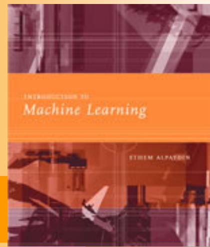


Lecture Slides for

INTRODUCTION TO

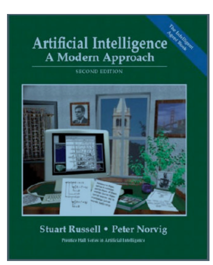
Machine Learning

ETHEM ALPAYDIN

alpaydin@boun.edu.tr
http://www.cmpe.boun.edu.tr/~ethem/i2ml

Lab Class and literature

- Friday, 9.00 – 10.00, Harburger Schloßstr. 20, 215
- First Lab Class will be announced on
  http://www.sts.tu-harburg.de/teaching/ss-09/FMDM/fmdm-ss-09.html
- Mainly used book



Official title: Software Architectures

Software Analysis

Zeit:
Freitag 15:15 bis 16:00, Freitag 16:15 bis 18:00

Semester:
SS 09

Nächster Termin:
Fr., 17.04.2009, 16:15 - 18:00, (DE17 2019)

Vorbesprechung:
keine

Veranstaltungsort:
Fr. wöchentlich 16:15-18:00 (DE17 2019) *Lecture*
Fr. wöchentlich 15:15-16:00 (SBS95 E2054P4b) *Exercise*

DozentIn:
Prof. Dr. Sibylle Schupp

TutorInnen:
- Volker Menrad
- Gustav Munkby

Persönlicher Status:
Sie sind nicht als TeilnehmerIn der Veranstaltung eingetragen.

Berechtigungen:
Lesen:
Schreiben:

Veranstaltungstyp:
Vorlesung in der Kategorie Lehre

Kommentar/Beschreibung:
Many software-engineering activities---in particular for assessing or improving software quality---start out as analytical problem: a testing technique, for example, might require an understanding of the control flow of software; a verification task might require knowledge of the state space; refactoring might need an understanding of the intended dependencies; and optimization might depend on a good approximation of the call graph. It is therefore crucial to have analytical methods available that can extract the desired information about a software system, and ideally can do so automatically.

In this course, we discuss the classical techniques of static program analysis. We will study the core ideas and their theoretical underpinnings, and see their workings in practical software-engineering settings, with a particular focus on testing. If you are a practitioner or future software engineer, or have for some other reason been faced with questions like "When can I stop testing?", "Is my model good enough?", or "Have I now inadvertently changed the program?", this course might be of interest to you. The course also provides a good preparation for a variety of advanced courses in the field of software engineering.

Overview (12 lectures total):
* Data-flow analysis (3 lectures)
* Symbolic execution and abstract interpretation (3 lectures)
* Control-flow analysis (3 lectures)
* Type and effect systems (2 lectures)
* Wrap-up

- Data-Flow Analysis I
- Data-Flow Analysis II
- Data-Flow Analysis III
- Symbolic Execution and Abstract Interpretation I
- Symbolic Execution and Abstract Interpretation II
- Symbolic Execution and Abstract Interpretation III
- Control-Flow Analysis I
- Control-Flow Analysis II
- Control-Flow Analysis III
- Type and Effect Systems I
- Type and Effect Systems II

Studienbereich:
Technische Universität Hamburg-Harburg (TUHH) > Studiendekanat Elektrotechnik und Informations

Heimat-Einrichtung:
Institut für Softwaresysteme (E-16)

Anzahl der Teilnehmenden: 14
Postings: 1
DozentInnen: 1
TutorInnen: 2
Sonstige: 11

Auch geeignet für Studierende,
die einen Schein für den Seminartyp "Anatomie eines I+K-Systems" benötigen

Softwaresysteme: Informatikseminar
Software-Implemented Hardware Fault Tolerance (SIHFT) Techniques

Zeit:
Freitag 16:00 bis 17:30, Termine am 17.4. 15:00 - 16:30

Semester:
SS 09

Nächster Termin:
Fr., 17.04.2009, 15:00 - 16:30, (SBS95 E2054P4b)

Vorbesprechung:
keine

Veranstaltungsort:
Fr. wöchentlich 16:00-17:30 k.A. *Sitzung*
Fr. 17.04.2009 15:00-16:30 (SBS95 E2054P4b)

DozentIn:
Prof. Dr. Sibylle Schupp

TutorInnen:
- Volker Menrad
- Gustav Munkby

Persönlicher Status:
Sie sind nicht als TeilnehmerIn der Veranstaltung eingetragen.

Berechtigungen:
Lesen:
Schreiben: (Registrierungsmail beachten!)

Veranstaltungstyp:
Seminar in der Kategorie Lehre

Kommentar/Beschreibung:
Computer-based systems that must function properly even in the presence of hardware errors traditionally use hardware mechanisms for their protection, for example, special-purpose hardware or redundant hardware. Software-Implemented Hardware Fault Tolerance (SIHFT) is a recent design paradigm for detecting and possibly correcting those hardware faults solely by means of software. SIHFT techniques can therefore provide a cost-efficient alternative for safety-critical systems in various application areas, including the automotive and telecommunication domain. In this seminar, we will get an overview of the major techniques known today.

Topics
* Basic Terminology
* Hardening Data
* Hardening the Control Flow
* Fault Tolerance
* Fault Injection

See "Literature" for the accompanying textbook by Goloubeva et al. This textbook is on-line available within the TUHH intranet.

Language: English or German, or both, depending on the audience

Studienbereich:
Technische Universität Hamburg-Harburg (TUHH) > Studiendekanat Elektrotechnik und Informationstechnik

Heimat-Einrichtung:
Institut für Softwaresysteme (E-16)

Anzahl der Teilnehmenden: 9
Postings: 1
Dokumente: keine
DozentInnen: 1
TutorInnen: 2
Sonstige: 6

CHAPTER 1:

Introduction

Why "Learn" ?

- Machine learning is programming computers to optimize a *performance criterion* using example data or past experience.
- There is no need to "learn" to calculate payroll
- Learning is used when:
  - Human expertise does not exist (navigating on Mars),
  - 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)

What We Talk About When We Talk About "Learning"

- Learning general models from a data of particular examples
- Data is cheap and abundant (data warehouses, data marts); knowledge is expensive and scarce.
- 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.

Data Mining

- Retail: Market basket analysis, Customer relationship management (CRM)
- Finance: Credit scoring, fraud detection
- Manufacturing: Optimization, troubleshooting
- Medicine: Medical diagnosis
- Telecommunications: Quality of service optimization
- Bioinformatics: Motifs, alignment
- Web mining: Search engines
- ...

What is Machine Learning?

- Optimize a performance criterion using example data or past experience.
- Role of Statistics: Building mathematical models, core task is inference from a sample
- Role of Computer science: Efficient algorithms to
  - Solve the optimization problem
  - Representing and evaluating the model for inference

Sample of ML Applications

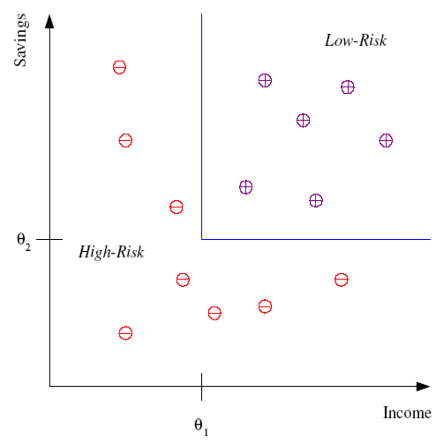
- Learning Associations
- Supervised Learning
  - Classification
  - Regression
- Unsupervised Learning
- Reinforcement Learning

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:
  - $P(X \mid Y, D)$
    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

Classification

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



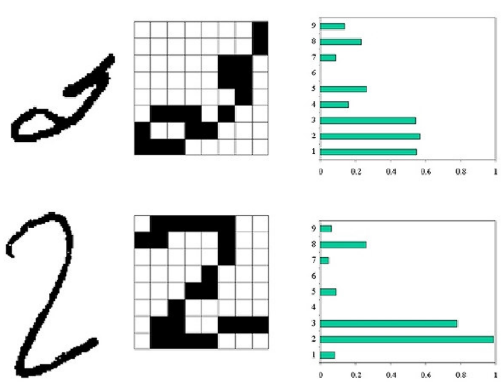
Discriminant: IF *income* > θ_1 AND *savings* > θ_2
THEN low-risk ELSE high-risk

Classification: Applications

- Aka Pattern recognition
- Character recognition: Different handwriting styles.
- Face recognition: Pose, lighting, occlusion (glasses, beard), make-up, hair style
- Speech recognition: Temporal dependency.
  - Use of a dictionary or the syntax of the language.
  - Sensor fusion: Combine multiple modalities; eg, visual (lip image) and acoustic for speech
- Medical diagnosis: From symptoms to illnesses
- Brainwave understanding: From signals to “states” of thought
- Reading text:
- ...

Character Recognition

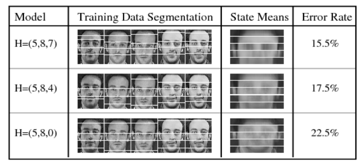
Want to learn how to recognize characters, even if written in different ways by different people



Face Recognition

Training examples of a person

Test images

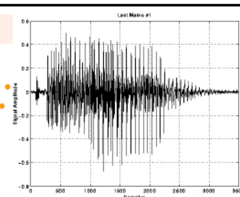


Model	Training Data Segmentation	State Means	Error Rate
H=(5,8,7)			15.5%
H=(5,8,4)			17.5%
H=(5,8,0)			22.5%

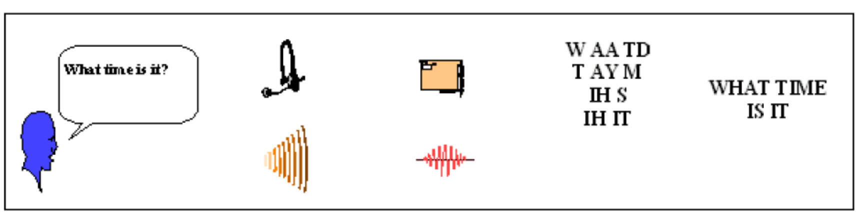
Figure 4.11: Segmentation and feature results for varying M

AT&T Laboratories, Cambridge UK

Example Pattern Recognition: Speech Recognition



USER	MICROPHONE	SOUND CARD	SPEECH RECOGNITION ENGINE	SPEECH-AWARE APPLICATION

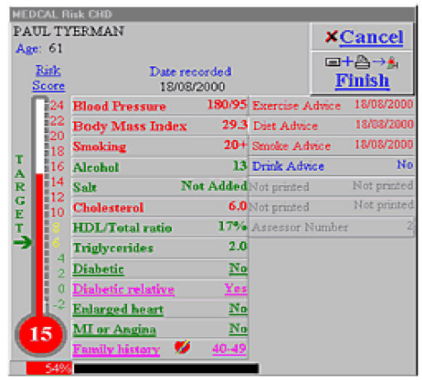


User speaks into the microphone.	Microphone captures sound waves and generates electrical impulses.	Sound card converts acoustical signal to digital signal.	Speech recognition engine coverts digital signal to phonemes, then words.	Application processes words as text input.

Medical diagnosis

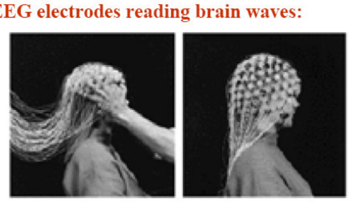
Inputs: relevant info about patient, symptoms, test results, etc.

Output: Expected illness or risk factors

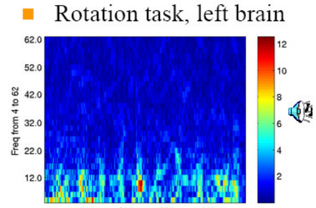


Example Pattern Recognition: Interpreting Brainwaves

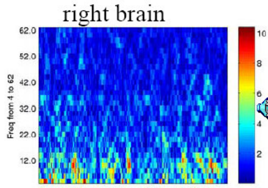
EEG electrodes reading brain waves:



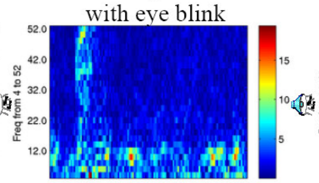
- Rotation task, left brain



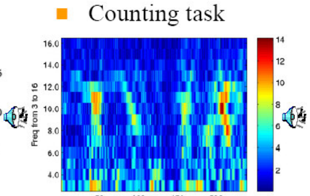
- Rotation task, right brain



- Resting task, with eye blink



- Counting task



Example Pattern Recognition: Reading text

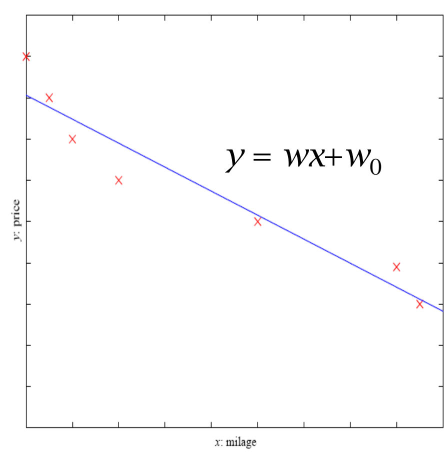
- Can you read this?
  - Aircndcog to a rseerhcaer at Cbiardmge Urensvitiy, it dsoen't mtetar in waht oderr the letrtes in a wrod are, the olny ipnaotmrt tihng is taht the fsrit and lsat lteter be at the rgiht plcae. The rset can be a toatl mses and you can slitl raed it wutohit porlebm. Tehy spectluae taht tihs is bseuace the hmaun mnid deos not raed erevy leettr by iesltf but the wrod as a whloe. Wtehehr tihs is ture or not is a ponit of deabte.

- Clearly, the brain has learned syntax and semantics of language, including contextual dependencies, to make sense of of this ☺

- For fun: Here’s a web page where you can create your own jumbled text: http://www.stevesachs.com/jumbler.cgi

Regression

- Example: Price of a used car
- x : car attributes
  y : price
  $$y = g(x \mid \theta)$$
  $g(\,)$ model,
  θ parameters



Supervised Learning: Uses

- 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

Unsupervised Learning

- Learning "what normally happens"
- No output (we do not know the right answer)
- Clustering: Grouping similar instances
- Example applications
  - Customer segmentation in CRM
    - Company may have different marketing approaches for different groupings of customers
  - Image compression: Color quantization
    - Instead of using 24 bits to represent 16 million colors, reduce to 6 bits and 64 colors, if the image only uses those 64 colors
  - Bioinformatics: Learning motifs
  - Document Classification in unknown Domains.

Reinforcement Learning

- Learning a policy: A sequence of actions/outputs
- No supervised output but delayed reward
- Credit assignment problem
- Game playing
- Robot in a maze
- Multiple agents, partial observability, ...

An Extended Example

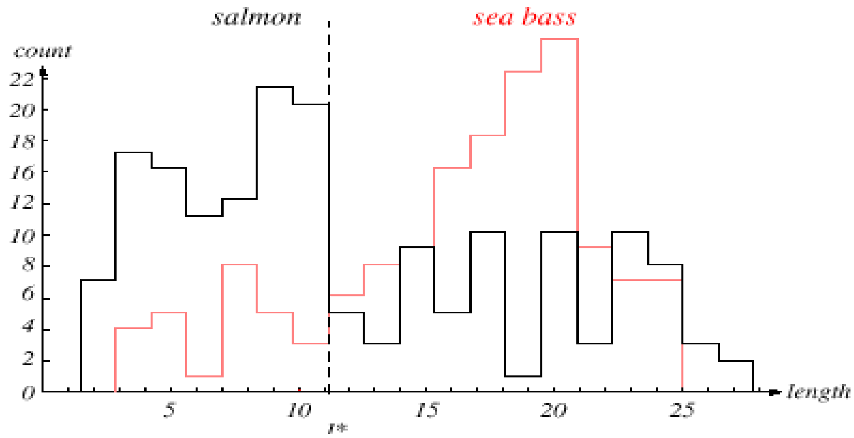
- "Sorting incoming Fish on a conveyor according to species using optical sensing"

Species
- Sea bass
- Salmon

- Problem Analysis
  - 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!

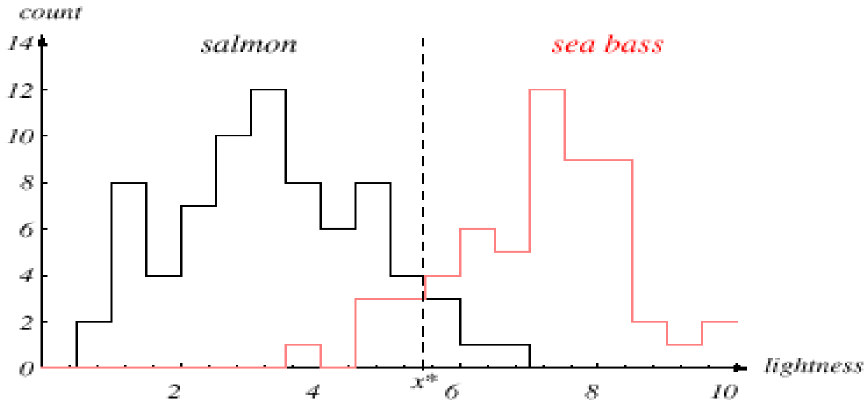
- Preprocessing
  - Use a segmentation operation to isolate fishes from one another and from the background
- Information from a single fish is sent to a feature extractor whose purpose is to reduce the data by measuring certain features
- The features are passed to a classifier

- 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”

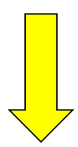


The length is a poor feature alone!
→ Cost of decision

Select the lightness as a possible feature.



- 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!)

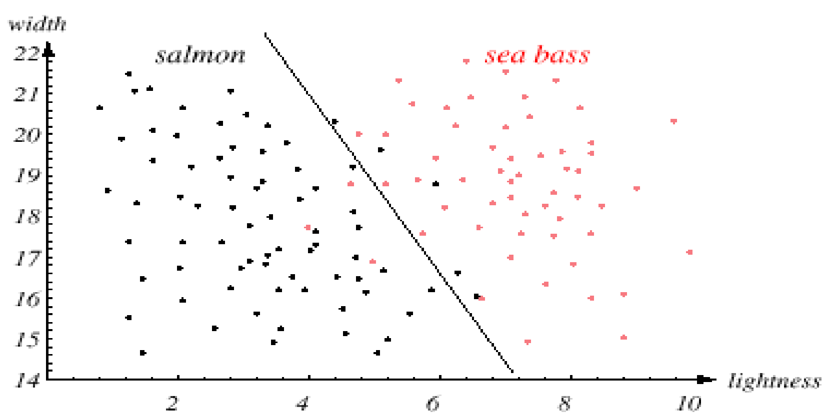


Task of decision theory

- Adopt the lightness and add the width of the fish

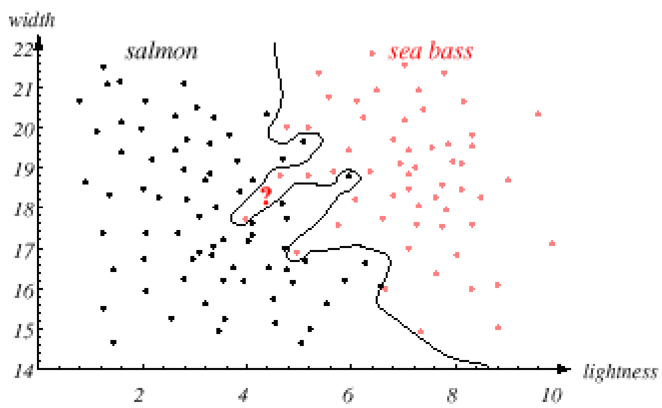
Fish → $x^T = [x_1, x_2]$

Lightness **Width**



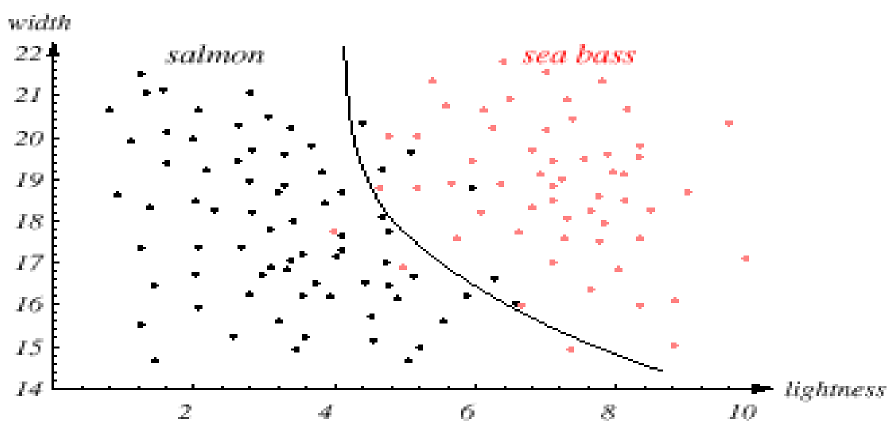
- We might add other features that are not correlated with the ones we already have. A precaution should be taken not to reduce the performance by adding such “noisy features”

- Ideally, the best decision boundary should be the one which provides an optimal performance such as in the following figure:



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

Issue of generalization!



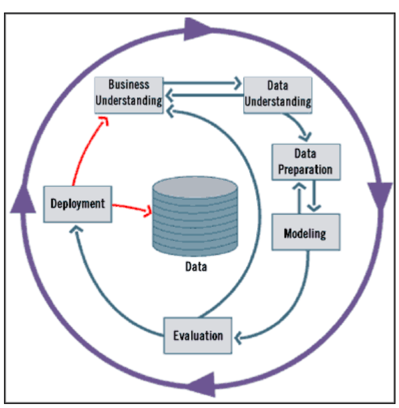
CRISP-DM

CRoss-Industry Standard Process
for Data Mining

STONY BROOK
STATE UNIVERSITY OF NEW YORK

Standard data mining life cycle

- CRISP (Cross-Industry Standard Process)
- It is an iterative process with phase dependencies
- IT consists of six (6) phases:



see
www.crisp-dm.org
for more
information

CRISP_DM

- Cross-industry standard developed in 1996
  - Analysts from SPSS/ISL, NCR, Daimler-Benz, OHRA
- Funding from European Commission
- Important Characteristics:
  - Non-proprietary
  - Application/Industry neutral
  - Tool neutral
  - General problem-solving process
  - Process with six phases but missing:
    - Saving results and updating the model

CRISP-DM Phases (1)

- Business Understanding
  - Understand project objectives and requirements
  - Formulation of a data mining problem definition
- Data Understanding
  - Data collection
  - Evaluate the quality of the data
  - Perform exploratory data analysis
- Data Preparation
  - Clean, prepare, integrate, and transform the data
  - Select appropriate attributes and variables

1. Business Understanding Phase

DaimlerChrysler's objectives are to reduce costs associated with warranty claims and improve customer satisfaction. Through conversations with plant engineers, who are the technical experts in vehicle manufacturing, the researchers are able to formulate specific business problems, such as the following:

- Are there interdependencies among warranty claims?
- Are past warranty claims associated with similar claims in the future?
- Is there an association between a certain type of claim and a particular garage?

The plan is to apply appropriate data mining techniques to try to uncover these and other possible associations.

2. Data Understanding Phase

The researchers make use of DaimlerChrysler's Quality Information System (QUIS), which contains information on over 7 million vehicles and is about 40 gigabytes in size. QUIS contains production details about how and where a particular vehicle was constructed, including an average of 30 or more sales codes for each vehicle. QUIS also includes warranty claim information, which the garage supplies, in the form of one of more than 5000 possible potential causes.

The researchers stressed the fact that the database was entirely unintelligible to domain nonexperts: "So experts from different departments had to be located and consulted; in brief a task that turned out to be rather costly." They emphasize that analysts should not underestimate the importance, difficulty, and potential cost of this early phase of the data mining process, and that shortcuts here may lead to expensive reiterations of the process downstream.

3. Data Preparation Phase

The researchers found that although relational, the QUIS database had limited SQL access. They needed to select the cases and variables of interest manually, and then manually derive new variables that could be used for the modeling phase. For example, the variable *number of days from selling date until first claim* had to be derived from the appropriate date attributes.

They then turned to proprietary data mining software, which had been used at DaimlerChrysler on earlier projects. Here they ran into a common roadblock—that the data format requirements varied from algorithm to algorithm. The result was further exhaustive preprocessing of the data, to transform the attributes into a form usable for model algorithms. The researchers mention that the data preparation phase took much longer than they had planned.

CRISP-DM Phases (2)

- Modeling
  - Select and apply appropriate modeling techniques
  - Calibrate model parameters to optimize results
  - If necessary, return to data preparation phase to satisfy model's data format
- Evaluation
  - Determine if model satisfies objectives set in phase 1
  - Identify business issues that have not been addressed
- Deployment
  - Organize and present the model to the "user"
  - Put model into practice
  - Set up for continuous mining of the data

4. Modeling Phase

Since the overall business problem from phase 1 was to investigate dependence among the warranty claims, the researchers chose to apply the following techniques: (1) Bayesian networks and (2) association rules. Bayesian networks model uncertainty by explicitly representing the conditional dependencies among various components, thus providing a graphical visualization of the dependency relationships among the components. As such, Bayesian networks represent a natural choice for modeling dependence among warranty claims. The mining of association rules is covered in Chapter 10. Association rules are also a natural way to investigate dependence among warranty claims since the confidence measure represents a type of conditional probability, similar to Bayesian networks.

The details of the results are confidential, but we can get a general idea of the type of dependencies uncovered by the models. One insight the researchers uncovered was that a particular combination of construction specifications doubles the probability of encountering an automobile electrical cable problem. DaimlerChrysler engineers have begun to investigate how this combination of factors can result in an increase in cable problems.

The researchers investigated whether certain garages had more warranty claims of a certain type than did other garages. Their association rule results showed that, indeed, the confidence levels for the rule "If garage *X*, then cable problem," varied considerably from garage to garage. They state that further investigation is warranted to reveal the reasons for the disparity.

5. Evaluation Phase

The researchers were disappointed that the support for sequential-type association rules was relatively small, thus precluding generalization of the results, in their opinion. Overall, in fact, the researchers state: "In fact, we did not find any rule that our domain experts would judge as interesting, at least at first sight." According to this criterion, then, the models were found to be lacking in effectiveness and to fall short of the objectives set for them in the business understanding phase. To account for this, the researchers point to the "legacy" structure of the database, for which automobile parts were categorized by garages and factories for historic or technical reasons and not designed for data mining. They suggest adapting and redesigning the database to make it more amenable to knowledge discovery.

6. Deployment Phase

The researchers have identified the foregoing project as a pilot project, and as such, do not intend to deploy any large-scale models from this first iteration. After the pilot project, however, they have applied the lessons learned from this project, with the goal of integrating their methods with the existing information technology environment at DaimlerChrysler. To further support the original goal of lowering claims costs, they intend to develop an intranet offering mining capability of QUIS for all corporate employees.

Fallacies of Data Mining (1)

- Fallacy 1: There are data mining tools that automatically find the answers to our problem
  - Reality: There are no automatic tools that will solve your problems "while you wait"
- Fallacy 2: The DM process require little human intervention
  - Reality: The DM process require human intervention in all its phases, including updating and evaluating the model by human experts
- Fallacy 3: Data mining have a quick ROI
  - Reality: It depends on the startup costs, personnel costs, data source costs, and so on

Fallacies of Data Mining (2)

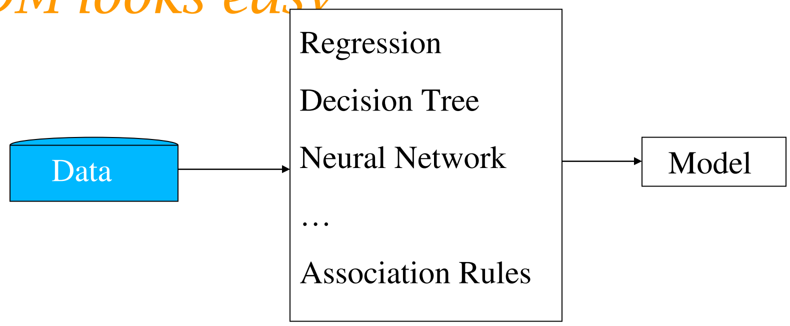
- Fallacy 4: DM tools are easy to use
  - Reality: Analysts must be familiar with the model
- Fallacy 5: DM will identify the causes to the business problem
  - Reality: DM tool only identify patterns in your data, analysts must identify the cause
- Fallacy 6: Data mining will clean up a data repository automatically
  - Reality: Sequence of transformation tasks must be defined by an analysts during early DM phases

* Fallacies described by Jen Que Louie, President of Nautilus Systems, Inc.

Remember

- Problems suitable for Data Mining:
  - Require to discover knowledge to make right decisions
  - Current solutions are not adequate
  - Expected high-payoff for the right decisions
  - Have accessible, sufficient, and relevant data
  - Have a changing environment
- IMPORTANT:
  - **ENSURE privacy if personal data is used!**
  - **Not every data mining application is successful!**

DM looks easy



Data Mining Method

- But it is not easy

- Real-world is complicate

Methods and Techniques

- Cluster Analysis (tasks: clustering)
- Association Rules (tasks: association)
- Decision trees (tasks: prediction, classification)
- Neural networks (tasks: prediction, classification)
- K-nearest neighbor (tasks: prediction, classification, clustering)
- Regression analysis (task: estimation, prediction)

CHAPTER 2:

Supervised Learning

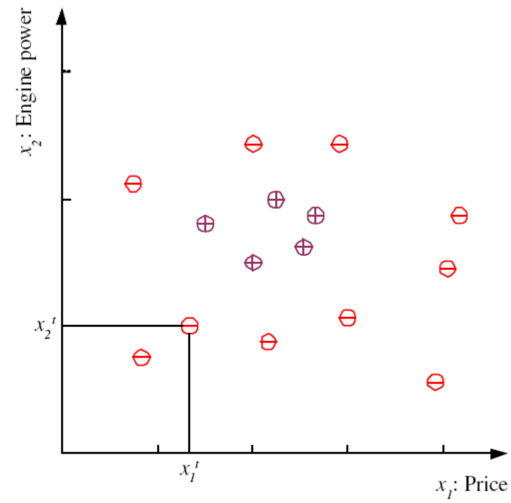
Learning a Class from Examples

- 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

Training set $\mathcal{X}$



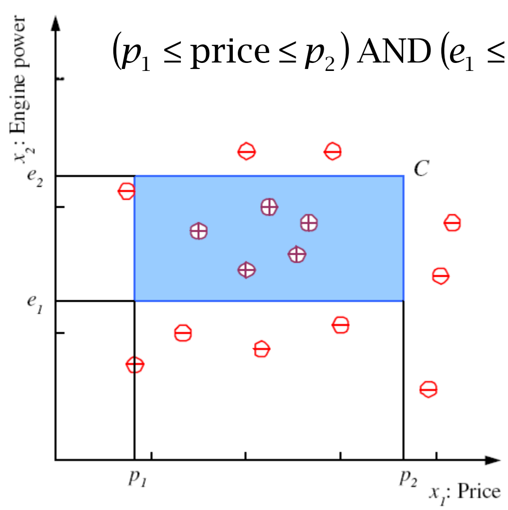
$$\mathcal{X} = \{\mathbf{x}^t, r^t\}_{t=1}^N$$

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

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

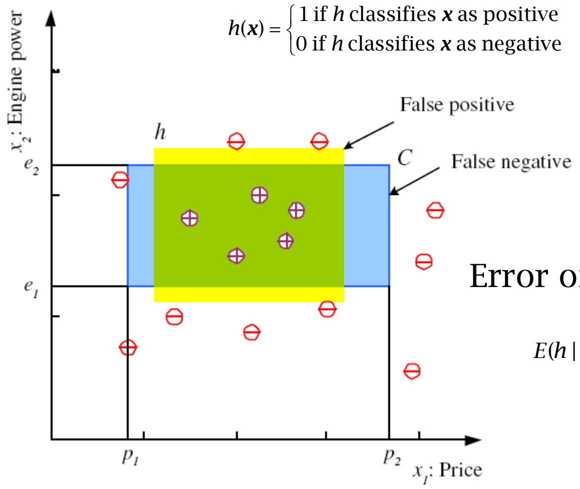
Class C

$$(p_1 \le \text{price} \le p_2) \text{ AND } (e_1 \le \text{engine power} \le e_2)$$



Hypothesis class $\mathcal{H}$

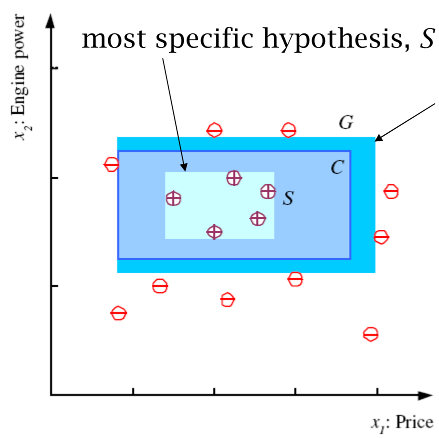
$$h(\mathbf{x}) = \begin{cases} 1 \text{ if } h \text{ classifies } \mathbf{x} \text{ as positive} \\ 0 \text{ if } h \text{ classifies } \mathbf{x} \text{ as negative} \end{cases}$$



Error of h on $\mathcal{H}$

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

S, G, and the Version Space



$h \in \mathcal{H}$, between S and G is consistent

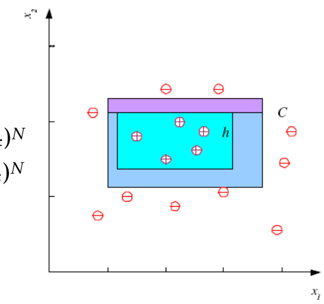
and make up the version space

(Mitchell, 1997)

Probably Approximately Correct (PAC) Learning

- How many training examples N should we have, such that with probability at least $1 - \delta$, h has error at most ε ?
(Blumer et al., 1989)

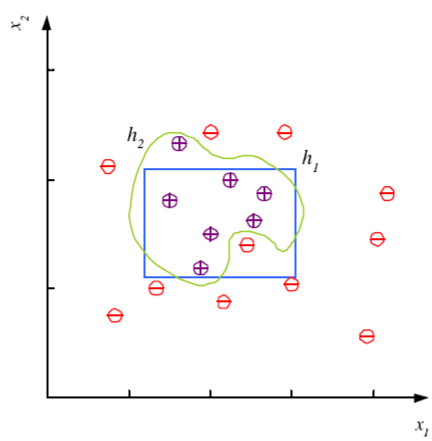
- Upper bound for each strip should be $\varepsilon/4$
➔ $4(\varepsilon/4) = \varepsilon$
- Pr that we miss one strip $1 - \varepsilon/4$
- Pr that N instances miss one strip $(1 - \varepsilon/4)^N$
- Pr that N instances miss 4 strips $4(1 - \varepsilon/4)^N$
- $4(1 - \varepsilon/4)^N \le \delta$ use $(1 - x) \le e^{-x} = \exp(-x)$
- $4\exp(-\varepsilon N/4) \le \delta$ and $N \ge (4/\varepsilon)\log(4/\delta)$



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)

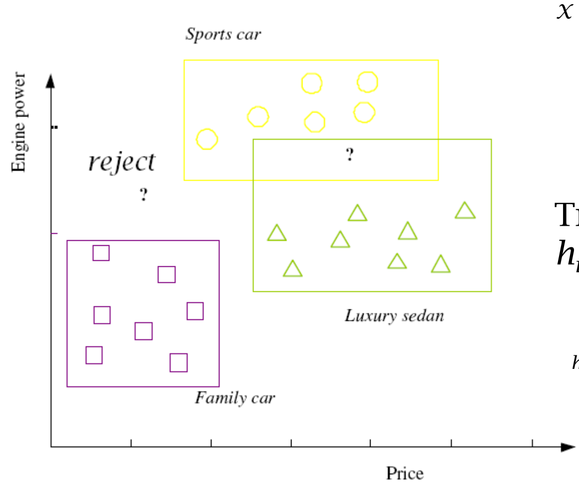


Lecture Notes for E Alpaydın 2004 Introduction to Machine Learning © The MIT Press (V1.1)

Multiple Classes, C_i $i=1,...,K$

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

$$r_i^t = \begin{cases} 1 & \text{if } \mathbf{x}^t \in C_i \\ 0 & \text{if } \mathbf{x}^t \in C_j, j \neq i \end{cases}$$



Train hypotheses $h_i(\mathbf{x})$, $i = 1,...,K$:

$$h_i(\mathbf{x}^t) = \begin{cases} 1 & \text{if } \mathbf{x}^t \in C_i \\ 0 & \text{if } \mathbf{x}^t \in C_j, j \neq i \end{cases}$$

Regression

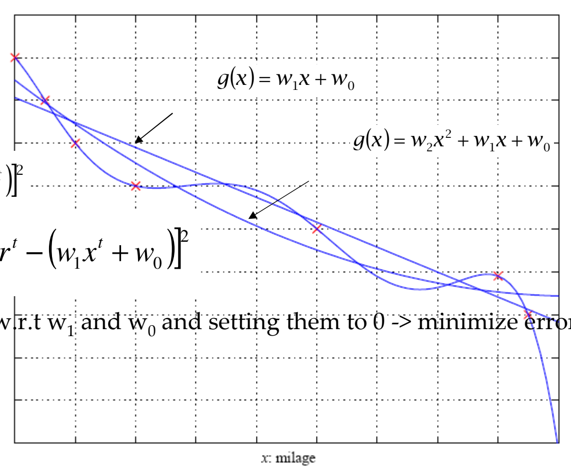
$$\mathcal{X} = \{x^t, r^t\}_{t=1}^N$$

$$r^t \in \Re$$

$$r^t = f(x^t)$$

$$E(g \mid \mathcal{X}) = \frac{1}{N}\sum_{t=1}^N \left[r^t - g(x^t)\right]^2$$

$$E(w_1, w_0 \mid \mathcal{X}) = \frac{1}{N}\sum_{t=1}^N \left[r^t - (w_1 x^t + w_0)\right]^2$$



Partial derivatives of E w.r.t w_1 and w_0 and setting them to 0 -> minimize error

$$w_1 = \frac{\sum_t x^t r^t - \overline{x}\overline{r}N}{\sum_t (x^t)^2 - N\overline{x}^2}$$

$$w_0 = \overline{r} - w_1 \overline{x}$$

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 or f
- Underfitting: $\mathcal{H}$ less complex than C or f

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 $E\uparrow$

Cross-Validation

- To estimate generalization error, we need data unseen during training. We split the data as
  - Training set (50%)
  - Validation set (25%)
  - Test (publication) set (25%)
- Resampling when there is few data

Date for Lab Class

- **Friday**, 8.30 - 9.30, Harburger Schloßstr., 215
- **Friday**, 9.45 - 10.30, SBS 95, H0.03
- **Tuesday**,8.30 - 9.30, Harburger Schloßstr., 215
- **Monday**,8.30 - 9.30, Harburger Schloßstr., 215