

COMP37332

Introduction
to classification techniques

Goran Nenadic
School of Computer Science

1

Aims

- Understand basic principles of classification
- Understand main classification techniques
 - decision trees
 - SVM
- Discuss possible applications

2

Plan

- Introduction to classification
- Classification techniques
 - overview
 - decision trees
 - SVM classification
- Evaluation of classification
- Applications
- Main issues with classification

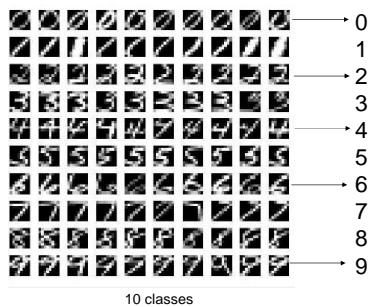
3

Classification

- Mapping data into predefined set of classes
 - classes are known and fixed
- Constructs a model based on a training set and the values (*class labels*) in a classifying attribute
 - supervised learning
- Uses the learnt model in classifying new data
- Typical applications
 - credit approval, target marketing, medical diagnosis

4

An example: digit recognition

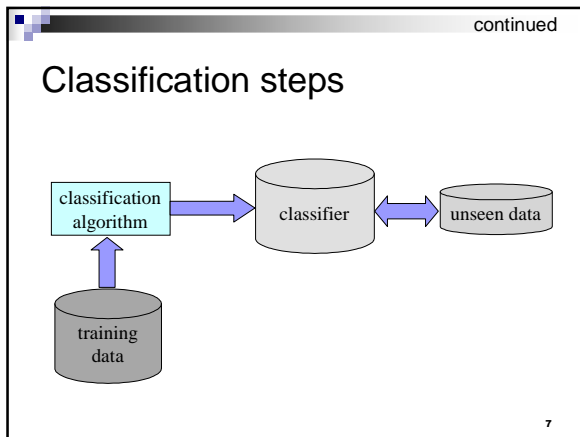


5

Classification steps

1. **Model construction from a training set**
 - training set: set of objects used for model construction
 - using features of training objects, describe a set of predetermined classes
2. **Model usage: classifying new objects**
 - predict a class for a given object

6



- continued
- ## Classification steps
1. **Model construction from a training set**
 - each object from the training set has its class label
 - model is represented as either classification rules, decision trees, or mathematical formulae
 2. **Model usage: classifying new objects**
 - for classifying future or unseen objects
 - classification outcome: class membership
- 8

example

Training examples

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31..40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31..40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31..40	medium	no	excellent	yes
31..40	high	yes	fair	yes
>40	medium	no	excellent	no

A training set for "buys computer"

9

Unseen/test data examples

age	income	student	credit rating	buys computer
<=30	low	yes	fair	?
<=30	high	no	excellent	?
31...40	medium	yes	fair	?
>40	medium	no	excellent	?
>40	low	yes	fair	?
>40	low	yes	fair	?
31...40	high	yes	excellent	?
<=30	medium	no	fair	?
<=30	low	yes	excellent	?
>40	medium	yes	fair	?
<=30	medium	no	excellent	?
31...40	high	no	excellent	?
31...40	low	yes	fair	?
>40	medium	no	excellent	?

Classification outcomes

- Outcomes are discrete values
 - not continued-valued attributes
[recall regression: numeric predictions]
- **Binary vs. multi-class** classification
 - binary: only two classes (yes/no; pos/neg; +/-)
 - multi-class: **membership** in a given class
 - digit recognition (10 classes)
 - credit scoring class (e.g. 5 classes)
 - customer profiles

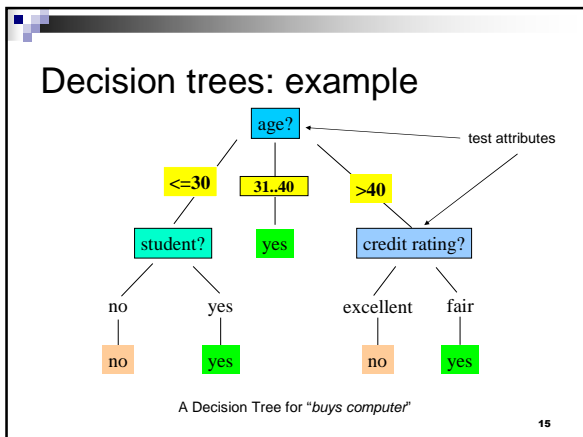
Classification techniques

- **Decision trees**
- Bayesian classification
- Neural networks
- Case-based reasoning
- Genetic algorithms
- **Support vector machines (SVM)**
- ...

Decision trees

13

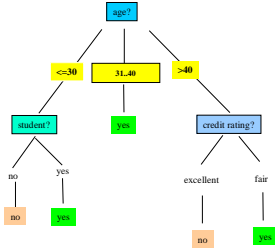
- ## Decision trees
- so, not manually created by an analyst
- A flow-chart-like tree structure, **automatically** derived from data
 - Internal node denotes a test on an attribute
 - Branch represents an outcome of the test
 - Leaf nodes represent class labels or class distribution
 - Traversing the tree from the root to a leaf gives an answer i.e. classification outcome
- 14



Decision trees: classification

■ Class prediction

- target entity
 - age: 48
 - income: low
 - student: no
 - credit rating: fair
 - buys computer ??



Notes

- Attributes are **categorical**
 - if continuous-valued (e.g. age), discretised in advance
 - e.g. map age in 3 classes (<=30, 31..40, >40)
- Not all attributes (from the DB) are relevant to a given classification problem
 - previous example does not use income
 - feature selection (see previous lectures):
 - select attributes that are important
 - noise and irrelevant attributes should be filtered

Decision tree induction

- Tree is constructed **automatically** in a top-down recursive manner (typically)
- Also, tree pruning may be performed
 - identify and remove branches that reflect noise or outliers
- Various algorithms for constructing DTs
 - ID3 algorithm
 - C4.5

ID3 algorithm – general outline

1. At the beginning, all the training examples are at the root of the decision tree being built
2. Examples are partitioned recursively based on selected attributes, which are chosen on the basis of “best splitting” of training examples
3. Conditions for stopping further partitioning
 - all samples for a given node belong to the same class
 - there are no samples left
 - there are no remaining attributes for further partitioning [majority voting is then employed for classifying the leaf]

19

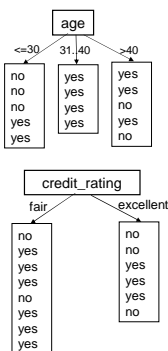
ID3 algorithm: main issues

- “Best splitting” criteria: how to measure splitting success for a chosen test attribute?
- Aims:
 - minimize the information needed to classify data
 - minimize the number of tests in the tree
- Various information measures for attribute selection
 - information gain

20

Splitting the sample

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no



21

Information gain

- Decision tree produces a **message**, which is the classification outcome
- Information gain: a measure used to select a test attribute at each node of the tree
 - estimation based on **entropy** (measures randomness)
 - estimate the gain in information from a particular partitioning of the dataset i.e. find the attribute with the highest information gain or greatest **entropy reduction**

22

Information gain and entropy

- Entropy measures randomness, structure, disorder, homogeneity in set of data points
 - **low entropy**: structured sets (not so random)
 - **high entropy**: unstructured sets, without order
 - so, we need more information to describe them
- The attribute with the greatest **entropy reduction** would do good partitioning and would minimize the information needed to classify the samples
 - this aims at providing a simpler tree for classification, i.e. reduction of the number of tests

23

Entropy

optional material

you can skip it

- Also known as "information content"
- From information theory
 - a message with probability p needs $-\log_2 p$ bits to be encoded
- In binary classification problems, expected number of bits to encode positive (+) or negative (-) outcome of a random member is:
$$p_+ (-\log_2 p_+) + p_- (-\log_2 p_-) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$
 - p_+ is probability of + examples
 - p_- is probability of - examples

24

continued

Entropy

$$p_+ (-\log_2 p_+) + p_- (-\log_2 p_-) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$

- if $p_+ = p_- = 1/2$, then entropy (information content) is
 - $1/2 * \log_2 1/2 - 1/2 * \log_2 1/2 = 1/2 + 1/2 = 1$
- 1 bit is needed to encode such information

25

continued

Entropy

- **S** - example set, S_1, \dots, S_n - classes

$$I(S) = -\sum_{i=1}^n p_i \cdot \log_2 p_i$$

- p_i is probability of class S_i (i.e. probability that a random sample belongs to class S_i)
- Probabilities are estimated by frequencies
 - $p_i = (\text{number of samples in } S_i) / (\text{total number of samples})$

26

Entropy using attribute A

- Assume that by using attribute *A*, a set *S* will be partitioned into sets $\{S_1, S_2, \dots, S_m\}$
- Each S_i may contain samples from many classes
 - s_{ki} = number of samples from class *k* in S_i (i.e. the number of elements in S_{ki})
 - $p_{ki} = s_{ki} / s_i$ (s_i = total number of samples in S_i)
- Information need for each S_i is then

$$I(S_1, \dots, S_m) = -\sum_{i=1}^m p_i \log_2 p_i$$

27

Entropy using attribute A

- probability of S_i is $p_{S_i} = (s_{i1} + \dots + s_{in})/s$
- finally, entropy associated with attribute A

$$E(A) = \sum_{i=1}^n p_{S_i} \cdot I(S_{i1}, \dots, S_{in})$$

Information gain

- If we use attribute A to partition the training set, the encoding information that would be gained by branching on A is

$$Gain(A) = I(S_1, \dots, S_n) - E(A)$$

$I(S_1, \dots, S_n)$ = information need of S_1, \dots, S_n

$E(A)$ = entropy using attribute A

- Choose attribute that gives the greatest information gain

Example

- binary problem: $S_{YES}; S_{NO}$

$$I(S_{YES}, S_{NO}) = -\frac{9}{14} \log \frac{9}{14} - \frac{5}{14} \log \frac{5}{14} = 0.94$$

income: $S_{high}, S_{medium}, S_{low}$

$$p_{high} = p_{low} = 4/14; p_{medium} = 6/14$$

age: $S_{<=30}, S_{31..40}, S_{>40}$

$$p_{<=30} = p_{>40} = 5/14; p_{31..40} = 4/14$$

student: S_{yes}, S_{no}

$$p_{yes} = p_{no} = 7/14$$

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	<no
<=30	high	no	excellent	<no
31..40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	<no
31..40	low	yes	excellent	yes
<=30	medium	no	fair	<no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31..40	medium	no	excellent	yes
31..40	high	yes	fair	yes
>40	medium	no	excellent	<no

continued

Example

- information need for *income* attribute:

$$I(S_{highYES}, S_{highNO}) = -\frac{2}{4} \log \frac{2}{4} - \frac{2}{4} \log \frac{2}{4} = 1 = I(2,2)$$

$$I(S_{medYES}, S_{medNO}) = -\frac{4}{6} \log \frac{4}{6} - \frac{2}{6} \log \frac{2}{6} = 0.918 = I(4,2)$$

$$I(S_{lowYES}, S_{lowNO}) = -\frac{3}{4} \log \frac{3}{4} - \frac{1}{4} \log \frac{1}{4} = 0.811 = I(3,1)$$

31

continued

Example

- entropy for *income* attribute:

$$E(\text{income}) = p_{high} \cdot I(S_{highYES}, S_{highNO}) +$$

$$p_{med} \cdot I(S_{medYES}, S_{medNO}) +$$

$$p_{low} \cdot I(S_{lowYES}, S_{lowNO})$$

$$= \frac{4}{14} \cdot 1 + \frac{6}{14} \cdot 0.918 + \frac{4}{14} \cdot 0.811 = 0.911$$

$$\text{Gain}(\text{income}) = I(S_{YES}, S_{NO}) - E(\text{income}) = 0.94 - 0.911 = 0.029$$

32

continued

Example

- entropy for *age* attribute:

$$E(\text{age}) = p_{<=30} \cdot I(S_{<=30;YES}, S_{<=30;NO}) +$$

$$p_{31..40} \cdot I(S_{31..40;YES}, S_{31..40;NO}) +$$

$$p_{>40} \cdot I(S_{>40;YES}, S_{>40;NO})$$

$$= \frac{5}{14} \cdot I(2,3) + \frac{4}{14} \cdot I(4,0) + \frac{5}{14} \cdot I(3,2) = 0.69$$

$$\text{Gain}(\text{age}) = I(S_{YES}, S_{NO}) - E(\text{age}) = 0.94 - 0.69 = 0.25$$

33

continued

Example

- $Gain(age) = 0.25$
- $Gain(income) = 0.029$
- $Gain(student) = 0.151$
- $Gain(credit_rating) = 0.048$

choose *age* as a testing/partitioning attribute, since it has the greatest information gain

34

continued

ID3 algorithm – general outline

1. At the beginning, all the training examples are at the root
2. Examples are partitioned recursively based on selected "best splitting" attributes
3. Conditions for stopping further partitioning
 - all samples for a given node belong to the same class
 - there are no samples left
 - there are no remaining attributes for further partitioning [majority voting is then employed for classifying the leaf]

35

continued

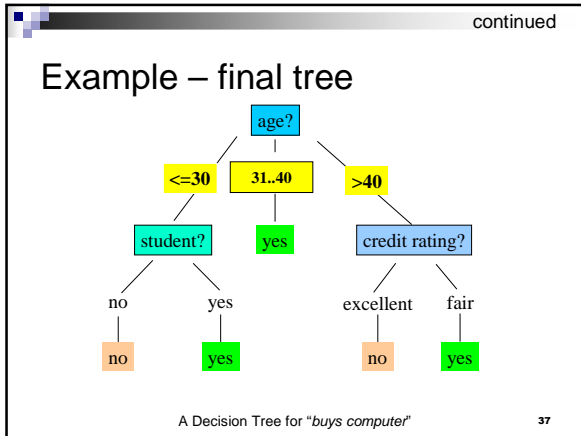
Example

```

graph TD
    A[age?] -- "<=30" --> B[ ]
    A -- "31..40" --> C[yes]
    A -- ">40" --> D[ ]
  
```

- repeat the process for **each** of the three branches
- discard the used attribute (*age*), use the other 3 attributes
- stopping conditions:
 - all samples for a given node belong to the same class
 - no remaining attributes for further partitioning: majority voting is employed for classifying the leaf
 - no samples left

36



- ### Extracting classification rules from decision trees
- Represent the classification model in the form of IF-THEN rules
 - IF *age* = "31..40" THEN *buys_computer* = "yes"
 - Methodology
 - a rule is created for each path from the root to a leaf
 - each attribute-value pair along a path forms a conjunction
 - the leaf node holds the class prediction
- 38

- example
- ### Extracting classification rules from decision trees
- Example
 - IF *age* = "<=30" AND *student* = "no" THEN *buys_computer* = "no"
 - IF *age* = "<=30" AND *student* = "yes" THEN *buys_computer* = "yes"
 - IF *age* = "31...40" THEN *buys_computer* = "yes"
 - IF *age* = ">40" AND *credit_rating* = "excellent" THEN *buys_computer* = "yes"
 - IF *age* = ">40" AND *credit_rating* = "fair" THEN *buys_computer* = "no"
- 39

Decision trees: application examples

40

Weka - example

age	spectacle-prescrip	astigmatism	tear-prod-rate	contact-lenses
young	myope	no	reduced	none
young	myope	no	normal	soft
young	myope	yes	reduced	none
young	myope	yes	normal	hard
young	hypermetropic	no	reduced	none
young	hypermetropic	no	normal	soft
young	hypermetropic	yes	reduced	none
young	hypermetropic	yes	normal	hard
pre-presbyopic	myope	no	reduced	none
pre-presbyopic	myope	no	normal	soft
pre-presbyopic	myope	yes	reduced	none
pre-presbyopic	myope	yes	normal	hard
pre-presbyopic	hypermetropic	no	reduced	none
pre-presbyopic	hypermetropic	no	normal	soft
pre-presbyopic	hypermetropic	yes	reduced	none
pre-presbyopic	hypermetropic	yes	normal	none
presbyopic	myope	no	reduced	none
presbyopic	myope	no	normal	none
presbyopic	myope	yes	reduced	none
presbyopic	myope	yes	normal	hard
presbyopic	hypermetropic	no	reduced	none
presbyopic	hypermetropic	no	normal	soft
presbyopic	hypermetropic	yes	reduced	none
presbyopic	hypermetropic	yes	normal	none
presbyopic	hypermetropic	yes	reduced	none
presbyopic	hypermetropic	yes	normal	none

41

Weka - example

```

graph TD
    A[tear_prod_rate] -- normal --> B[NONE]
    A -- reduced --> C[astigmatism]
    C -- no --> D[age]
    C -- yes --> E[...]
    D -- presbyopic --> F[spec_prescript]
    D -- pre-presbyopic --> G[SOFT]
    D -- young --> H[SOFT]
    F -- myope --> I[NONE]
    F -- hypermetropic --> J[SOFT]
  
```

Weka - example

Selecting classification algorithm

- several algorithms developed
- C4.5: improved Id3 algorithm, widely used
 - dealing with numeric attributes
 - missing values
 - noisy data
- implemented in Weka as J48

43

Decision trees in Oracle

DMRFCASE_ID	OCCUPATION1	CUST_GENDE	CUST_MARITA	PREDICTION	PROBABILITY	NODE
100,001	Exec	F	NeverM	0	0.9195	9
100,002	Prof	F	NeverM	0	0.9095	8
100,003	Sales	M	NeverM	0	0.9195	9
100,004	Sales	F	Divorc.	0	0.9195	9
100,005	Crafts	M	Married	0	0.5943	6
100,006	Prof	F	NeverM	0	0.9954	10
100,007	Other	F	Divorc.	0	0.9195	9
100,008	Crafts	M	NeverM	0	0.9195	9
100,009	Prof	M	Married	1	0.7557	5
100,010	Crafts	M	Married	0	0.8537	7
100,011	Farming	M	NeverM	0	0.9195	9

44

Decision trees – summary

- A method for **automatic** mining of decision trees (i.e. classification rules) from a dataset
 - traversing the tree from the root to a leaf gives a classification outcome for a given data point
- ID3 algorithm
 - **automatic** construction of a DT, in a top-down recursive manner
 - test attributes are selected on the basis of "best splitting" of the sample set
 - information gain: a measure used to select a test attribute at each node of the tree

45

SVM classification

SVM = support vector machine

46

SVM classification

- Data is represented as vectors
 - (young, 3, 145.45, 321, employed)
- SVM = an **automatic** method that can be used to learn a **separation boundary** between two classes
 - in the testing phase, find the best 'line' that separates the data into two classes
- Typically used
 - for binary classification
 - when attributes are (mostly) numeric

47

example

SVM classification

48

SVM classification

- Aim: find an optimal “separation line”, i.e. a decision boundary that is as far away from the data of both classes as possible
 - maximize the margin between classes
 - find a (linear) function that describes the boundary
- Once the **decision function** is determined, new objects are classified by calculating their distance to the decision boundary

Recall: classification steps

1. **Model construction from a training set**
 - training set: set of objects used for model construction
 - model is represented as classification rules, decision trees, or mathematical formulae
2. **Model usage: classifying new or unknown objects**
 - predict a class for a given object

Main idea behind SVMs

- Blend of linear modelling and instance-based learning
- Idea: find a set of *critical boundary points* (called support vectors) that are used for (linear) separation
- These points should provide “maximum margin hyperplane”

continued

Main idea behind SVMs

52

Constructing SVMs

- A set of training “vectors” (i.e. data points)

$$x_i = (x_{i1}, \dots, x_{im}) \quad x_i, \quad i = 1, \dots, l$$
- Define vectors y_i that define class membership

$$y_i = 1 \text{ if } x_i \text{ in class A}$$

$$y_i = -1 \text{ if } x_i \text{ not in class A}$$

53

continued

you can skip this

Constructing SVM

A separating hyperplane: $w^T x + b = 0$

$$w^T x + b = \begin{bmatrix} +1 \\ 0 \\ -1 \end{bmatrix}$$

$$(w^T x_i) + b > 0 \quad \text{if } y_i = 1$$

$$(w^T x_i) + b < 0 \quad \text{if } y_i = -1$$

54

continued

Constructing SVM

- Many possible choices for w and b
- Find ones that maximise the distance to training examples

$$\min_{w,b} \frac{1}{2} w^T w$$

subject to $y_i((w^T x_i) + b) \geq 1,$
 $i = 1, \dots, l.$

55

Classification using SVM

- Once we find w and b , **decision function** is

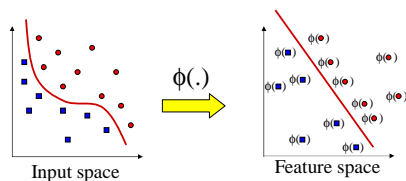
$$f(x) = \text{sign}(w^T x + b)$$

- x is an unseen data point
- $f(x) = 1$, if x in class A
- $f(x) = -1$, if x not in class A

56

SVM classification: kernels

- How about not linearly separable data?



kernel: transform data to a higher dimensional space to "make life easier"
Use linear model to implement non-linear class boundaries

57

SVM classification

■ Strengths

- training is relatively easy
- also, non-numeric data like strings and trees can be used as input to SVM
- scales relatively well to high dimensional data

■ "Weaknesses"

- needs a "good" kernel function if data is not separable (which is typically the case)

SVM classifiers

■ svm^{light}

- http://www.cs.cornell.edu/People/tj/svm_light/

■ SVM LIB

- <http://www.csie.ntu.edu.tw/~cjlin/libsvm>

Example

• Training

```
./svm-train train.1
.....*
optimization finished, #iter = 6131
nu = 0.606144
obj = -1061.528899, rho = -0.495258
nSV = 3053, nBSV = 724
Total nSV = 3053
```

• Testing

```
./svm-predict test.1 train.1.model
test.1.predict
Accuracy = 66.925% (2677/4000)
```

SVM vs. decision trees

- Decision trees are based on a series of discrete “questions” to classify an object
 - test features values
- SVMs use a mathematical function
 - calculate a **function** using feature values
- Rules in decision tree are sometimes easier to “understand”
- SVMs are difficult to “understand”
- SVMs scale better: can handle thousands features

61

Oracle data mining

- Provides the following classification algorithms
 - decision trees
 - SVM
 - Naïve Bayes
 - Generalised Linear Models (GLM)

62

Evaluation of classification

- Estimate accuracy of the model
 - test set must be independent of training set
 - the known label of a test example is compared with the classified result from the model
 - correctly and incorrectly classified instances
- **Measures**
 - precision
 - recall

63

continued

Evaluation of classification

- In case of binary classification, possible outcomes are
 - *true positives* (TP) = examples that belong to class A, and are classified as belonging to A
 - *true negatives* (TN) = examples that do not belong to class A, and are classified as not belonging to A
 - *false positives* (FP) = examples that do not belong to class A, but are classified as belonging to A
 - *false negatives* (FN) = examples that belong to class A, but are classified as not belonging to A

64

continued

Evaluation of classification

65

continued

Evaluation of classification

		actual class	
		yes	no
predicted class	yes	true positives	false positives
	no	false negatives	true negatives

66

continued

Evaluation of classification

- **Precision** estimates the accuracy of classification:
 - how many elements have been classified correctly

Precision = $\frac{\# \text{ True positives}}{\# \text{ True positives} + \# \text{ False positives}}$

= number of testing examples

67

continued

Evaluation of classification

- **Recall** estimates the coverage of classification:
 - how many elements from a given class the model has been able to recognise/predict

Recall = $\frac{\# \text{ True positives}}{\# \text{ True positives} + \# \text{ False negatives}}$

= number of examples that belong to the given class

68

continued

Evaluation of classification

- Recall and precision are inversely related:
 - improving precision, typically reduces recall and vice versa
- **F-measure**: a combination of precision and recall

$$F_1(r, p) = \frac{2rp}{r + p}$$

false positives and false negatives
- Other measures
 - **error rate**: the ratio of false predictions to the total number of predictions
 - **overall accuracy rate**: the ratio of true predictions to the total number of predictions

true positives and true negatives

69

Classification outcomes in Weka

Correctly Classified Instances 51 89.4737 %
Incorrectly Classified Instances 6 10.5263 %
Kappa statistic 0.7635
Mean absolute error 0.1053
Root mean squared error 0.3244
Relative absolute error 23.0111 %
Root relative squared error 67.9505 %
Total Number of Instances 57

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.8	0.054	0.889	0.8	0.842	bad
0.946	0.2	0.897	0.946	0.921	good

=== Confusion Matrix ===

a	b	<- classified as
16	4	a = bad
2	35	b = good

70

Application areas

■ Typical applications

- credit/loan approval
- target marketing
- medical diagnosis
- bioinformatics
- document classification
- image classification

71

example

Classification in bioinformatics

- Given gene expression data (i.e. some experimental data) predict function of the gene, its location and processes it is involved in
 - possible functions, locations, processes are known
- Training set is selected from genes with known functions, locations and processes
 - use them to train a classifier
 - "dry experiments"
- Apply on new genes with unknown functions
 - verify results by further "wet" experiments

72

Main issues in classification

- Selection of classification features
 - find representative attributes (when numerous features are present)
- Training data may be noisy
- “Contradicting” data
 - e.g. customers with the same values for attributes, but different outcomes
- Over-fitting
- Time to construct and use the model
 - speed and scalability issues
- Interpretability
 - understanding and insight provided by the model

73

Summary

- Classification: arrangement of a set of objects into a predefined set of classes
- Supervised learning
 - learn from a training set
- Example techniques
 - decision trees
 - SVMs
- Mainly used to predict characteristics of a new object

74

Reading for this lecture

- Chapter 27 (27.3) in [Elmasri and Navathe]

Also:

- On-line materials, tutorials and software

<http://personalpages.manchester.ac.uk/staff/G.Nenadic/COMP37332/>

- Go through the Classification tutorial (Weka)

75
