

For written notes on this lecture, please read chapter 3 of *The Practical Bioinformatician*. Alternatively, please read "Rule-Based Data Mining Methods for Classification Problems in Biomedical Domains", a tutorial at *PKDD04* by Jinyan Li and Limsoon Wong, September 2004. <http://www.comp.nus.edu.sg/~wongli/talks/pkdd04/>

CS2220: Introduction to Computational Biology Lecture 2: Essence of Knowledge Discovery

Limsoon Wong



Outline



- **Overview of Supervised Learning**
 - Decision Trees
- **Decision Trees Ensembles**
 - Bagging
- **Other Methods**
 - K-Nearest Neighbour
 - Support Vector Machines
 - Bayesian Approach
 - Hidden Markov Models

Copyright 2011 © Limsoon Wong

Overview of Supervised Learning



Supervised Learning



- Also called **classification**
- Learn from past experience, and use the learned knowledge to classify new data
- Knowledge learned by **intelligent algorithms**
- **Examples:**
 - Clinical diagnosis for patients
 - Cell type classification

Copyright 2011 © Limsoon Wong

Data



- **Classification application involves > 1 class of data. E.g.,**
 - Normal vs disease cells for a diagnosis problem
- **Training data is a set of instances (samples, points) with known class labels**
- **Test data is a set of instances whose class labels are to be predicted**

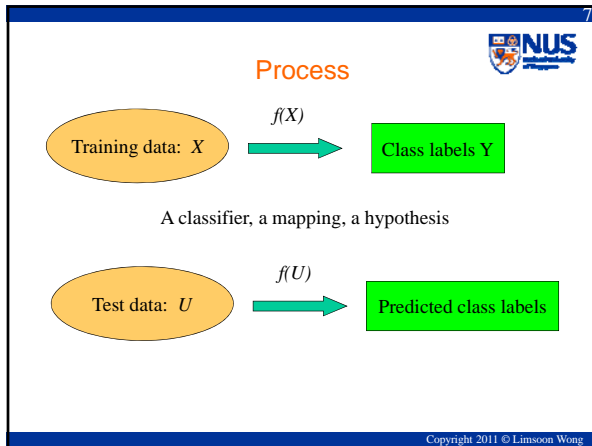
Copyright 2011 © Limsoon Wong

Typical Notations



- **Training data**
 $\{\langle x_1, y_1 \rangle, \langle x_2, y_2 \rangle, \dots, \langle x_m, y_m \rangle\}$
 where x_j are n -dimensional vectors and y_j are from a discrete space Y .
 E.g., $Y = \{\text{normal, disease}\}$
- **Test data**
 $\{\langle u_1, ? \rangle, \langle u_2, ? \rangle, \dots, \langle u_k, ? \rangle, \}$

Copyright 2011 © Limsoon Wong



8

Relational Representation of Gene Expression Data

n features (order of 1000)

		gene ₁	gene ₂	gene ₃	gene ₄	...	gene _n	class
m samples	x ₁₁	x ₁₂	x ₁₃	x ₁₄	...	x _{1n}	→P	
	x ₂₁	x ₂₂	x ₂₃	x ₂₄	...	x _{2n}	→N	
	x ₃₁	x ₃₂	x ₃₃	x ₃₄	...	x _{3n}	→P	
	x _{m1}	x _{m2}	x _{m3}	x _{m4}	...	x _{mn}	→N	

Copyright 2011 © Limsoon Wong

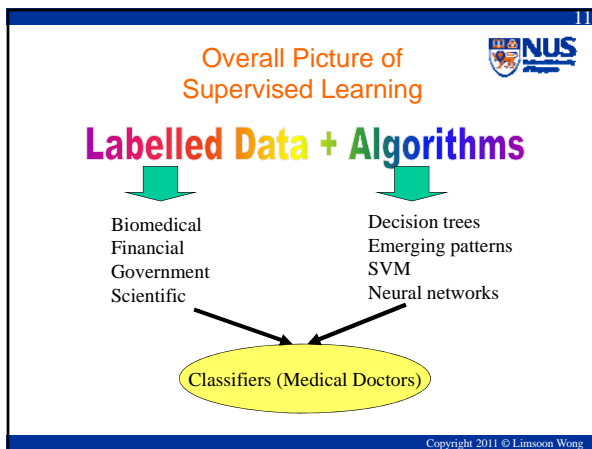
- 9
- ### Features (aka Attributes)
- **Categorical features**
 - color = {red, blue, green}
 - **Continuous or numerical features**
 - gene expression
 - age
 - blood pressure
 - **Discretization**
- Copyright 2011 © Limsoon Wong

10

An Example

Outlook	Temp	Humidity	Windy	class
Sunny	75	70	true	Play
Sunny	80	90	true	Don't
Sunny	85	85	false	Don't
Sunny	72	95	true	Don't
Sunny	69	70	false	Play
Overcast	72	90	true	Play
Overcast	83	78	false	Play
Overcast	64	65	true	Play
Overcast	81	75	false	Play
Rain	71	80	true	Don't
Rain	65	70	true	Don't
Rain	75	80	false	Play
Rain	68	80	false	Play
Rain	70	96	false	Play

Copyright 2011 © Limsoon Wong



- 12
- ### Recap: Evaluation of a Classifier
- Performance on independent blind test data
 - K-fold cross validation: Given a dataset, divide it into k even parts, $k-1$ of them are used for training, and the rest one part treated as test data
 - LOOCV, a special case of K-fold CV
 - Accuracy, error rate
 - False positive rate, false negative rate, sensitivity, specificity, precision
- Copyright 2011 © Limsoon Wong

Requirements of Biomedical Classification

- High accuracy/sensitivity/specificity/precision
- High comprehensibility

13

Copyright 2011 © Limsoon Wong

Importance of Rule-Based Methods

- Systematic selection of a small number of features used for the decision making
- ⇒ Increase comprehensibility of the knowledge patterns
- C4.5 and CART are two commonly used rule induction algorithms---a.k.a. decision tree induction algorithms

14

Copyright 2011 © Limsoon Wong

Structure of Decision Trees

- If $x_1 > a_1$ & $x_2 > a_2$, then it's A class
- C4.5, CART, two of the most widely used
- Easy interpretation, but accuracy generally unattractive

15

Copyright 2011 © Limsoon Wong

Elegance of Decision Trees

Every path from root to a leaf forms a decision rule

16

Copyright 2011 © Limsoon Wong

Brief History of Decision Trees

```

    graph TD
      CLS[CLS (Hunt et al. 1966) --- cost driven] --> CART[CART (Breiman et al. 1984) --- Gini Index]
      CLS --> ID3[ID3 (Quinlan, 1986) --- Information-driven]
      ID3 --> C45[C4.5 (Quinlan, 1993) --- Gain ratio + Pruning ideas]
  
```

17

Copyright 2011 © Limsoon Wong

A Simple Dataset

Outlook	Temp	Humidity	Windy	class
Sunny	75	70	true	Play
Sunny	80	90	true	Don't
Sunny	85	85	false	Don't
Sunny	72	95	true	Don't
Sunny	69	70	false	Play
Overcast	72	90	true	Play
Overcast	83	78	false	Play
Overcast	64	65	true	Play
Overcast	81	75	false	Play
Rain	71	80	true	Don't
Rain	65	70	true	Don't
Rain	75	80	false	Play
Rain	68	80	false	Play
Rain	70	96	false	Play

9 Play samples
5 Don't
A total of 14.

18

Copyright 2011 © Limsoon Wong

19

A Decision Tree

- Construction of a tree is equiv to determination of root node of the tree and root nodes of its sub-trees

Exercise: What is the accuracy of this tree?

Copyright 2011 © Limsoon Wong

20

An Example

Source: Anthony Tung

Copyright 2011 © Limsoon Wong

21

Most Discriminatory Feature

- Every feature can be used to partition the training data
- If the partitions contain a pure class of training instances, then this feature is most discriminatory

Copyright 2011 © Limsoon Wong

22

Example of Partitions

- Categorical feature**
 - Number of partitions of the training data is equal to the number of values of this feature
- Numerical feature**
 - Two partitions

Copyright 2011 © Limsoon Wong

23

Instance #	Outlook	Temp	Humidity	Windy	class
1	Sunny	75	70	true	Play
2	Sunny	80	90	true	Don't
3	Sunny	85	85	false	Don't
4	Sunny	72	95	true	Don't
5	Sunny	69	70	false	Play
6	Overcast	72	90	true	Play
7	Overcast	83	78	false	Play
8	Overcast	64	65	true	Play
9	Overcast	81	75	false	Play
10	Rain	71	80	true	Don't
11	Rain	65	70	true	Don't
12	Rain	75	80	false	Play
13	Rain	68	80	false	Play
14	Rain	70	96	false	Play

Copyright 2011 © Limsoon Wong

24

A categorical feature is partitioned based on its number of possible values

- Outlook = sunny: 1,2,3,4,5 (P,D,D,D,P)
- Outlook = overcast: 6,7,8,9 (P,P,P,P)
- Outlook = rain: 10,11,12,13,14 (D, D, P, P, P)

Copyright 2011 © Limsoon Wong

25

Total 14 training instances

A numerical feature is generally partitioned by choosing a "cutting point"

Temperature ≤ 70 (5, 8, 11, 13, 14 P, P, D, P, P)

Temperature > 70 (1, 2, 3, 4, 6, 7, 9, 10, 12 P, D, D, D, P, P, P, D, P)

Copyright 2011 © Limsoon Wong

26

Steps of Decision Tree Construction

- Select the "best" feature as root node of the whole tree
- Partition dataset into subsets using this feature so that the subsets are as "pure" as possible
- After partition by this feature, select the best feature (wrt the subset of training data) as root node of this sub-tree
- Recursively, until the partitions become pure or almost pure

Copyright 2011 © Limsoon Wong

27

Let's Construct a Decision Tree Together

Outlook	Temp	Humidity	Windy	class
Sunny	75	70	true	Play
Sunny	80	90	true	Don't
Sunny	85	85	false	Don't
Sunny	72	95	true	Don't
Sunny	69	70	false	Play
Overcast	72	90	true	Play
Overcast	83	78	false	Play
Overcast	64	65	true	Play
Overcast	81	75	false	Play
Rain	71	80	true	Don't
Rain	65	70	true	Don't
Rain	75	80	false	Play
Rain	68	80	false	Play
Rain	70	96	false	Play

Ask the class to pick root node and construct the tree recursively with them... How good is that tree?

Copyright 2011 © Limsoon Wong

28

Three Measures to Evaluate Which Feature is Best

- Gini index
- Information gain
- Information gain ratio

Copyright 2011 © Limsoon Wong

29

Gini Index

$$gini(S) = \frac{\text{diff of two arbitrary specimen in } S}{\text{mean specimen in } S}$$

$$= \text{prob}(\text{getting two specimen of diff class in } S)$$

$$= 1 - \text{prob}(\text{getting two specimen of same class in } S)$$

$$= 1 - \sum_i \text{prob}(\text{getting specimen of class } i \text{ in } S)^2$$

- Gini index is the expected value of the ratio of the diff of two arbitrary specimens to the mean value of all specimens
- Closer to 1 means similar to "background distribution". Closer to 0, means feature is "unexpected"

Copyright 2011 © Limsoon Wong

30

Gini Index

Let $U = \{C_1, \dots, C_k\}$ be all the classes. Suppose we are currently at a node and D is the set of those samples that have been moved to this node. Let f be a feature and $d[f]$ be the value of the feature f in a sample d . Let S be a range of values that the feature f can take. Then the Gini index for f in D for the range S is defined as

$$gini_f^D(S) = 1 - \sum_{c_i \in U} \left(\frac{|\{d \in D \mid d \in C_i, d[f] \in S\}|}{|D|} \right)^2$$

The purity of a split of the value range S of an attribute f by some split-point into subranges S_1 and S_2 is then defined as

$$gini_f^D(S_1, S_2) = \sum_{s \in \{S_1, S_2\}} \frac{|\{d \in D \mid d[f] \in s\}|}{|D|} * gini_f^D(s)$$

we choose the feature f and the split point p that minimizes $gini_f^D(S_1, S_2)$ over all possible alternative features and split-points.

Copyright 2011 © Limsoon Wong

31

Gini Index of Outlook

Outlook	Temp	Humidity	Windy	class
Sunny	75	70	true	Play
Sunny	80	90	true	Don't
Sunny	85	85	false	Don't
Sunny	72	95	true	Don't
Sunny	69	70	false	Play
Overcast	72	90	true	Play
Overcast	83	78	false	Play
Overcast	64	65	true	Play
Overcast	81	75	false	Play
Rain	71	80	true	Don't
Rain	65	70	true	Don't
Rain	75	80	false	Play
Rain	68	80	false	Play
Rain	70	96	false	Play

$$gini^D(S) = 1 - \sum_{C_i \in \mathcal{C}} \left(\frac{|d \in D \mid d \in C_i, d \in S|}{|D|} \right)^2$$

$$gini^D(S_1, S_2) = \sum_{S \in \{S_1, S_2\}} \frac{|d \in D \mid d \in S|}{|D|} \cdot gini^D(S)$$

- $gini(\text{Sunny}) = 1 - (2/5)^2 - (3/5)^2 = 0.48$
- $gini(\text{Overcast}) = 1 - (4/4)^2 - (0/5)^2 = 0$
- $gini(\text{Rain}) = 1 - (3/5)^2 - (2/5)^2 = 0.48$
- $gini(\text{Outlook}) = 5/14 * 0.48 + 4/14 * 0 + 5/14 * 0.48 = 0.34$

Copyright 2011 © Limsoon Wong

32

Characteristics of C4.5/CART Trees

- Single coverage of training data (elegance)
- Divide-and-conquer splitting strategy
- Fragmentation problem \Rightarrow Locally reliable but globally insignificant rules

- Miss many globally significant rules; mislead system

Copyright 2011 © Limsoon Wong

33

Example Use of Decision Tree Methods: Proteomics Approaches to Biomarker Discovery

- In prostate and bladder cancers (Adam et al. *Proteomics*, 2001)
- In serum samples to detect breast cancer (Zhang et al. *Clinical Chemistry*, 2002)
- In serum samples to detect ovarian cancer (Petricoin et al. *Lancet*; Li & Rao, *PAKDD* 2004)

Copyright 2011 © Limsoon Wong

Decision Tree Ensembles

35

Motivating Example

- h_1, h_2, h_3 are indep classifiers w/ accuracy = 60%
- C_1, C_2 are the only classes
- t is a test instance in C_1
- $h(t) = \operatorname{argmax}_{C \in \{C_1, C_2\}} |\{h_j \in \{h_1, h_2, h_3\} \mid h_j(t) = C\}|$
- Then $\operatorname{prob}(h(t) = C_1)$

$$= \operatorname{prob}(h_1(t)=C_1 \ \& \ h_2(t)=C_1 \ \& \ h_3(t)=C_1) +$$

$$\operatorname{prob}(h_1(t)=C_1 \ \& \ h_2(t)=C_1 \ \& \ h_3(t)=C_2) +$$

$$\operatorname{prob}(h_1(t)=C_1 \ \& \ h_2(t)=C_2 \ \& \ h_3(t)=C_1) +$$

$$\operatorname{prob}(h_1(t)=C_2 \ \& \ h_2(t)=C_1 \ \& \ h_3(t)=C_1)$$

$$= 60\% * 60\% * 60\% + 60\% * 60\% * 40\% +$$

$$60\% * 40\% * 60\% + 40\% * 60\% * 60\% = 64.8\%$$

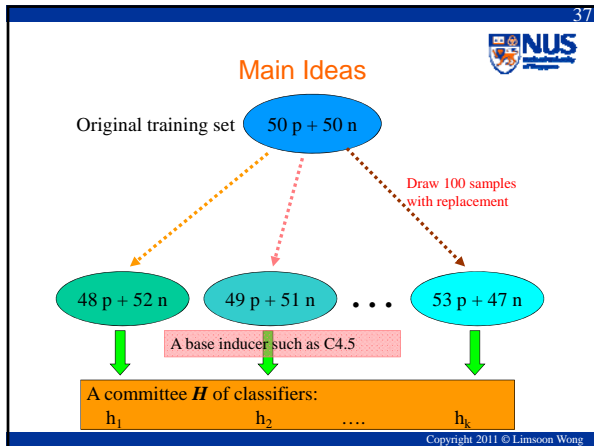
Copyright 2011 © Limsoon Wong

36

Bagging

- Proposed by Breiman (1996)
- Also called **Bootstrap aggregating**
- Make use of randomness injected to training data

Copyright 2011 © Limsoon Wong



38

Decision Making by Bagging

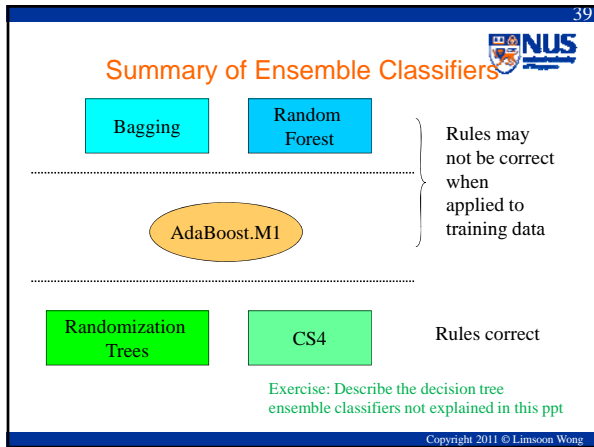
Given a new test sample T

$$bagged(T) = \operatorname{argmax}_{C_j \in U} |\{h_i \in H \mid h_i(T) = C_j\}|$$

where $U = \{C_1, \dots, C_r\}$

Exercise: What does the above formula mean?

Copyright 2011 © Limsoon Wong



Other Machine Learning Approaches

- 41
- Outline**
- K-Nearest Neighbour
 - Support Vector Machines
 - Bayesian Approach
 - Hidden Markov Models
- Exercise: Name and describe one other commonly used machine learning method
- Copyright 2011 © Limsoon Wong

K-Nearest Neighbours

43

How kNN Works

- Given a new case
- Find k “nearest” neighbours, i.e., k most similar points in the training data set
- Assign new case to the same class to which most of these neighbours belong

- A common “distance” measure betw samples x and y is

$$\sqrt{\sum_f (x_f - y_f)^2}$$
 where f ranges over features of the samples

Exercise: What does the formula above mean?

Copyright 2011 © Limsoon Wong

44

Illustration of kNN (k=8)

Image credit: Zaki

Copyright 2011 © Limsoon Wong

45

Some Issues

- Simple to implement
- But need to compare new case against all training cases
 - ⇒ May be slow during prediction
- No need to train
- But need to design distance measure properly
 - ⇒ May need expert for this
- Can't explain prediction outcome
 - ⇒ Can't provide a model of the data

Copyright 2011 © Limsoon Wong

46

Example Use of kNN: Ovarian Cancer Diagnosis Based on SELDI Proteomic Data

- Li et al, *Bioinformatics* 20:1638-1640, 2004
- Use kNN to diagnose ovarian cancers using proteomic spectra
- Data set is from Petricoin et al., *Lancet* 359:572-577, 2002

Number of top-ranked m/z ratios

percentage of correct prediction

Image credit: Zaki

Copyright 2011 © Limsoon Wong

Support Vector Machines

48

Basic Idea

Image credit: Zien

- (a) Linear separation not possible w/o errors
- (b) Better separation by nonlinear surfaces in input space
- (c) Nonlinear surface corr to linear surface in feature space.

Map from input to feature space by “kernel” function Φ

⇒ “Linear learning machine” + kernel function as classifier

Copyright 2011 © Limsoon Wong

49

Linear Learning Machines

- Hyperplane separating the x's and o's points is given by $(W \cdot X) + b = 0$, with $(W \cdot X) = \sum_j W[j] \cdot X[j]$
- \Rightarrow Decision function is $\text{llm}(X) = \text{sign}((W \cdot X) + b)$

Copyright 2011 © Limsoon Wong

50

Linear Learning Machines

- Solution is a linear combination of training points X_k with labels Y_k
 $W[j] = \sum_k \alpha_k \cdot Y_k \cdot X_k[j]$,
 with $\alpha_k > 0$, and $Y_k = \pm 1$
- $\Rightarrow \text{llm}(X) = \text{sign}(\sum_k \alpha_k \cdot Y_k \cdot (X_k \cdot X) + b)$

↑

"data" appears only in dot product!

Copyright 2011 © Limsoon Wong

51

Kernel Function

- $\text{llm}(X) = \text{sign}(\sum_k \alpha_k \cdot Y_k \cdot (X_k \cdot X) + b)$

- $\text{svm}(X) = \text{sign}(\sum_k \alpha_k \cdot Y_k \cdot (\Phi X_k \cdot \Phi X) + b)$
- $\Rightarrow \text{svm}(X) = \text{sign}(\sum_k \alpha_k \cdot Y_k \cdot K(X_k, X) + b)$
 where $K(X_k, X) = (\Phi X_k \cdot \Phi X)$

Copyright 2011 © Limsoon Wong

52

Kernel Function

- $\text{svm}(X) = \text{sign}(\sum_k \alpha_k \cdot Y_k \cdot K(X_k, X) + b)$
- $\Rightarrow K(A, B)$ can be computed w/o computing Φ
- In fact replace it w/ lots of more "powerful" kernels besides $(A \cdot B)$. E.g.,
 - $K(A, B) = (A \cdot B)^d$
 - $K(A, B) = \exp(-\|A - B\|^2 / (2 \cdot \sigma))$, ...

Copyright 2011 © Limsoon Wong

53

How SVM Works

- $\text{svm}(X) = \text{sign}(\sum_k \alpha_k \cdot Y_k \cdot K(X_k, X) + b)$
- To find α_k is a quadratic programming problem
 $\text{max: } \sum_k \alpha_k - 0.5 \cdot \sum_k \sum_h \alpha_k \cdot \alpha_h \cdot Y_k \cdot Y_h \cdot K(X_k, X_h)$
 subject to: $\sum_k \alpha_k \cdot Y_k = 0$
 and for all $\alpha_k, C \geq \alpha_k \geq 0$
- To find b , estimate by averaging
 $Y_h - \sum_k \alpha_k \cdot Y_k \cdot K(X_h, X_k)$
 for all $\alpha_k \geq 0$

Copyright 2011 © Limsoon Wong


54

Example Use of SVM: Recognition of Protein Translation Initiation Sites


- Zien et al., *Bioinformatics* 16:799-807, 2000
- Use SVM to recognize protein translation initiation sites from genomic sequences
- Raw data set is same as Liu & Wong, *JBCB* 1:139-168, 2003

Copyright 2011 © Limsoon Wong

Bayesian Approach



Bayes Theorem




$$P(h|d) = \frac{P(d|h) * P(h)}{P(d)}$$

- P(h) = prior prob that hypothesis h holds
- P(d|h) = prob of observing data d given h holds
- P(h|d) = posterior prob that h holds given observed data d

Copyright 2011 © Limsoon Wong

Bayesian Approach



- Let H be all possible classes. Given a test instance w/ feature vector $\{f_1 = v_1, \dots, f_n = v_n\}$, the most probable classification is given by


$$\underset{h \in H}{\operatorname{argmax}} P(h | f_1 = v_1, \dots, f_n = v_n)$$
- Using Bayes Theorem, rewrites to

$$\underset{h \in H}{\operatorname{argmax}} \frac{P(f_1 = v_1, \dots, f_n = v_n | h) * P(h)}{P(f_1 = v_1, \dots, f_n = v_n)}$$
- Since denominator is independent of h , this simplifies to

$$\underset{h \in H}{\operatorname{argmax}} P(f_1 = v_1, \dots, f_n = v_n | h) * P(h)$$

Copyright 2011 © Limsoon Wong

Naïve Bayes



- But estimating $P(f_1=v_1, \dots, f_n=v_n|h)$ accurately may not be feasible unless training data set is large
- “Solved” by assuming f_1, \dots, f_n are conditionally independent of each other
- Then

$$\underset{h \in H}{\operatorname{argmax}} P(f_1 = v_1, \dots, f_n = v_n | h) * P(h) = \underset{h \in H}{\operatorname{argmax}} \prod_i P(f_i = v_i | h) * P(h)$$
- where $P(h)$ and $P(f_i=v_i|h)$ can often be estimated reliably from typical training data set

Exercise: How do you estimate $P(h)$ and $P(f_i=v_i|h)$?

Copyright 2011 © Limsoon Wong

Abstractly, the probability model for a classifier is a conditional model

$p(C|F_1, \dots, F_n)$

over a dependent class variable C with a small number of outcomes or classes, conditional on several feature variables F_1 through F_n . The problem is that if the number of features n is large or when a feature can take on a large number of values, then basing such a model on probability tables is infeasible. We therefore reformulate the model to make it more tractable.

Using Bayes' theorem, we write

$$p(C|F_1, \dots, F_n) = \frac{p(C) p(F_1, \dots, F_n|C)}{p(F_1, \dots, F_n)}$$

In practice we are only interested in the numerator of that fraction, since the denominator does not depend on C and the values of the features F_i are given, so that the denominator is effectively constant. The numerator is equivalent to the joint probability model

$$p(C, F_1, \dots, F_n)$$

which can be rewritten as follows, using repeated applications of the definition of conditional probability:

$$\begin{aligned} p(C, F_1, \dots, F_n) &= p(C) p(F_1, \dots, F_n|C) \\ &= p(C) p(F_1|C) p(F_2, \dots, F_n|C, F_1) \\ &= p(C) p(F_1|C) p(F_2|C, F_1) p(F_3, \dots, F_n|C, F_1, F_2) \\ &= p(C) p(F_1|C) p(F_2|C, F_1) p(F_3|C, F_1, F_2) p(F_4, \dots, F_n|C, F_1, F_2, F_3) \end{aligned}$$

and so forth. Now the “naïve” conditional independence assumptions come into play: assume that each feature F_j is conditionally independent of every other feature F_k for $j \neq k$. This means that

$$p(F_i|C, F_j) = p(F_i|C)$$

and so the joint model can be expressed as


$$p(C, F_1, \dots, F_n) = p(C) p(F_1|C) p(F_2|C) p(F_3|C) \dots$$

$$= p(C) \prod_{i=1}^n p(F_i|C).$$

Source: Wikipedia

Copyright 2011 © Limsoon Wong

Independence vs Conditional Independence




- Independence: $P(A,B) = P(A) * P(B)$
- Conditional Independence: $P(A,B|C) = P(A|C) * P(B|C)$
- Indep does not imply conditional indep
 - Consider tossing a fair coin twice
 - A is event of getting head in 1st toss
 - B is event of getting head in 2nd toss
 - C is event of getting exactly one head
 - Then $A=\{HT, HH\}$, $B=\{HH, TH\}$ and $C=\{HT, TH\}$
 - $P(A,B|C) = P(\{HH\}|C) = 0$
 - $P(A|C) = P(A,C)/P(C) = P(\{HT\})/P(C) = (1/4)/(1/2) = 1/2$
 - Similarly, $P(B|C) = 1/2$

Copyright 2011 © Limsoon Wong

61

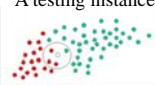
An Example

Training samples



Prior probability for OREEN = $\frac{\text{Number of OREEN objects}}{\text{Total number of objects}} = 40/60$
 Prior probability for RED = $\frac{\text{Number of RED objects}}{\text{Total number of objects}} = 20/60$

A testing instance X



Likelihood of X given OREEN = $\frac{\text{Number of OREEN in the vicinity of X}}{\text{Total number of OREEN cases}} = 1/40$
 Likelihood of X given RED = $\frac{\text{Number of RED in the vicinity of X}}{\text{Total number of RED cases}} = 3/20$

Posterior probability of X being OREEN = $\frac{4}{6} \times \frac{1}{40} = \frac{1}{60}$
 Prior probability of OREEN x Likelihood of X given OREEN
 Posterior probability of X being RED = $\frac{2}{6} \times \frac{3}{20} = \frac{1}{20}$
 Prior probability of RED x Likelihood of X given RED

we classify X as RED since its class membership achieves the largest posterior probability

Source: <http://www.stanford.edu/testbed/sumavob.html>
 Copyright 2011 © Limsoon Wong

62

Example Use of Bayesian: Design of Screens Macromolecular Crystallization

- Hennessy et al., *Acta Cryst* D56:817-827, 2000
- Xtallization of proteins requires search of expt settings to find right conditions for diffraction-quality xtals
- BMCD is a db of known xtallization conditions
- Use Bayes to determine prob of success of a set of expt conditions based on BMCD

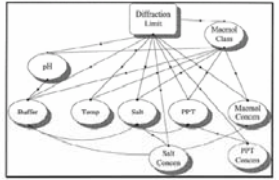



Figure 1
 Crystallization parameter dependency graph. The graph represents the parameters included in the calculation of the estimated probability of success and their dependencies. A connecting arc from pH to buffer indicates that the probability distribution for the buffer may depend on the value of the pH. The lack of a connecting arc between two parameters reflects conditional independence (the probability distribution for a parameter is independent of the value of the other parameter).

Copyright 2011 © Limsoon Wong

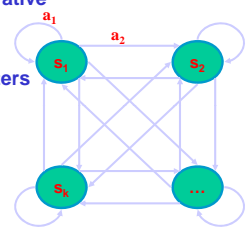
Hidden Markov Models



64

What is a HMM

- HMM is a stochastic generative model for seqs
- Defined by model parameters
 - finite set of states S
 - finite alphabet A
 - transition prob matrix T
 - emission prob matrix E
- Move from state to state as per T while emitting symbols as per E



Copyright 2011 © Limsoon Wong

65

Order of a HMM

- In *n*th order HMM, T & E depend on all *n* previous states
- E.g., for 1st order HMM, given emissions $X = x_1, x_2, \dots$, & states $S = s_1, s_2, \dots$, the prob of this seq is

$$Prob(X, S) = \prod_i Prob(x_i | s_i) = \prod_i E(x_i | s_i) * T(s_{i-1}, s_i)$$

Copyright 2011 © Limsoon Wong

66

Using HMM

- Given the model parameters, compute the probability of a particular output sequence. Solved by the **forward algorithm**
- Given the model parameters, find the most likely sequence of (hidden) states which could have generated a given output sequence. Solved by the **Viterbi algorithm**
- Given an output sequence, find the most likely set of state transition and output probabilities. Solved by the **Baum-Welch algorithm**

Exercise: Describe these algorithms
 Copyright 2011 © Limsoon Wong

67

Example: Dishonest Casino

- Casino has two dices:
 - Fair dice
 - $P(i) = 1/6, i = 1..6$
 - Loaded dice
 - $P(i) = 1/10, i = 1..5$
 - $P(i) = 1/2, i = 6$
- Game:
 - You bet \$1
 - You roll
 - Casino rolls
 - Highest number wins \$2
- Question: Suppose we played 2 games, and the sequence of rolls was 1, 6, 2, 6. Were we likely to have been cheated?

Copyright 2011 © Limsoon Wong

68

“Visualization” of Dishonest Casino

$E_1(\text{Fair}) = 1/6$	$E_1(\text{loaded}) = 1/10$
$E_2(\text{Fair}) = 1/6$	$E_2(\text{loaded}) = 1/10$
$E_3(\text{Fair}) = 1/6$	$E_3(\text{loaded}) = 1/10$
$E_4(\text{Fair}) = 1/6$	$E_4(\text{loaded}) = 1/10$
$E_5(\text{Fair}) = 1/6$	$E_5(\text{loaded}) = 1/10$
$E_6(\text{Fair}) = 1/6$	$E_6(\text{loaded}) = 1/2$

$T(\text{loaded} \text{loaded}) = 1/2$
$T(\text{loaded} \text{Fair}) = 1/2$
$T(\text{Fair} \text{loaded}) = 1/2$
$T(\text{Fair} \text{Fair}) = 1/2$
$T(\text{?} \text{Fair}) = 1.0$
$T(\text{?} \text{loaded}) = 0.0$

Copyright 2011 © Limsoon Wong

69

1, 6, 2, 6? We were probably cheated...

$$\begin{aligned}
 \text{Prob}(X, S = \text{Fair}, \text{Fair}, \text{Fair}, \text{Fair}) &= E(1|\text{Fair}) * T(?, \text{Fair}) * \\
 &E(6|\text{Fair}) * T(\text{Fair}, \text{Fair}) * \\
 &E(2|\text{Fair}) * T(\text{Fair}, \text{Fair}) * \\
 &E(6|\text{Fair}) * T(\text{Fair}, \text{Fair}) \\
 &= \frac{1}{6} * 1 * \frac{1}{6} * \frac{1}{6} * \frac{1}{6} * \frac{1}{6} * \frac{1}{6} \\
 &= 9.651 * 10^{-25}
 \end{aligned}$$

$$\begin{aligned}
 \text{Prob}(X, S = \text{Fair}, \text{Loaded}, \text{Fair}, \text{Loaded}) &= E(1|\text{Fair}) * T(?, \text{Fair}) * \\
 &E(6|\text{Loaded}) * T(\text{Fair}, \text{Loaded}) * \\
 &E(2|\text{Fair}) * T(\text{Loaded}, \text{Fair}) * \\
 &E(6|\text{Loaded}) * T(\text{Fair}, \text{Loaded}) \\
 &= \frac{1}{6} * 1 * \frac{1}{2} * \frac{1}{6} * \frac{1}{6} * \frac{1}{2} * \frac{1}{6} \\
 &= 8.636 * 10^{-3}
 \end{aligned}$$

Copyright 2011 © Limsoon Wong

70

Example Use of HMM: Protein Families Modeling

- Baldi et al., *PNAS* 91:1059-1063, 1994
- HMM is used to model families of biological sequences, such as kinases, globins, & immunoglobulins
- Bateman et al., *NAR* 32:D138-D141, 2004
- HMM is used to model 6190 families of protein domains in Pfam

Copyright 2011 © Limsoon Wong

Concluding Remarks...

72

What have we learned?

- Decision Trees
- Decision Trees Ensembles
 - Bagging
- Other Methods
 - K-Nearest Neighbour
 - Support Vector Machines
 - Bayesian Approach
 - Hidden Markov Models

Copyright 2011 © Limsoon Wong

Any Question?



- <http://www.cs.waikato.ac.nz/ml/weka>
- Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization.

Exercise: Download a copy of WEKA. What are the names of classifiers in WEKA that correspond to C4.5 and SVM?

Copyright 2011 © Limsoon Wong

Acknowledgements



- Most of the slides used in this ppt came from a tutorial that I gave with Jinyan Li at the *8th European Conference on Principles and Practice of Knowledge Discovery in Databases*, Pisa, Italy, 20-24 September 2004
- The dishonest casino example came from slides I inherited from Ken Sung
- The “indep vs conditional indep” example came from Kwok Pui Choi

Copyright 2011 © Limsoon Wong

References



- L. Breiman, et al. Classification and Regression Trees. Wadsworth and Brooks, 1984
- L. Breiman, Bagging predictors, *Machine Learning*, 24:123-140, 1996
- L. Breiman, Random forests, *Machine Learning*, 45:5-32, 2001
- J. R. Quinlan, Induction of decision trees, *Machine Learning*, 1:81-106, 1986
- J. R. Quinlan, C4.5: Program for Machine Learning. Morgan Kaufmann, 1993
- C. Gini, Measurement of inequality of incomes, *The Economic Journal*, 31:124-126, 1921
- Jinyan Li et al., Data Mining Techniques for the Practical Bioinformatician, *The Practical Bioinformatician*, Chapter 3, pages 35-70, WSPC, 2004

Copyright 2011 © Limsoon Wong

References



- Y. Freund, et al. Experiments with a new boosting algorithm, *ICML* 1996, pages 148-156
- T. G. Dietterich, An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization, *Machine Learning*, 40:139-157, 2000
- Naive Bayesian Classification, *Wikipedia*, http://en.wikipedia.org/wiki/Naive_Bayesian_classification
- Hidden Markov Model, *Wikipedia*, http://en.wikipedia.org/wiki/Hidden_Markov_model

Copyright 2011 © Limsoon Wong