Course overview

Heikki Mannila
Laboratory for Computer and Information Science
Helsinki University of Technology
Heikki.Mannila@tkk.fi

T-61.5060 Algorithmic methods of data mining (5 cr) L P

- T-61.5060 Tiedon louhinnan algoritmiset menetelmät (5 op) L P
- Data mining, also called knowledge discovery in databases (KDD)
- In Finnish: tiedon louhinta, tietämyksen muodostaminen
- Goal of the course: an overview of some of the basic algorithmic ideas in data mining
- Biased overview
- Theory and examples
- Course home page: http://www.cis.hut.fi/Opinnot/T-61.5060/
- Email: t615060@james.hut.fi

Contents of the course: current plan

- Introduction: what is knowledge discovery, types of tasks, etc.
- Counting and approximate counting
- Discovery of frequent patterns: association rules, frequent episodes
- Basic ideas of clustering, selected algorithmic themes in cluster analysis
- Dimension reduction: random projections and other methods
- Link analysis: basic ideas
- Significance testing
- Possibly also some other themes

Course organization

- Lectures: Mondays 10–12, Heikki Mannila
- Exercises: Thursday 16-18, Niko Vuokko, starting September 13
- Language of instruction: English
- Exam schedule: see http://tieto.tkk.fi/Opinnot/2007-08_en_sorted.html
- (Current information: December 21, 2007)

Material

- copies of slides available on the web during the course
- H. Mannila, H. Toivonen: Knowledge discovery in databases: the search for frequent patterns; available from the web page of the course. This give a detailed introduction to the search of frequent patterns.
- Background material: D. Hand, H. Mannila, P.Smyth: Principles of Data Mining, MIT Press 2001.
- Recent conference and journal papers in the area (details given later).

Prerequisites

- Basic algorithms: sorting, hashing, set manipulation
- Analysis of algorithms: O-notation and its variants, perhaps some recursion equations
- Programming: some programming language; ability to do small experiments reasonably quickly
- Probability: concepts of probability and conditional probability, binomial distribution and other simple distributions
- T-106.1220 - Tietorakenteet ja algoritmit T (5 op)
- A nice textbook covering these things — and many more: Kleinberg & Tardos: Algorithm Design.

Requirements for passing the course

- Exam: see
  http://tieto.tkk.fi/Opinnot/2007-08_en_sorted.html
- (Current information: December 21, 2007)
- A small practical assignment or essay

Data mining activities at TKK/CIS

- basic research: algorithms, theory
- applied research: genetics, ecology, ubiquitous computing, documents, natural language, ...
- FDK "From Data to Knowledge": Academy of Finland Center of Excellence 2002–2007; Algodan: Algorithmic Data analysis: Academy of Finland Center of Excellence 2008-2013
- HIIT: Helsinki Institute for Information Technology
- Adaptive Informatics Research Centre: Academy of Finland Center of Excellence

Chapter 1: Introduction

Chapter 1. Introduction

- Introduction
- What is data mining?
- Some examples
- Data mining vs. statistics and machine learning

What is data mining?

Goal: obtain useful knowledge
from large masses of data.

Data mining is the analysis of (often large) observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data analyst

- "Tell something interesting about this data."
- "Describe this data."
- exploratory data analysis on large data sets

Examples of discovered knowledge

- Frequent patterns: "there are lots of documents that contain the phrases 'association rules', 'data mining', and 'efficient algorithm'
- association rules:
  "80 % of customers who buy beer and sausage buy also mustard"
- rules: "if Age < 40 then Income < 10"
- functional dependencies:
  $A \to B$, i.e., "if $t[A] = u[A]$, then $t[B] = u[B]$"
- models: $Y = aX + b$
- clusterings: the documents in this collection form three different groups

Example: sales data

- 0/1 matrix D
- rows: customer transactions
- columns: products
- $t(A) = 1$ if and only if customer t contained product A
- thousands of columns, millions of rows
- easy to collect
- find something interesting from this?

Association rules

- mustard, sausage, beer $\Rightarrow$ chips
- conditional probability (*accuracy*) of the rule: 0.8
- *frequency* of the rule: 0.2
- arbitrary number of conjuncts on the left-hand side

Find all association rules with frequency
at least σ

Example: student/course data

- matrix D with $D(t, c) = 1$ iff student t has taken course c

```
1 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 1 0 0 0 1 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
0 1 0 0 1 1 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1
1 1 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```


Example: student/course data

- example rule:
  `if {Data Communications, Unix, Networks}`
  `then (graduation) (0.3)`

Example: words in documents

- Given a collection of documents
- Find out which words define topics, i.e., some sort of clusters of words that tend to occur together
- Topic 1: "mining", "data", "association", "rule", etc.
- Topic 2: "lecture", "exercise", "exam","material","assignment", etc.
- Some documents talk about topic 1, some about topic 2, some about both
- How to find the topics from data?

Example: SKICAT sky survey

approximately 1000 photographs of 13000x13000 pixels

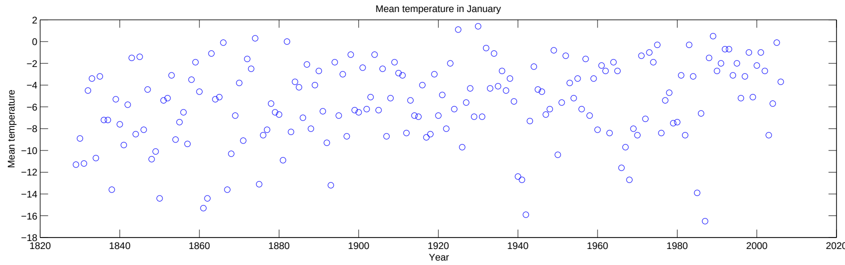
$\Rightarrow 3 \cdot 10^9$ smudges of light, each with 40 attributes

- task 1: classify the objects to
  - stars
  - galaxies
  - others
- task 2: find some interesting clusters of objects
  - quasars with high redshift
  - ...

machine learning? pattern recognition?

Example: climate data

- Long time series on temperature



Example: protein-protein interactions

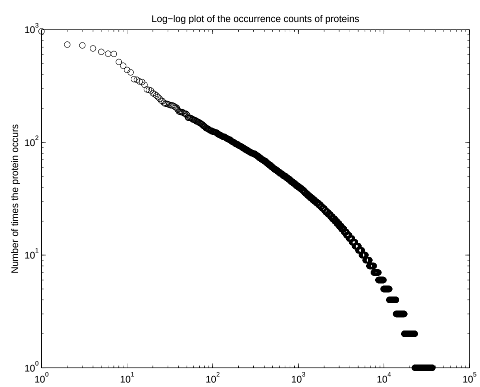
- Which proteins have something to do with each other?
- A small dataset http://www.ebi.ac.uk/intact/site/index.jsf
- Which protein interacts with what, how the interaction was verified, how certain it is, etc.
- 127452 interactions among 36760 proteins
- What is the structure of this network?

Fields and an example record in the data

ID interactor A, ID interactor B, Alt. ID interactor A, Alt. ID interactor B, Alias(es) interactor A, Alias(es) interactor B, Interaction detection method(s), Publication 1st author(s), Publication Identifier(s), Taxid interactor A, Taxid interactor B, Interaction type(s), Source database(s), Interaction identifier(s), Confidence value(s)

uniprotkb:Q9P2S5, genbank-protein-gi:4501917, uniprotkb:WDR8(gene name), - , - , - , MI:0006(anti bait coip), - , pubmed:17353931, taxid:9606(human) , taxid:9606(human) , MI:0218(physical interaction) , MI:0469(intact), intact:EBI-1062786—intact:EBI-1062786, -

Number of times each protein occurs in the data



Why data mining?

- raw data is easy to collect, expensive to analyze
- more data than can be analyzed using traditional methods
- successful applications
- methods: algorithms, machine learning, statistics, databases
- a tool for exploratory data analysis
- can and has to be used in combination with traditional methods
- data analysis is one of the strongest research themes in computer science

The KDD process

- understanding the domain,
- preparing the data set,
- discovering patterns, clusters, models, etc.
- postprocessing of discovered regularities, and
- putting the results into use

Where is the effort?

iteration, interaction

Phases of finding patterns, models, clusters, etc.

- What is the task? What do we want to find out?
- What is the model, or pattern class?
- What is the score function, i.e., how do we know which model fits the data well?
- What is the algorithm? How do we find the model?

Data mining tasks

- Pattern discovery: find interesting patterns that occur frequently in the data
- Prediction: predict the value of a variable in the data
- Clustering: group the observations (or variables) into groups of similar objects
- Outlier detection: find observations that are unusual
- ...

Example: words in documents

- 0-1 matrix representation of documents
- Rows: document
- Columns: words
- Entry M_{ij} has a 1 if document i contains word j, 0 otherwise
- "Bag-of-words" representation
- Loses a lot of information, but a lot is retained

```
1 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 1 0 0 0 1 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
0 1 0 0 1 1 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1
1 1 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```


Data mining tasks in document data

- Find collections of word that occur frequently together in the same documents
- Find topics from data
- Cluster similar documents together
- Will term x occur in the document
- Which are unusual documents?

Another view of data mining tasks

- Exploratory data analysis
- Descriptive modeling
- Predictive modeling
- Discovering patterns and rules
- Retrieval by content

Data mining and related areas

- How does data mining relate to statistics
- How does data mining relate to machine learning?
- Other related areas?

Data mining vs. machine learning

- machine learning methods are used for data mining
  - classification, clustering, ...
- amount of data makes a difference
  - accessing examples can be a problem
- data mining has more modest goals:
  - automating tedious discovery tasks, not aiming at human performance in real discovery
  - helping user, not replacing them
- this course: data mining $\setminus$ machine learning

Data mining vs. statistics

"tell me something interesting about this data" – what else is this than statistics?

- the goal is similar
- different types of methods
- in data mining one investigates a lot of possible hypotheses
- amount of data
- exploratory data analysis

Data mining and databases

- ordinary database usage: deductive
- knowledge discovery: "inductive"
- new requirements for database management systems
- novel data structures, algorithms, and architectures are needed
- SQL queries; OLAP queries; data mining

Data mining and algorithms

- Lots of nice connections
- A wealth of interesting research questions
- Some will be treated later in the course

Why is data mining an interesting research area?

- practically relevant
- easy theoretical questions
- the whole spectrum: from theoretical issues to systems issues to concrete data cleaning to novel discoveries
- easy to cooperate with other areas and with industry

2. Counting and approximate counting

The simplest of data analysis

- Given a stream or set of numbers (identifiers, etc.)
- How many numbers are there?
- How many distinct numbers are there?
- What are the most frequent numbers?
- How many numbers appear at least K times?
- How many numbers appear only once?
- Etc.

Counting numbers and unique numbers

- Given a file data.txt, one identifier per line
- How to count how many identifiers there are? How many unique identifiers?
- Quick solutions using Unix commands:
  `wc -l data.txt`
- Unique numbers:
  `sort < data.txt | uniq -c | wc -l`
- What are the running times of these?
- Linear time and $O(n \log n)$ (sorting)
- (Can you observe the $\log n$ factor in real data?)

Experimental results

- Data generation
  gawk -vn=1000000 'BEGIN {for (i=1;i<=n;i++) {print int(n*rand())}}' > data1000000
- Counting the numbers in the file (on a laptop)

n	time	
10^5	0.007s	
10^6	0.051s	
10^7	0.397s	
10^8	58.772s	(real time; user time 6.744)

- Counting unique numbers: sort < data100000 | uniq -c | wc -l

n	result	time	
10^5	63220	0.245s	
10^6	631706	3.147s	
10^7	6320776	40.456s	
10^8	63206198	723.120s	(real; user time 461.193s)

Thus 36780 numbers did not occur at all in the first experiment.

Remarks on the results

- The count of unique numbers seems to be about 0.63 of the size of the data
- The running time increases nicely at first, and then suddenly a lot
- What are the reasons for these phenomena?

The count of distinct identifiers

- Given n random integers between 1 and n
- How many distinct number will there be, on the average?
- What is the probability that a specific number k occurs at least once in the data?
- Probability that k does not occur:

$$(1-1/n)^n \to \frac{1}{e} \approx 0.367879$$

Probability that k does occur: $1-1/e \approx 0.632121$

The coupon collector's problem

- There are m different items
- Sample one at a time at random
- How many samples does one have to do in order to see every item at least once?
- Do an experimental study

How many numbers occur i times?

i	items	
8	1	
7	4	
6	50	
5	283	What would be an explanation?
4	1573	
3	6205	
2	18238	
1	36866	
0	36780	

Explanation

- Probability that an item occurs exactly k times:

$$Pr(n,k) = \binom{n}{k} p^k (1-p)^{n-k)},$$

where $p = 1/n$ is the probability that it occurs in a single row.

- Poisson approximation says that this is about

$$\frac{e^{-np}(np)^k}{k!}$$

- Comparison

i	Poisson	data
8	0.9	1
7	7.3	4
6	51	50
5	306	283
4	1533	1573
3	6131	6205
2	18394	18238
1	36788	36866
0	36788	36780

The running times

- Is there a good match with the predicted $n \log n$ behavior?
- Why does the running time of counting lines suddenly jump up?
- Main memory vs. disk
- Disk I/O is very slow compared to operations on main memory
- Main memory operations are very slow compared to cache operations
- Is there a good match with the predicted $n \log n$ behavior?

Finding the frequency of each item

- Sorting gives us this information
- Only after we have seen all of the data
- Using hashing is another way: implement associative arrays
- Identifier ℓ; write something of the form `increment`$[\ell, 1]$
- Works directly in `awk` and perl; many languages have suitable packages
- In awk: {c[$1]++} END {for (x in c) {print x, c[x]}}
- Prints the first identifiers in each row in the data and their frequencies

Associative arrays

- An array that can be indexed by using an arbitrary object (e.g., a string)
- Built using hashing functions
- Cost of accessing the array element for identifier $[\ell]$ is constant (more or less $\cdots$)
- A large literature on this

Data streams

- What if the data is so large that we cannot keep an array element for each identifier occurring in the data?
- This seldom is the case
- The data streams model
- IP routers, other telecom applications, etc.
- Lots of very nice theoretical and practical results

Finding the dominant element

- A neat problem, but not very important
- A stream of identifiers; one of them occurs more than 50% of the time
- How to find it using no more that a few memory locations?
- A = the first item; count=1
- for each subsequent item B

```
    if A==B then count++ else {
        count - -
        if count==0 {A=B; count=1}
```

- Does this work correctly?

Finding the missing item

- Another simple – and not very general – trick
- The input file has $N-1$ distinct numbers from 1 to N
- How to find out which one does not occur? Using $O(1)$ storage?
- What if there are two missing numbers? K missing ones?

Finding a number in the top half

- Given a set of N numbers on disk; N is large
- Find a number x such that x is likely to be larger than the median of the numbers
- Using just a small number of operations
- Solution: randomly sample K numbers from the file
- Output their maximum
- Probability of failure is $(1/2)^K$

Approximate counting

- What if it is not necessary to find the exact frequencies of the items?
- Approximations are sufficient
- How to get approximate counts of the frequencies of items in the data file?
- Take a sample
- How? How large?

Basic techniques of sampling

- Sampling from a file
- Given a file of N records $t_1, \ldots, tn_n$, we wish to choose K from them
- With replacement or without replacement
- With replacement:
  - for $i = 1$ to K do:
    - * generate a random integer b between 1 and N
    - * output record t_b
  - Or sort the generated random integers into order and read the data once

Sampling without replacement, basic method

- Keep a bit vector of N bits
- Generate random integers b between 1 and N and mark bit b, if it is not already marked
- Until K bits have been marked
- Read through the bit vector and the data file, and output the selected records

Sampling without replacement, sequential method

$T := K$; (how many items are needed)
$M := N$; (how many elements still to be seen)
$i := 1$;
while $T > 0$ **do**
  let b be a random number from $[0, 1]$;
  if $b < T/M$ **then**
    output record t_i;
    $T := T - 1$;
    $M := M - 1$;
  else
    $M := M - 1$;
  end;
  $i := i + 1$; **end**;

Correctness

- By induction on N; for $N = 0$ and $N = 1$, the correctness is clear
- Assume the algorithm works for $N = N'$; we show that it works for $N = N' + 1$
- The first element of the file will be selected with probability K/N, as required
- What about the next elements? two cases: the first element was selected or it wasn't
- Probability that an element will be selected is

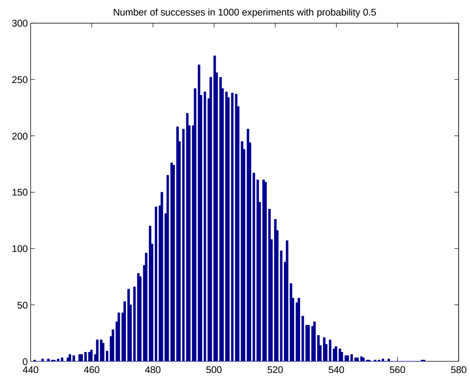
$$\frac{K}{N}\frac{K-1}{N-1} + \frac{N-K}{N}\frac{K}{N-1} = \frac{K}{N}$$

What is a good sample size

- Depends on what we want to do
- Assume we want to estimate the fraction of rows t in the data that satisfy a condition $Q(t)$
- n rows in total, pn rows t in the data with $Q(t)$ true
- How large an error will we make in estimating p by taking a sample of size s?
- Approaches: normal approximation, Poisson approximation, Chernoff bounds, brute force

Brute force estimation of the error

- Generate (say) 10000 datasets
- Each dataset has s elements; each element is independely 0 or 1, $Pr(1) = p$
- Count the number of datasets in which the frequency of 1s is larger than a given deviation from the mean



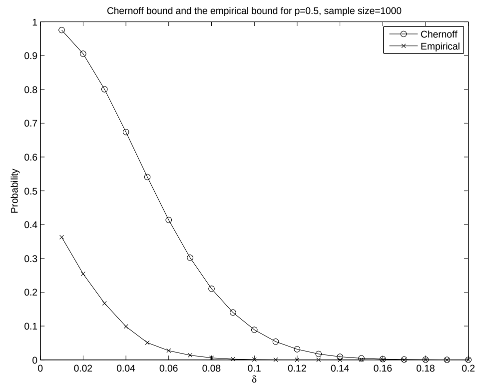
Chernoff bounds

- Normal approximation and Poisson approximation have some assumptions
- Brute force does not work for very small p
- Chernoff bounds: useful when p is small, n is huge, and np is large
- Error of at least $\delta > 0$ in the sample frequency

$$Pr(X > (1+\delta)sp) < \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{sp}$$

- For $0 < \delta < 1$

$$Pr(X > (1+\delta)sp) < \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{sp} < e^{-sp\delta^2/3}.$$


Chernoff bound and the empirical bound for p=0.5, sample size=1000
Probability
δ
Chernoff
Empirical
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.02
0.04
0.06
0.08
0.12
0.14
0.16
0.18
0.2