DAT630
Classification
Alternative Techniques

Introduction to Data Mining, Chapter 5

09/10/2017

Darío Garigliotti | University of Stavanger

Recall

Attribute set (**x**) → **Classification Model** → Class label (*y*)

Outline

- Alternative classification techniques
  - Rule-based
  - Nearest neighbors
  - Naive Bayes
  - Ensemble methods
- Class imbalance problem
- Multiclass problem

Rule-based classifier

Rule-based Classifier

- Classifying records using a set of **"if… then…"** rules
- Example

  R1: (Give Birth = no) ∧ (Can Fly = yes) → Birds
  R2: (Give Birth = no) ∧ (Live in Water = yes) → Fishes
  R3: (Give Birth = yes) ∧ (Blood Type = warm) → Mammals
  R4: (Give Birth = no) ∧ (Can Fly = no) → Reptiles
  R5: (Live in Water = sometimes) → Amphibians

- R is known as the **rule set**

Classification Rules

- Each classification rule can be expressed in the following way

$$r_i : (Condition_i) \rightarrow y_i$$

rule antecedent (or **precondition**)

rule consequent

Classification Rules

- A rule r **covers** an instance x if the attributes of the instance satisfy the condition of the rule

R1: (Give Birth = no) ∧ (Can Fly = yes) → Birds
R2: (Give Birth = no) ∧ (Live in Water = yes) → Fishes
R3: (Give Birth = yes) ∧ (Blood Type = warm) → Mammals
R4: (Give Birth = no) ∧ (Can Fly = no) → Reptiles
R5: (Live in Water = sometimes) → Amphibians

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
hawk	warm	no	yes	no	?
grizzly bear	warm	yes	no	no	?

Which rules cover the "hawk" and the "grizzly bear"?

Classification Rules

- A rule r **covers** an instance x if the attributes of the instance satisfy the condition of the rule

R1: (Give Birth = no) ∧ (Can Fly = yes) → Birds
R2: (Give Birth = no) ∧ (Live in Water = yes) → Fishes
R3: (Give Birth = yes) ∧ (Blood Type = warm) → Mammals
R4: (Give Birth = no) ∧ (Can Fly = no) → Reptiles
R5: (Live in Water = sometimes) → Amphibians

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
hawk	warm	no	yes	no	?
grizzly bear	warm	yes	no	no	?

The rule R1 covers a hawk => Bird
The rule R3 covers the grizzly bear => Mammal

Rule Coverage and Accuracy

- **Coverage** of a rule
  - Fraction of records that satisfy the antecedent of a rule
- **Accuracy** of a rule
  - Fraction of records that satisfy both the antecedent and consequent of a rule

Tid	Refund	Marital Status	Taxable Income	Class
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

(Status=Single) → No

Coverage = 40%, Accuracy = 50%

How does it work?

R1: (Give Birth = no) ∧ (Can Fly = yes) → Birds
R2: (Give Birth = no) ∧ (Live in Water = yes) → Fishes
R3: (Give Birth = yes) ∧ (Blood Type = warm) → Mammals
R4: (Give Birth = no) ∧ (Can Fly = no) → Reptiles
R5: (Live in Water = sometimes) → Amphibians

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
lemur	warm	yes	no	no	?
turtle	cold	no	no	sometimes	?
dogfish shark	cold	yes	no	yes	?

A lemur triggers rule R3, so it is classified as a mammal
A turtle triggers both R4 and R5
A dogfish shark triggers none of the rules

Properties of the Rule Set

- Mutually exclusive rules
  - Classifier contains mutually exclusive rules if the rules are independent of each other
  - Every record is covered by at most one rule
- Exhaustive rules
  - Classifier has exhaustive coverage if it accounts for every possible combination of attribute values
  - Each record is covered by at least one rule
- These two properties ensure that every record is covered by exactly one rule

When these Properties are not Satisfied

- Rules are not mutually exclusive
  - A record may trigger more than one rule
  - Solution?
    - Ordered rule set
    - Unordered rule set – use voting schemes
- Rules are not exhaustive
  - A record may not trigger any rules
  - Solution?
    - Use a default class (assign the majority class from the training records)

Ordered Rule Set

- Rules are rank ordered according to their priority
  - An ordered rule set is known as a *decision list*
- When a test record is presented to the classifier
  - It is assigned to the class label of the highest ranked rule it has triggered
  - If none of the rules fired, it is assigned to the default class

R1: (Give Birth = no) ∧ (Can Fly = yes) → Birds
R2: (Give Birth = no) ∧ (Live in Water = yes) → Fishes
R3: (Give Birth = yes) ∧ (Blood Type = warm) → Mammals
R4: (Give Birth = no) ∧ (Can Fly = no) → Reptiles
R5: (Live in Water = sometimes) → Amphibians

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
turtle	cold	no	no	sometimes	?

Rule Ordering Schemes

- Rule-based ordering
  - Individual rules are ranked based on some quality measure (e.g., accuracy, coverage)
- Class-based ordering
  - Rules that belong to the same class appear together
  - Rules are sorted on the basis of their class information (e.g., total description length)
  - The relative order of rules within a class does not matter

Rule Ordering Schemes

Rule-based Ordering

(Refund=Yes) ==> No

(Refund=No, Marital Status={Single,Divorced}, Taxable Income<80K) ==> No

(Refund=No, Marital Status={Single,Divorced}, Taxable Income>80K) ==> Yes

(Refund=No, Marital Status={Married}) ==> No

Class-based Ordering

(Refund=Yes) ==> No

(Refund=No, Marital Status={Single,Divorced}, Taxable Income<80K) ==> No

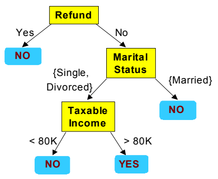
(Refund=No, Marital Status={Married}) ==> No

(Refund=No, Marital Status={Single,Divorced}, Taxable Income>80K) ==> Yes

How to Build a Rule-based Classifier?

- Direct Method
  - Extract rules directly from data
- Indirect Method
  - Extract rules from other classification models (e.g. decision trees, neural networks, etc)

From Decision Trees To Rules



Classification Rules

(Refund=Yes) ==> No

(Refund=No, Marital Status={Single,Divorced}, Taxable Income<80K) ==> No

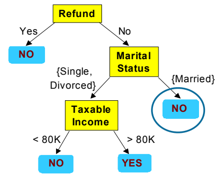
(Refund=No, Marital Status={Single,Divorced}, Taxable Income>80K) ==> Yes

(Refund=No, Marital Status={Married}) ==> No

Rules are mutually exclusive and exhaustive

Rule set contains as much information as the tree

Rules Can Be Simplified



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

Initial Rule: **(Refund=No) $\wedge$ (Status=Married) $\rightarrow$ No**

Simplified Rule: **(Status=Married) $\rightarrow$ No**

Summary

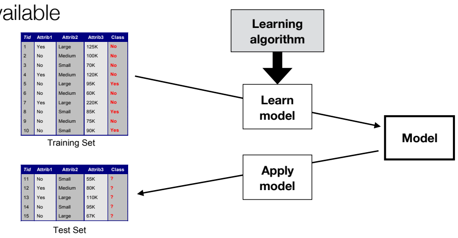
- Expressiveness is almost equivalent to that of a decision tree
- Generally used to produce descriptive models that are easy to interpret, but gives comparable performance to decision tree classifiers
- The class-based ordering approach is well suited for handling data sets with imbalanced class distributions

Exercise

Nearest Neighbors

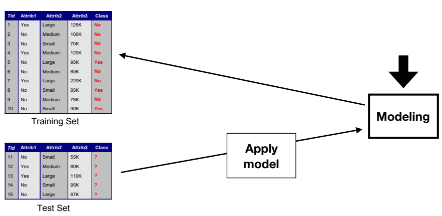
So far

- **Eager learners**
  - Decision trees, rule-base classifiers
  - Learn a model as soon as the training data becomes available

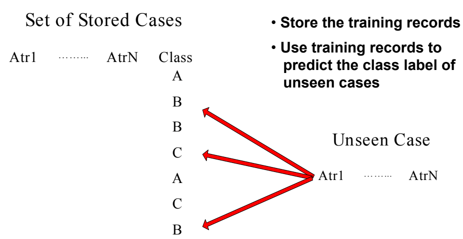


Opposite strategy

- **Lazy learners**
  - Delay the process of modeling the data until it is needed to classify the test examples



Instance-Based Classifiers



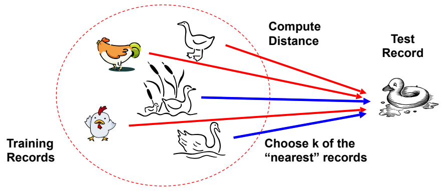
- **Store the training records**
- **Use training records to predict the class label of unseen cases**

Instance Based Classifiers

- Rote-learner
  - Memorizes entire training data and performs classification only if attributes of record match one of the training examples exactly
- Nearest neighbors
  - Uses k “closest” points (nearest neighbors) for performing classification

Nearest neighbors

- Basic idea
  - "If it walks like a duck, quacks like a duck, then it’s probably a duck"

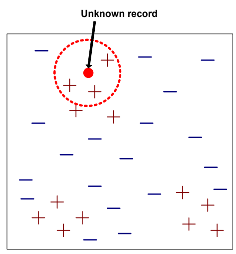


Nearest-Neighbor Classifiers

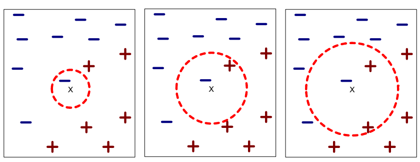
- Requires three things
  - The set of stored records
  - Distance Metric to compute distance between records
  - The value of k, the number of nearest neighbors to retrieve

Nearest-Neighbor Classifiers

- To classify an unknown record
  - Compute distance to other training records
  - Identify k-nearest neighbors
  - Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)



Definition of Nearest Neighbor



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

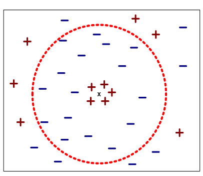
K-nearest neighbors of a record x are data points that have the k smallest distance to x

Choices to make

- Compute distance between two points
  - E.g., Euclidean distance
  - See Chapter 2
- Determine the class from nearest neighbor list
  - Take the majority vote of class labels among the k-nearest neighbors
  - Weigh the vote according to distance
- Choose the value of k

Choosing the value of k

- If k is too small, sensitive to noise points
- If k is too large, neighborhood may include points from other classes



Summary

- Part of a more general technique called instance-based learning
  - Use specific training instances to make predictions without having to maintain an abstraction (model) derived from data
- Because there is no model building, classifying a test example can be quite expensive
- Nearest-neighbors make their predictions based on local information
  - Susceptible to noise

Bayes Classifier

Bayes Classifier

- In many applications the relationship between the attribute set and the class variable is **non-deterministic**
  - The label of the test record cannot be predicted with certainty even if it was seen previously during training
- A probabilistic framework for solving classification problems
  - Treat **X** and Y as random variables and capture their relationship probabilistically using P(Y|**X**)

Example

- Football game between teams A and B
  - Team A won 65% team B won 35% of the time
  - Among the games Team A won, 30% when game hosted by B
  - Among the games Team B won, 75% when B played home
- Which team is more likely to win if the game is hosted by Team B?

Probability Basics

- Conditional probability

$$P(X,Y) = P(X|Y)P(Y) = P(Y|X)P(X)$$

- Bayes' theorem

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

Example

- Probability Team A wins: P(win=A) = 0.65
- Probability Team B wins: P(win=B) = 0.35
- Probability Team A wins when B hosts: P(hosted=B|win=A) = 0.3
- Probability Team B wins when playing at home: P(hosted=B|win=B) = 0.75
- Who wins the next game that is hosted by B?
  P(win=B|hosted=B) = ?
  P(win=A|hosted=B) = ?

Solution

- Using:

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

- P(win=B|hosted=B) = 0.5738
- P(win=A|hosted=B) = 0.4262
- See book page 229

Bayes' Theorem for Classification

Class-conditional probability   Prior probability

$$P(Y|\mathbf{X}) = \frac{P(\mathbf{X}|Y)P(Y)}{P(\mathbf{X})}$$

Posterior probability   The evidence

Bayes' Theorem for Classification

Class-conditional probability   Prior probability

$$P(Y|\mathbf{X}) = \frac{P(\mathbf{X}|Y)P(Y)}{P(\mathbf{X})}$$

Posterior probability

The evidence
Constant (same for all classes), **can be ignored**

Bayes' Theorem for Classification

Prior probability

Can be computed from training data (fraction of records that belong to each class)

Class-conditional probability

$$P(Y|\mathbf{X}) = \frac{P(\mathbf{X}|Y)P(Y)}{P(\mathbf{X})}$$

Posterior probability

The evidence

Bayes' Theorem for Classification

Class-conditional probability

Two methods: Naive Bayes, Bayesian belief network

Prior probability

$$P(Y|\mathbf{X}) = \frac{P(\mathbf{X}|Y)P(Y)}{P(\mathbf{X})}$$

Posterior probability

The evidence

Naive Bayes

Estimation

- Mind that **X** is a vector

$$\mathbf{X} = \{X_1, \dots, X_n\}$$

- Class-conditional probability

$$P(\mathbf{X}|Y) = P(X_1, \dots, X_n|Y)$$

- "Naive" assumption: attributes are independent

$$P(\mathbf{X}|Y) = \prod_{i=1}^{n} P(X_i|Y)$$

Naive Bayes Classifier

- Probability that X belongs to class Y

$$P(Y|\mathbf{X}) \propto P(Y) \prod_{i=1}^{n} P(X_i|Y)$$

- Target label for record X

$$y = \arg\max_{y_j} P(Y = y_j) \prod_{i=1}^{n} P(X_i|Y = y_j)$$

Estimating class-conditional probabilities

- **Categorical attributes**
  - The fraction of training instances in class Y that have a particular attribute value x_i

$$P(X_i = x_i|Y = y) = \frac{n_c}{n}$$

number of training instances where $X_i=x_i$ and Y=y

number of training instances where Y=y

- **Continuous attributes**
  - Discretizing the range into bins
  - Assuming a certain probability distribution

Conditional probabilities for categorical attributes

- The fraction of training instances in class Y that have a particular attribute value X_i
- P(Status=Married|No)=?
- P(Refund=Yes|Yes)=?

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

Conditional probabilities for continuous attributes

- Discretize the range into bins, or
- Assume a certain form of probability distribution
  - Gaussian (normal) distribution is often used

$$P(X_i = x_i|Y = y_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} \exp^{-\frac{(x_i-\mu_{ij})^2}{2\sigma_{ij}^2}}$$

  - The parameters of the distribution are estimated from the training data (from instances that belong to class y_j)
  - sample mean μ_{ij} and variance σ_{ij}^2

Example

Tid	Refund	Marital Status	Taxable Income	Class
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

Example

X={Refund=No, Marital st.=Married, Income=120K}

Tid	Refund	Marital Status	Taxable Income	Class
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

	P(C)	P(Refund=x|Y)		P(Marital=x|Y)			Ann. income	
		No	Yes	Single	Divorced	Married	mean	var
class=No	7/10	4/7	3/7	2/7	1/7	4/7	110	2975
class=Yes	3/10	3/3	3/3	2/3	1/3	0/3	90	25

Example
classifying a new instance

X={Refund=No, Marital st.=Married, Income=120K}

	P(C)	P(Refund=x|Y)		P(Marital=x|Y)			Ann. income	
		No	Yes	Single	Divorced	Married	mean	var
class=No	7/10	4/7	3/7	2/7	1/7	4/7	110	2975
class=Yes	3/10	3/3	3/3	2/3	1/3	0/3	90	25

P(Class=No|X) = P(Class=No) **7/10**
× P(Refund=No|Class=No) **4/7**
× P(Marital=Married| Class=No) **4/7**
× P(Income=120K| Class=No) **0.0072**

Example
classifying a new instance

X={Refund=No, Marital st.=Married, Income=120K}

	P(C)	P(Refund=x|Y)		P(Marital=x|Y)			Ann. income	
		No	Yes	Single	Divorced	Married	mean	var
class=No	7/10	4/7	3/7	2/7	1/7	4/7	110	2975
class=Yes	3/10	3/3	0/3	2/3	1/3	0/3	90	25

P(Class=Yes|X) = P(Class=Yes) **3/10**
× P(Refund=No|Class=Yes) **3/3**
× P(Marital=Married| Class=Yes) **0/3**
× P(Income=120K| Class=Yes) **$1.2 \cdot 10^{-9}$**

Can anything go wrong?

$$P(Y|\mathbf{X}) \propto P(Y) \prod_{i=1}^{n} P(X_i|Y)$$

What if this probability is zero?

- If one of the conditional probabilities is zero, then the entire expression becomes zero!

Probability estimation

- **Original**

$$P(X_i = x_i | Y = y) = \frac{n_c}{n}$$

number of training instances where $X_i=x_i$ and Y=y

number of training instances where Y=y

- **Laplace smoothing**

$$P(X_i = x_i | Y = y) = \frac{n_c + 1}{n + c}$$

c is the number of classes

Probability estimation (2)

- **M-estimate**

$$P(X_i = x_i | Y = y) = \frac{n_c + mp}{n + m}$$

- **p** can be regarded as the prior probability
- **m** is called equivalent sample size which determines the trade-off between the observed probability n_c/n and the prior probability p
- E.g., p=1/3 and m=3

Summary

- Robust to isolated noise points
- Handles missing values by ignoring the instance during probability estimate calculations
- Robust to irrelevant attributes
- Independence assumption may not hold for some attributes

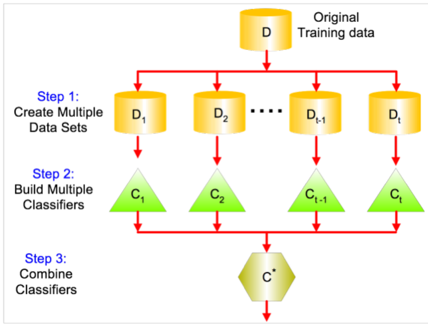
Exercise

Ensemble Methods

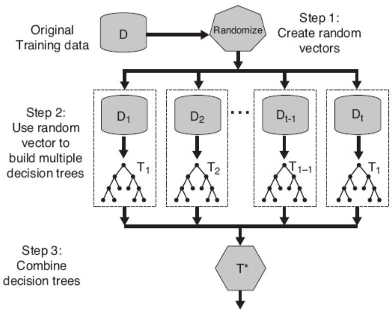
Ensemble Methods

- Construct a set of classifiers from the training data
- Predict class label of previously unseen records by aggregating predictions made by multiple classifiers

General Idea



Random Forests



Class Imbalance Problem

Class Imbalance Problem

- Data sets with imbalanced class distributions are quite common in real-world applications
  - E.g., credit card fraud detection
- Correct classification of the rare class has often greater value than a correct classification of the majority class
- The accuracy measure is not well suited for imbalanced data sets
- **We need alternative measures**

Confusion Matrix

		Predicted class	
		Positive	Negative
Actual class	Positive	True Positives (TP)	False Negatives (FN)
	Negative	False Positives (FP)	True Negatives (TN)

Additional Measures

- **True positive rate** (or **sensitivity**)
  - Fraction of positive examples predicted correctly

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

- **True negative rate** (or **specificity**)
  - Fraction of negative examples predicted correctly

$$TNR = \frac{TN}{TN + FP}$$

Additional Measures

- **False positive rate**
  - Fraction of negative examples predicted as positive

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

- **False negative rate**
  - Fraction of positive examples predicted as negative

$$FNR = \frac{FN}{TP + FN}$$

Additional Measures

- **Precision**
  - Fraction of positive records among those that are classified as positive

$$P = \frac{TP}{TP + FP}$$

- **Recall**
  - Fraction of positive examples correctly predicted (same as the true positive rate)

$$R = \frac{TP}{TP + FN}$$

Additional Measures

- **F1-measure**
  - Summarizing precision and recall into a single number
  - Harmonic mean between precision and recall

$$F1 = \frac{2RP}{R + P}$$

Multiclass Problem

Multiclass Classification

- Many of the approaches are originally designed for binary classification problems
- Many real-world problems require data to be divided into more than two categories
- Two approaches
  - One-against-rest (1-r)
  - One-against-one (1-1)
- Predictions need to be combined in both cases

One-against-rest

- $Y=\{y_1, y_2, \ldots y_K\}$ classes
- For each class y_i
  - Instances that belong to y_i are positive examples
  - All other instances are negative examples
- Combining predictions
  - If an instance is classified positive, the positive class gets a vote
  - If an instance is classified negative, all classes except for the positive class receive a vote

Example

target class → y_1

total votes	
y_1	●●●●
y_2	●●
y_3	●●
y_4	●●

- 4 classes, $Y=\{y_1, y_2, y_3, y_4\}$
- Classifying a given test instance

y_1	+	●
y_2	-	
y_3	-	
y_4	-	
class	+	

y_1	-	●
y_2	+	
y_3	-	●
y_4	-	●
class	-	

y_1	-	●
y_2	-	●
y_3	+	
y_4	-	●
class	-	

y_1	-	●
y_2	-	●
y_3	-	●
y_4	+	
class	-	

One-against-one

- $Y=\{y_1, y_2, \dots y_K\}$ classes
- Construct a binary classifier for each pair of classes (y_i, y_j)
  - $K(K-1)/2$ binary classifiers in total
- Combining predictions
  - The positive class receives a vote in each pairwise comparison

Example

- 4 classes, $Y=\{y_1, y_2, y_3, y_4\}$
- Classifying a given test instance

target class

	total votes	
→	y_1	●●
	y_2	●
	y_3	●
→	y_4	●●

y_1	+	●
y_2	-	
class	+	

y_1	+	●
y_3	-	
class	+	

y_1	+	
y_4	-	●
class	-	

y_2	+	●
y_3	-	
class	+	

y_2	+	
y_4	-	●
class	-	

y_3	+	●
y_4	-	
class	+	