Machine Learning and Data Mining

Today's slides partly taken from E. Alpaydin

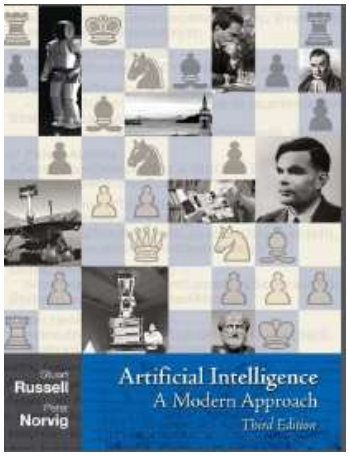
1

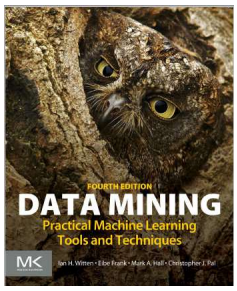
Organization

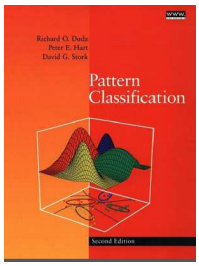
- Lecture: Tuesday, 8:00 – 9:30
  PDF of the slides will be uploaded before the lecture
- Exercise: Tuesday, 9:45 – 11:15 starts on 22nd of April
- Questions of the exercises are from the week before
- Exercise sheets will be uploaded on Tuesday
- Solution on exercises will be uploaded after the exercise session.

2

Literature





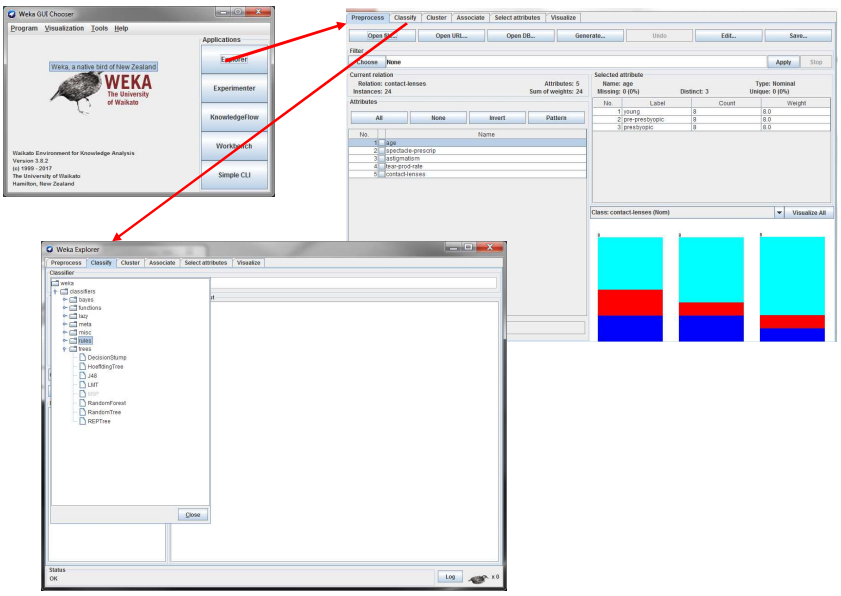


http://aima.cs.berkeley.edu
- with code repository
- further readings

https://www.cs.waikato.ac.nz/ml/weka/

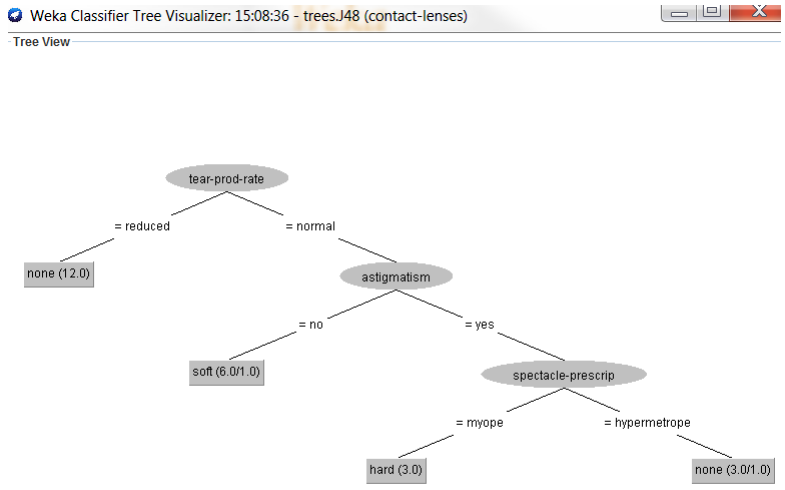
3

Weka



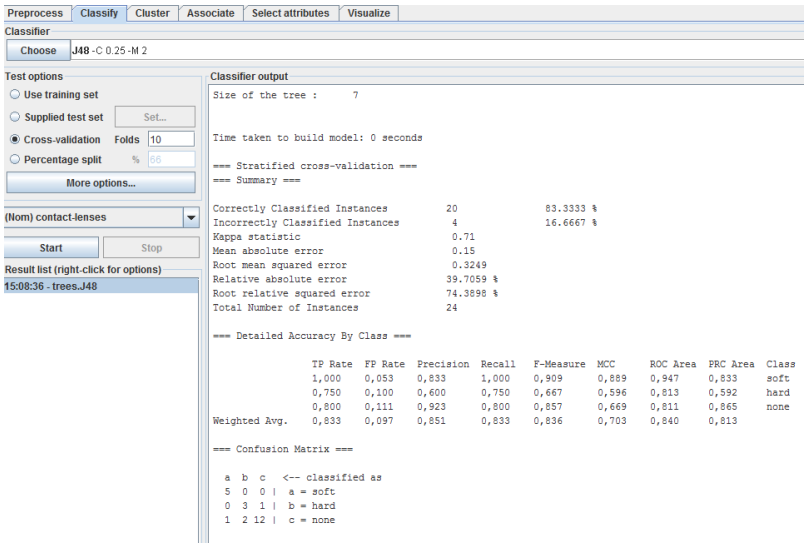
4

Weka


Weka Classifier Tree Visualizer: 15:08:36 - trees.J48 (contact-lenses)
Tree View
tear-prod-rate
= reduced
= normal
none (12.0)
astigmatism
= no
= yes
soft (6.0/1.0)
spectacle-prescrip
= myope
= hypermetrope
hard (3.0)
none (3.0/1.0)

5

Weka


Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) contact-lenses
Start
Stop
Result list (right-click for options)
15:08:36 - trees.J48
Classifier output
Size of the tree : 7
Time taken to build model: 0 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 20 83.3333 %
Incorrectly Classified Instances 4 16.6667 %
Kappa statistic 0.71
Mean absolute error 0.15
Root mean squared error 0.3249
Relative absolute error 39.7059 %
Root relative squared error 74.3898 %
Total Number of Instances 24
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
1,000 0,053 0,833 1,000 0,909 0,889 0,947 0,833 soft
0,750 0,100 0,600 0,750 0,667 0,596 0,813 0,592 hard
0,800 0,111 0,923 0,800 0,857 0,669 0,811 0,865 none
Weighted Avg. 0,833 0,097 0,851 0,833 0,836 0,703 0,840 0,813
=== Confusion Matrix ===
a b c <-- classified as
5 0 0 | a = soft
0 3 1 | b = hard
1 2 12 | c = none

6

Main topics covered

1. Introduction, Validation
2. Regression
3. Decision Trees
4. Version Spaces
5. Bayesian Networks
6. *K-N*earest *N*eighbor (KNN)
7. Logistic regression
8. Neuronal Networks
9. Support Vector Machines
10. Ensemble Learning
11. Clustering
12. Reinforcement Learning
13. Learning Association Rules

7

Machine Learning?

- „Machine learning is programming computers to optimize a *performance criterion* using example data.“
- Learning is used when:
  - Human expertise does not exist (navigating on planet X),
  - Humans are unable to explain their expertise (speech recognition)
  - Solution changes in time (routing on a computer network)
  - Solution needs to be adapted to particular cases (user biometrics)
  - …

8

Machine Learning

- Data is cheap and abundant;
  knowledge is expensive and scarce.
- Learning *general models* from data of particular examples
- Example in retail: Customer transactions to consumer behavior:

  People who bought "Da Vinci Code" also bought "The Five People You Meet in Heaven" (www.amazon.com)
- Build a model that is *a good and useful approximation* to the data.

Classes of ML Applications

- Learning Associations
- Supervised Learning
  - Classification
  - Regression
- Unsupervised Learning
- Reinforcement Learning

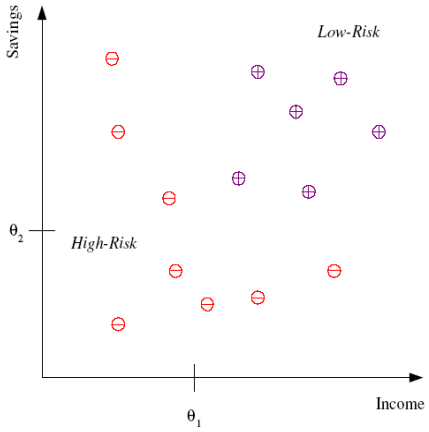
Learning Associations

- Basket analysis:
  $P(Y \mid X)$ probability that somebody who buys X also buys Y where X and Y are products/services.
  Example: P (chips | beer) = 0.7
- If we know more about customers or make a distinction among them:
  - $P(Y \mid X, D)$
    where D is the customer profile (age, gender, martial status, …)
  - In case of a Web portal, items correspond to links to be shown/prepared/downloaded in advance

11

Classification

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

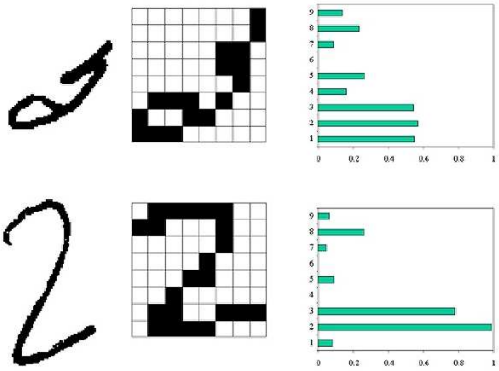


Discriminant: IF *income* > θ_1 AND *savings* > θ_2
THEN low-risk ELSE high-risk

12

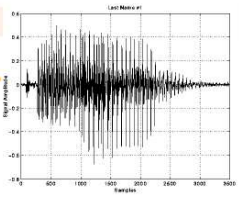
Character Recognition

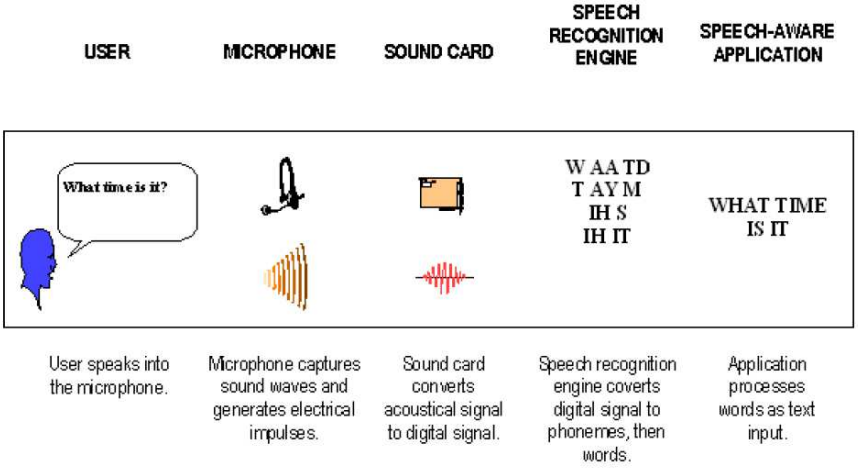
Want to learn how to recognize characters, even if written in different ways by different people



13

Example Pattern Recognition: Speech Recognition



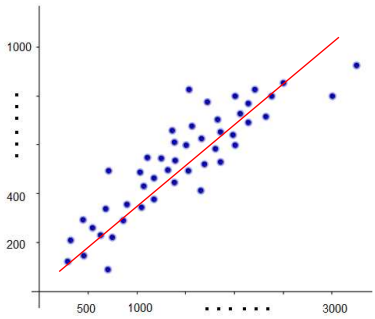


14

Regression

- Want to predict a continuous value not classes

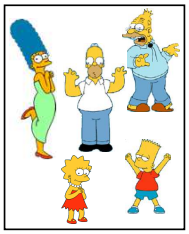
Value to be predicted



feature/attribute

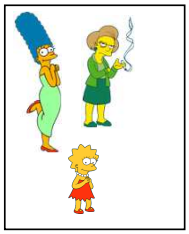
15

Unsupervised Learning

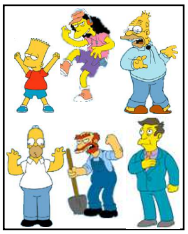


Simpson's Family

School Employees



Females

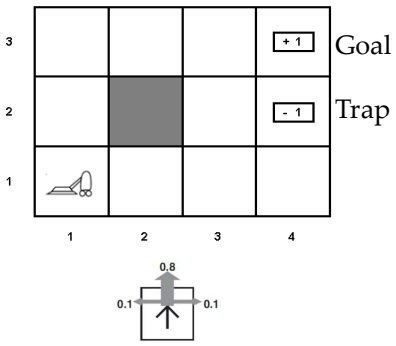


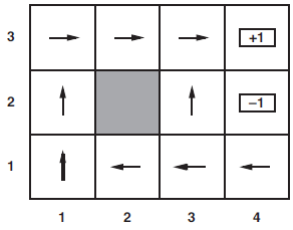
Males

16

Reinforcement Learning

- Robot navigation with uncertain actions.
- Have to find a plan for each possible situation.

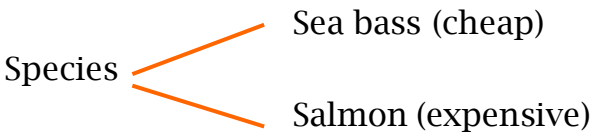




17

The ML process by example

- "Sorting incoming Fish on a conveyor according to species using optical sensing"



18

- Gathering data
  - Set up a camera and take some sample images to extract features
    - Length
    - Lightness
    - Width
    - Number and shape of fins
    - Position of the mouth, etc…
    - This is the set of all suggested features to explore for use in our classifier!

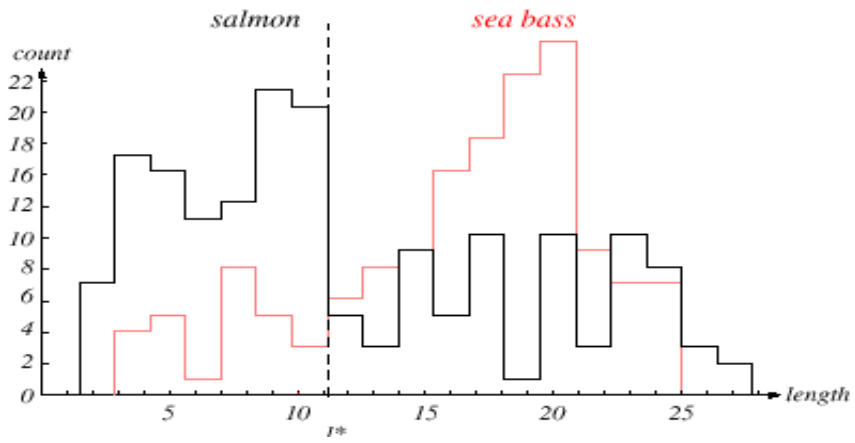
19

- Preprocessing
  - Use a segmentation operation to isolate fishes from one another and from the background
  - Information from a single fish is sent to a feature extractor whose purpose is to reduce the data by measuring certain features
  - The features are passed to a classifier

20

- Classification
  - Now we need (expert) information to find features that enables us to distinguish the species.
  - "Select the length of the fish as a possible feature for discrimination"
- Use the smallest number of features! Why?

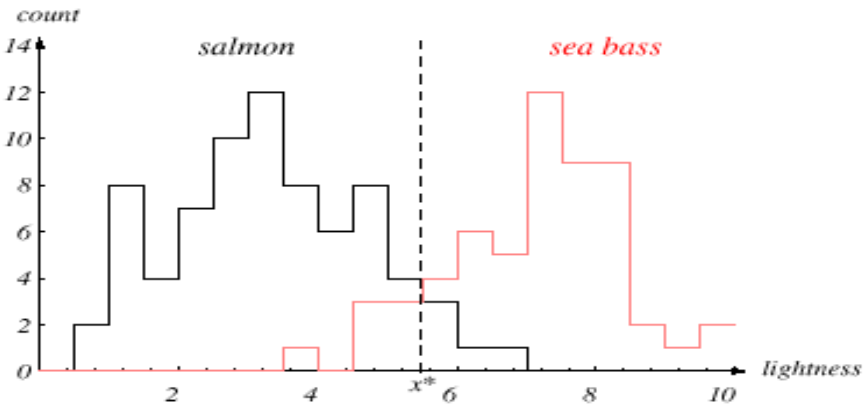
21



The length is a poor feature alone!

22

Select the lightness as a possible feature.



23

- Threshold decision boundary and cost relationship
  - Move our decision boundary toward smaller values of lightness in order to minimize the cost (reduce the number of sea bass that are classified salmon!)

Task of decision theory

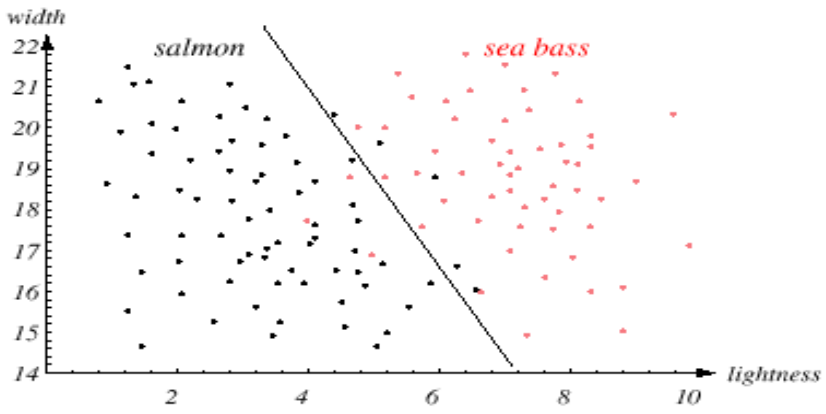
24

- Adopt the lightness and add the width of the fish

Fish → $x^T = [x_1, x_2]$

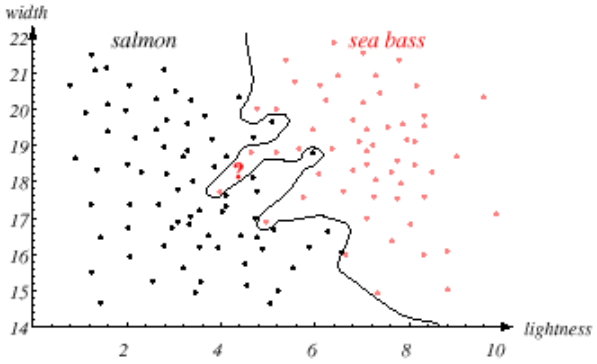
x_1: **Lightness** x_2: **Width**

25



26

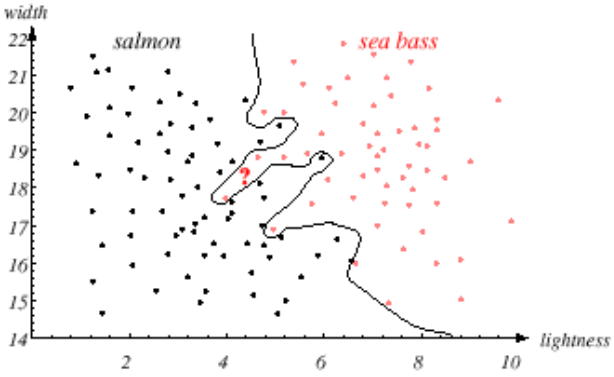
- We might add other features that are not correlated with the ones we already have. Precaution should be taken not to reduce the performance by adding "noisy features"
- Ideally, the best decision boundary should be the one which provides an optimal performance such as in the following figure:



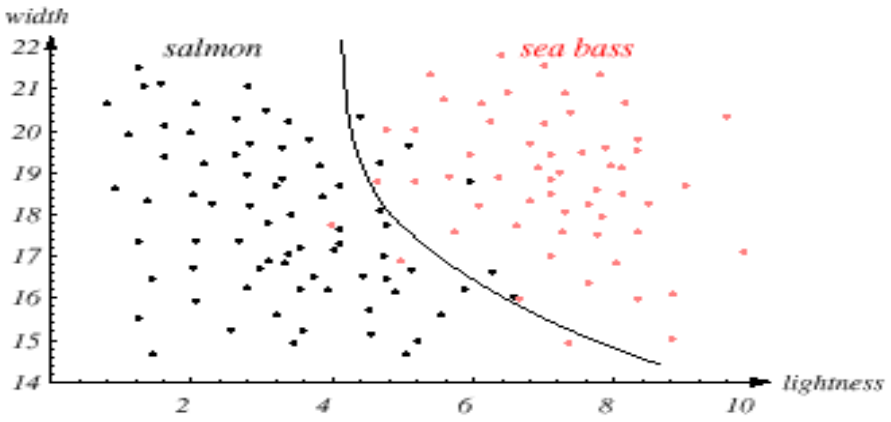
27

- However, our satisfaction is premature because the central aim of designing a classifier is to correctly classify novel input

Issue of generalization!



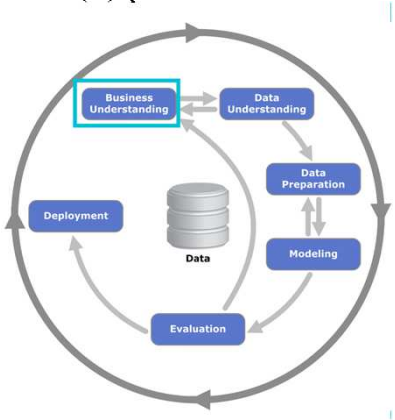
28



29

Standard data mining life cycle

- It is an iterative process with phase dependencies
- Consists of six (6) phases:



Cross-Industry Standard Process for Data Mining (CRISP-DM)

30

Learning a Class from Examples

- Class C of a "family car"
  - Prediction: Is car x a family car?
  - Knowledge extraction: What do people expect from a family car?
- Output:

  Positive (+) and negative (–) examples
- Input representation:

  x_1: price, x_2 : engine power

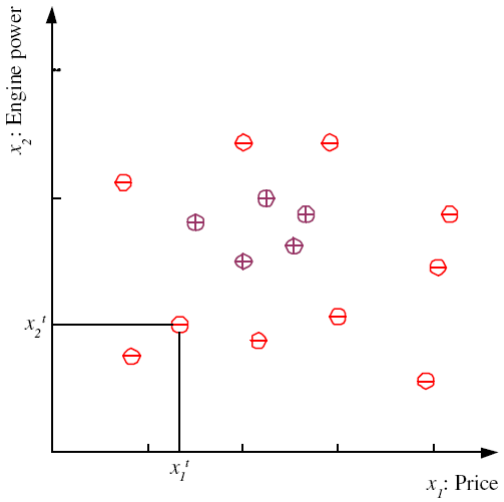
31

Training set $\mathcal{X}$

$$\mathcal{X} = \{\boldsymbol{x}^t, r^t\}_{t=1}^N$$

$$r = \begin{cases} 1 \text{ if } \boldsymbol{x} \text{ is positive} \\ 0 \text{ if } \boldsymbol{x} \text{ is negative} \end{cases}$$

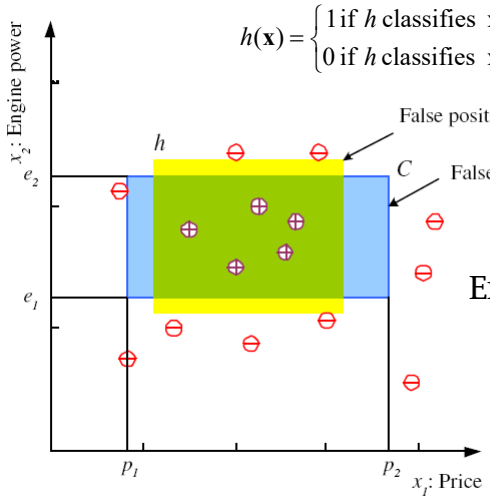
$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$



32

Hypothesis class $\mathcal{H}$

$$h(\mathbf{x}) = \begin{cases} 1 \text{ if } h \text{ classifies } \mathbf{x} \text{ as positive} \\ 0 \text{ if } h \text{ classifies } \mathbf{x} \text{ as negative} \end{cases}$$



Error of h on $\mathcal{H}$

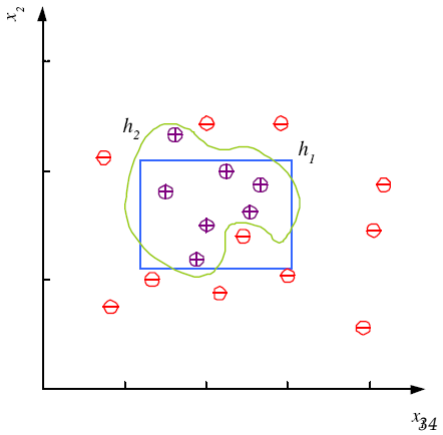
$$E(h \mid \mathcal{X}) = \sum_{t=1}^{N} l\left(h\left(\mathbf{x}^t\right) \neq r^t\right)$$

l(a ≠ b) = 1 if a ≠ b and is 0 if a=b.

Noise and Model Complexity

Use the simpler one because

- Simpler to use (lower computational complexity)
- Easier to train (lower space complexity)
- Easier to explain (more interpretable)
- Generalizes better (lower variance - Occam's razor)



Model Selection & Generalization

- Learning is an ill-posed problem; data is not sufficient to find a unique solution
- The need for inductive bias, assumptions about $\mathcal{H}$
- Generalization: How well a model performs on new data
- Overfitting: $\mathcal{H}$ more complex than C
- Underfitting: $\mathcal{H}$ less complex than C

35

Triple Trade-Off

- There is a trade-off between three factors (Dietterich, 2003):
  1. Complexity of $\mathcal{H}$, $c(\mathcal{H})$,
  2. Training set size, N,
  3. Generalization error, E, on new data
- As $N \uparrow$, $E \downarrow$
- As $c(\mathcal{H}) \uparrow$, first $E \downarrow$ and then $E \uparrow$

How to evaluate a model?

36

Evaluation of models: Holdout estimation

- The holdout method reserves a certain amount for testing and uses the remainder for training
  - Usually: one third for testing, the rest for training
  - Sample with replacement
- Problem: the samples might not be representative
  - Example: a class might be missing in the test data
- Advanced version uses stratification
  - Ensures that each class is represented with approximately equal proportions in both subsets

37

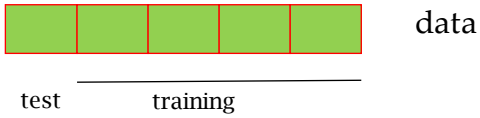
Repeated holdout method

- Holdout estimate can be made more reliable by repeating the process with different subsamples
  - In each iteration, a certain proportion is randomly selected for training (possibly with stratification)
  - The error rates on the different iterations are averaged to yield an overall error rate
- This is called the repeated holdout method
- Still not optimum: the different test sets may overlap
  - Can we prevent overlapping?

38

Cross-validation

- *Cross-validation* avoids overlapping test sets
  - First step: split data into subsets of equal size
  - Second step: use each subset in turn for testing, the remainder for training



  - Here we get 5 models and average the error

Cross-validation

- *Cross-validation* avoids overlapping test sets
  - First step: split data into **k** subsets of equal size
  - Second step: use each subset in turn for testing, the remainder for training
- Called ***k-fold*** *cross-validation*
- Often the subsets are stratified before the cross-validation is performed
- The error estimates are averaged to yield an overall error estimate

More on cross-validation

- Standard method for evaluation: stratified ten-fold cross-validation
- Why ten?
  - Extensive experiments have shown that this is the best choice to get an accurate estimate
  - There is also some theoretical evidence for this
- Even better: repeated stratified cross-validation
  - E.g. ten-fold cross-validation is repeated ten times and results are averaged (reduces the variance)

41

The bootstrap

- CV uses sampling without replacement
  - The same instance, once selected, can not be selected again for a particular training/test set
- The bootstrap uses sampling with replacement to form the training set
  - Sample a dataset of *n* instances with replacement to form a new dataset of *n* instances
  - Use this data as the training set
  - Use the instances from the original dataset that **don't occur** in the new training set for testing

42

The bootstrap

- Also called the 0.632 bootstrap
- A particular instance has a probability of $1 - \frac{1}{n}$ of **not** being picked
- Thus its probability of ending up in the test data is:

$$(1 - \frac{1}{n})^n \approx e^{-1} \approx 0.368$$

- This means the training data will contain approximately 63.2% of the instances

43

Estimating error with the bootstrap

- The error estimate on the test data will be very pessimistic
  - Trained on just ~63% of the instances
- Therefore, combine it with the resubstitution error:

$$err = 0.632 \times e_{\text{test instances}} + 0.368 \times e_{\text{training\_instances}}$$

- The resubstitution error gets less weight than the error on the test data
- Repeat process several times average the results

44

Confusion Matrix: Classification

- Assume a binary classification. Four classification outcomes are possible, which can be displayed in a confusion matrix:

True class

Predicted class	Positive	Negative
Positive	a	b
Negative	c	d

True positives (TP): class members classified as class members : a
True negatives (TN): class non-members classified as non-members : d
False positives (FP): class non-members classified as class members : c
False negatives (FN): class members classified as class non-members : b

45

Accuracy/error rate

- The error rate is sometimes an inadequate measure of the performance of an algorithm, it doesn't take into account the cost of making wrong decisions.

$$\frac{|TP| + |TN|}{|TP| + |TN| + |FP| + FN|}$$

- Example: Based on chemical analysis of the water try to detect an oil slick in the sea.

  False positive: wrongly identifying an oil slick if there is none.
  False negative: fail to identify an oil slick if there is one.

- Here, *false negative* (environmental disasters) are much more costly than *false positive* (false alarms).

46

Precision and Recall

- Precision: number of class members classified correctly over total number of instances classified as class members.

$$\text{Precision} = \frac{|TP|}{|TP|+|FP|}$$

"What proportion of positive identifications was actually correct?"

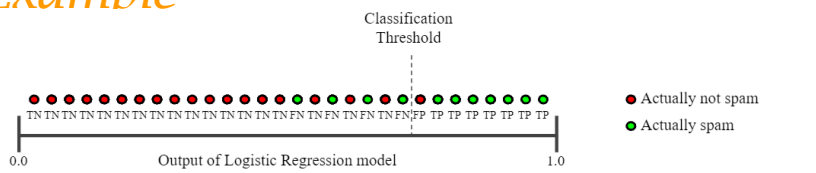
- Recall: number of class members classified correctly over total number of class members.'

$$\text{Recall} = \frac{|TP|}{|TP|+|FN|}$$

"What proportion of actual positives was identified correctly?"

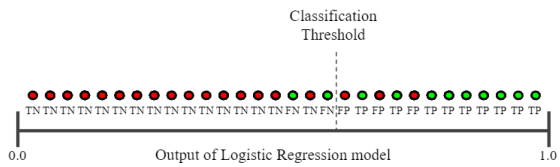
47

Example



$$\text{Precision} = \frac{TP}{TP+FP} = \frac{7}{7+1} = 0.88$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{7}{7+4} = 0.64$$



$$\text{Precision} = \frac{TP}{TP+FP} = \frac{9}{9+3} = 0.75$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{9}{9+2} = 0.82$$

48

F-measure

- Precision and recall can be combined in the F-measure:

$$\text{F-measure} = 2 \times \frac{precision \times recall}{precision + recall}$$

Example:
For the oil slick scenario, we want to maximize *recall* (avoiding environmental disasters). Maximizing *precision* (avoiding false alarms) is less important.

The F-measure can be used if precision and recall are equally important.

49

Evaluating Numeric Prediction

- Error rate, precision, recall are used for classification tasks. They are less suitable for tasks were a numeric quantity has to be predicted.

  Examples: predict the arrival time of a vessel. We want to measure how close the model is to the actual time.

  $p_1,\ldots, p_n$: predicted values of the for instances $1,\ldots,n$
  $a_1,\ldots,a_n$: actual values of the for instances $1,\ldots,n$

  There are several measure that compare a_i and p_i.

50

(Root) Mean Squared Error

- Mean squared error measures the mean difference between actual and predicted values:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(a_i - p_i)^2$$

- Often also the root mean squared error is used:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(a_i - p_i)^2}$$

- Mean absolute error

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|a_i - p_i|$$

51

Summary: Dimensions of a Supervised Learner

1. Model : $g(\boldsymbol{x} \mid \theta)$
2. Loss function: $E(\theta \mid \mathcal{X}) = \sum_t L(\boldsymbol{r}^t, g(\boldsymbol{x}^t \mid \theta))$
3. Optimization procedure:

$$\theta^* = \arg\min_{\theta} E(\theta \mid \mathcal{X})$$

52