Week 3: Experiment Design

Data Science Certificate Program

Ryerson University

Bora Caglayan

29 Jan, 2015

Outline

- Overview of data mining
- Experiment design
    - Problem definition
    - Preparation of Data
    - Identify performance measures
    - Training and testing
    - Interpretation of results
- Dealing with missing data
- Applications with R
    - Cleaning and preparing data for experiments
    - Handling missing data
- Lab

Overview of Data Mining

Data Mining

Retail: Market basket analysis, Customer relationship management (Cl

Finance: Credit scoring, fraud detection

Manufacturing: Control, robotics, troubleshooting

Medicine: Medical diagnosis

Telecommunications: Spam filters, intrusion detection

Bioinformatics: Motifs, alignment

Web mining: Search engines

Big Data

Widespread use of personal computers and wireless communication leads to “big data”.

We are both producers and consumers of data.

Data is not random, it has structure, e.g., customer behaviour.

We need “big theory” to extract that structure from data for:
(a) Understanding the process.
(b) Making predictions for the future.

Applications

- Association
- Supervised Learning
  - Classification
  - Regression
- Unsupervised Learning
- Reinforcement Learning

Learning Associations

Goal: Identify strong rules discovered in databases using different measures of interestingness.

Basket analysis:
P (Y | X) probability that somebody who buys X also buys Y where X and Y are products/services.

Example: P (chips | beer) = 0.7

Learning Associations

$p(x_1, x_2)$	$x_2 = 1$	$x_2 = 2$
$x_1 = 1$	0.3	0.3
$x_1 = 2$	0.1	0.3

1. Find the following quantities
   - Marginals: $p(x_1)$, $p(x_2)$
   - Conditionals: $p(x_1|x_2)$, $p(x_2|x_1)$
   - Posterior: $p(x_1, x_2 = 2)$, $p(x_1|x_2 = 2)$
   - Evidence: $p(x_2 = 2)$
   - $p(\{\})$
   - Max: $p(x_1^*) = \max_{x_1} p(x_1|x_2 = 1)$
   - Mode: $x_1^* = \arg\max_{x_1} p(x_1|x_2 = 1)$
   - Max-marginal: $\max_{x_1} p(x_1, x_2)$
2. Are x_1 and x_2 independent ? (i.e., Is $p(x_1, x_2) = p(x_1)p(x_2)$?)

Learning Associations

$p(x_1, x_2)$	$x_2 = 1$	$x_2 = 2$
$x_1 = 1$	0.3	0.3
$x_1 = 2$	0.1	0.3

- Marginals:

$p(x_1)$	
$x_1 = 1$	0.6
$x_1 = 2$	0.4

$p(x_2)$	$x_2 = 1$	$x_2 = 2$
	0.4	0.6

- Conditionals:

$p(x_1 \| x_2)$	$x_2 = 1$	$x_2 = 2$
$x_1 = 1$	0.75	0.5
$x_1 = 2$	0.25	0.5

$p(x_2 \| x_1)$	$x_2 = 1$	$x_2 = 2$
$x_1 = 1$	0.5	0.5
$x_1 = 2$	0.25	0.75

Learning Associations

$p(x_1, x_2)$	$x_2 = 1$	$x_2 = 2$
$x_1 = 1$	0.3	0.3
$x_1 = 2$	0.1	0.3

- Posterior:

$p(x_1, x_2 = 2)$	$x_2 = 2$
$x_1 = 1$	0.3
$x_1 = 2$	0.3

$p(x_1 \| x_2 = 2)$	$x_2 = 2$
$x_1 = 1$	0.5
$x_1 = 2$	0.5

- Evidence:

$$p(x_2 = 2) = \sum_{x_1} p(x_1, x_2 = 2) = 0.6$$

- Normalisation constant:

$$p(\{\}) = \sum_{x_1} \sum_{x_2} p(x_1, x_2) = 1$$

Learning Associations

$p(x_1, x_2)$	$x_2 = 1$	$x_2 = 2$
$x_1 = 1$	0.3	0.3
$x_1 = 2$	0.1	0.3

- Max: (get the value)

$$\max_{x_1} p(x_1 | x_2 = 1) = 0.75$$

- Mode: (get the index)

$$\operatorname*{argmax}_{x_1} p(x_1 | x_2 = 1) = 1$$

- Max-marginal: (get the "skyline") $\max_{x_1} p(x_1, x_2)$

$\max_{x_1} p(x_1, x_2)$	$x_2 = 1$	$x_2 = 2$
	0.3	0.3

Learning Associations

- Possible Problems
  - We may not have a perfect probability distribution.
  - Probabilities might change
    - We may need to update our probability distributions
  - Very ***large*** and ***sparse*** probability distributions
  - We may need to control a lot of variables:
    - Example: (Sleep vs health)

Supervised Learning

Prediction of future cases: Use the rule to predict the output for future inputs

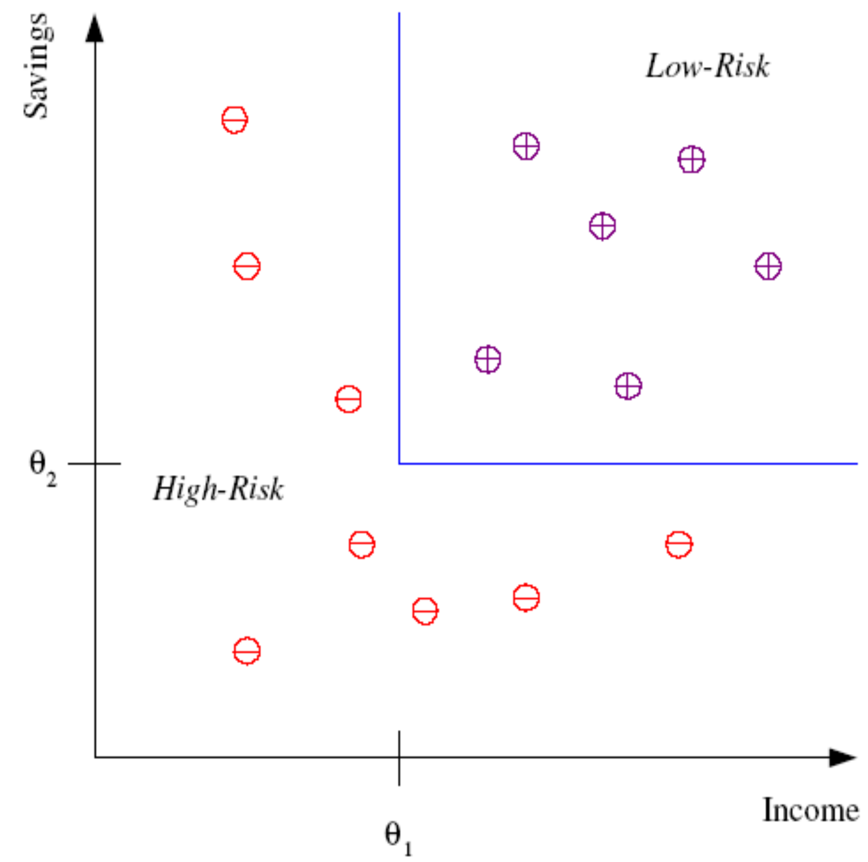
Knowledge extraction: The rule is easy to understand

Compression: The rule is simpler than the data it explains

Outlier detection: Exceptions that are not covered by the rule, e.g., fraud

Classification

Example: Credit scoring
Differentiating between low-risk and high-risk customers from their *income* and *savings*



Discriminant: IF *income* > θ_1 AND *savings* > θ_2
THEN low-risk ELSE high-risk

Regression Example

Example: Price of a used car

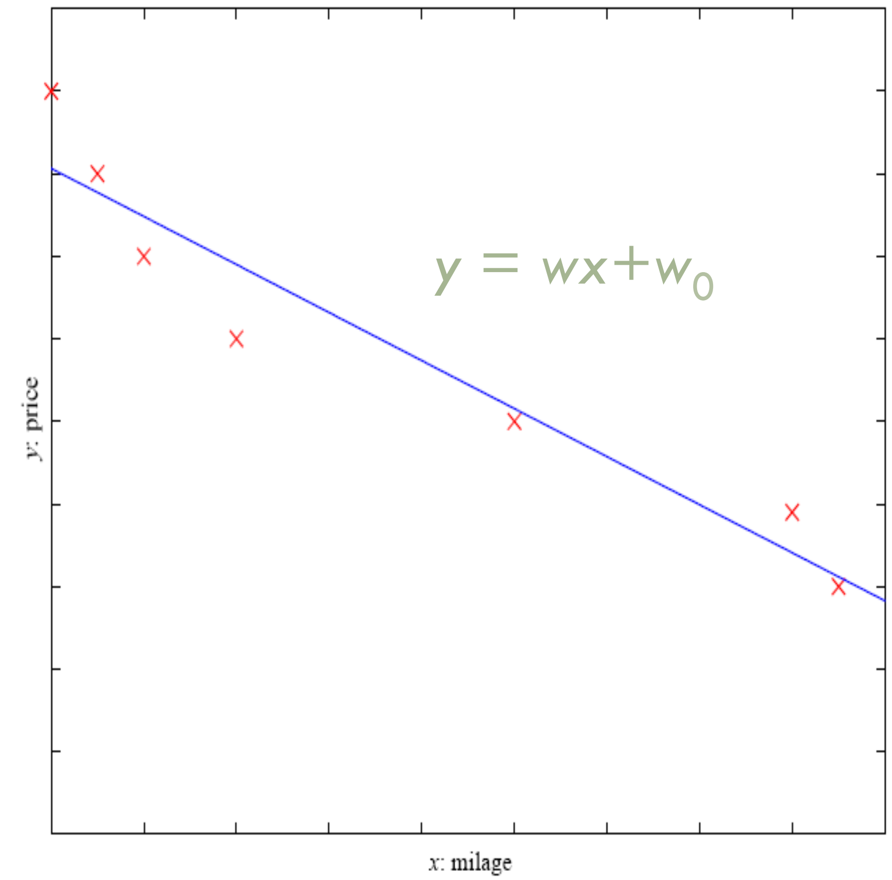
x : car attributes

y : price

$y = g(x \mid \theta)$

g ( ) model,

θ parameters



Unsupervised Learning

Learning “what normally happens”

No output

Clustering: Grouping similar instances

Example applications:

- Customer segmentation in CRM
- Image compression: Color quantization
- Bioinformatics: Learning motifs

Reinforcement Learning

Learning a policy: A sequence of outputs

No supervised output but delayed reward

- Credit assignment problem
- Game playing
- Robot in a maze
- Multiple agents, partial observability

Experiment Design

Definition of the Problem

Informal Definition:

*A computer program is said to learn from experience **E** with respect to some class of tasks **T** and performance measure **P**, if its performance at tasks in **T**, as measured by **P**, improves with experience **E**.*

*Initially we need to define **T, E** and **P.***

Definition of the Problem

Example

- Task (T): Classify a tweet that has not been published as going to get retweets or not.
- Experience (E): A corpus of tweets for an account where some have retweets and some do not.
- Performance (P): Classification accuracy, the number of tweets predicted correctly out of all tweets considered as a percentage.

Preparation of Data

Each instance is described by a fixed predefined set of features, its “attributes”

- Number of attributes may vary in practice.
- Possible solution: “irrelevant value” flag

Possible attribute types (“levels of measurement”):

- Nominal, ordinal, interval and ratio

Preparation of Data

Nominal Values

Values are distinct symbols

- Values themselves serve only as labels or names Nominal comes from the Latin word for name
- No relation is implied among nominal values (no ordering or distance measure)
- Only equality tests can be performed

Example: attribute “outlook” from weather data

Values: “sunny”,”overcast”, and “rainy”

Preparation of Data

Ordinal Values

- Known order of values but unknown distance between values.

- Note: addition and subtraction don't make sense

- Distinction between nominal and ordinal not always clear.

Example: attribute "temperature" in weather data

Values: "hot" > "mild" > "cool"

Preparation of Data

Interval Values

- Interval quantities are not only ordered but measured in fixed and equal units
- Difference of two values makes sense
- Sum or product doesn’t make sense
- Zero point is not defined!

Example 1: attribute “year”

Preparation of Data

Ratio Values

- Ratio quantities are ones for which the measurement scheme defines a zero point
- All mathematical operations are allowed
- But: is there an “inherently” defined zero point?
- Answer depends on scientific knowledge (e.g. Fahrenheit knew no lower limit to temperature)
- Distance between an object and itself is zero Ratio quantities are treated as real numbers

Example: “distance” in kilometers

Preparation of Data

We may have to convert nominal data to numeric in some models.

A conversion method from nominal to binary:

Outlook
Sunny
Rainy
Overcast

Sunny	Rainy	Overcast
1	0	0
0	1	0
0	0	1

Preparation of Data

Possible Problems:

- Which features to include?
  - Identify relevant features.
- How many instances should we use?
- How to deal with collinearity?

Training and Testing

Training phase: You present your data from your "gold standard" and train your model, by pairing the input with expected output.

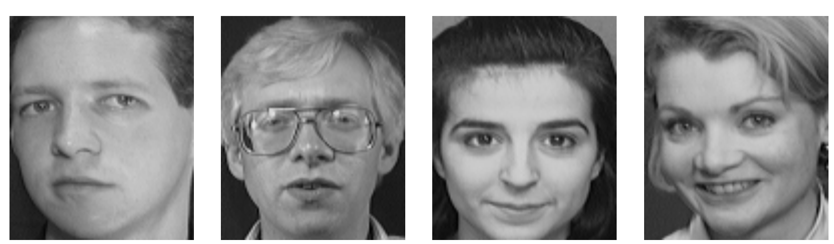
Validation/Test phase: In this phase, trained model is tested on an independent dataset with the same features.

Application phase: Now we apply the freshly-developed model to the real-world data and get the results.

Training and Testing

Training examples of a person

Test images

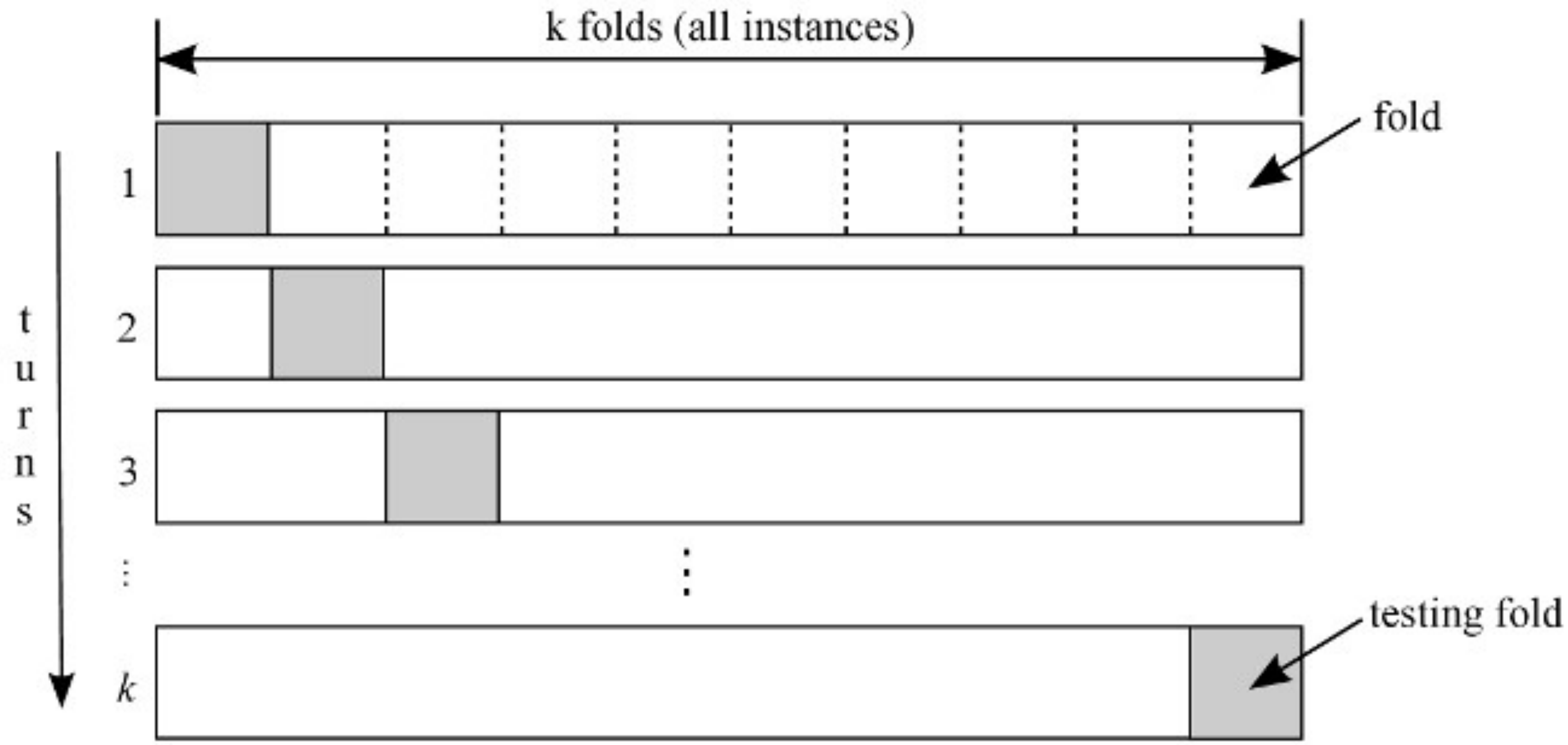


ORL dataset,
AT&T Laboratories, Cambridge UK

k-fold Cross Validation

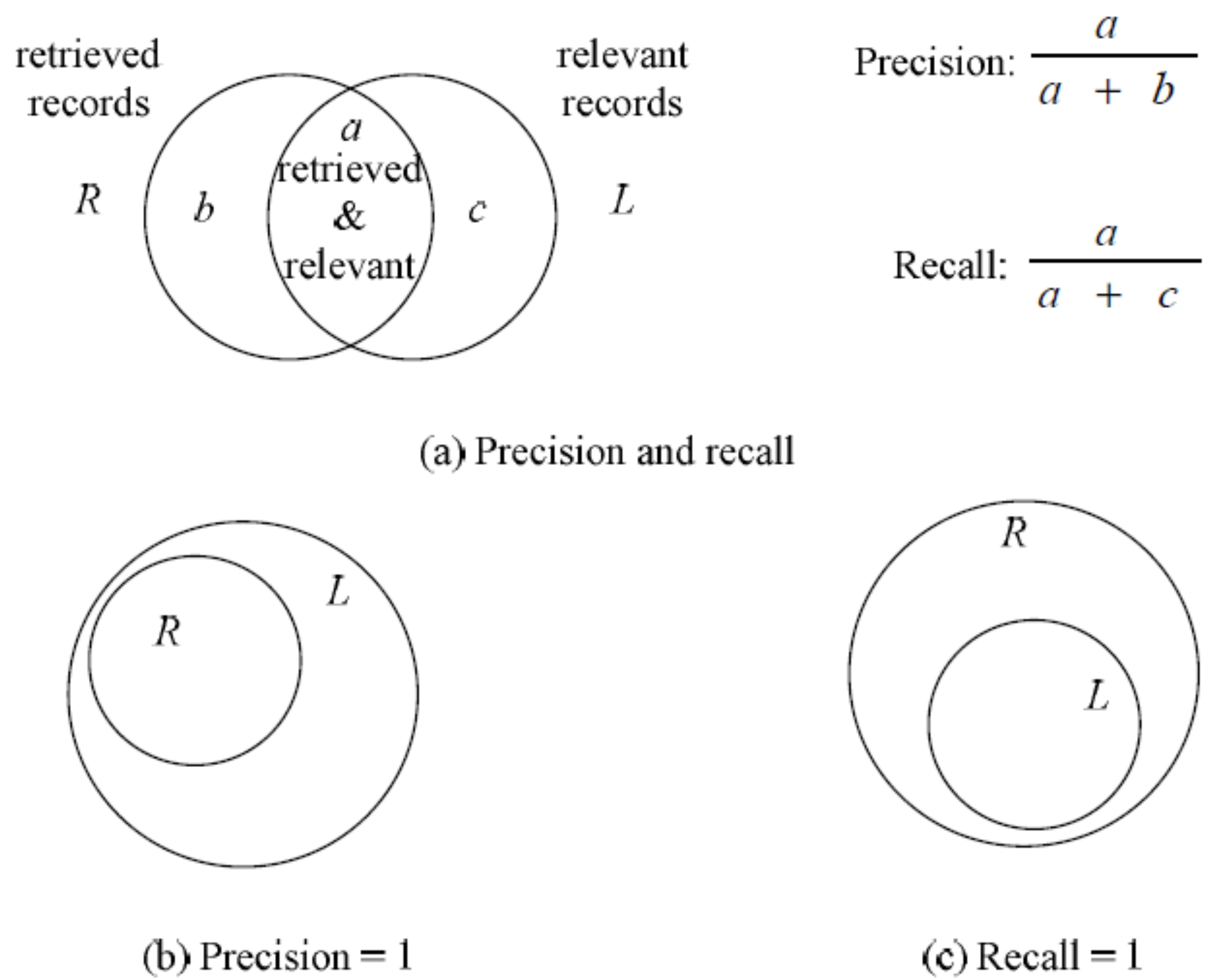
- The dataset is divided into training and test datasets k turns.

  - Each turn #instances/k are used for testing and the rest is used for training.



Performance Evaluation

Precision and Recall



Performance Evaluation

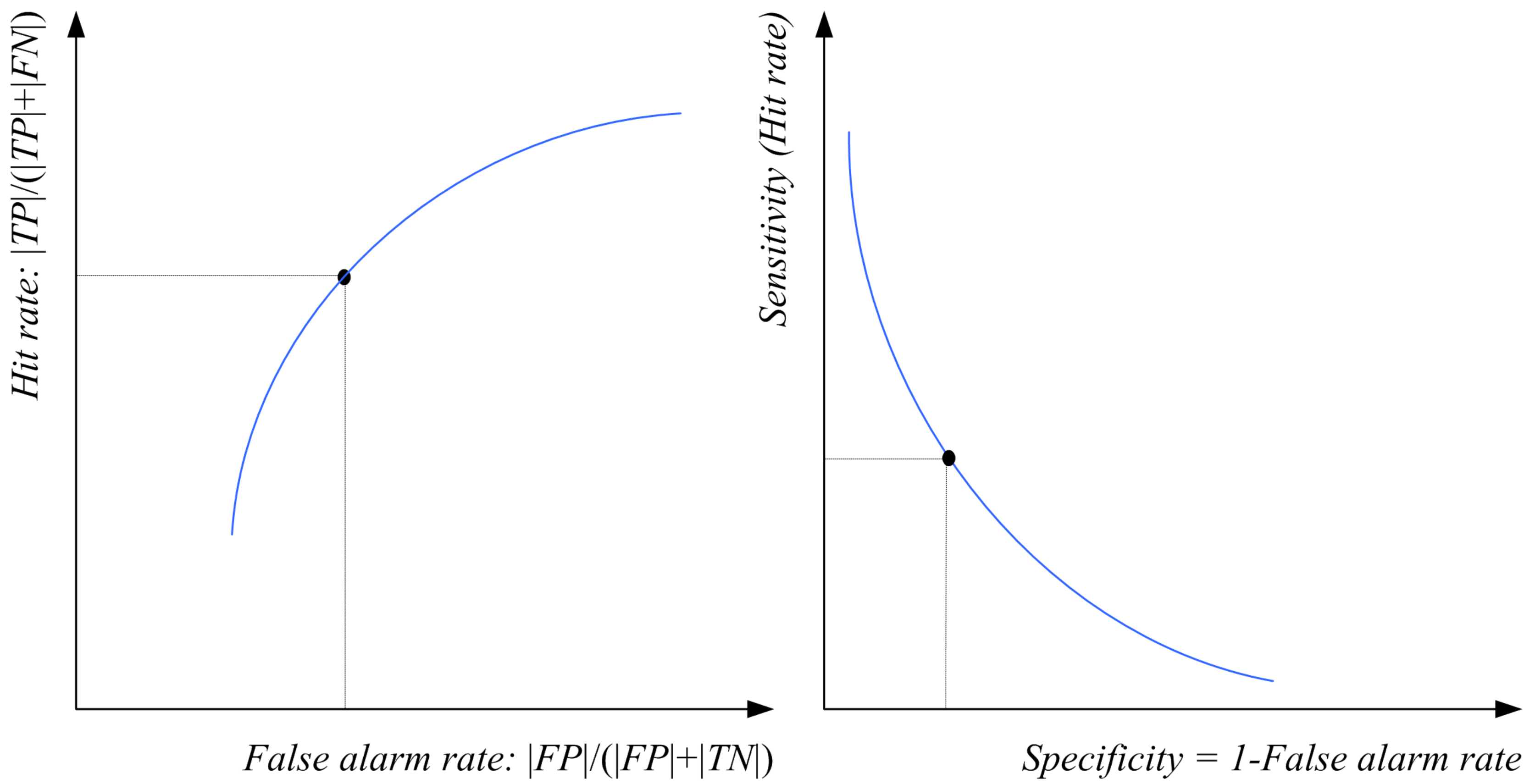
Classification Performance Measures

	Predicted class	
True Class	Yes	No
Yes	TP: True Positive	FN: False Negative
No	FP: False Positive	TN: True Negative

- Error rate = # of errors / # of instances = (FN+FP) / N
- Recall = # of found positives / # of positives
  = TP / (TP+FN) = sensitivity = hit rate
- Precision = # of found positives / # of found
  = TP / (TP+FP)
- Specificity = TN / (TN+FP)
- False alarm rate = FP / (FP+TN) = 1 - Specificity

Performance Evaluation

ROC Curve



Performance Evaluation

Comparing two algorithms

Sign test: Count how many times *A* beats *B* over *N* datasets, and check if this could have been by chance if A and B did have the same error rate

Comparing multiple algorithms

Kruskal-Wallis test: Calculate the average rank of all algorithms on N datasets, and check if these could have been by chance if they all had equal error.

If KW rejects, we do pairwise posthoc tests to find which ones have significant rank difference

Interpretation of Results

- Generalization issues:
    - If the model has as many degrees of freedom as the data, it can fit the training data perfectly but the objective is generalization.
    - Different models might perform best under different scenarios.

- If the performance is too low:
    - Can we increase the information content in the dataset?
    - Might using more instances help?

Estimation of Benefit

Estimation of the benefit

- Might be more important (and *harder to find)* than any other performance measure.
- Depends on the problem
  - Lives saved
  - Money made
  - Customer satisfaction
- We need a benefit estimation model.

Example of a Benefit Analysis

- The programmer-hours saved by using a software fault estimation model can be estimated.
- From this number savings of the company can be calculated by using the average programmer salary.

Automation of Experiments

- By automating the experiment the practitioner can save a lot of time.

- An ideal experiment should be automated completely. Execution of one script should finish all the steps (dataset formation, model runs) and generate a report.

- The versions of the data, model parameters and results may be tracked with a version control system. Using a database might also be helpful.

Dealing with Missing Data

Reasons for Missing Data

Certain attributes are available for only for a subset of the samples.

	bankname	bank	year	quarter	quarters	beta	leverage	roa	r_rwa	rwa_assets
178	Sparebank SMN	3	2011	4	2011q4	.7119	12.9143	.00277	.003803	.739655
179	Sparebank SMN	3	2012	1	2012q1	.0361	12.528	.002714	.003588	.773668
180	Sparebank SMN	3	2012	2	2012q2	.6157	12.3613	.002302	.003043	.740516
181	Sparebank SMN	3	2012	3	2012q3	.3987	12.5357	.002801	.003756	.751244
182	Sparebank SMN	3	2012	4	2012q4	.4382	11.5395	.002388	.003153	.763566
183	Sparebank SMN	3	2013	1	2013q1	.804	11.436	.002935	.00389	.745497
184	Sparebank Vest	4	1998	1	1998q1	.4144	.	.	.	.
185	Sparebank Vest	4	1998	2	1998q2	.1306	.	.	.	.
186	Sparebank Vest	4	1998	3	1998q3	.1818	.	.	.	.
187	Sparebank Vest	4	1998	4	1998q4	.3931	.	.	.	.
188	Sparebank Vest	4	1999	1	1999q1	-.3533	.	.004946	.	.
189	Sparebank Vest	4	1999	2	1999q2	.4742	15.9602	.002861	.004298	.66559
190	Sparebank Vest	4	1999	3	1999q3	-.113	.	.002546	.	.
191	Sparebank Vest	4	1999	4	1999q4	.4135	14.2458	.004057	.006104	.648981
192	Sparebank Vest	4	2000	1	2000q1	.1378	.	.002616	.	.
193	Sparebank Vest	4	2000	2	2000q2	.0917	15.2056	.00157	.002422	.64823
194	Sparebank Vest	4	2000	3	2000q3	-.1545	15.6238	.00257	.003951	.652837
195	Sparebank Vest	4	2000	4	2000q4	.2499	15.2741	.001703	.002639	.638658
196	Sparebank Vest	4	2001	1	2001q1	.4581	15.2077	.000838	.001301	.649364
197	Sparebank Vest	4	2001	2	2001q2	.2473	15.5972	.001753	.002712	.64303
198	Sparebank Vest	4	2001	3	2001q3	.674	15.7925	.001156	.001774	.660849
199	Sparebank Vest	4	2001	4	2001q4	.0563	15.146	.000589	.000876	.682053
200	Sparebank Vest	4	2002	1	2002q1	-.0516	15.4266	.002142	.003115	.693378

Handling Missing Data

Throw away cases with missing values

- In some data sets, most cases get thrown away
- If missing not random, throwing away cases can bias sample towards certain kinds of cases

Treat "missing" as a new attribute value

- What value should we use to code for missing with continuous or ordinal attributes?
- If missing causally related to what is being predicted?
  - Option 1: Fill-in with mean, median, or most common value
  - Option 2: Predict missing values using machine learning

References

- Designing experiments: https://www.cs.purdue.edu/homes/neville/courses/573/readings/08_design-and-analysis-expts.pdf
- Stats QA site: http://stats.stackexchange.com/

Week 3 Application Part

January 29, 2015

Identification of Missing Values

```r
a <- cbind(x=c(1,2,NA), y=c(NA,1,2))
a
```

```
##       x  y
## [1,]  1 NA
## [2,]  2  1
## [3,] NA  2
```

```r
complete.cases(a)
```

```
## [1] FALSE  TRUE FALSE
```

Identification of Missing Values

```r
a <- cbind(x=c(1,2,NA), y=c(NA,1,2))
a[!complete.cases(a),]
```

```
##       x  y
## [1,]  1 NA
## [2,] NA  2
```

Removing Instances with Missing Values

```
a <- cbind(x=c(1,2,NA), y=c(NA,1,2))
na.omit(a)
```

```
##      x y
## [1,] 2 1
## attr(,"na.action")
## [1] 3 1
## attr(,"class")
## [1] "omit"
```

Changing Missing Values

```r
a <- cbind(x=c(1,2,NA), y=c(NA,1,2))
a[is.na(a)] <- 0
a
```

```
##      x y
## [1,] 1 0
## [2,] 2 1
## [3,] 0 2
```

Creation of Test and Training Data

Create random split

```
rn_train <- sample(nrow(iris), floor(nrow(iris)*0.7))
train <- iris[rn_train,]
test <- iris[-rn_train,]
```

Creation of Test and Training Data

Cross validation:

```
library(caret)
```

```
## Loading required package: lattice
## Loading required package: ggplot2
```

```
library(mlbench)
data(Sonar)
folds <- createFolds(Sonar$Class)
```

Creation of Test and Training Data

Cross validation:

```
str(folds)
```

```
## List of 10
##  $ Fold01: int [1:20] 2 26 29 49 54 56 86 89 97 115 ...
##  $ Fold02: int [1:20] 7 16 43 53 61 70 83 93 95 116 ...
##  $ Fold03: int [1:21] 1 12 20 34 45 46 55 62 65 75 ...
##  $ Fold04: int [1:21] 4 21 33 35 51 67 72 79 81 82 ...
##  $ Fold05: int [1:21] 6 9 30 37 64 68 69 77 80 92 ...
##  $ Fold06: int [1:21] 17 19 36 44 47 52 57 59 63 96 ...
##  $ Fold07: int [1:21] 3 5 8 15 24 32 66 71 76 78 ...
##  $ Fold08: int [1:20] 11 18 23 25 39 42 58 60 87 98 ...
##  $ Fold09: int [1:22] 13 14 27 28 31 50 74 85 90 91 ...
##  $ Fold10: int [1:21] 10 22 38 40 41 48 73 84 88 94 ...
```

Creation of Test and Training Data

Cross validation:

```
for (f in folds){
  train <- Sonar[-f,]
  test <- Sonar[f,]
  #do stuff
  }
```

Lab Section - 1

1. Find the instances in airquality dataset with missing values.
2. Remove instances with missing data in airquality dataset.
3. Randomly split mtcars dataset to 80% training samples and 20% test samples.
4. Create 10 fold cross validation pairs for iris dataset.

Lab Section - 2

Find the following probabilities in mtcars dataset (hint: use prop.tables from week 2)

1. P(4cylinder | 3 gears)
2. P(4cylinder,4 gears)
3. P(4cylinder)

Homework

1. Individual work
2. Send code+report through blackboard.
3. No paper report required
4. Deadline: 22 Feb 2015 (hard)