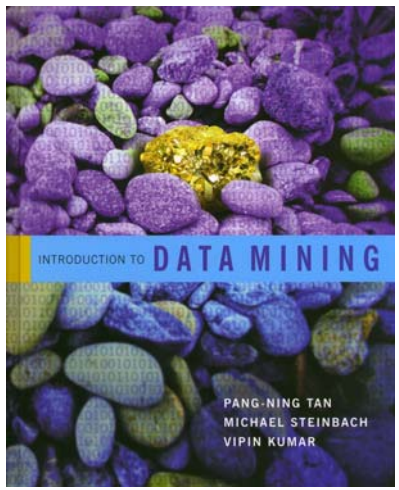
**The assignment is posted at**

**http://www.stats202.com/homework.html**

**Important: If using email, please submit only a single file (word or pdf) with your name and chapters in the file name. Also, include your name on the first page. Finally, please put your name and the homework # in the subject of the email.**

# Introduction to Data Mining

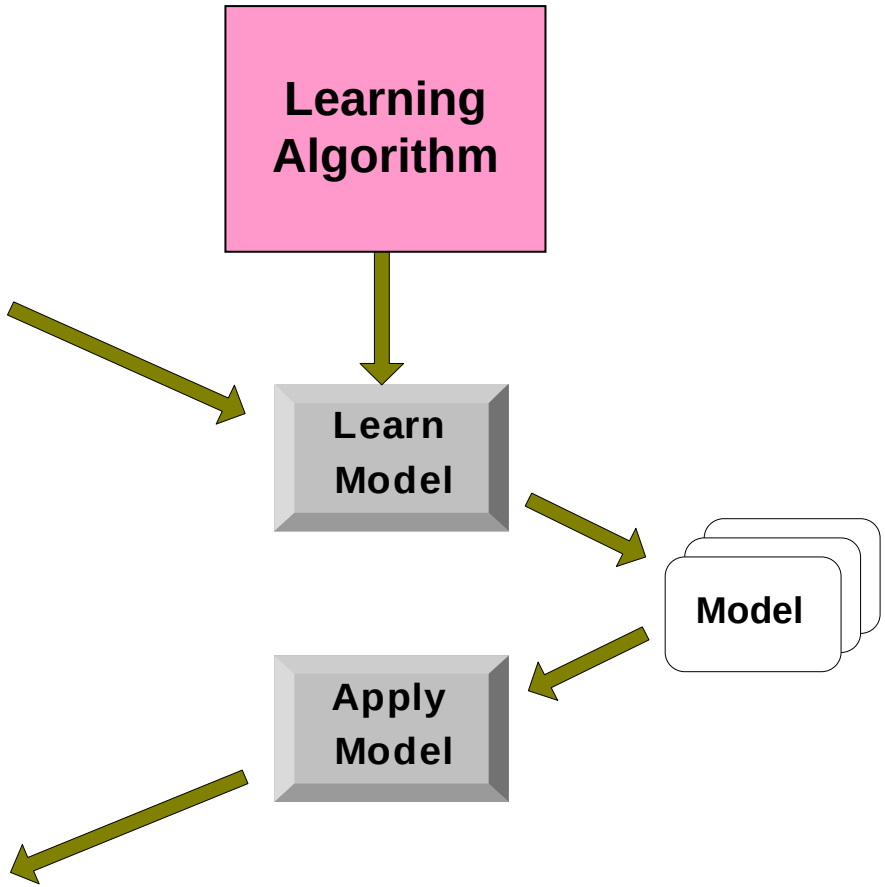
by
Tan, Steinbach, Kumar




# Chapter 4: Classification: Basic Concepts, Decision Trees, and Model Evaluation

# Illustration of the Classification Task:

| *Tid* | Attrib1 | Attrib2 | Attrib3 | Class |
|---|---|---|---|---|
| 1 | Yes | Large | 125K | No |
| 2 | No | Medium | 100K | No |
| 3 | No | Small | 70K | No |
| 4 | Yes | Medium | 120K | No |
| 5 | No | Large | 95K | Yes |
| 6 | No | Medium | 60K | No |
| 7 | Yes | Large | 220K | No |
| 8 | No | Small | 85K | Yes |
| 9 | No | Medium | 75K | No |
| 10 | No | Small | 90K | Yes |




| *Tid* | Attrib1 | Attrib2 | Attrib3 | Class |
|---|---|---|---|---|
| 11 | No | Small | 55K | ? |
| 12 | Yes | Medium | 80K | ? |
| 13 | Yes | Large | 110K | ? |
| 14 | No | Small | 95K | ? |
| 15 | No | Large | 67K | ? |

# Classification: Definition

- **Given a collection of records (*training set*)**
  - **Each record contains a set of *attributes (x)*, with one additional attribute which is the *class (y)*.**
- **Find a *model* to *predict* the class as a function of the values of other attributes.**
- **Goal: previously unseen records should be assigned a class as accurately as possible.**
  - **A *test set* is used to determine the accuracy of the model. Usually, the given data set is divided into training and test sets, with training set used to build the model and test set used to validate it.**

# Classification Techniques

- **There are many techniques/algorithms for carrying out classification**

- **In this chapter we will study only *decision trees***

- **In Chapter 5 we will study other techniques, including some very modern and effective techniques**

# An Example of a Decision Tree

| Tid | Refund | Marital Status | Taxable Income | Cheat |
|---|---|---|---|---|
| | | | | |
| 1 | Yes | Single | 125K | No |
| 2 | No | Married | 100K | No |
| 3 | No | Single | 70K | No |
| 4 | Yes | Married | 120K | No |
| 5 | No | Divorced | 95K | Yes |
| 6 | No | Married | 60K | No |
| 7 | Yes | Divorced | 220K | No |
| 8 | No | Single | 85K | Yes |
| 9 | No | Married | 75K | No |
| 10 | No | Single | 90K | Yes |

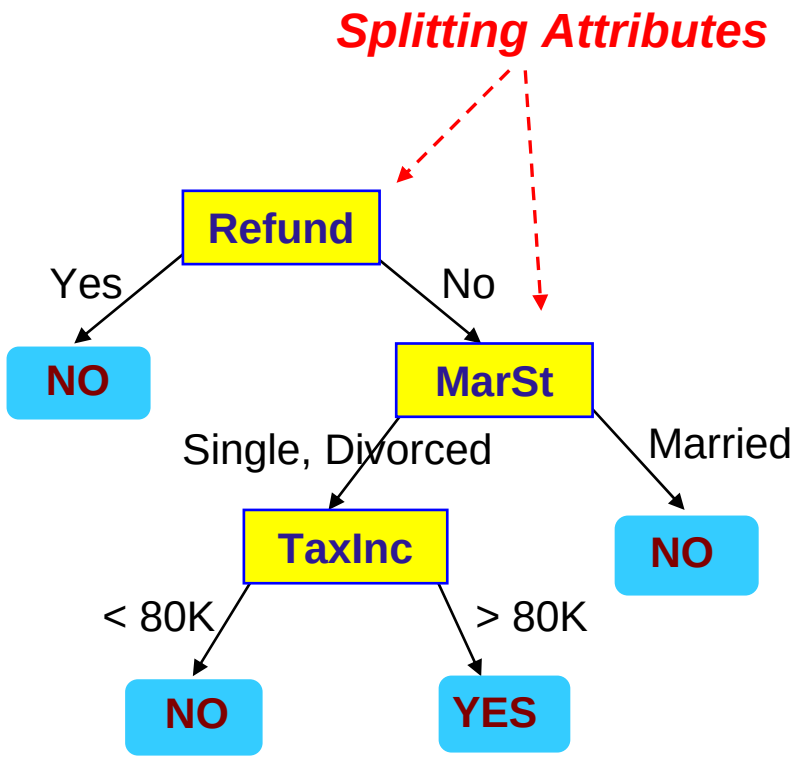
categorical

categorical

continuous

class

**Training Data**




**Model: Decision Tree**

# How are Decision Trees Generated?

- Many algorithms use a version of a "top-down" or "divide-and-conquer" approach known as Hunt's Algorithm (Page 152):

Let $D_t$ be the set of training records that reach a node t

- If $D_t$ contains records that belong the same class $y_t$, then t is a leaf node labeled as $y_t$
- If $D_t$ contains records that belong to more than one class, use an attribute test to split the data into smaller subsets. Recursively apply the procedure to each subset.

# How to Apply Hunt's Algorithm

- **Usually it is done in a "greedy" fashion.**

- **"Greedy" means that the optimal split is chosen at each stage according to some criterion.**

- **This may not be optimal at the end even for the same criterion, as you will see in your homework.**

- **However, the greedy approach is computational efficient so it is popular.**

# How to Apply Hunt's Algorithm (continued)

- Using the greedy approach we still have to decide 3 things:

  #1) What attribute test conditions to consider

  #2) What criterion to use to select the "best" split

  #3) When to stop splitting

- For #1 we will consider only binary splits for both numeric and categorical predictors as discussed on the next slide

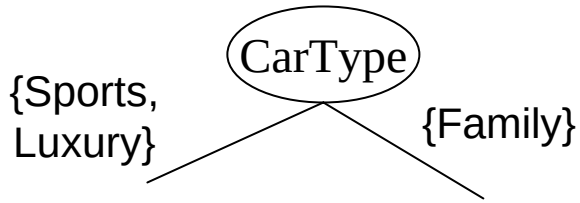
- For #2 we will consider misclassification error, Gini index and entropy

- #3 is a subtle business involving model selection. It is tricky because we don't want to overfit or underfit.

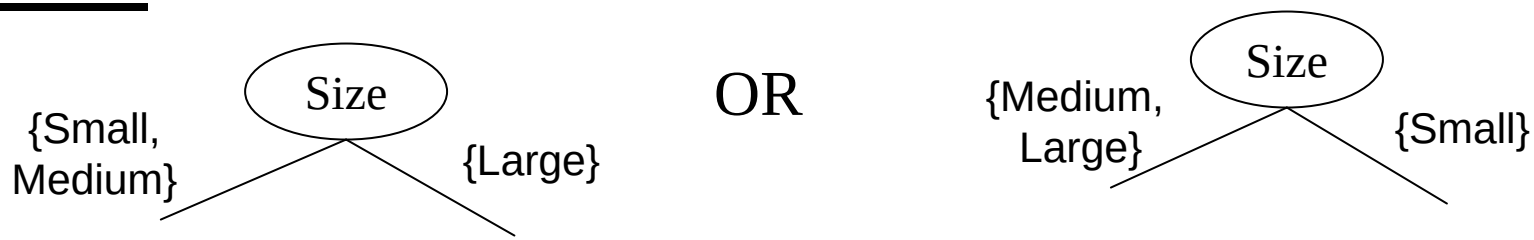
# #1) What Attribute Test Conditions to Consider (Section 4.3.3, Page 155)

- **We will consider only binary splits for both numeric and categorical predictors as discussed, but your book talks about multiway splits also**

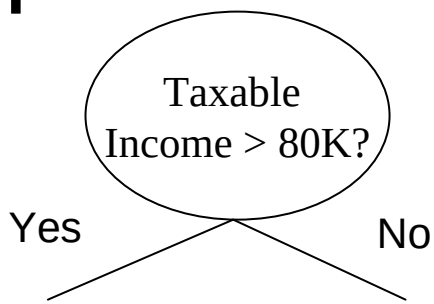
- **Nominal**




- **Ordinal – like nominal but don't break order with split**




- **Numeric – often use midpoints between numbers**



# #2) What criterion to use to select the "best" split (Section 4.3.4, Page 158)

- **We will consider misclassification error, Gini index and entropy**

**Misclassification Error:** $$Error(t) = 1 - \max_i P(i \mid t)$$

**Gini Index:** $$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

**Entropy:** $$Entropy(t) = -\sum_j p(j \mid t) \log_2 p(j \mid t)$$

# Misclassification Error

$$Error(t) = 1 - \max_i P(i \mid t)$$

- **Misclassification error is usually our final metric which we want to minimize on the test set, so there is a logical argument for using it as the split criterion**

- **It is simply the fraction of total cases misclassified**

- **1 - Misclassification error = "Accuracy" (page 149)**

# In class exercise #36:

## This is textbook question #7 part (a) on page 201.

7. The following table summarizes a data set with three attributes $A$, $B$, $C$ and two class labels $+$, $-$. Build a two-level decision tree.

| A | B | C | Number of Instances | |
|---|---|---|---|---|
| | | | + | − |
| T | T | T | 5 | 0 |
| F | T | T | 0 | 20 |
| T | F | T | 20 | 0 |
| F | F | T | 0 | 5 |
| T | T | F | 0 | 0 |
| F | T | F | 25 | 0 |
| T | F | F | 0 | 0 |
| F | F | F | 0 | 25 |

(a) According to the classification error rate, which attribute would be chosen as the first splitting attribute? For each attribute, show the contingency table and the gains in classification error rate.

# Gini Index

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

- This is commonly used in many algorithms like CART and the rpart() function in R

- After the Gini index is computed in each node, the overall value of the Gini index is computed as the weighted average of the Gini index in each node

$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

# Gini Examples for a Single Node

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

| | |
|---|---|
| C1 | **0** |
| C2 | **6** |

**P(C1) = 0/6 = 0  P(C2) = 6/6 = 1**

**Gini = 1 – P(C1)² – P(C2)² = 1 – 0 – 1 = 0**

| | |
|---|---|
| C1 | **1** |
| C2 | **5** |

**P(C1) = 1/6  P(C2) = 5/6**

**Gini = 1 – (1/6)² – (5/6)² = 0.278**

| | |
|---|---|
| C1 | **2** |
| C2 | **4** |

**P(C1) = 2/6  P(C2) = 4/6**

**Gini = 1 – (2/6)² – (4/6)² = 0.444**

# In class exercise #37:

## This is textbook question #3 part (f) on page 200.

3. Consider the training examples shown in Table 4.2 for a binary classification problem.

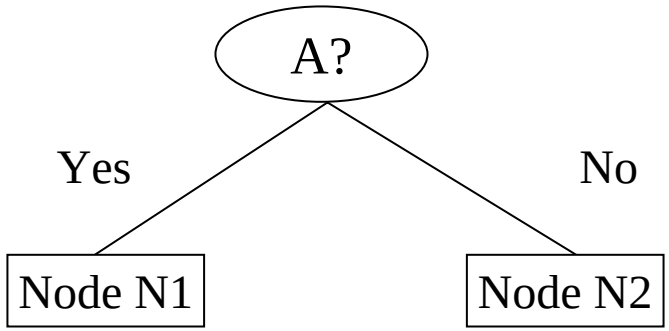
**Table 4.2.** Data set for Exercise 3.

| Instance | $a_1$ | $a_2$ | $a_3$ | Target Class |
|---|---|---|---|---|
| 1 | T | T | 1.0 | + |
| 2 | T | T | 6.0 | + |
| 3 | T | F | 5.0 | − |
| 4 | F | F | 4.0 | + |
| 5 | F | T | 7.0 | − |
| 6 | F | T | 3.0 | − |
| 7 | F | F | 8.0 | − |
| 8 | T | F | 7.0 | + |
| 9 | F | T | 5.0 | − |

(f) What is the best split (between $a_1$ and $a_2$) according to the Gini index?

# Misclassification Error Vs. Gini Index

| | Parent |
|---|---|
| C1 | 7 |
| C2 | 3 |
| Gini = 0.42 | |




**Gini(N1)**
**= 1 – (3/3)² – (0/3)²**
**= 0**

**Gini(N2)**
**= 1 – (4/7)² – (3/7)²**
**= 0.490**

**Gini(Children)**
**= 3/10 * 0**
**+ 7/10 * 0.49**
**= 0.343**

| | N1 | N2 |
|---|---|---|
| C1 | 3 | 4 |
| C2 | 0 | 3 |

- **The Gini index decreases from .42 to .343 while the misclassification error stays at 30%. This illustrates why we often want to use a *surrogate* loss function like the Gini index even if we really only care about misclassification.**

# Entropy

$$Entropy(t) = -\sum_j p(j \mid t) \log_2 p(j \mid t)$$

- Measures purity similar to Gini
- Used in C4.5
- After the entropy is computed in each node, the overall value of the entropy is computed as the weighted average of the entropy in each node as with the Gini index
- The decrease in Entropy is called “information gain” (page 160)

$$GAIN_{split} = Entropy(p) - \left( \sum_{i=1}^{k} \frac{n_i}{n} Entropy(i) \right)$$

# Entropy Examples for a Single Node

| C1 | 0 |
|---|---|
| C2 | 6 |

**P(C1) = 0/6 = 0** **P(C2) = 6/6 = 1**

**Entropy = – 0 log 0 – 1 log 1 = – 0 – 0 = 0**

| C1 | 1 |
|---|---|
| C2 | 5 |

**P(C1) = 1/6** **P(C2) = 5/6**

**Entropy = – (1/6) $\log_2$ (1/6) – (5/6) $\log_2$ (1/6) = 0.65**

| C1 | 2 |
|---|---|
| C2 | 4 |

**P(C1) = 2/6** **P(C2) = 4/6**

**Entropy = – (2/6) $\log_2$ (2/6) – (4/6) $\log_2$ (4/6) = 0.92**

# In class exercise #38:

## This is textbook question #5 part (a) on page 200.

5. Consider the following data set for a binary class problem.

| A | B | Class Label |
|---|---|---|
| T | F | + |
| T | T | + |
| T | T | + |
| T | F | − |
| T | T | + |
| F | F | − |
| F | F | − |
| F | F | − |
| T | T | − |
| T | F | − |

(a) Calculate the information gain when splitting on $A$ and $B$. Which attribute would the decision tree induction algorithm choose?

# In class exercise #39:

**This is textbook question #3 part (c) on page 199. It is part of your homework so we will not do all of it in class.**

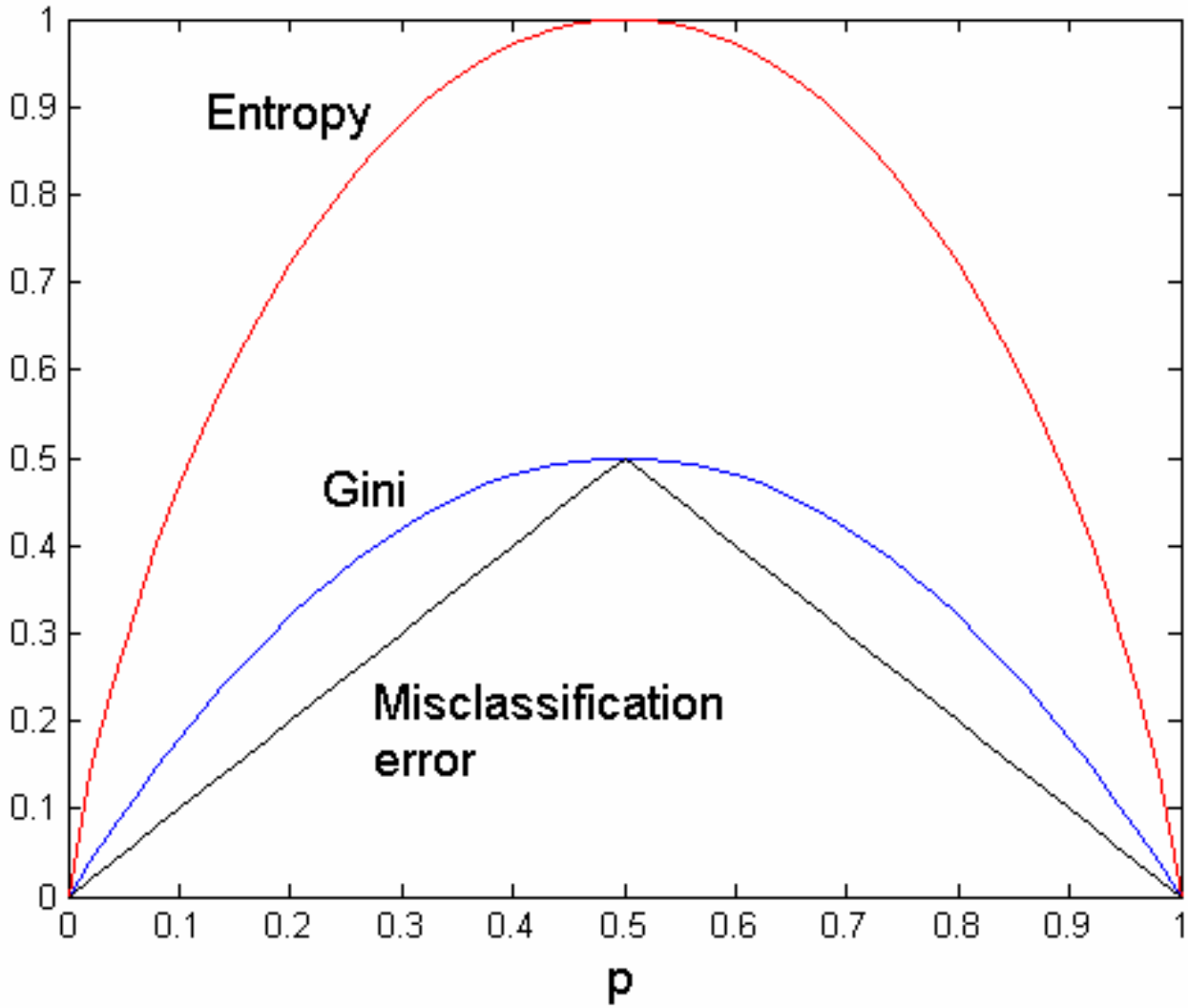
3. Consider the training examples shown in Table 4.2 for a binary classification problem.

**Table 4.2.** Data set for Exercise 3.

| Instance | $a_1$ | $a_2$ | $a_3$ | Target Class |
|---|---|---|---|---|
| 1 | T | T | 1.0 | + |
| 2 | T | T | 6.0 | + |
| 3 | T | F | 5.0 | − |
| 4 | F | F | 4.0 | + |
| 5 | F | T | 7.0 | − |
| 6 | F | T | 3.0 | − |
| 7 | F | F | 8.0 | − |
| 8 | T | F | 7.0 | + |
| 9 | F | T | 5.0 | − |

(c) For $a_3$, which is a continuous attribute, compute the information gain for every possible split.

# A Graphical Comparison



# #3) When to stop splitting

- **This is a subtle business involving model selection. It is tricky because we don't want to overfit or underfit.**

- **One idea would be to monitor misclassification error (or the Gini index or entropy) on the test data set and stop when this begins to increase.**

- **"Pruning" is a more popular technique.**

# Pruning

- **"Pruning" is a popular technique for choosing the right tree size**

- **Your book calls it post-pruning (page 185) to differentiate it from prepruning**

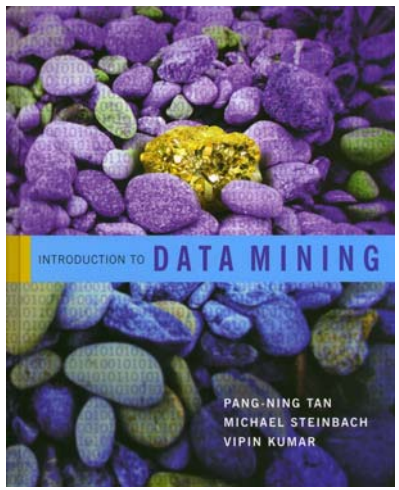
- **With (post-) pruning, a large tree is first grown top-down by one criterion and then trimmed back in a bottom up approach according to a second criterion**

- **Rpart() uses (post-) pruning since it basically follows the CART algorithm**

**(Breiman, Friedman, Olshen, and Stone, 1984,**
***Classification and Regression Trees*)**

# Introduction to Data Mining

by
Tan, Steinbach, Kumar




## Chapter 5: Classification: Alternative Techniques

# The Class Imbalance Problem (Sec. 5.7, p. 204)

- **So far we have treated the two classes equally. We have assumed the same loss for both types of misclassification, used 50% as the cutoff and always assigned the label of the majority class.**

- **This is appropriate if the following three conditions are met**

**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**

# The Class Imbalance Problem (Sec. 5.7, p. 204)

- If any one of these three conditions is not true, it may be desirable to "turn up" or "turn down" the number of observations being classified as the positive class.

- This can be done in a number of ways depending on the classifier.

- Methods for doing this include choosing a probability different from 0.5, using a threshold on some continuous confidence output or under/over-sampling.

# Recall and Precision (page 297)

- **When dealing with class imbalance it is often useful to look at *recall* and *precision* separately**

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

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

# The *F* Measure (page 297)

- ***F*** **combines recall and precision into one number**
- ***F*** **=** $\frac{2rp}{r+p} = \frac{2TP}{2TP+FP+FN}$
- **It equals the harmonic mean of recall and precision**

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

- **Your book calls it the** $F_1$ **measure because it weights both recall and precision equally**
- **See http://en.wikipedia.org/wiki/Information_retrieval**

# The ROC Curve (Sec 5.7.2, p. 298)

- **ROC stands for Receiver Operating Characteristic**

- **Since we can "turn up" or "turn down" the number of observations being classified as the positive class, we can have many different values of true positive rate (TPR) and false positive rate (FPR) for the same classifier.**

$$\text{TPR} = \frac{TP}{TP + FN} \qquad \text{FPR} = \frac{FP}{FP + TN}$$

- **The ROC curve plots TPR on the y-axis and FPR on the x-axis**

# The ROC Curve (Sec 5.7.2, p. 298)

- The ROC curve plots TPR on the y-axis and FPR on the x-axis
- The diagonal represents random guessing
- A good classifier lies near the upper left
- ROC curves are useful for comparing 2 classifiers
- The better classifier will lie on top more often
- The Area Under the Curve (AUC) is often used a metric

# In class exercise #40:

**This is textbook question #17 part (a) on page 322. It is part of your homework so we will not do all of it in class. We will just do the curve for $M_1$.**

You are asked to evaluate the performance of two classification models, $M_1$ and $M_2$. The test set you have chosen contains 26 binary attributes, labeled as $A$ through $Z$.

Table 5.5 shows the posterior probabilities obtained by applying the models to the test set. (Only the posterior probabilities for the positive class are shown). As this is a two-class problem, $P(-) = 1 - P(+)$ and $P(-|A, \ldots, Z) = 1 - P(+|A, \ldots, Z)$. Assume that we are mostly interested in detecting instances from the positive class.

| Instance | True Class | $P(+\|A, \ldots, Z, M_1)$ | $P(+\|A, \ldots, Z, M_2)$ |
|---|---|---|---|
| 1 | + | 0.73 | 0.61 |
| 2 | + | 0.69 | 0.03 |
| 3 | − | 0.44 | 0.68 |
| 4 | − | 0.55 | 0.31 |
| 5 | + | 0.67 | 0.45 |
| 6 | + | 0.47 | 0.09 |
| 7 | − | 0.08 | 0.38 |
| 8 | − | 0.15 | 0.05 |
| 9 | + | 0.45 | 0.01 |
| 10 | − | 0.35 | 0.04 |

(a) Plot the ROC curve for both $M_1$ and $M_2$. (You should plot them on the same graph.) Which model do you think is better? Explain your reasons.

# In class exercise #41:

## This is textbook question #17 part (b) on page 322.

(b) For model $M_1$, suppose you choose the cutoff threshold to be $t = 0.5$. In other words, any test instances whose posterior probability is greater than $t$ will be classified as a positive example. Compute the precision, recall, and F-measure for the model at this threshold value.