Algorithmics for Data Mining

Master in Innovation and Research in Informatics
FIB, UPC

Department of Computer Science

Spring 2020

0. Course Presentation

Personnel

José Luis Balcázar

- jose.luis.balcazar@upc.edu
- Omega 255 (2nd floor), 93 413 7847

Additionally, we plan for Prof. Josep Carmona to cover Business Process Mining.

Logistics

Schedule in the Racó with the initial plans.

- Quite low registration this year (about half the usual).
- The two half groups seem overkill, but the timing is not compatible.
- Could we reach an agreement on a more sensible decision?

Logistics

Schedule in the Racó with the initial plans.

- Quite low registration this year (about half the usual).
- The two half groups seem overkill, but the timing is not compatible.
- Could we reach an agreement on a more sensible decision?

Additional personal conversations as needed:

- Usually available after each of our sessions;
- recommended (but not enforced) to warn me in advance by email;
- many alternative slots for appointments, again by email.

Written Support

Link to the evolving slides:
`www.cs.upc.edu/~balqui/slidesADM2020.pdf`

Link will be made available also from the Racó.

Several books available in the Main Library BRGF

(please take initiative, look for them, browse through them...)

and also freely online (like `this one`, or also `that one`...).

Mainly, individually agreed research papers for state-or-the-art advances on each topic.

Evaluation, I

Papers to turn in, one optional oral presentation

I want four intermediate grades from each student in order to decide the final grade.

- Four written assignments ("your papers"), or

Evaluation, I

Papers to turn in, one optional oral presentation

I want four intermediate grades from each student in order to decide the final grade.

- Four written assignments ("your papers"), or
- Three written assignments ("your papers") and one oral presentation of one of them.

Evaluation, I

Papers to turn in, one optional oral presentation

I want four intermediate grades from each student in order to decide the final grade.

- Four written assignments ("your papers"), or
- Three written assignments ("your papers") and one oral presentation of one of them.
  - Which option?

Evaluation, I

Papers to turn in, one optional oral presentation

I want four intermediate grades from each student in order to decide the final grade.

- Four written assignments ("your papers"), or
- Three written assignments ("your papers") and one oral presentation of one of them.
  - Which option?
  - Which topics?

Evaluation, I

Papers to turn in, one optional oral presentation

I want four intermediate grades from each student in order to decide the final grade.

- Four written assignments ("your papers"), or
- Three written assignments ("your papers") and one oral presentation of one of them.
  - Which option?
  - Which topics?
  - You negotiate all this with me!

Evaluation, I

Papers to turn in, one optional oral presentation

I want four intermediate grades from each student in order to decide the final grade.

- Four written assignments ("your papers"), or
- Three written assignments ("your papers") and one oral presentation of one of them.
  - Which option?
  - Which topics?
  - You negotiate all this with me!
- There will be no formal exam.

Evaluation, I

Papers to turn in, one optional oral presentation

I want four intermediate grades from each student in order to decide the final grade.

- Four written assignments ("your papers"), or
- Three written assignments ("your papers") and one oral presentation of one of them.
  - Which option?
  - Which topics?
  - You negotiate all this with me!
- There will be no formal exam.
- Papers to be uploaded through the Racó.

Evaluation, I

Papers to turn in, one optional oral presentation

I want four intermediate grades from each student in order to decide the final grade.

- Four written assignments ("your papers"), or
- Three written assignments ("your papers") and one oral presentation of one of them.
  - Which option?
  - Which topics?
  - You negotiate all this with me!
- There will be no formal exam.
- Papers to be uploaded through the Racó.
  - The first one, around one month from now

Evaluation, I

Papers to turn in, one optional oral presentation

I want four intermediate grades from each student in order to decide the final grade.

- Four written assignments ("your papers"), or
- Three written assignments ("your papers") and one oral presentation of one of them.
  - Which option?
  - Which topics?
  - You negotiate all this with me!
- There will be no formal exam.
- Papers to be uploaded through the Racó.
  - The first one, around one month from now (probably, march 11);

Evaluation, I

Papers to turn in, one optional oral presentation

I want four intermediate grades from each student in order to decide the final grade.

- Four written assignments ("your papers"), or
- Three written assignments ("your papers") and one oral presentation of one of them.
  - Which option?
  - Which topics?
  - You negotiate all this with me!
- There will be no formal exam.
- Papers to be uploaded through the Racó.
  - The first one, around one month from now (probably, march 11);
  - the second one, just before the Easter break;

Evaluation, I

Papers to turn in, one optional oral presentation

I want four intermediate grades from each student in order to decide the final grade.

- Four written assignments ("your papers"), or
- Three written assignments ("your papers") and one oral presentation of one of them.
  - Which option?
  - Which topics?
  - You negotiate all this with me!
- There will be no formal exam.
- Papers to be uploaded through the Racó.
  - The first one, around one month from now (probably, march 11);
  - the second one, just before the Easter break;
  - the deadlines for third and fourth will depend on whether you give a presentation.

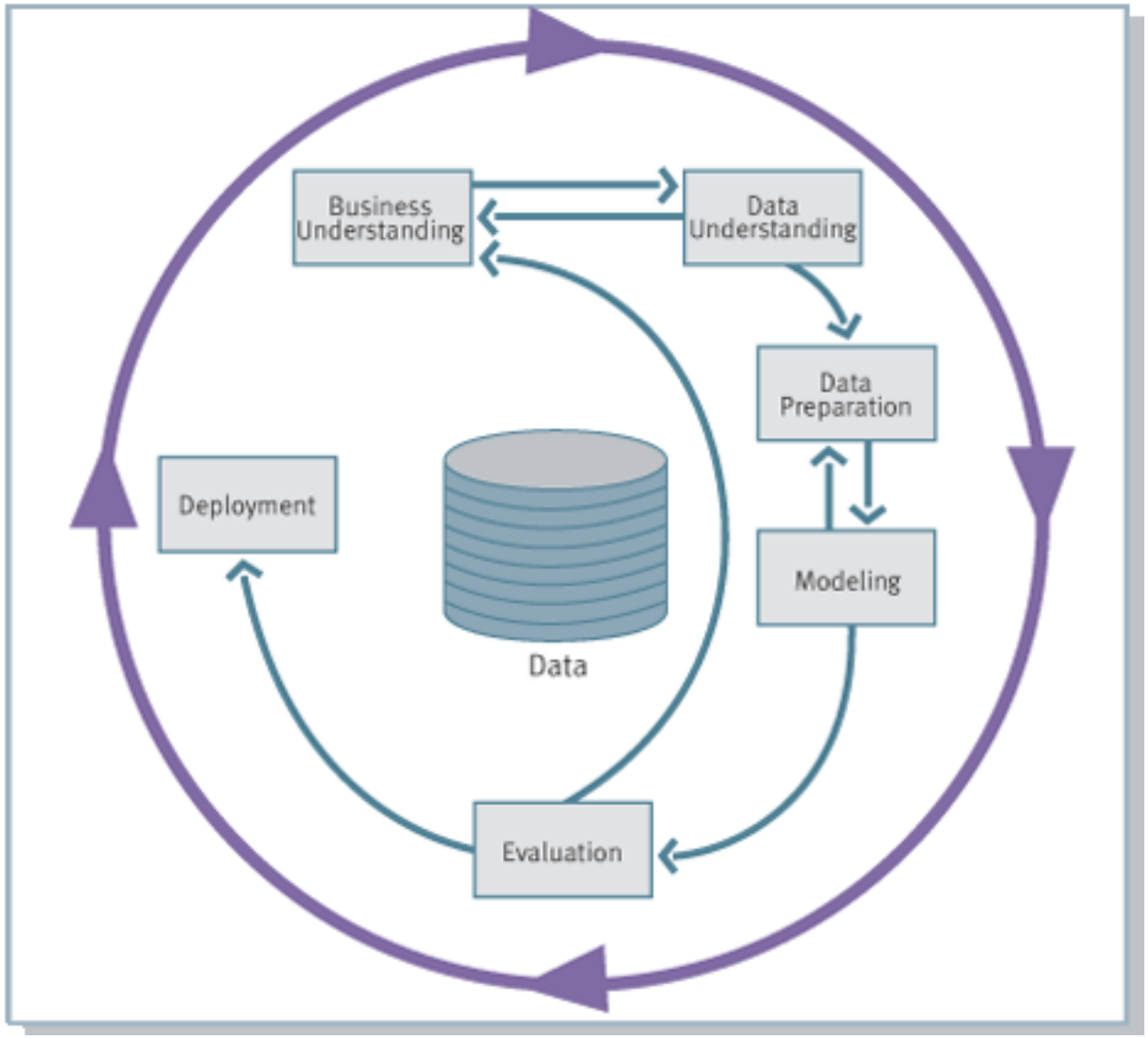
Evaluation, II

Expected characteristics

- "Your papers" must have a substantial content related to the topic of the course.
- Teamwork allowed, but:
  - Not for a paper that acts as basis of an oral presentation, and
  - your sets of coworkers on different papers must be disjoint.
- At least one of them (recommended: the first one) is to be on usage of a Data Mining tool for some Data Mining task.
- Under the previous conditions, the more your papers resemble original research papers, the better.
- Ask me if in need of clarification or if you want to propose some justified variant (I am likely to accept it).

CRISP-DM

Industry-designed diagram (1996)



Course Contents

Difficulty

Some of you may be attending, or have already listened to, courses similar to this one.

- We all must accept that there will be duplicities.
- Want most of these to still turn out to be useful!
  - By refreshing known but forgotten content,
  - By expanding the understanding,
  - By deepening the understanding.

Approximate topic guidance

- Book: The "Top Ten" Algorithms in Data Mining, `http://crcpress.com/product/isbn/9781420089646`,
- Preceding survey paper with same title, `http://link.springer.com/article/10.1007/s10115-007-0114-2`,
- plus a few variations and deeper considerations.

Taxonomy of Modeling Tools in Data Mining

Careful: not universal

- Predictive Models (always "supervised"):
  - Classification (Discrimination): non-numeric, unstructured prediction space
  - Categorization and Multiclassification: non-numeric, structured prediction space
  - Ranking: non-numeric prediction on a total ordering
  - Regression (Interpolation): numeric prediction space
    - Linear,
    - Polynomial,
    - ...
- Descriptive Models (possibly "unsupervised"):
  - Humanly interpretable predictors,
  - Clustering,
  - Pattern mining:
    - Frequent sets, frequent closures,
    - Association rule mining,
    - Pattern set mining...

Relational Data

Most common for starters

Relational data:

- Structured in tuples of attribute/value pairs.
- Akin to a SQL table.
- Often reformulated as a cloud of points in R^n.
- To predict: the value of one chosen "class" attribute.

Toy Relational Data

A simple and somewhat famous example that probably you have seen before

outlook	temperature	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

(Source today: Witten & Frank "Data Mining".)

Transactional Data, I

Alternative context, usual for pattern mining

Each observation is seen as a data structure on itself.

On the basis of a set of atomic items:

- Simplest (and most common) case: each observation is a set. (Analogy: documents as sets of terms.)
- Slight sophistication: multiplicity is relevant (but is likely to need adjustements; analogy: tfidf-like weights. . .).
- Further sophistications!

We will return to transactional data every now and then; but, for the time being, we work mostly with relational data.

Missing Topics

(Some of) The most important notions we are not discussing

- Time Series (very important in practice);
- Visual Analytics;
- OLAP;
- Data Streams;
- Neural Models (hint at connection at the approprate time);
- ...

Practical Data Analysis, I

Tools: Programming, GUIs, and workflows

Approaches

- Programming or CLI's: mostly "verbal", visualization basically reduced to graphics of the results of analysis;

Practical Data Analysis, I

Tools: Programming, GUIs, and workflows

Approaches

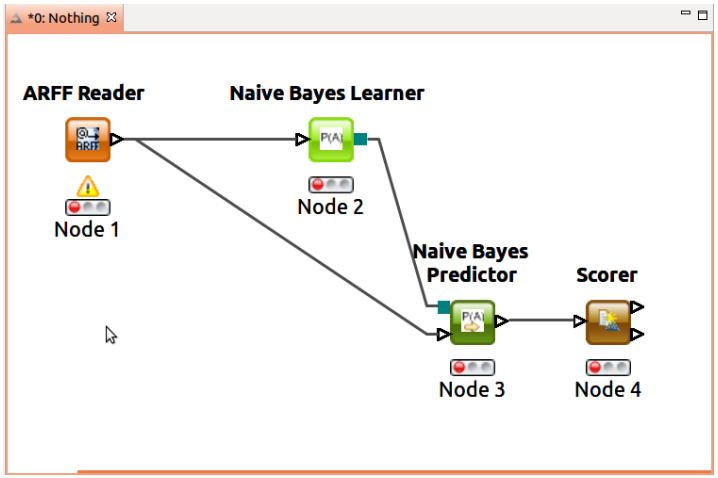
- Programming or CLI's: mostly "verbal", visualization basically reduced to graphics of the results of analysis;
- Relatively static, traditional GUIs (declining): buttons to load data and run algorithms, configuration tabs...

Practical Data Analysis, I

Tools: Programming, GUIs, and workflows

Approaches

- Programming or CLI's: mostly "verbal", visualization basically reduced to graphics of the results of analysis;
- Relatively static, traditional GUIs (declining): buttons to load data and run algorithms, configuration tabs...
- Workflows: very much visual; everything (or almost) is handled graphically: movable icons, contextual menus to configure... — may be successful with managers.



Practical Data Analysis, II

Tools: Specific proposals

Who's who

`Recent poll` from `http://www.kdnuggets.com` (or navigate `http://www.kdnuggets.com` → `Polls`...)

- Tools with a different originary purpose:
  - Python, R, EXCEL, SQL...
- More or less traditional GUI:
  - Weka Explorer, FRIDA...
- Workflow-based:
  - KNIME, RapidMiner, Weka Knowledge Flows, Orange...
  - Cloud-supported `clowdflows`, not very mature yet but you are welcome to give it a try.
- Omitted from this course: Visual Analytics tools (Tableau, Spotfire, Qlik...)

Practical Data Analysis, III

About most datasets

To keep in mind:

Blindly feeding the data into your data analysis tools is unlikely to work well!

A substantial amount of reading and thinking must be spent in preprocessing and transformation.

`https://www.kdnuggets.com/2015/05/data-science-inconvenient-truth.html`

Practical Data Analysis, IV

Where to explore for datasets

Main dataset sources:

- `mldata.org`,
- `https://www.kaggle.com/competitions`,
- the classical `archive.ics.uci.edu/ml/`:
  - Car evaluation (synthetic),
  - Mushroom (semi-synthetic),
  - Adult (a.k.a. "census income"),
  - Congressional Voting Records,
  - Contraceptive Method Choice,
  - Covertype,
  - (Statlog) German Credit Scoring,
  - (Statlog) Shuttle...

Additional data sources for the politically motivated:
`http://databank.worldbank.org`

(and plenty of others out there!)

1. Intro to KNIME

Lab Session 1, I

http://KNIME.org

Get KNIME working on your machine!

- On Linux, only installation necessary is uncompressing the tarball.
- Self-installer on Windows: run it, keep going...
- Folder for your workflows: maybe on cloud?

Lab Session 1, IV

KNIME Nodes

Learn to:

- read in data;
- transform data matrices:
  - handle sorting criteria for visualizing tables,
  - identify and change the types of columns,
  - perform other data manipulation operations:
    column/row filters, group-by, join, sampling...
  - handle collection columns;
- get a glimpse of the basic statistics of your data;
- visualize and plot data;
  - create interactive tables, hilite instances, and propagate the highlighter marks,
  - create and manipulate scatter plots,
  - handle colors, sizes, and shapes,
  - create histograms, line plots, box plots...

Count on a bit of help from the instructor when necessary.

2. Brief Probability Review

Probabilistic Tools

Recap

1. Probability space, events, random variables;

Probabilistic Tools

Recap

1. Probability space, events, random variables;
2. Empirical frequency as approximate probability,

Probabilistic Tools

Recap

1. Probability space, events, random variables;
2. Empirical frequency as approximate probability,
   - support, $supp(A)$: number of observations where A holds;

Probabilistic Tools

Recap

1. Probability space, events, random variables;
2. Empirical frequency as approximate probability,
   - support, $supp(A)$: number of observations where A holds;
   - probability $Pr(A)$, normalized support in $[0, 1]$: divide by total number of observations;

Probabilistic Tools

Recap

1. Probability space, events, random variables;
2. Empirical frequency as approximate probability,
   - support, $supp(A)$: number of observations where A holds;
   - probability $Pr(A)$, normalized support in $[0, 1]$: divide by total number of observations;
   - sometimes we normalize into $[0, 100]$ and indicate %.

Probabilistic Tools

Recap

1. Probability space, events, random variables;
2. Empirical frequency as approximate probability,
   - support, $supp(A)$: number of observations where A holds;
   - probability $Pr(A)$, normalized support in $[0, 1]$: divide by total number of observations;
   - sometimes we normalize into $[0, 100]$ and indicate %.
3. Conditional probability,

$$conf(A \to B) = Pr(B \mid A) = \frac{supp(AB)}{supp(A)};$$

Probabilistic Tools

Recap

1. Probability space, events, random variables;
2. Empirical frequency as approximate probability,
   - support, $supp(A)$: number of observations where A holds;
   - probability $Pr(A)$, normalized support in $[0, 1]$: divide by total number of observations;
   - sometimes we normalize into $[0, 100]$ and indicate %.
3. Conditional probability,
   - confidence: empirical approximation to the conditional probability in "implicational" form,

$$conf(A \to B) = Pr(B \mid A) = \frac{supp(AB)}{supp(A)};$$

Probabilistic Tools

Recap

1. Probability space, events, random variables;
2. Empirical frequency as approximate probability,
   - support, $supp(A)$: number of observations where A holds;
   - probability $Pr(A)$, normalized support in $[0, 1]$: divide by total number of observations;
   - sometimes we normalize into $[0, 100]$ and indicate %.
3. Conditional probability,
   - confidence: empirical approximation to the conditional probability in "implicational" form,

$$conf(A \to B) = Pr(B \mid A) = \frac{supp(AB)}{supp(A)};$$

4. Independence:

Probabilistic Tools

Recap

1. Probability space, events, random variables;
2. Empirical frequency as approximate probability,
   - support, $supp(A)$: number of observations where A holds;
   - probability $Pr(A)$, normalized support in $[0, 1]$: divide by total number of observations;
   - sometimes we normalize into $[0, 100]$ and indicate %.
3. Conditional probability,
   - confidence: empirical approximation to the conditional probability in "implicational" form,

$$conf(A \to B) = Pr(B \mid A) = \frac{supp(AB)}{supp(A)};$$

4. Independence:

$$Pr(A \wedge B) = Pr(A) * Pr(B),$$

Probabilistic Tools

Recap

1. Probability space, events, random variables;
2. Empirical frequency as approximate probability,
   - support, $supp(A)$: number of observations where A holds;
   - probability $Pr(A)$, normalized support in $[0, 1]$: divide by total number of observations;
   - sometimes we normalize into $[0, 100]$ and indicate %.
3. Conditional probability,
   - confidence: empirical approximation to the conditional probability in "implicational" form,

$$conf(A \to B) = Pr(B \mid A) = \frac{supp(AB)}{supp(A)};$$

4. Independence:

$$Pr(A \wedge B) = Pr(A) * Pr(B),$$
$$Pr(A \mid B) = Pr(A),$$

Probabilistic Tools

Recap

1. Probability space, events, random variables;
2. Empirical frequency as approximate probability,
   - support, $supp(A)$: number of observations where A holds;
   - probability $Pr(A)$, normalized support in $[0, 1]$: divide by total number of observations;
   - sometimes we normalize into $[0, 100]$ and indicate %.
3. Conditional probability,
   - confidence: empirical approximation to the conditional probability in "implicational" form,

$$conf(A \to B) = Pr(B \mid A) = \frac{supp(AB)}{supp(A)};$$

4. Independence:

$$Pr(A \wedge B) = Pr(A) * Pr(B),$$
$$Pr(A \mid B) = Pr(A),$$
$$Pr(B \mid A) = Pr(B);$$

Probabilistic Tools

Recap

1. Probability space, events, random variables;
2. Empirical frequency as approximate probability,
   - support, $supp(A)$: number of observations where A holds;
   - probability $Pr(A)$, normalized support in $[0, 1]$: divide by total number of observations;
   - sometimes we normalize into $[0, 100]$ and indicate %.
3. Conditional probability,
   - confidence: empirical approximation to the conditional probability in "implicational" form,

$$conf(A \to B) = Pr(B \mid A) = \frac{supp(AB)}{supp(A)};$$

4. Independence:

$$Pr(A \wedge B) = Pr(A) * Pr(B),$$
$$Pr(A \mid B) = Pr(A),$$
$$Pr(B \mid A) = Pr(B);$$

5. Bayes Rule: $Pr(A|B) = Pr(B|A) * Pr(A)/Pr(B)$.

Numerical Spaces and Expectation

Main property: Linearity

If random outcomes allow for the operations of addition and of multiplication by a real number (for instance, real vectors), we can use probabilities to compute expectations, that is, weighted averages:

$$E[A] = \sum_x (x * Pr[A = x])$$

Numerical Spaces and Expectation

Main property: Linearity

If random outcomes allow for the operations of addition and of multiplication by a real number (for instance, real vectors), we can use probabilities to compute expectations, that is, weighted averages:

$$E[A] = \sum_x (x * Pr[A = x])$$

Properties:

- Linearity of expectation: $E[\sum_i \alpha_i * A_i] = \sum_i (\alpha_i * E[A_i])$.
- For independent events, commuting with product: $E[A * B] = E[A] * E[B]$ provided $Pr(A \wedge B) = Pr(A) * Pr(B)$.

Counterintuitive Facts About Probability, I

"Rosencrantz and Guildenstern are dead" (Link)

Some context: http://en.wikisource.org/wiki/The_Tragedy_of_Hamlet,_Prince_of_Denmark/Act_5

Counterintuitive Facts About Probability, I

"Rosencrantz and Guildenstern are dead" (Link)

Some context: http://en.wikisource.org/wiki/The_Tragedy_of_Hamlet,_Prince_of_Denmark/Act_5

Starting scene of the movie: ‘‘Heads’’.

Counterintuitive Facts About Probability, I

"Rosencrantz and Guildenstern are dead" (Link)

Some context: http://en.wikisource.org/wiki/The_Tragedy_of_Hamlet,_Prince_of_Denmark/Act_5

Starting scene of the movie: ‘‘Heads’’.

Recap:

- 79 times (92 in the theater play), a fair coin has been tossed along the way.
- All of them came up heads.
- Surely the probability of the next cointoss is higher for tails!

Counterintuitive Facts About Probability, I

"Rosencrantz and Guildenstern are dead" (Link)

Some context: http://en.wikisource.org/wiki/The_Tragedy_of_Hamlet,_Prince_of_Denmark/Act_5

Starting scene of the movie: ‘‘Heads’’.

Recap:

- 79 times (92 in the theater play), a fair coin has been tossed along the way.
- All of them came up heads.
- Surely the probability of the next cointoss is higher for tails!

  Actually, no.

  They are independent events!
- Related:
  - http://en.wikipedia.org/wiki/Ludic_fallacy.
  - "Bayesian" point of view: infer that the coins are not fair.

Counterintuitive Facts About Probability, II

Three doors in TV

Monty Hall paradox:

There are three doors. All participants know the rules:

- Behind one door there is a prize ("the car"). Behind the others, less desirable items ("big pumpkins", "goats").
- You choose one door.
- Monty Hall opens one door, different from the one you have chosen: the prize is not there.
- Then he asks you: do you want to switch?

Counterintuitive Facts About Probability, II

Three doors in TV

Monty Hall paradox:

There are three doors. All participants know the rules:

- Behind one door there is a prize ("the car"). Behind the others, less desirable items ("big pumpkins", "goats").
- You choose one door.
- Monty Hall opens one door, different from the one you have chosen: the prize is not there.
- Then he asks you: do you want to switch?

Is it better to switch? Is it better to stick?

Counterintuitive Facts About Probability, II

Three doors in TV

Monty Hall paradox:

There are three doors. All participants know the rules:

- Behind one door there is a prize ("the car"). Behind the others, less desirable items ("big pumpkins", "goats").
- You choose one door.
- Monty Hall opens one door, different from the one you have chosen: the prize is not there.
- Then he asks you: do you want to switch?

Is it better to switch? Is it better to stick?

The first correct answer right away is actually another question:

Counterintuitive Facts About Probability, II

Three doors in TV

Monty Hall paradox:

There are three doors. All participants know the rules:

- Behind one door there is a prize ("the car"). Behind the others, less desirable items ("big pumpkins", "goats").
- You choose one door.
- Monty Hall opens one door, different from the one you have chosen: the prize is not there.
- Then he asks you: do you want to switch?

Is it better to switch? Is it better to stick?

The first correct answer right away is actually another question: What do we mean by "better"?

But, for a sensible notion of "better", it is better to switch.

Counterintuitive Facts About Probability (III)

Expectation of linearity

Simpson's Paradox:

Somebody has performed a survey.

- All along North Alderonia, vegetarians are more common among blue-eyed people than among non-blue-eyed people.

Counterintuitive Facts About Probability (III)

Expectation of linearity

Simpson's Paradox:

Somebody has performed a survey.

- All along North Alderonia, vegetarians are more common among blue-eyed people than among non-blue-eyed people.
- Also all along South Alderonia, vegetarians are more common among blue-eyed people than among non-blue-eyed people, as well.

Counterintuitive Facts About Probability (III)

Expectation of linearity

Simpson's Paradox:

Somebody has performed a survey.

- All along North Alderonia, vegetarians are more common among blue-eyed people than among non-blue-eyed people.
- Also all along South Alderonia, vegetarians are more common among blue-eyed people than among non-blue-eyed people, as well.

We can infer that, all along both Alderonias, vegetarianism occurs more often among blue-eyed people than among the rest.

Counterintuitive Facts About Probability (III)

Expectation of linearity

Simpson's Paradox:

Somebody has performed a survey.

- All along North Alderonia, vegetarians are more common among blue-eyed people than among non-blue-eyed people.
- Also all along South Alderonia, vegetarians are more common among blue-eyed people than among non-blue-eyed people, as well.

We can infer that, all along both Alderonias, vegetarianism occurs more often among blue-eyed people than among the rest.

No! We cannot make that inference. It is possible that the comparison of the ratios gets reversed upon considering the whole population.

Counterintuitive Facts About Probability (IV)

Don't place too much confidence on confidence

Dataset CMC (Contraceptive Method Choice)

A "partial implication" of over 10% support and 90% confidence:

`near-low-wife-education` `no-contraception-method`

$\longrightarrow$

`good-media-exposure`

Seems like a reliable "partial implication".

Counterintuitive Facts About Probability (IV)

Don't place too much confidence on confidence

Dataset CMC (Contraceptive Method Choice)

A "partial implication" of over 10% support and 90% confidence:

`near-low-wife-education` `no-contraception-method`

$\longrightarrow$

`good-media-exposure`

Seems like a reliable "partial implication".

But the support of "`good-media-exposure`" is over 92%.

The "correlation" is actually negative!

3. Predictors and their Evaluation

Probabilistic Prediction

Probability-based predictive models

Probabilistic prediction

In a merely frequentist sense: counting;

- when is the prediction to be issued?
  - before seeing anything?

Probabilistic prediction

In a merely frequentist sense: counting;

- when is the prediction to be issued?
  - before seeing anything?
    "a priori" predictor: the most common value for the class (*ZeroR* predictor);

Probabilistic prediction

In a merely frequentist sense: counting;

- when is the prediction to be issued?
  - before seeing anything?
    "a priori" predictor: the most common value for the class
    (*ZeroR* predictor);
  - after seeing all values for all non-class attributes?
    "a posteriori" predictor: the most common value for the class, conditioned to the values seen
    (*MAP* predictor, for "maximum a posteriori").

$$\arg\max_C \{Pr(C|A_1 \ldots A_n)\}$$

MAP Prediction

Unfortunately infeasible

A small case:

Task of binary classification:

- Assume ten attributes with four values each;
- Then we need to store 2^{20} conditional probabilities;
- and we need to estimate 2^{20} conditional probabilities.

Rule of thumb:

Ten or more observations per parameter to estimate might be still far from sufficient, but are necessary anyway; with less, don't even dream.

Conditional Independence Assumption

One way out

Bayes rule

Applied to arg $\max_C \{Pr(C|A_1 \ldots A_n)\}$:

$$Pr(C|A_1 \ldots A_n) =$$
$$Pr(A_1 \ldots A_n|C) * Pr(C)/Pr(A_1 \ldots A_n)$$

We can forget about the divisor, as it is the same for all values of C and does not modify the max.

Now we assume independence conditioned to the class value:

$$Pr(A_1 \ldots A_n|C) * Pr(C) =$$
$$Pr(A_1|C) * \ldots * Pr(A_n|C) * Pr(C)$$

Naïve Bayes

Rather good for such a simple approach

Precompute $Pr(A_i|C)$ for each value of each attribute conditioned to the class value; do it through the empirical frequency.

Instead of predicting

$$\arg\max_C\{Pr(C|A_1 \ldots A_n)\},$$

we predict

$$\arg\max_C\{Pr(A_1|C) * \ldots * Pr(A_n|C) * Pr(C)\}$$

Variant: the "Laplace correction" makes up for cases that might be potentially missing; some tools (like Weka) apply it (without warning).

Autonomous Learning Topics, I

Proposals to explore on yourself

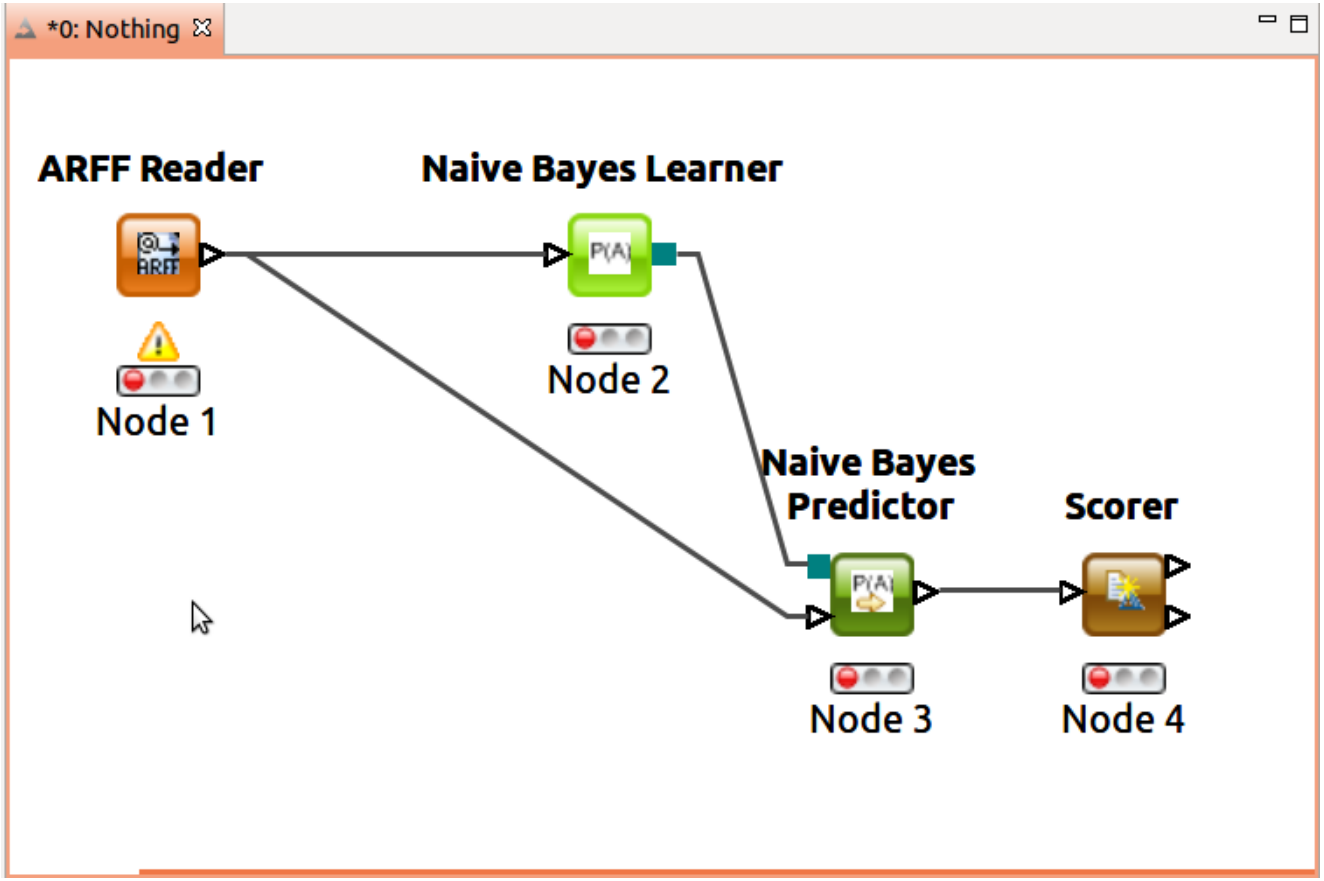
Some of these, if worked out in sufficient depth, may provide us with one of your four papers for the evaluation of the course.

1. Learn all details of the usage of Naïve Bayes predictors in various systems like R, KNIME, scikit-learn... (including the notion and usage of the Laplace correction).
2. Write your own implementation of MAP and Naïve Bayes in your favorite programming language (or, even better, in a programming language you don't master yet but want to practice further with).

How to Test a Predictor, I

On the original data?

Resubstitution error



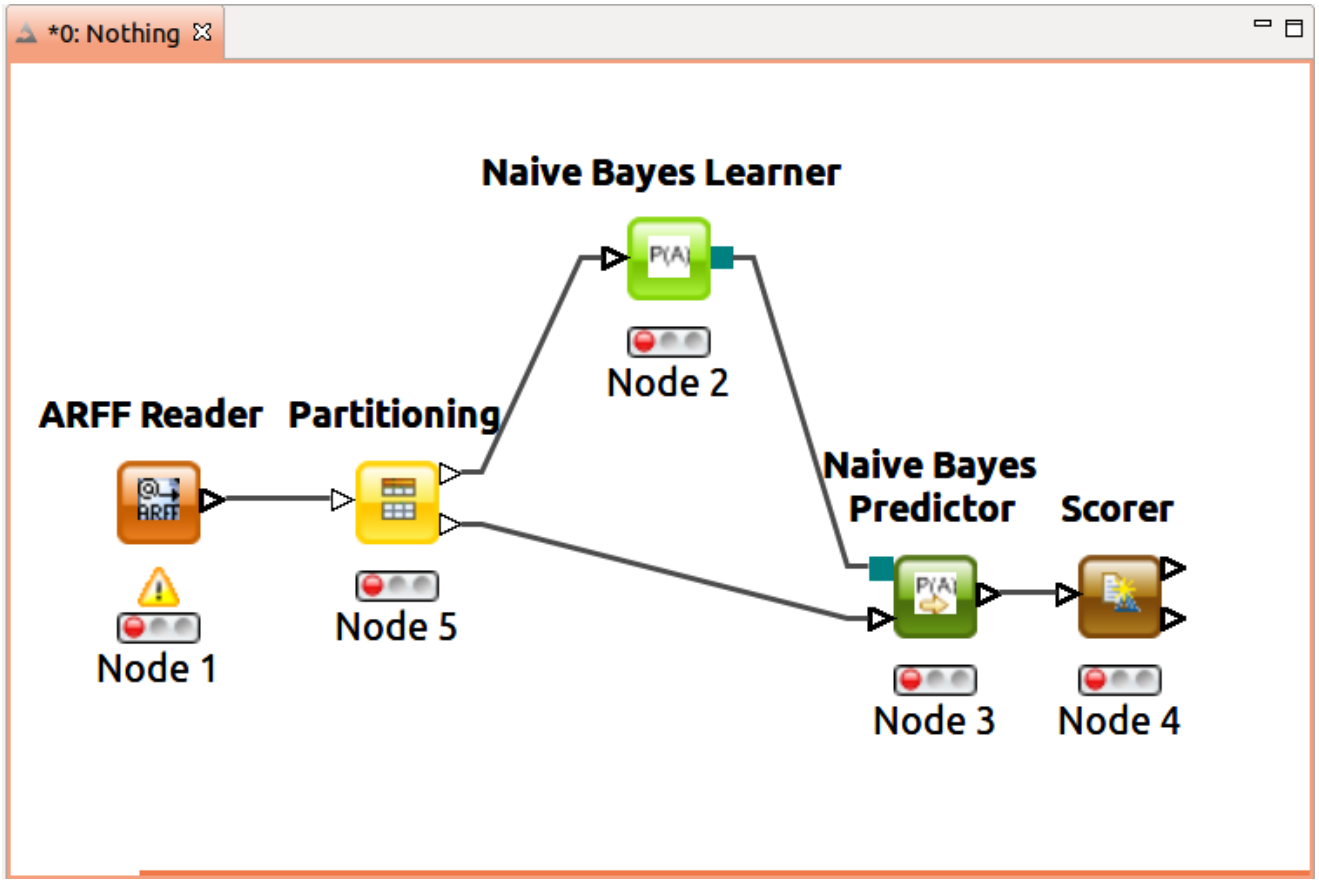
Far too optimistic!

How to Test a Predictor, II

On holdout data?

Test error

after training on a different subset.



How to Test a Predictor, III

Advantages and disadvantages

Resubstitution error

- Employs data to the maximum.

How to Test a Predictor, III

Advantages and disadvantages

Resubstitution error

- Employs data to the maximum.
- However, it cannot detect overfitting:
  - A predictor overfits when it adjusts very closely to peculiarities of the specific instances used for training.
  - Overfitting may hinder predictions on unseen instances.

How to Test a Predictor, III

Advantages and disadvantages

Resubstitution error

- Employs data to the maximum.
- However, it cannot detect overfitting:
  - A predictor overfits when it adjusts very closely to peculiarities of the specific instances used for training.
  - Overfitting may hinder predictions on unseen instances.

Holdout data

- Requires us to balance scarce instances into two tasks: training and test.
- Usual: train with 2/3 of the instances — but, which ones?
- It does not sound fully right that some available data instances are never seen for training.

How to Test a Predictor, III

Advantages and disadvantages

Resubstitution error

- Employs data to the maximum.
- However, it cannot detect overfitting:
  - A predictor overfits when it adjusts very closely to peculiarities of the specific instances used for training.
  - Overfitting may hinder predictions on unseen instances.

Holdout data

- Requires us to balance scarce instances into two tasks: training and test.
- Usual: train with 2/3 of the instances — but, which ones?
- It does not sound fully right that some available data instances are never seen for training.
- It sounds even worse that some are never used for testing.

How to Test a Predictor, IV

The idea of x-val

Cross-validation

The key intuitions to get the maximum information from scoring the predictor:

- To run a scorer on a maximum of data instances, we wish exactly one prediction per training instance.
- Let's make sure that each instance is used exactly once for testing.
- Let's run several prediction rounds: each instance will be used for testing in exactly one round.
- The instances used for testing in one round are used for training in all the other rounds.

Cross-validation, I

Basic description

Partition the available data

into N disjoint subsets called folds. (Often, $N = 10$.)

- Each instance goes exactly into one fold.
- Run the learner and predictor N times.
- For each fold i ($1 \leq i \leq N$), the learned is trained on the union of all folds except fold i, and is then used to obtain predictions on all the instances of fold i as test.

Cross-validation, II

Options

Further precisions:

- How many folds?
  - Leave-One-Out X-validation:
    - one fold per instance;
    - most often unacceptably slow;
    - exhibits often too large variance to be reliable.
  - Very standard approach: 10 folds.
- Construct folds as instances come in? Randomize instead?
  - Keep present reproducibility!
- Potential problems if some values of the class attribute are infrequent. How to solve this?

Cross-validation, III

Stratification

Stratified X-validation

means that the folds are constructed in such a way that all the values of the class attribute are as evenly split as possible.

- Ensures even presence of all labels in all folds.
- Turns out to reduce the variance of the computed approximate accuracy.

Predictor Evaluation, I

Simplest case first: binary accuracy

Confusion matrix

(also known as Contingency matrix):

- True positives (positive prediction, hit)
- False positives (positive prediction, fail: false alarm)
- True negatives (negative prediction, hit)
- False negatives (negative prediction, fail)

Accuracy, hit ratio:

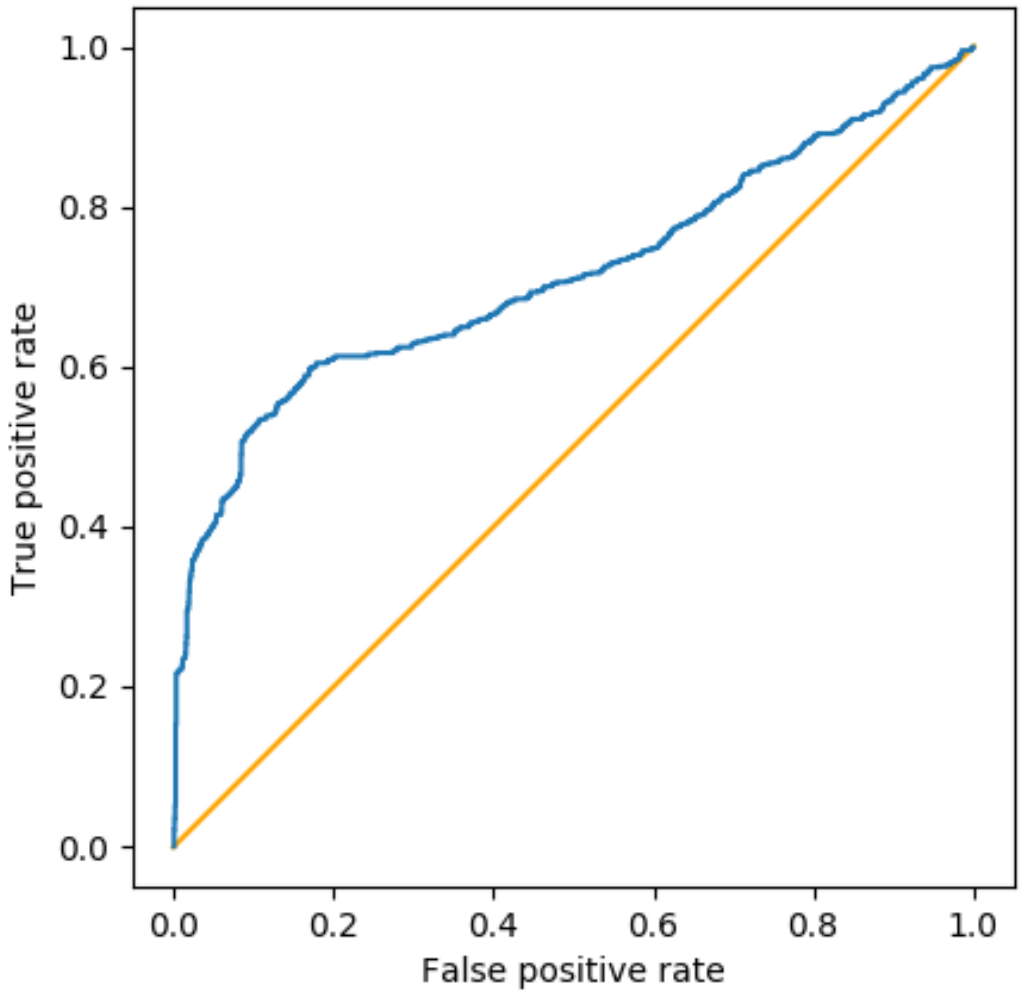
Number of hits divided by total number of predictions.

Warnings:

- Note that our reference to the true label is only indirect.
- Simple generalization to an $n \times n$ confusion matrix if the problem at hand consists of n class values.
- Some problems may suggest (or require) to weight differently the false positives and the false negatives.

ROC space and ROC curves

The curve is formalized subsequently



ROC Space

Predictors lead to points in ROC space

Consider the unit square:

Top left will mean performing quite well.

- The x coordinate is the false positive rate: the ratio of false positives to negative labels.
- The y coordinate is the true positive rate: ratio of true positives to positive labels.

The various regions of ROC space

Each has an intuitive meaning:

- Half-square below the main diagonal,
- around the center,
- near the corners...

The ROC Curve, I

Some predictors provide further information

Ranked predictions:

Predictors that may "bet" on pairs of observations, effectively sorting them.

- For instance, MAP and Naïve Bayes have several options:
  - Higher probability for the "positive" class value;
  - Larger difference of probabilities with respect to other class values...
- Regression-based predictors inherit the real line ordering;
- *Information Retrieval* algorithms are often able to order observations according to the expected relevance.

The ROC Curve, II

For predictors that are able to rank their observations

Tweak the predictor (usually by thresholding or by sorting all the n observations), so as to classify as negative exactly k points.

ROC curve:
(Receiver/Relative Operating Characteristics).

for each k from 0 to n,

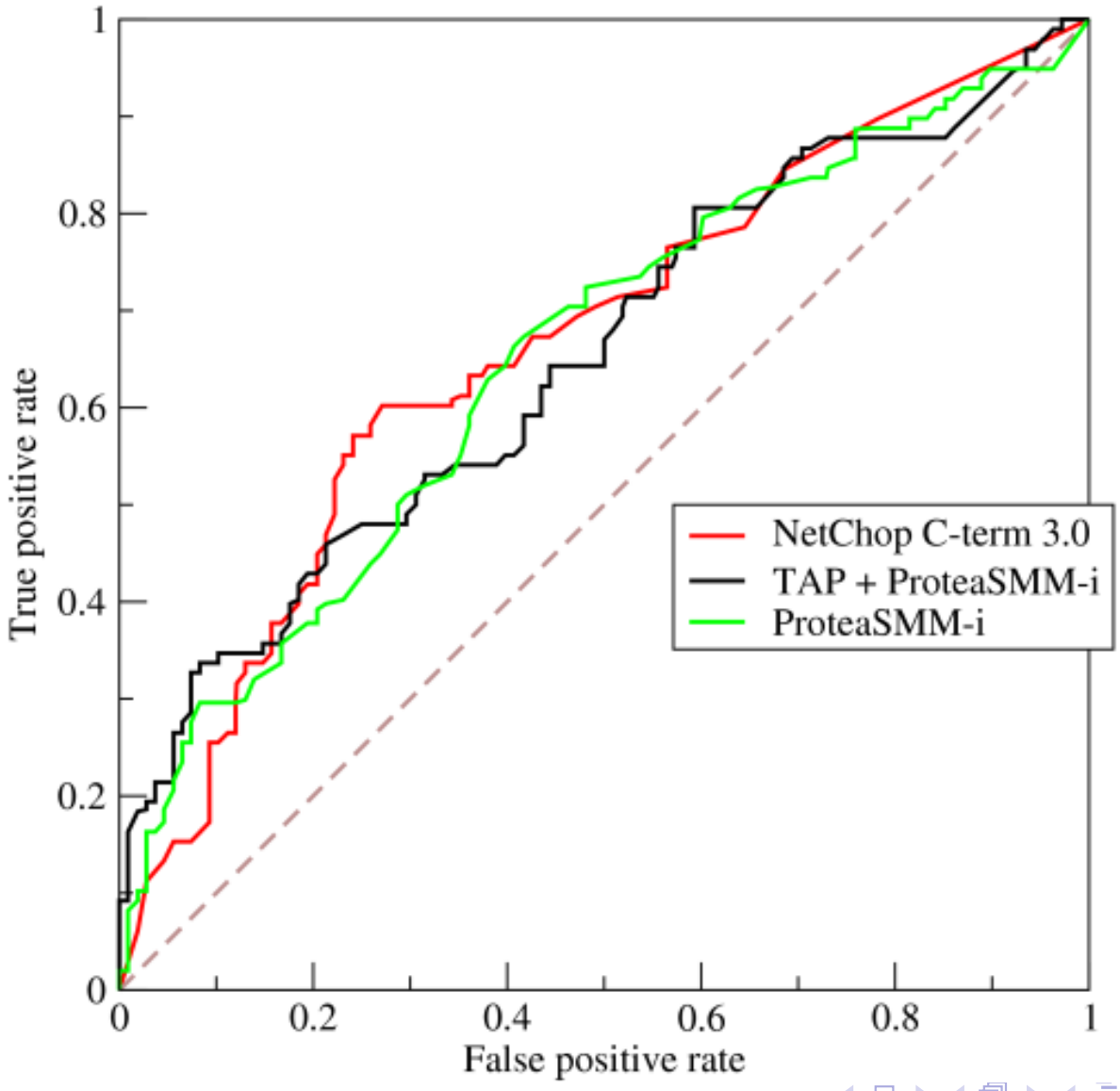
plot the ROC space point corresponding to predicting negatively to k cases (e.g. the k lowest-ranked observations).

We get a curve from (0,0), where we reject everything and there are no false positives, all the way to (1,1) where we accept everything and there are no false negatives.

The ROC Curve, III

Source: Wikipedia, 2009

Further Examples of ROC curves:



The Area Under the ROC Curve, AUC

Fashionable but dangerous

Motivation:

ROC Curves often do not lead to a clear winner among several choices of a classifier.

- AUC reduces each classifier's performance on a dataset to a single number.
- Thus allowing us to compare classifiers.
- However, it corresponds to weighting differently the false positive errors than the false negative errors,

The Area Under the ROC Curve, AUC

Fashionable but dangerous

Motivation:

ROC Curves often do not lead to a clear winner among several choices of a classifier.

- AUC reduces each classifier's performance on a dataset to a single number.
- Thus allowing us to compare classifiers.
- However, it corresponds to weighting differently the false positive errors than the false negative errors,
- and the weights depend on the classifier.

The Area Under the ROC Curve, AUC

Fashionable but dangerous

Motivation:

ROC Curves often do not lead to a clear winner among several choices of a classifier.

- AUC reduces each classifier's performance on a dataset to a single number.
- Thus allowing us to compare classifiers.
- However, it corresponds to weighting differently the false positive errors than the false negative errors,
- and the weights depend on the classifier.
- Thus, we should avoid that usage.
- See Hand (Machine Learning Journal, 2009) for further explanations and alternatives.

Predictor Evaluation, II

Sometimes accuracy is insufficient

Alternative quantities:

- Confidence of "positive label" $\Longrightarrow$ "positive prediction":

  Sensitivity (recall in *IR*): ratio of true positives to all positively labeled cases;

- Confidence of "positive prediction" $\Longrightarrow$ "positive label":

  Precision: ratio of true positives to all positively predicted cases;

- Confidence of "negative label" $\Longrightarrow$ "negative prediction":

  Specificity: ratio of true negatives to all negatively labeled cases.

For you to think about:

- How do these notions connect with the axes in ROC space?
- Express accuracy as a linear combination of sensitivity and specificity, and interpret the weights.

Predictor Evaluation, III

Train set, test set, and validation set

Even if we evaluate just accuracy, we may have two frequent reasons for the evaluation.

- We may just want to know (an approximate assessment of) the accuracy of our predictor.
  - A hold-out train/test approach will do,
  - but cross-validation will do better.
- Or, we may want to have (approximate assessments of) the accuracies of several predictors, in order to choose the one with best accuracy.
- But: what if we want both?
  - Then the cross-validation accuracy is unreliable!
  - We chose the best among several possibilities, hence it is "biased towards optimism".
  - We should combine both: hold out a subset for final validation, and run cross-validation on the rest.

Autonomous Learning Topics, II

Proposals to explore on yourself

3. Find out `why` people tend to use 10-fold X-validation schemes (link to a research paper of 1995; also the `presentation slides` are available and quite interesting).
4. Write your own implementation of a ROC curve visualizer.
5. There are several possibilities to handle ties in the construction of a ROC curve. Find bibliography on ROC curves and investigate whether these references mention this issue.
6. Ask the instructor for the link to the paper by David Hand which critizices the AUC measure and offers an alternative, and try to construct your own summary and intuitive explanation of the research findings reported there.

Prediction on Transactional Data

Still on simple predictors

Our discussion of predictors has assumed relational data.

(Often, real-valued vectors.)

What if we are to predict on transactional data?

A very common application: classification on texts.

Like:

- Spam detection,
- sentiment analysis

  (movie reviews, tweets. . .),
- news classification. . .

From Texts to Transactional Data

A wide set of issues

Preprocessing:

Very important! However, not covered in this course.

- Stemming: mapping each word to its stem.
- Feature selection: only keep things that matter
  (remove stop words, maybe punctuation...).
- ...

Terms as attributes:

With values *true* or *false*: binary representation.

Terms as items:

Texts as transactions: but,

- do repetitions matter?
- does order matter? (Sequence mining!, not covered either.)

Bernouilli Naïve Bayes

The direct transformation for binary representation

Terms as attributes with boolean values:

- $N =$ total number of terms,
- x_i is boolean (presence or absence of term),
- $P(x_i|y)$ is the probability of finding term i in a training observation labeled y, if $x_i =$ True,
- $P(x_i|y)$ is the probability of not finding term i in a training observation labeled y, if $x_i =$ False.

Bernouilli Naïve Bayes

The direct transformation for binary representation

Terms as attributes with boolean values:

- N = total number of terms,
- x_i is boolean (presence or absence of term),
- $P(x_i|y)$ is the probability of finding term i in a training observation labeled y, if x_i = True,
- $P(x_i|y)$ is the probability of not finding term i in a training observation labeled y, if x_i = False.

That is:
if the ratio of observations labeled y that have term i, among the total of observations labeled y, is p, then $P(x_i|y)$ is either

- p, for x_i = True, or
- $1 - p$, for x_i = False.

Multinomial Naïve Bayes

Appropriate for bag-of-words representation

Count only occurrences:

- Ignore absence of the items/terms,
- ignore also their order, but
- take into account repetitions:
  - $N =$ size of the observation (transaction) on which we predict,
  - x_i are the actual items in that observation,
  - some of them may be the same,
  - $P(x_i|y)$ is the proportion of item x_i among all items in all transactions of class y.

$$P(x_i|y) = \frac{\text{frequency of item } x_i \text{ in transactions of class } y}{\text{total of items in transactions of class } y}$$

Additional Considerations, I

Must be taken into account!

Upon implementation

A couple of relevant ideas:

- Multiplying together probabilities: very small numbers, high risk of underflow!

  Work instead with their logarithms (often negative!):

  addition instead of multiplication.

- Always apply a Laplace correction, unless you are pretty sure that all counts are positive

  (it is like adding implicitly to the training observations new, artificial ones that make sure that no counts remain at zero).

Additional Considerations, II

Numerical attributes?

Naive Bayes Learner View - 0:4 - Naive Bayes Learner

File

Class counts for Score

Class:	bad	good
Count:	300	700

Total count: 1000

Threshold to used for zero probabilities: 1.0E-4

P(Foreign worker | class=?)

Class/Foreign worker	No	Yes
bad	4	296
good	33	667
Rate:	4%	96%

P(Personal status and sex | class=?)

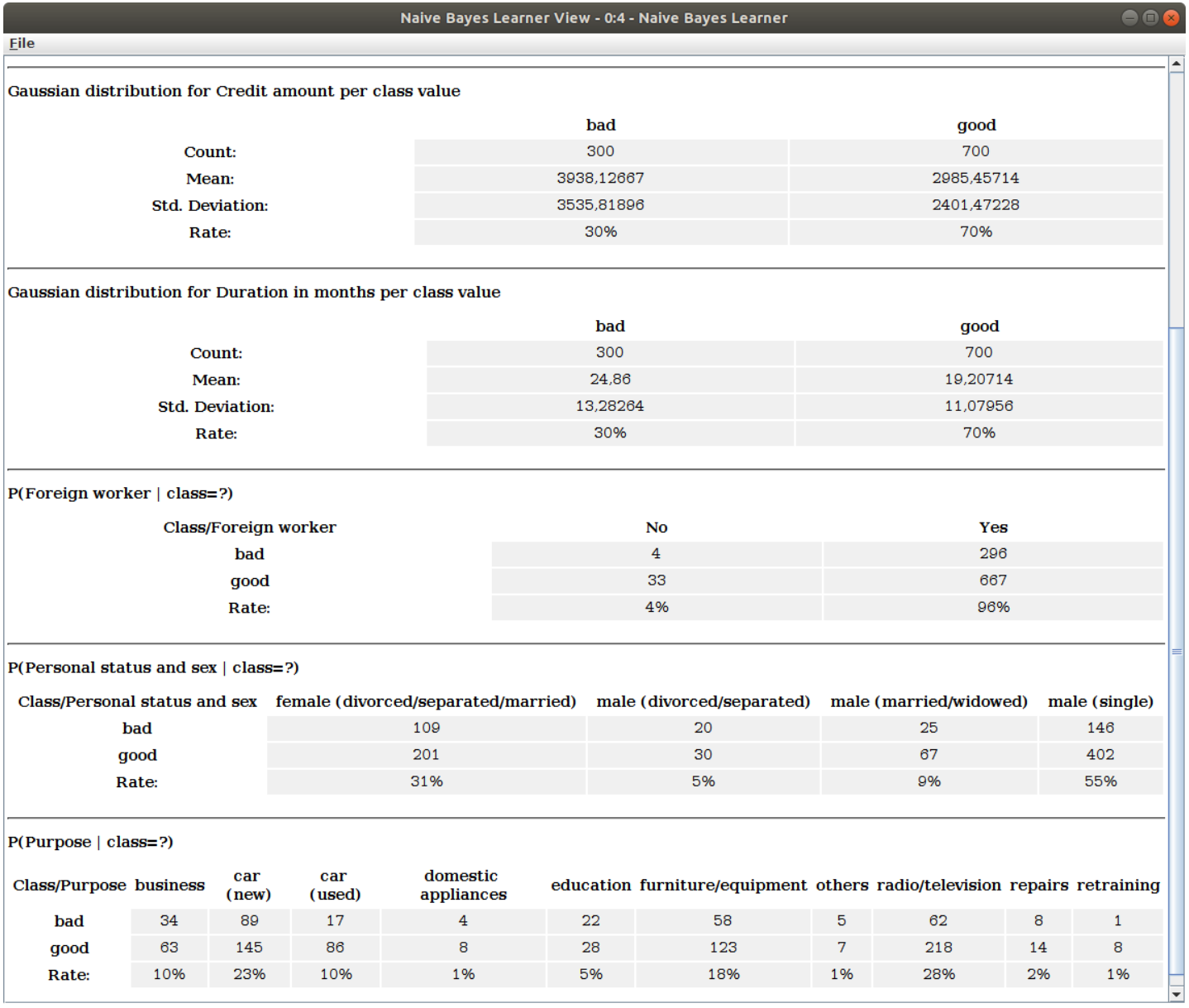
Class/Personal status and sex	female (divorced/separated/married)	male (divorced/separated)	male (married/widowed)	male (single)
bad	109	20	25	146
good	201	30	67	402
Rate:	31%	5%	9%	55%

P(Purpose | class=?)

Class/Purpose	business	car (new)	car (used)	domestic appliances	education	furniture/equipment	others	radio/television	repairs	retraining
bad	34	89	17	4	22	58	5	62	8	1
good	63	145	86	8	28	123	7	218	14	8
Rate:	10%	23%	10%	1%	5%	18%	1%	28%	2%	1%

Additional Considerations, II

Numerical attributes, standard approach: parametric view assuming Gaussians



Naive Bayes Learner View - 0:4 - Naive Bayes Learner

File

Gaussian distribution for Credit amount per class value

	bad	good
Count:	300	700
Mean:	3938,12667	2985,45714
Std. Deviation:	3535,81896	2401,47228
Rate:	30%	70%

Gaussian distribution for Duration in months per class value

	bad	good
Count:	300	700
Mean:	24,86	19,20714
Std. Deviation:	13,28264	11,07956
Rate:	30%	70%

P(Foreign worker | class=?)

Class/Foreign worker	No	Yes
bad	4	296
good	33	667
Rate:	4%	96%

P(Personal status and sex | class=?)

Class/Personal status and sex	female (divorced/separated/married)	male (divorced/separated)	male (married/widowed)	male (single)
bad	109	20	25	146
good	201	30	67	402
Rate:	31%	5%	9%	55%

P(Purpose | class=?)

Class/Purpose	business	car (new)	car (used)	domestic appliances	education	furniture/equipment	others	radio/television	repairs	retraining
bad	34	89	17	4	22	58	5	62	8	1
good	63	145	86	8	28	123	7	218	14	8
Rate:	10%	23%	10%	1%	5%	18%	1%	28%	2%	1%

Additional Considerations, III

Alternative approaches related to Naïve Bayes

Good candidates for your papers!

- A proposal to discretize numerical attributes into "bins" instead of fitting a Gaussian (and how to do that smart);
- trying to find out explicitly dependencies among the attributes: Bayesian networks;
- trying to account for dependencies among the attributes "implicitly": Hidden Naïve Bayes.

4. Regression, Bias, and Variance

Approximating a Real Value

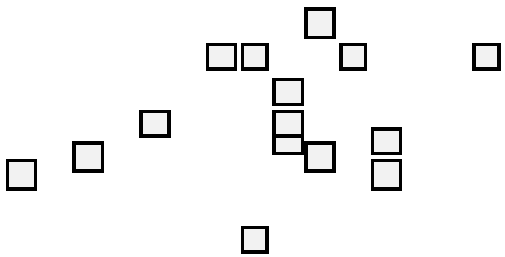
Analyzing sources of error

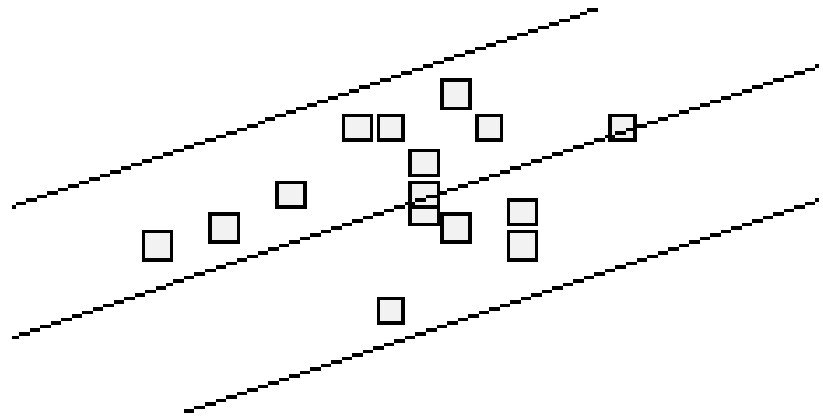
A rather common need:

- Regression tasks: predicting $f(x)$ given x and a sample of points $(x_i, f(x_i))$;
- Estimating the expected accuracy of a given classifier on unseen data, given how well it performs on sample data.
- Estimating other quantities related to properties of predictors.
- Common difficulty: the value to estimate depends on a number of explicit or implicit variables.

Regression: Prediction of Real Values

Rather: floats





Linear regression using minimum square error

The most classical and venerable predictor

A linear predictor is a line in 2D space
or a hyperplane in higher dimensions.

Absolute error is the difference between the value given by the hyperplane and the actual value.

By differentiating the expression that sums all the squares of the absolute errors and equating to zero, we can solve for the "best" hyperplane.

(Other options exist: minimum margin for one.)

The Intuition Of Variance

One source of prediction error

Why may the result be incorrect?

Variance:

Risk arising from the data.

- Data is seen as a sample;
- one cannot rule out the risk that the sample is a particularly bad one, just due to sheer bad luck;
- different samples may lead to different predictions — how different? Can the answer vary very much?
- This question is modeled by variance in the good old statistics sense: expected squared difference between the obtained values and their own mean.
- If the outcome is very concentrated around the mean outcome, there is little risk of being misled due to hitting a bad sample.

The Intuition Of Bias

Another source of prediction error

Why may the result be incorrect?

Bias:

Risk arising from your family of hypotheses.

- In a poor family of hypothesis,

  even the best one might not be very good;
- besides, the reference hypothesis is not the best one: it is the expected one with respect to the sample data; how good is it?

The Intuition Of Bias

Another source of prediction error

Why may the result be incorrect?

Bias:

Risk arising from your family of hypotheses.

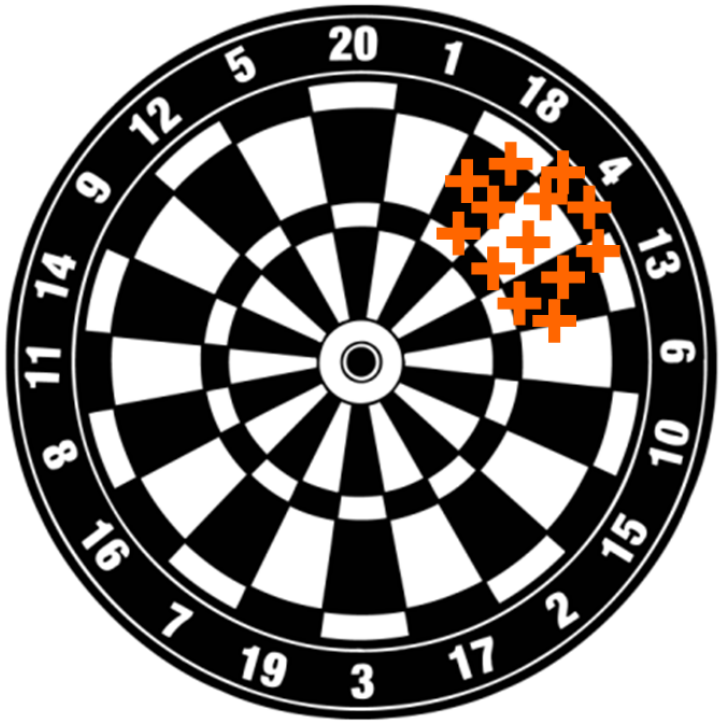
- In a poor family of hypothesis,

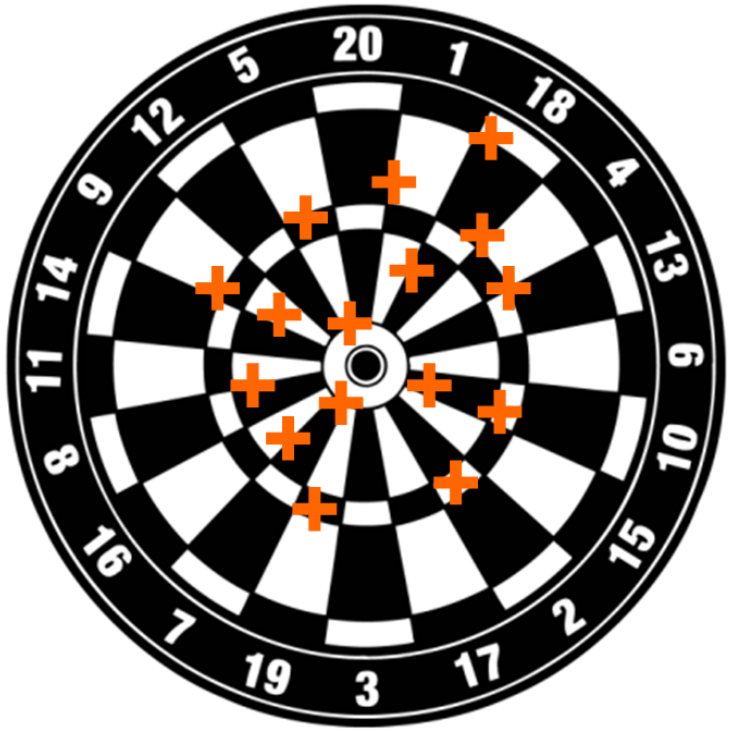
  even the best one might not be very good;
- besides, the reference hypothesis is not the best one: it is the expected one with respect to the sample data; how good is it?

But: avoiding bias error requires using rich families of hypotheses... which leads to high variance!

Visual Intuition

Of bias versus variance





Source: R. Gavaldà

A Very Simple Example

That may help understanding bias versus variance

A case with a single additional parameter.

We wish to estimate the average of a Gaussian

from a sample from it.

- It can be proved that the best estimation is the average of the sample:

  We receive a couple dozen *floats*,

  we are told they come from our Gaussian,

  we estimate the true average by the empirical average...

A Very Simple Example

That may help understanding bias versus variance

A case with a single additional parameter.

We wish to estimate the average of a Gaussian

from a sample from it.

- It can be proved that the best estimation is the average of the sample:

  We receive a couple dozen *floats*,

  we are told they come from our Gaussian,

  we estimate the true average by the empirical average...

  ... using how many decimal places?

The data could be y values for very close x values if we are facing a standard regression problem with Gaussian noise.

Is it better to use very few decimal places? We use a `Python (actually web-based Brython) program` to try a few cases.

Bias and Variance: Formalization, I

Main ingredients at play

Context:

- A real value we want to predict, y;
- A sample s which reveals some information about y;
- An estimator $e(s)$ that tries to pinpoint y after seeing s.

As $e(s)$ depends on sample s, it is actually a random variable.

But note that y does not depend on s: from the point of view of the sample, y is a constant.

Bias and Variance: Formalization, II

Formalizing bias and variance

Variance:

Quadratic average error of $e(s)$ used as estimator of its own average $E[e(s)]$: $E[(e(s) - E[e(s)])^2]$.

Bias and Variance: Formalization, II

Formalizing bias and variance

Variance:

Quadratic average error of $e(s)$ used as estimator of its own average $E[e(s)]$: $E[(e(s) - E[e(s)])^2]$.

Bias:

Absolute expected error of e with respect to the true target: $|E[e(s)] - y|$. Note: $E[e(s)]$ and similar quantities, as well as the difference with y, are again independent of s.

Note the different "scale":
we will square the bias to compensate for this.

Error Descomposition

Error is made of bias and variance

Let's add up variance and bias squared:

$$
\begin{aligned}
&E[(e(s) - E[e(s)])^2] + \\
&\qquad (E[e(s)] - y)^2 = \\
&E\left[e(s)^2 - 2E[e(s)]e(s) + E[e(s)]^2\right] + \\
&\qquad E[e(s)]^2 - 2yE[e(s)] + y^2 = \\
&E[e(s)^2] - 2E[e(s)]E[e(s)] + E[e(s)]^2 + \\
&\qquad E[e(s)]^2 - 2yE[e(s)] + y^2 = \\
&E[e(s)^2] - 2yE[e(s)] + y^2 = \\
&E[e(s)^2] - E[2y\, e(s)] + E[y^2] = E[e(s)^2 - 2y\, e(s) + y^2] = \\
&E[(e(s) - y)^2]
\end{aligned}
$$

Error Descomposition

Error is made of bias and variance

Let's add up variance and bias squared:

$$
\begin{aligned}
&E[(e(s) - E[e(s)])^2] + \\
&\qquad (E[e(s)] - y)^2 = \\
&E\left[e(s)^2 - 2E[e(s)]e(s) + E[e(s)]^2\right] + \\
&\qquad E[e(s)]^2 - 2yE[e(s)] + y^2 = \\
&E[e(s)^2] - 2E[e(s)]E[e(s)] + E[e(s)]^2 + \\
&\qquad E[e(s)]^2 - 2yE[e(s)] + y^2 = \\
&E[e(s)^2] - 2yE[e(s)] + y^2 = \\
&E[e(s)^2] - E[2y\, e(s)] + E[y^2] = E[e(s)^2 - 2y\, e(s) + y^2] = \\
&E[(e(s) - y)^2]
\end{aligned}
$$

They add up to the average quadratic error!

Consequences

Why prediction is difficult

Mean square error, bias, and variance:

- Rigid estimators, that is, with relatively limited possibilities for the result, risk converging to a result rather far away from the truth:

  high error caused by high bias.

- Flexible estimators, with very many possibilities for the result, are likely to converge to the true value (low bias error) but even small sample perturbations will change the outcome:

  high error caused by high variance.

- How to strike the best balance in terms of rigidity or flexibility of the estimators?
- Variance can be reduced if we have large datasets, but in many practical cases available datasets are very far from large enough.

Lab Session 2, I

First couple of predictors

MAP and Naïve Bayes

- Brief recap:
  - *MAP* predictor:

    $$\arg\max_C \{Pr(C|A_1 \ldots A_n)\}$$

  - *Naïve Bayes* predictor:

    $$\arg\max_C \{Pr(A_1|C) * \ldots * Pr(A_n|C) * Pr(C)\}$$

- Today: Watch them running!
  - Check confusion matrices,
  - view the internal parameters,
  - check ROC curves,
  - run comparisons...

Lab Session 2, I

First couple of predictors

MAP and Naïve Bayes

- Brief recap:
  - *MAP* predictor:

    $$\arg\max_C \{Pr(C|A_1 \ldots A_n)\}$$

  - *Naïve Bayes* predictor:

    $$\arg\max_C \{Pr(A_1|C) * \ldots * Pr(A_n|C) * Pr(C)\}$$

- Today: Watch them running!
  - Check confusion matrices,
  - view the internal parameters,
  - check ROC curves,
  - run comparisons...
- Run them on additional datasets.
- See the MAP predictor fail.

Lab Session 2, II

Using Predictors

File: `http://www.cs.upc.edu/~balqui/LabADM20200226.zip`

Parts of the Python code depend on `matplotlib` and are not runnable if you don't have it.

Explore first `LabADM20200226.py` which loads in the dataset `weatherNominalTr.txt` (in the `datasets` folder) and calls a predictor.

- Understand the code,
- uncomment `pr.show()`,
- uncomment the `print...` line inside the `for` loop,
- swap predictors to use `NaiveBayes` instead;
- now explore `compare.py` and understand the comparison.
- Explore on your own the code for both predictors and see how it fits what we discussed so far.

Lab Session 2, III

Comparing Predictors

Move on to other datasets.

- Change the main program so as to load in the `titanicTr` dataset instead;
- Redo the whole thing with both predictors: which one seems better?
- Ask the predictors to predict on new tuples (see bottom of file `maxapost.py`):

  ```
  print pr.predict(('Class:1st','Sex:Female','Age:Child'))
  print pr.predict(('Class:Crew','Sex:Female','Age:Child'))
  ```

  Explain what happens!
- Try predicting other attributes.
- Explore the other datasets:
  - which predictor seems better for each?
  - Try to make the MAP predictor fail again like before.

Lab Session 2, IV

Evaluating Predictors

Train/test split

- In the Python `LabADM...` file, replace the declaration `Data(filename)` with `Data(filename,75)` (or a different figure in [0,100]).
- Understand what happens; then move to `compare.py` and work likewise on it.
- Explore this train/test decomposition for other datasets on yourself, using `naivebayes.py`, `maxapost.py`, `compare.py`...

ROC Curves

If you have `matplotlib`, proceed to exploring ROC curves through the source `roc.py` using both predictors and varying the dataset (and the value of the label to analyze).

Make sure to understand what happens!

Lab Session 2, V

Back to KNIME

On KNIME

redo parts of what you have done today (see Slides 35, 36):

- Manage to read in some dataset.
- Find the nodes for Naïve Bayes Learning and for Naïve Bayes Prediction (that uses the model learned).
- Find the Scorer node, that implements confusion matrices and accuracy evaluation, and compute resubstitution error.
- Add a Partitioning node to split the data into training set and test set, and compute the test set error.
- Try several partitioning strategies to check whether the test set error is stable.
- Find the ROC Curve node and show ROC curves of your predictors on your datasets.

Autonomous Learning Topics, III

Proposals to explore on yourself

7. Design and explore in practice cases of bias-variance trade-offs:
   - according to the degree, interpolating polynomials may incur high bias or high variance;
   - according to the number of units, neural networks may incur high bias or high variance...
8. The bias-variance trade-off has been shown for estimating a real value. Is there a way of analyzing in similar ways binary predictors? (Yes, of course; the problem is, there are several of them... Explore the literature!).

5. Additional Predictors

Nearest Neighbors, I

The data is the model

Assumption:

Similar observations lead to similar responses.

- Keep all the data in an appropriate data structure, and
- predict the most common response among the k nearest neighbors of a new observation to predict on.
- (Lots of demos on youtube.)

Nearest Neighbors, I

The data is the model

Assumption:

Similar observations lead to similar responses.

- Keep all the data in an appropriate data structure, and
- predict the most common response among the k nearest neighbors of a new observation to predict on.
- (Lots of demos on youtube.)
- Essentially, we are assuming a "bias of continuity"!
  - Often, the continuity assumption is correct.
  - Often, it is not.

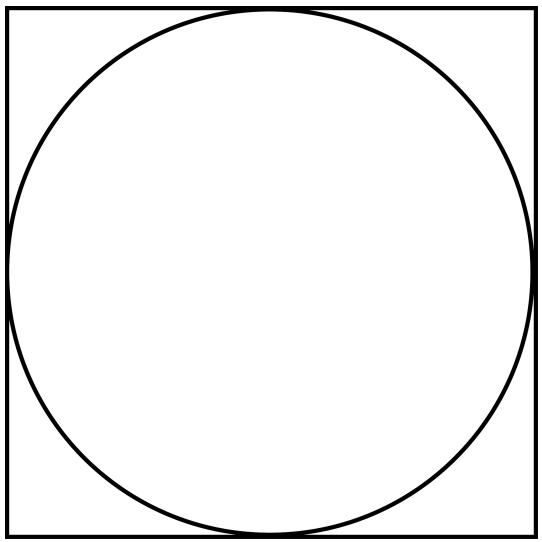
Careful!

In high dimensionality, "everything is far away".

- Is the difference between the closest neighbors and the farthest ones significative?
- `Hardly ever the case` beyond a couple dozen attributes! `(Alternative link.)`

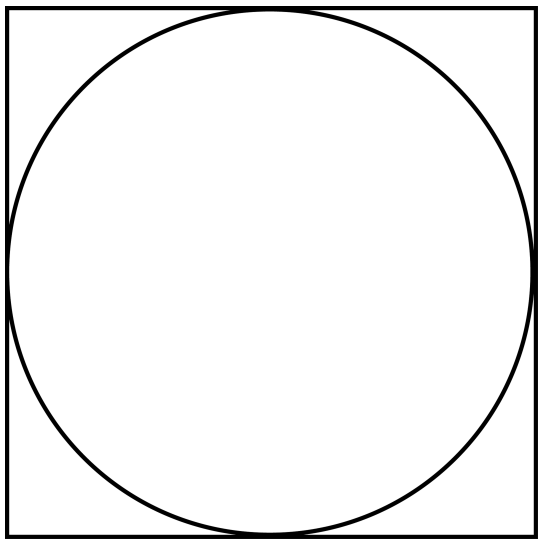
From 2 to 3 dimensions

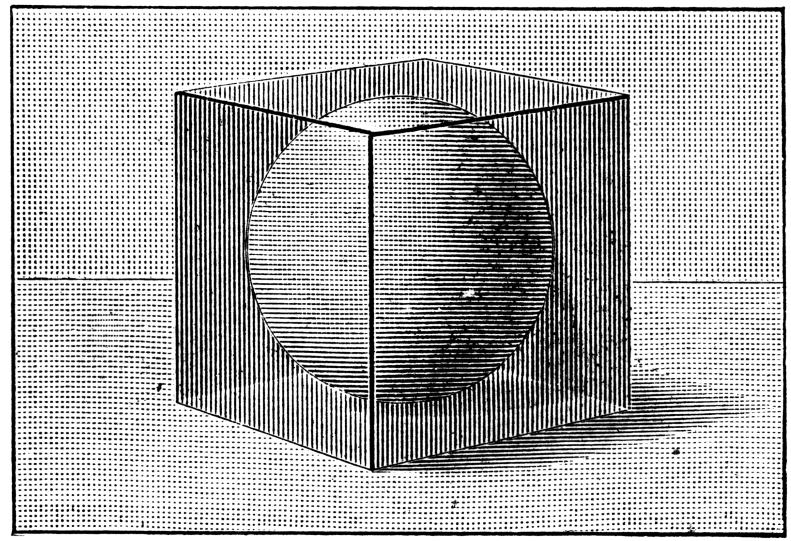
To grasp the trend



From 2 to 3 dimensions

To grasp the trend





Source: FCIT

Nearest Neighbors, II

Variations

Value of k?

(Odd for 2-class problems.)

Scale into similar intervals all the numeric attributes? Imagine:

- one attribute is age,
- another is annual salary in euros...

Options:

$$x' = \frac{x - \mu}{\sigma} \qquad x' = \frac{x - min}{max - min}$$

Weighted majority instead of plain majority?

(Then can use largish k.)

Nearest Neighbors, III

Data structures

In high dimensions, finding out the k nearest neighbors is computationally nontrivial.

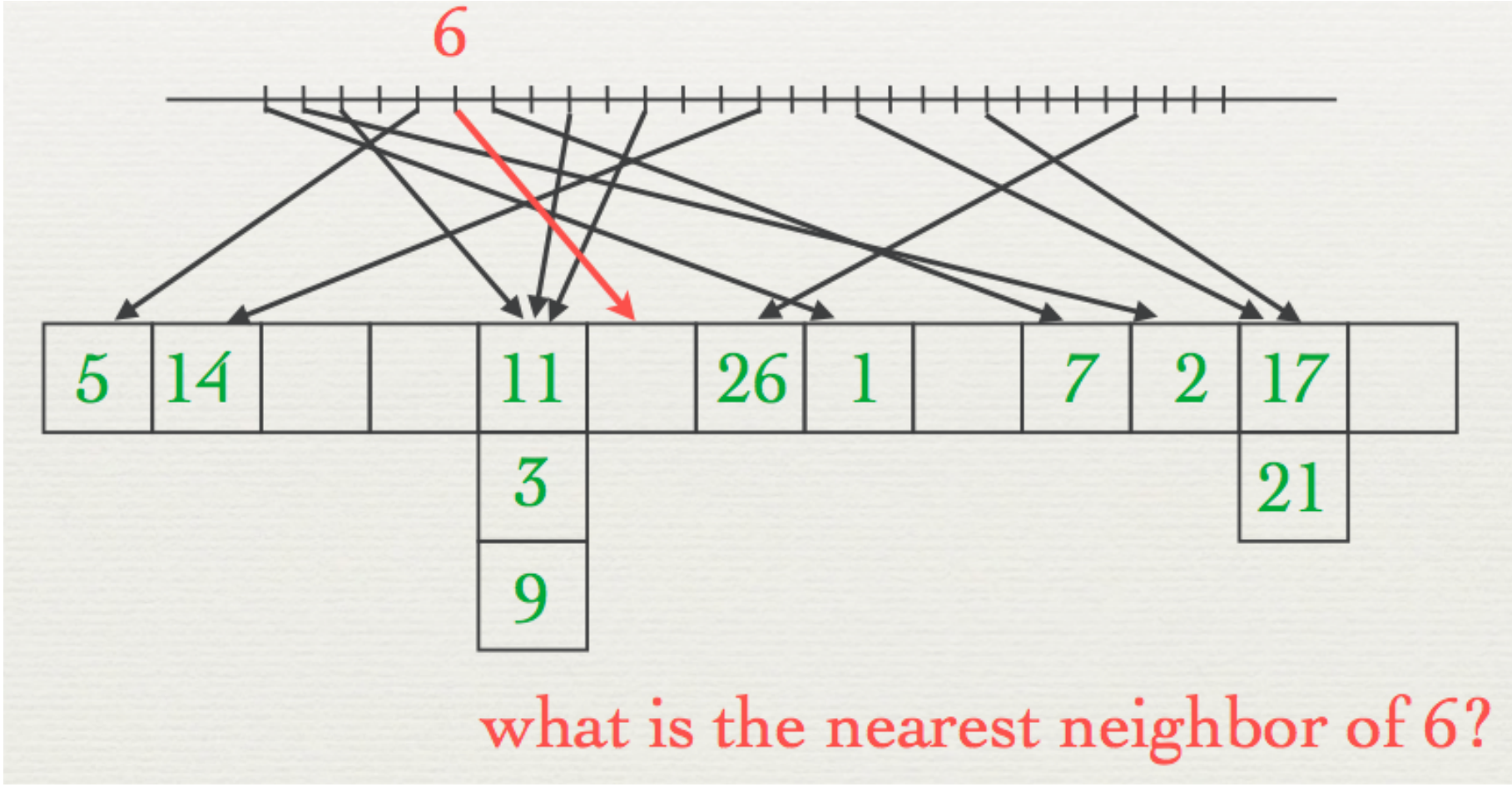
Computational options:

- Multidimensional search trees:
  - k-d-trees,
  - metric trees,
  - cover trees,
  - ball-trees;
- proximity graphs;
- locality-sensitive hashing

  (we study this one a bit more);
- ...

Usual Schemes for Hashing

Do not preserve any locality



Locality sensitive hashing functions, I

Definition

Goal:

- Collision probability for similar objects is high enough.
- Collision probability for dissimilar objects is rather low.

(Exercise: how is the algorithm, once we have this?)

Locality sensitive hashing functions, I

Definition

Goal:

- Collision probability for similar objects is high enough.
- Collision probability for dissimilar objects is rather low.

(Exercise: how is the algorithm, once we have this?)

Let $c < 1$ and $0 \leq p_1 < p_2 \leq 1$; function $s(x, y)$ measures how similar objects x and y are in a scale $[0, 1]$.

A family $\mathcal{F}$ is called $(s, c \cdot s, p_1, p_2)$-sensitive if for any two objects x and y we have:

- if $s(x, y) \geq s$, then $P[h(x) = h(y)] \geq p_2$,
- if $s(x, y) \leq c \cdot s$, then $P[h(x) = h(y)] \leq p_1$,

where the probability is taken over chosing h from $\mathcal{F}$.

Locality sensitive hashing functions, II

An example for bit-vectors

Consider the following context and hashing family:

- Objects are vectors in $\{0,1\}^d$.
- Distances are measured using Hamming distance

$$d(x,y) = \sum_{i=1}^{d} |x_i - y_i|.$$

- Similarity is measured as

$$s(x,y) = 1 - \frac{d(x,y)}{d}.$$

(Example: if $x = 10010$ and $y = 11011$, then $d(x,y) = 2$ and $s(x,y) = 1 - 2/5 = 0.6$.)

Then: the i-th hashing function just samples the i-th bit.

Locality sensitive hashing functions, III

Playing with the probabilities

The probability of collision is

$$P[h(x) = h(y)] = s(x, y).$$

Locality sensitive hashing functions, III

Playing with the probabilities

The probability of collision is

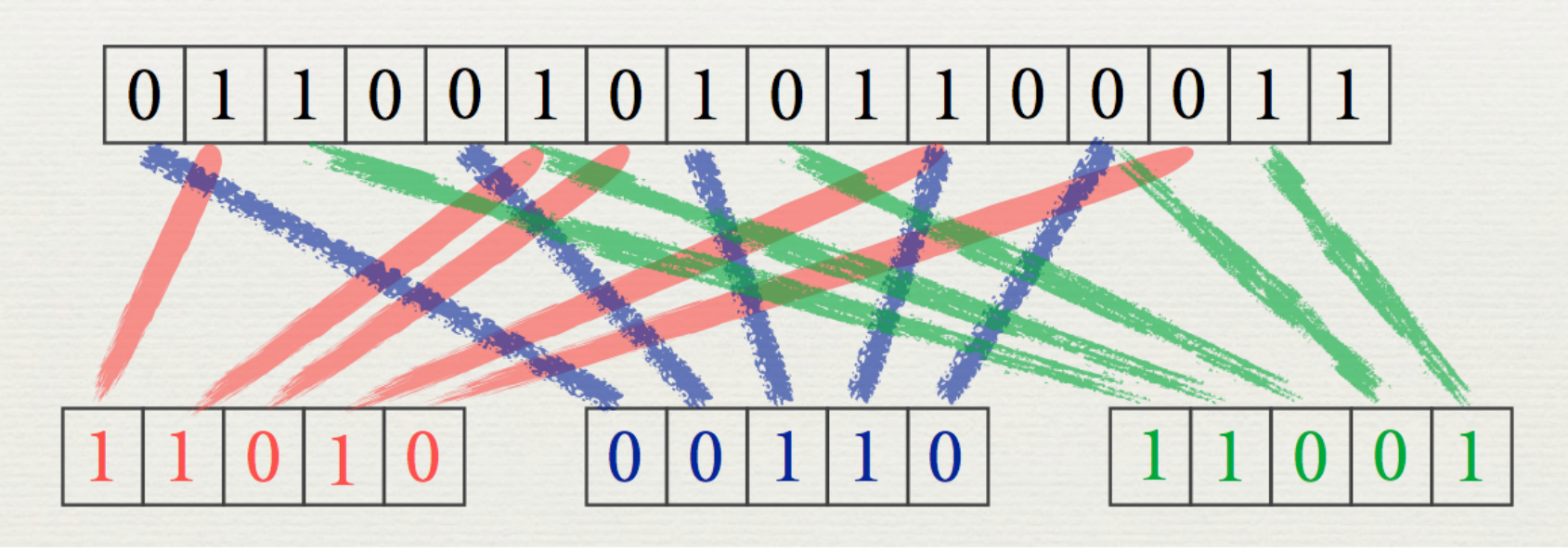
$$P[h(x) = h(y)] = s(x, y).$$

Amplifying the gap:

- By stacking together k hash functions:
  - $h(x) = (h_1(x), .., h_k(x))$ where $h_i \in \mathcal{F}$;
  - probability of collision of similar objects decreases to ${p_2}^k$;
  - probability of collision of dissimilar objects decreases even more to ${p_1}^k$.
- By repeating the process m times:
  - Probability of collision of similar objects increases to $1 - (1 - p_2)^m$.

Locality sensitive hashing functions, IV

Illustrating the bit-vector particular case



Lab Session 3

KNIME Prediction Nodes

Refresh:

- read in, transform, explore, and visualize data;
- clarify its basic statistic properties;

Today:

- Find some relational datasets,
- create workflows to run, evaluate, and compare several varied predictors on them:
  - partitioning into train/test data,
  - cross-validating. . .

  Count on a bit of help from the instructor when necessary.

Alternatively:

- Keep working hard on your first (or second!) deliverable.

Decision Trees, I

Can you imagine explaining your NB predictor to your boss?

Can we make do by checking a single attribute?

If not. . .

Decision Trees, I

Can you imagine explaining your NB predictor to your boss?

Can we make do by checking a single attribute?

If not... recurse!

Decision Trees, I

Can you imagine explaining your NB predictor to your boss?

Can we make do by checking a single attribute?

If not... recurse!

- Measure somehow the "heterogeneity" of the observations, and
- Pick one "test" of the value of an attribute so that the split reduces the "joint heterogeneity".

Decision Trees, I

Can you imagine explaining your NB predictor to your boss?

Can we make do by checking a single attribute?

If not... recurse!

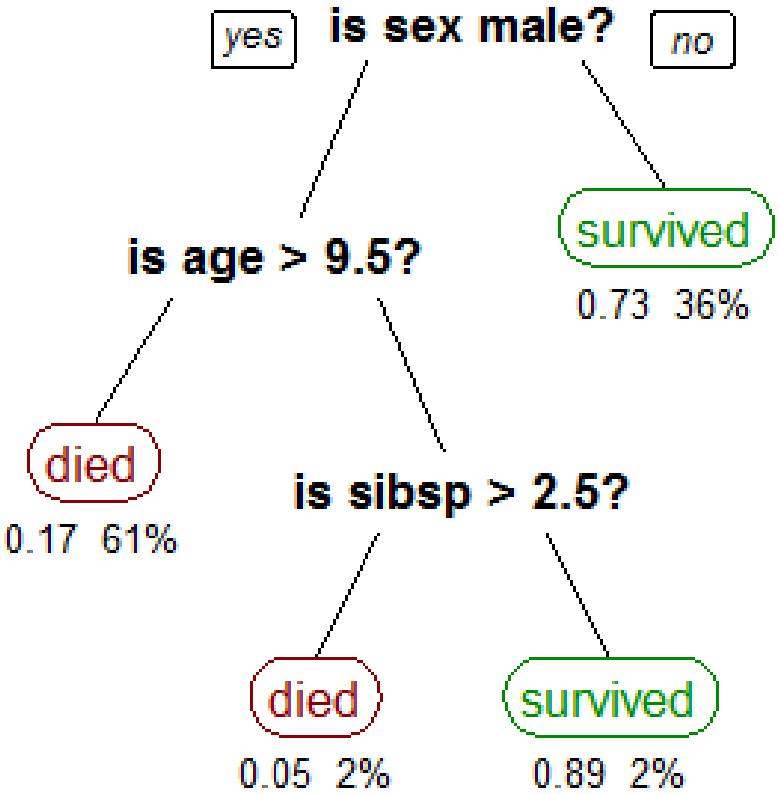
- Measure somehow the "heterogeneity" of the observations, and
- Pick one "test" of the value of an attribute so that the split reduces the "joint heterogeneity".

Several variants of this idea (ID3, C4.5, j48, C5.0, CART):

- the prediction follows a decomposition of the input space in "axis-parallel cuboids", but
- "tests" can be made in different ways, and
- there are several possible notions of "heterogeneity".

Decision Trees, II

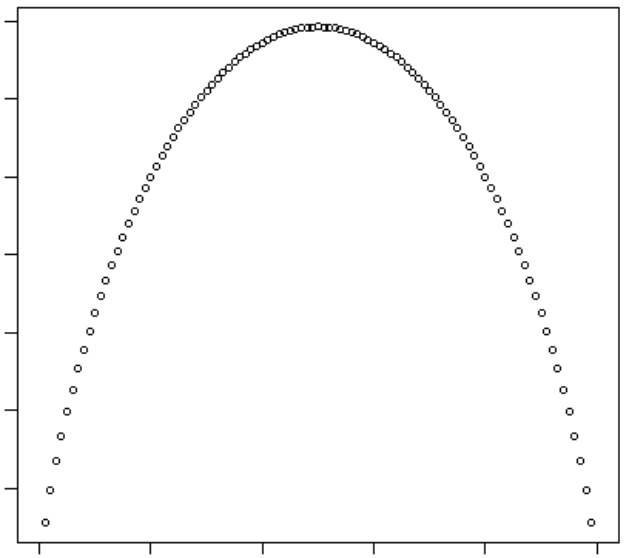
How do they look like?

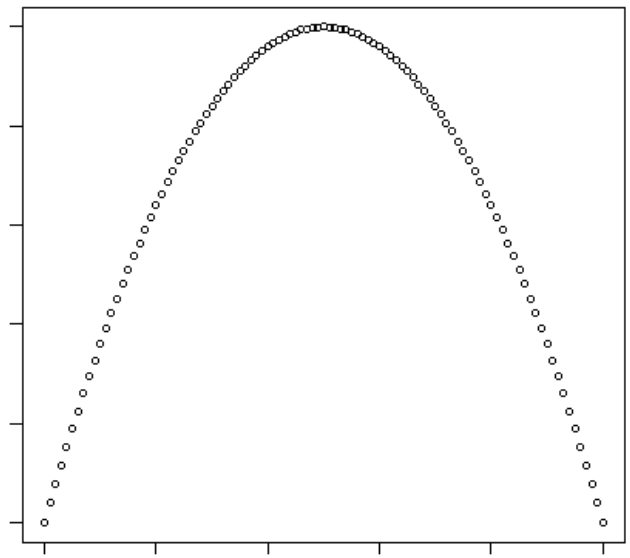


A CART example tree. Source: Wikipedia, 2014.

Heterogeneity

Shannon information versus Gini index (2-valued case)





Decision Rules, I

Related to decision trees

Decision rules: Decision trees explained verbally instead of depicted. (Horn-clause-like syntax!) `Example.`

Both:

- Are predictive and descriptive models;
- Easy to understand (as long as they are small enough).
- Low bias, but very high variance, hence low tolerance to "noise";
- Occassional slight advantage of decision rules over decision trees.
- A frequent outcome: branches or rules carve out small, very well-predictable niches but fail to get global patterns.
- Syntactic allusion to a suggested causality connection (*policy-makers* love it!)

Decision Rules, II

How to select them? How to apply them?

Several options for several choices,

hence many slightly different algorithms:
CN2, IREP, Ripper (JRip)...

- Which rules build up the predictor?
- At the time of applying the predictor,
  - what if no rule fires?
  - what if several rules fire, and lead to different predictions?
- One easy way out: rules are the branches of a decision tree.
- Other criteria:
  - coverage per rule,
  - accuracy per rule,
  - *default* rule,
  - exceptions...

Exploring and Extending KNIME

Many extra possibilities and modules

Learn to:

- Extend KNIME: suggestions of interesting imports are the Text Processing extension and the Weka extension.
- Use meta-nodes:
  - Find them in their appropriate place — do not confuse their place with the place for "nodes that are used to construct meta-nodes".
  - Practice with the cross-validation meta-node.
  - Practice with the feature elimination meta-node.
- Use variables:
  - Show variable ports,
  - inject variables defined through Java snippets,
  - use them in the corresponding tab,
  - explore further nodes handling variables...
- Program loops!
  - Explore nodes for loop programming;
  - combine them with variables such as the current iteration.

Feature Elimination

A process for reducing dimensionality

Works as follows:

- Combines a dataset with your predictor of choice.
- Repeats the following process:
  - For each attribute in turn:
    - Remove the column temporarily,
    - train with the rest,
    - test or x-validate;
  - The attribute that got worst accuracy along the loop is eliminated.
- Keep like this until reaching a fixed accuracy threshold or a given number of attributes.

KNIME Assignments

To work on your own as a means of mastering the tool

(These assignments are not compulsory. They are just proposals, expected to facilitate one or more of your coming evaluable papers.)

1. Create KNIME workflows that try several values of some parameter of some predictive model in search of good accuracies.
2. The `shuttle` dataset from the UCI Irvine repository has been reported to exhibit the following property: at some ratios of training/test split, larger training sets lead to an unexpected decrease of accuracy of the Naïve Bayes classifier. Create a KNIME workflow to try and replicate this experiment.
3. Repeat the experiment by playing with the number of folds in x-val instead of the ratio of training/test sets, and/or employing other classifiers.
4. Learn to program your own KNIME nodes!

Classifier Border Repertory

Class-separation shapes

- Decision Stumps:
  - axis-parallel hyperplanes,
- Decision Trees:
  - unions thereof,
- kNN, NB:
  - complex shapes...

Classifier Border Repertory

Class-separation shapes

- Decision Stumps:
  - axis-parallel hyperplanes,
- Decision Trees:
  - unions thereof,
- kNN, NB:
  - complex shapes...
- Linear predictors:
  - Separating hyperplanes

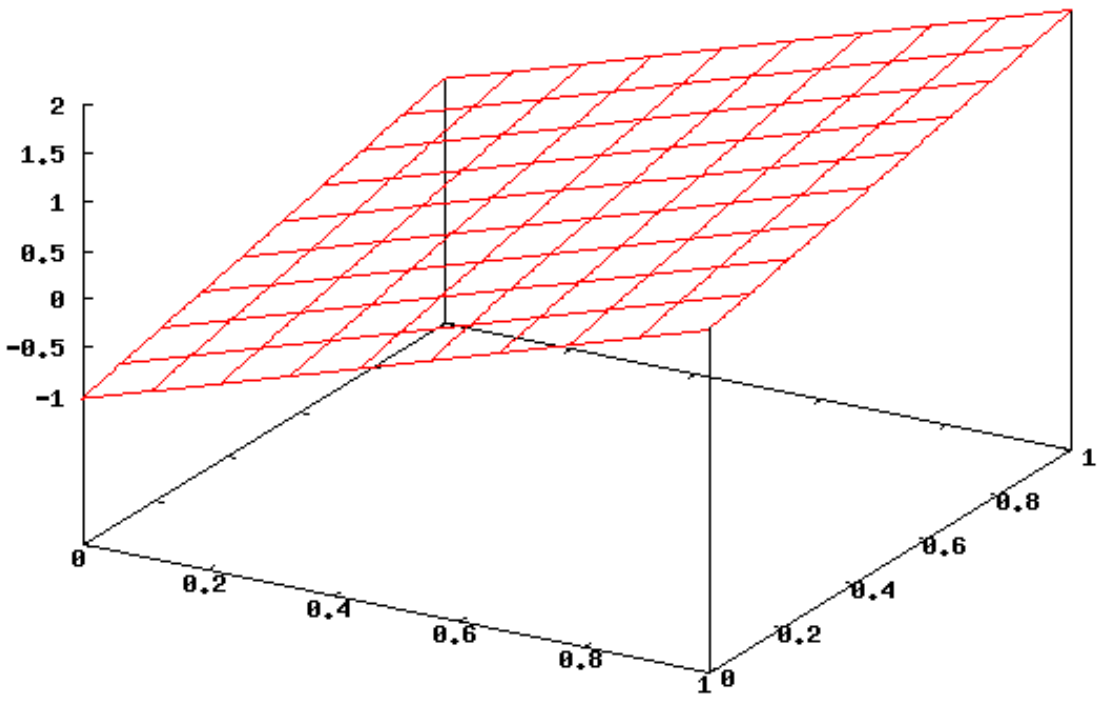
Classifier Border Repertory

Class-separation shapes

- Decision Stumps:
  - axis-parallel hyperplanes,
- Decision Trees:
  - unions thereof,
- kNN, NB:
  - complex shapes...
- Linear predictors:
  - Separating hyperplanes (not necessarily in the same space!)
    - Hard threshold,
    - Soft threshold.

A linear separator

In R^3: $2x + y - 1$



Support Vector Machines, I

SVM: the modern linear predictors

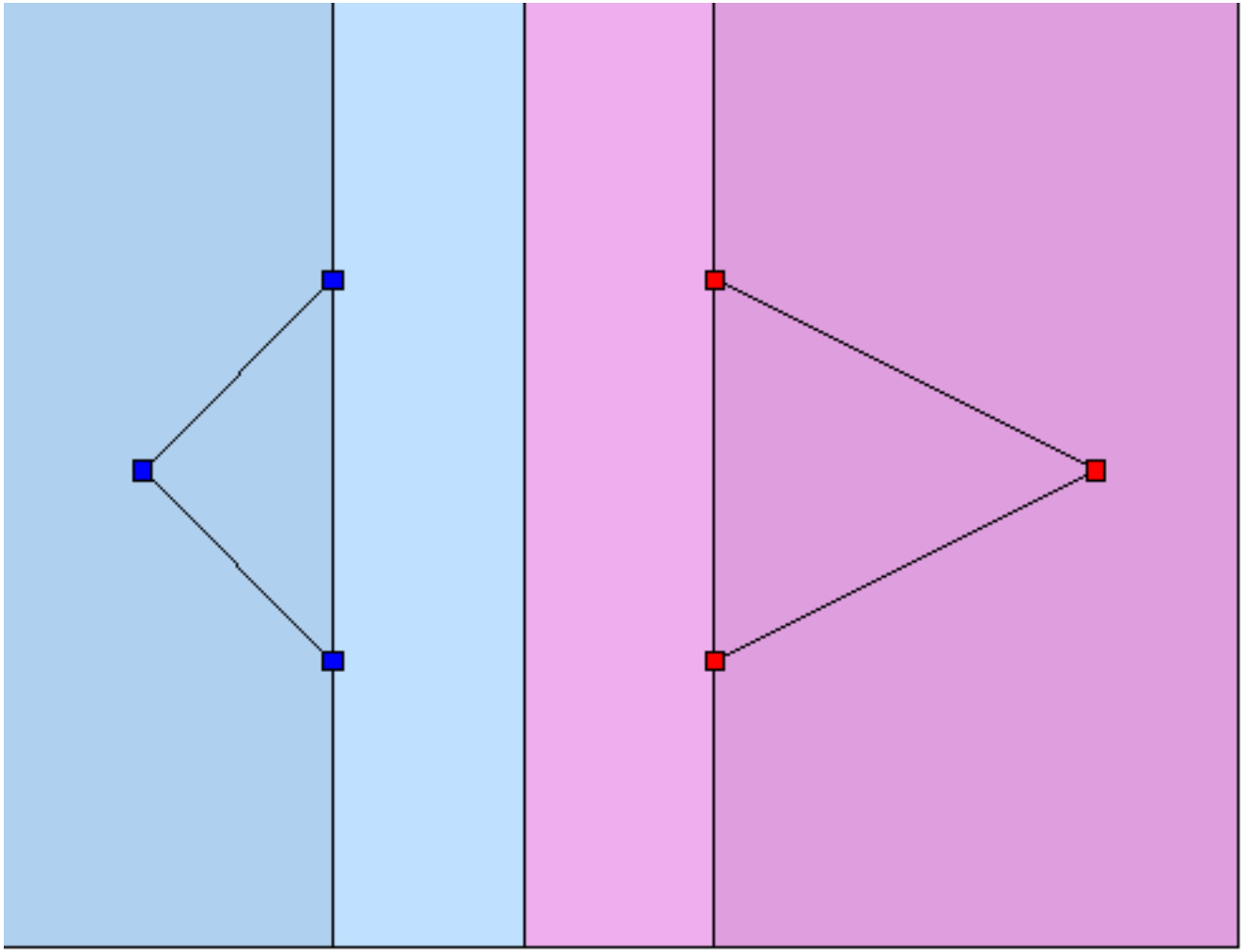
Slogan: maximal margin; don't get closer to any of the classes more than absolutely necessary.

- Hard margin: requires linear separability.
- Two alternatives for coping with nonlinearly separable data:
  - soft margin and
  - expanding the data into a feature space with a kernel.

We start with a couple of little toy implementations from `Suverat` and from `LIBSVM`.

Maximal-Margin Hyperplane, I

Linearly separable cases



Support Vector Machines, II

Some related hints

Optimization rendering:

Maximize m, under the constraints: $y_i \frac{(w^T x_i + b)}{||w||} \geq m$.

(Plus a funny trick on the scaling!)

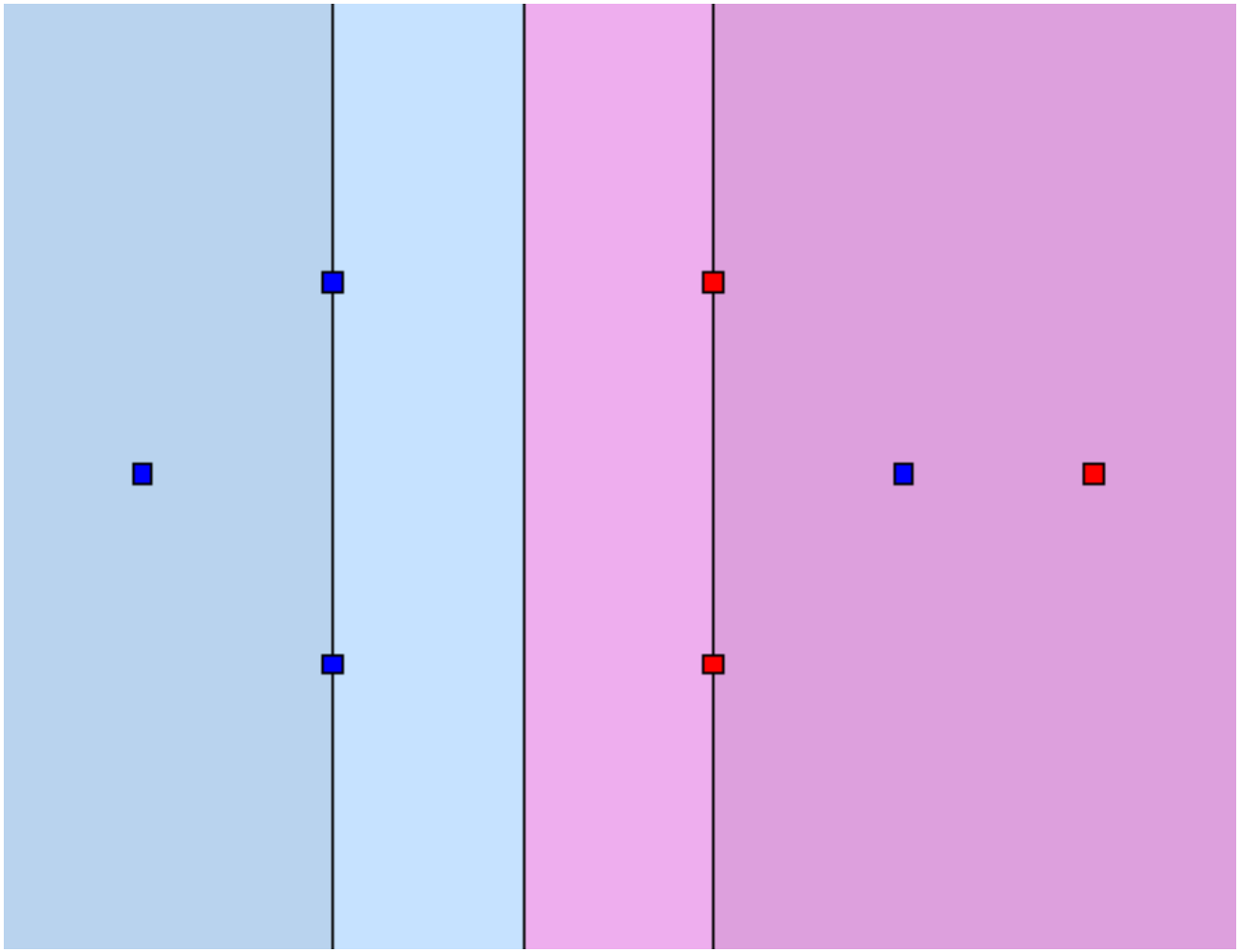
Duality and Lagrange multipliers:

The separating hyperplane is defined as linear combination of the data points.

- Coefficient is zero for many points! Nonzero coefficients only for the support vectors that lie exactly at the margin or (in the nonseparable case) within it.
- Polynomial-time semidefinite programming solution.
- Only operation required on the data points: their scalar product.

Maximal-Margin Hyperplane, II

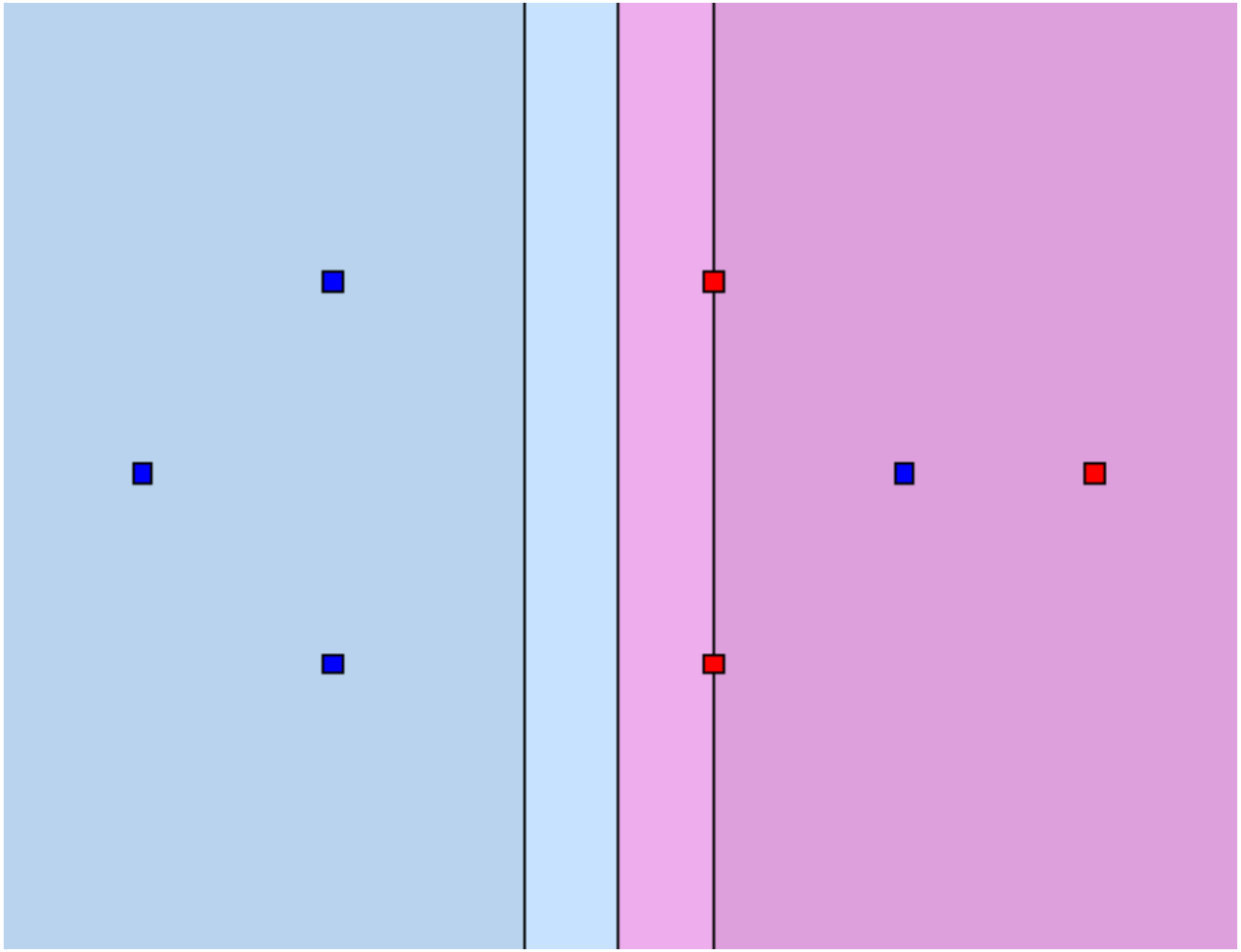
Linearly inseparable cases: not the same hyperplane due to misclassified instances



Hyperparameter to balance margin amount and mistakes made.

Maximal-Margin Hyperplane, III

Linearly inseparable cases: misclassified instances push the separator



Intuition: convex hulls! (Bennet and Bredensteiner.)

Support Vector Machines, III

Several implementations out there

Alternative algorithmics:

- Reduction to general-purpose semi-definite quadratic programming (QP) software — usual cost $O(n^2 m^2)$ for m points in n dimensions.
- Decomposition methods:

  they concentrate on just a subset of the points at each time.
  - The extreme case is Sequential Minimal Optimization: nontrivial to do it well, see `LIBSVM` (and, in particular, the paper linked there in the ACM Transactions on Intelligent Systems).
  - Reweighting scheme able to go down to $O(n^3 \log m)$, better when n much smaller than m, based on the Simple Sampling Lemma of Gärtner and Welzl
    (to demo it, we go back to our little `toy implementation` of this scheme).

Kernels

Switch to a richer space

Reproducing Kernel Hilbert Spaces:

Can be obtained through scalar products.

A two-dimensional conic:

$$w_1 x_1^2 + w_2 x_2^2 + w_s x_1 x_2 + w_4 x_1 + w_5 x_2 + w_6$$

is the scalar product of the weights $(w_1, w_2, w_3, w_4, w_5, w_6)$ with a "transformed" input point (x_1, x_2) a R^6:

$$f(x_1, x_2) = (x_1^2, x_2^2, x_1 x_2, x_1, x_2, 1)$$

(Please compute $((x_1, x_2)(y_1, y_2) + 1)^2$.)

`Search for 'svm' on youtube.`

Scheme may be made to work even on infinite-dimensional feature spaces!

Ensemble Methods

Improving weak but fast predictors

Often, getting good predictions require slow training algorithms.

Simple predictors are fast to train, but often weak — but, how weak? Would do they better than random guessing?

- Quite small decision trees,
- decision stumps (that is, minimally small decision trees),
- quite small decision rules,
- Naïve Bayes and variants...

Can we enhance the predictions of a simple predictor by combining several of them?

- Bagging (Random Forests, Random Naïve Bayes...),
- boosting (mainly Adaboost but there are many others),
- stacking...

Bagging: bootstrapping aggregates

Assume a relational dataset with n observations.

Fix some weak but fast-to-train predictive model.

- Sample $m \leq n$ observations (*bootstrap*), most often $m = n$.

Bagging: bootstrapping aggregates

Assume a relational dataset with n observations.

Fix some weak but fast-to-train predictive model.

- Sample $m \leq n$ observations (*bootstrap*), most often $m = n$.
- With replacement!

Bagging: bootstrapping aggregates

Assume a relational dataset with n observations.

Fix some weak but fast-to-train predictive model.

- Sample $m \leq n$ observations (*bootstrap*), most often $m = n$.
- With replacement!
- Train a predictor on the sample.
- Repeat k times.

Bagging: bootstrapping aggregates

Assume a relational dataset with n observations.

Fix some weak but fast-to-train predictive model.

- Sample $m \leq n$ observations (*bootstrap*), most often $m = n$.
- With replacement!
- Train a predictor on the sample.
- Repeat k times.

Predict: according to majority.

Newly available measure of accuracy: Error OOB (*Out-Of-Bag*), predicting on each x, by majority only of those predictors that "did not see" x upon being trained.

Random Forests

Apply bagging:

- on a form of decision trees,
- where each bifurcation is on the best splitting attribute among a small sample of them,
- and applying no pruning.

Properties:

- Several "success stories".
- Fast construction as few attributes examined per split.
- Some lucky nodes that catch very discriminative attributes compensate for the rest.
- Often still overfits.
- Not very good in the presence of many irrelevant attributes.

Random Naïve Bayes

Tricky!

Bagging on Naïve Bayes... but...

Recall that each Naïve Bayes prediction is actually a number:

- Obtained as a product of probabilities.
- For standalone Naïve Bayes, we "discretize" by looking for the class value that maximizes the probability.

Trick: instead, use them as weights that quantify how reliable the prediction is, obtaining the final prediction as a weighted majority.

Boosting

Formally, "boosting" means that there is a theorem that tells you how to combine bounded-but-large-error predictors so as to obtain small-error predictors.

Many people confuse Boosting (as a general property) with the most famous boosting algorithm: AdaBoost.

AdaBoost is similar to *bagging* but more sophisticate: sampling is potentially not uniform.

AdaBoost, I

Applies to binary classification.

New ingredient D: explicitly maintained probability distribution on the n data points.
(Initially uniform: $1/n$ probability mass per point.)

Algorithm:

- Train a predictor by taking the weights $D(x)$ into account:
  - Sample by independently choosing each x according to $D(x)$.
  - Train the predictor on the sample.
- Assign a weight to that predictor for the later weighted-majority prediction.
- Recompute D: increase the weight of errors (and normalize, of course).
- Repeat as long as the predictor obtained works better than random guessing.

AdaBoost, II

Tuning the scheme to recompute weights

Each round provides us with a new predictor h:

- Compute ϵ, its weighted error
  - (replacement error on the sample used to construct it, that is, that sample is both training set and test set; and
  - weighted error is the sum of the weights $D(x)$ for those x in the sample where $h(x)$ is wrong).
  - If $\epsilon \geq \frac{1}{2}$ then discard h and finish the training process.
- Tune the weights:
  - let $d = \frac{1-\epsilon}{\epsilon}$ (note: $d > 1$);
  - let $D(x) := D(x)/d$ if h is correct on x;
  - let $D(x) := D(x) * d$ otherwise;
- and don't forget to normalize D.
- For the final weighted majority prediction, store $\log d$ as weight to be assigned to h.

AdaBoost, III

Properties guaranteed by mathematical theorems

After T rounds, let ϵ_t be the error of the weak predictor obtained at round t. (All $\epsilon_t < \frac{1}{2}$ due to the finishing condition!)
Then, the error of the weighted majority is bounded by

$$e^{-2\sum(\frac{1}{2}-\epsilon_t)^2}.$$

For example: if weak-predictor error is always under 40%, then reaching $T = 10$ gives error under 7%, and reaching $T = 20$ gives error under 0.7%.

"Difficult" data is easily spotted because AdaBoost terminates in just a few rounds on them.

Additionally, there are theorems bounding the generalization error and also the classification margin: these explain the observed phenomenon that AdaBoost often avoids overfitting.

Autonomous Learning Topics, IV

Proposals to explore on yourself

9. Study alternative data structures supporting Nearest Neighbors.
10. Deepen in your understanding of some predictor(s), such as Decision Trees or Support Vector Machines (consider doing it by building your own implementation).
11. Deepen in your understanding of some meta-predictor, such as AdaBoost or Random Forests (consider doing it by building your own implementation).
12. Learn how to program your own KNIME nodes.

6. Clustering

Clustering

Computer-achieved abstraction

Group observations:

Make up your mind about how to "see" the observations in your dataset grouped together.

- Treat similar cases similarly (e. g. marketing campaigns);
- Identify "approximately common" characteristics of population segments;
- Get a more succinct explanation of what is in your data such as representing each "cluster" by a single point.

Clustering Methods

Starring K-Means and Expectation-Maximization

Besides K-Means and EM, there are many more:

K-medoids, PAM, CLARANS, CobWeb, BIRCH, Chameleon, DBSCAN, OPTICS ...

- Spectral Clustering,
- Biclustering and Conceptual Clustering,
- Hierarchical Clustering
  - agglomerative,
  - divisive...

Clustering Methods

Starring K-Means and Expectation-Maximization

Besides K-Means and EM, there are many more:

K-medoids, PAM, CLARANS, CobWeb, BIRCH, Chameleon, DBSCAN, OPTICS ...

- Spectral Clustering,
- Biclustering and Conceptual Clustering,
- Hierarchical Clustering
  - agglomerative,
  - divisive...

But, what is this "clustering" really? Why so many different algorithms?

Clustering Intuitions

To keep in mind and keep re-interpreting

Optimize

some sort of objetive function in such a way that we get

- "short distances within" each cluster

  (Main condition that observations within the same cluster "look alike"),

- "long distances between" clusters

  (Secondary condition that observations lying in different clusters "do not look alike").

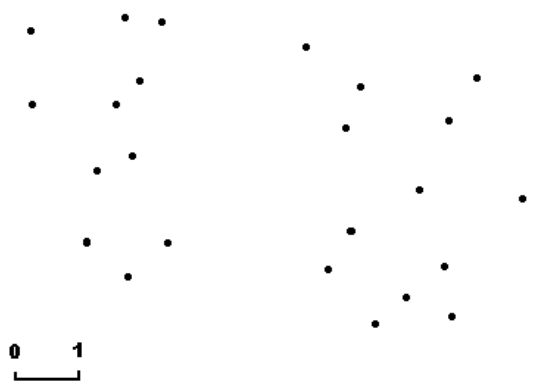
A Formal Approach

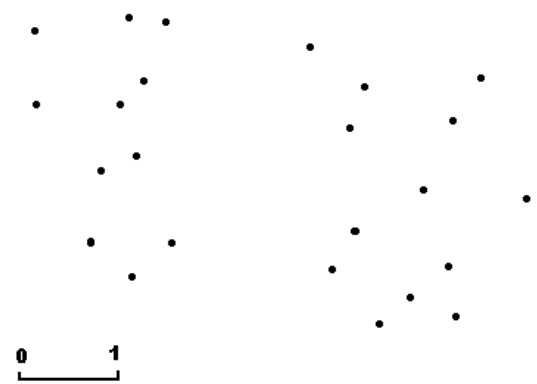
Trying to define "clustering"

Kleinberg axioms:

A very interesting proposal.

- Scale invariance: Where each observation lies matters, but not the unit length.
- Richness: No clustering is externally forbidden "a priori".
- Consistency: Reducing intra-cluster distances and/or enlarging inter-cluster distances does not change the clustering.





A Formal Approach

Trying to define "clustering"

Kleinberg axioms:

A very interesting proposal.

- Scale invariance: Where each observation lies matters, but not the unit length.
- Richness: No clustering is externally forbidden "a priori".
- Consistency: Reducing intra-cluster distance and/or enlarging inter-cluster distances does not change the clustering.

Theorem

No clustering algorithm at all can achieve all three properties.

A Formal Approach

Trying to define "clustering"

Kleinberg axioms:

A very interesting proposal.

- Scale invariance: Where each observation lies matters, but not the unit length.
- Richness: No clustering is externally forbidden "a priori".
- Consistency: Reducing intra-cluster distance and/or enlarging inter-cluster distances does not change the clustering.

Theorem

No clustering algorithm at all can achieve all three properties.

Choose your favorite target for disbelieving; it is easy now, that is "afterwards"...

What are the reasonable axioms then?

(Ben-David and others' work).

Hierarchical Agglomerative Clustering, I

Towards a simple algorithm that illustrates well the Kleinberg phenomenon

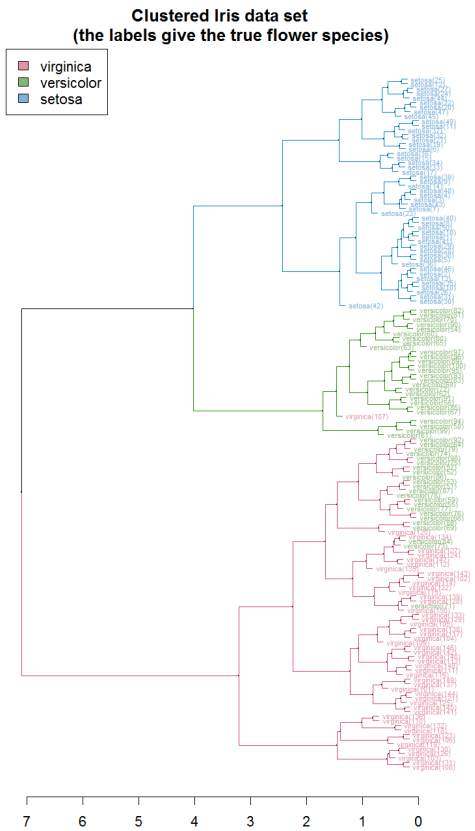
Initially: one cluster per point;

Repeat: merge two "closest" clusters;

(But... what do we mean by "closest"?)

Stop when: [...]

(many different alternatives).



(Image source: Wikipedia.)

Hierarchical Agglomerative Clustering, II

Interesting examples for Kleinberg's Axioms

Single Linkage

Initially: one cluster per point,
Repeat: merge the two clusters whose closest points are closest,
Until: stopping condition:

1. one given number of clusters is obtained;
2. the closest clusters are not "close enough" with respect to some absolute threshold;
3. the closest clusters are not "close enough" with respect to some threshold that is a fixed ratio to the largest distance between points.

We can choose to uphold any two of the Kleinberg's axioms... but will fail the third!

K-Means: Goal

Minimize the squared error

Geometry (working hypothesis):

Euclidean distance on the reals (parametric in disguise!).

- Data: n real vectors x_i, positive integer k;
- want: to split them into k clusters C_j;
- we will pick a real vector c_j representing each cluster C_j (its centroid);
- we want to minimize the average squared error:

$$\frac{1}{n} \sum_j \sum_{x_i \in C_j} d(x_i, c_j)^2$$

Note:

We do not require the c_j to be among the x_i.

K-Means: Goal

Minimize the squared error

Geometry (working hypothesis):

Euclidean distance on the reals (parametric in disguise!).

- Data: n real vectors x_i, positive integer k;
- want: to split them into k clusters C_j;
- we will pick a real vector c_j representing each cluster C_j (its centroid);
- we want to minimize the average squared error:

$$\frac{1}{n} \sum_j \sum_{x_i \in C_j} d(x_i, c_j)^2$$

Note:

We do not require the c_j to be among the x_i.

Bad news: Utterly infeasible

Complexity theorists say: *NP-hard*.

K-Means: Partial Approach

Let's think a bit more about it

If heavens would give us the centroids:

Then, constructing the clusters is easy: each point to its closest centroid, as otherwise the error increases.

K-Means: Partial Approach

Let's think a bit more about it

If heavens would give us the centroids:

Then, constructing the clusters is easy: each point to its closest centroid, as otherwise the error increases.

If heavens would give us the clusters:

Then, finding the centroids is easy: minimize $\sum_{x_i \in C} d(x_i, c)^2$, by forcing the derivative to zero; each centroid is set at the mass center of its cluster, as otherwise the error increases.

K-Means: HowTo

Stage-wise approximation

We alternate

among the two things we know how to do, starting from k initial centroid candidates:

- recompute the clusters,
- recompute the centroids,
- repeat.

Initial candidates:

- Random?
- One random, then further data points each as far as possible from the previous ones?
- Often advisable: try several runs!

`Applet by Karanveer Mohan.`

Probabilistic *K-Means*, I

Same alternating scheme, adapted procedures

Alternative k-means formulation: minimize

$$\frac{1}{n}\sum_j \sum_{x_i \in C_j} d(x_i, c_j)^2 = \frac{1}{n}\sum_j \sum_{x_i} [x_i \sim c_j]^m d(x_i, c_j)^2$$

- where $[x_i \sim c_j] \in \{0, 1\}$ represents $x_i \in C_j$, and
- each x_i is in a single C_j: thus, $\sum_j [x_i \sim c_j] = 1$ for all i.

Probabilistic *K-Means*, I

Same alternating scheme, adapted procedures

Alternative k-means formulation: minimize

$$\frac{1}{n}\sum_j \sum_{x_i \in C_j} d(x_i, c_j)^2 = \frac{1}{n}\sum_j \sum_{x_i} [x_i \sim c_j]^m d(x_i, c_j)^2$$

- where $[x_i \sim c_j] \in \{0, 1\}$ represents $x_i \in C_j$, and
- each x_i is in a single C_j: thus, $\sum_j [x_i \sim c_j] = 1$ for all i.

Now, for each data point, and for each centroid, instead of assigning the point fully to one single cluster, we maintain a probability that the point comes from the cluster:

minimize $\frac{1}{n}\sum_j \sum_{x_i} [x_i \sim c_j]^m d(x_i, c_j)^2$

where now

- $x_i \sim c_j = u_{i,j} \in [0, 1]$ instead, for a matrix $U = [u_{i,j}]$,
- with $\sum_j u_{i,j} = \sum_j [x_i \sim c_j] = 1$ for all i,
- and for some value $m \geq 1$.

Probabilistic *K-Means*, II

The *Fuzzy C-means* node in KNIME

Alternate:

- recompute the centroids:
  - They are still the means of the points;
  - instead of the plain average of the points that belong to the cluster (former condition $[x_i \sim c_j] = 1$),
  - they are the weighted average of all the points (hence, if $[x_i \sim c_j] = 0$ the point does not contribute at all, as before).
  - The weights are $u_{i,j}^m$ and we must remember to renormalize (dividing by the sum of all the weights);
- recompute the "clusters" or, rather now, the weights

  (formula omitted here, check it out at `https://en.wikipedia.org/wiki/Fuzzy_clustering`);
- repeat.

Most often $m \in [1, 30]$, defaulting to 2.
The lower the m, the "crisper" the clustering.

Probabilistic *K-Means*, III

The Conceptual Essence of *Expectation-Maximization*

Bias: weighted sum ("mixture") of Gaussians:

`Example`, `example`, `example`; $Pr(x_i) = \sum_j w_j \mathcal{N}_{\mu_j,\sigma_j}(x_i)$

Probabilistic *K-Means*, III

The Conceptual Essence of *Expectation-Maximization*

Bias: weighted sum ("mixture") of Gaussians:

`Example, example, example`; $Pr(x_i) = \sum_j w_j \mathcal{N}_{\mu_j,\sigma_j}(x_i)$

- Centroids: the Gaussians' means.
- Each data point assumed to come from one of them,
- but we don't know which one!

Probabilistic *K-Means*, III

The Conceptual Essence of *Expectation-Maximization*

Bias: weighted sum ("mixture") of Gaussians:

`Example, example, example`; $Pr(x_i) = \sum_j w_j \mathcal{N}_{\mu_j,\sigma_j}(x_i)$

- Centroids: the Gaussians' means.
- Each data point assumed to come from one of them,
- but we don't know which one!
- For each data point, for each centroid, we maintain a probability that the point comes from the Gaussian centered at that centroid.
- That probability depends on the parameters of the corresponding Gaussian and on its weight in the mixture.

`https://lovasoa.github.io/expectation-maximization/dist`

Probabilistic *K-Means*, IV

Suspiciously familiar alternation

What parameters are at work?

- Mean and standard deviation of each Gaussian, and
- weight with which each of them participates in the mixture.

How to handle them?

Probabilistic *K-Means*, IV

Suspiciously familiar alternation

What parameters are at work?

- Mean and standard deviation of each Gaussian, and
- weight with which each of them participates in the mixture.

How to handle them?

- If we knew the weights in the mixture, and the parameters of each Gaussian, it would be easy to assign to each point the probability contribution of each Gaussian.

Probabilistic *K-Means*, IV

Suspiciously familiar alternation

What parameters are at work?

- Mean and standard deviation of each Gaussian, and
- weight with which each of them participates in the mixture.

How to handle them?

- If we knew the weights in the mixture, and the parameters of each Gaussian, it would be easy to assign to each point the probability contribution of each Gaussian.
- If we knew the true source Gaussian of each point, with its corresponding probability, we would be able to estimate the weights and parameters.

Probabilistic *K-Means*, IV

Suspiciously familiar alternation

What parameters are at work?

- Mean and standard deviation of each Gaussian, and
- weight with which each of them participates in the mixture.

How to handle them?

- If we knew the weights in the mixture, and the parameters of each Gaussian, it would be easy to assign to each point the probability contribution of each Gaussian.
- If we knew the true source Gaussian of each point, with its corresponding probability, we would be able to estimate the weights and parameters.

(Sounds familiar?)

A Simple Example

Combination of two Gaussians

```
from random import gauss, random
for i in range(20):
    if random() < 0.333:
        print "%2.4f" % gauss(1,0.8)
    else:
        print "%2.3f" % gauss(-1,0.6)
```

```
-1.150      -1.410       1.0845     -1.451       1.5767
-2.105      -0.339      -2.888       0.248      -1.365
-0.974      -0.019       1.2649     -1.038      -1.0894
 1.7531     -1.570      -1.840       0.205      -1.713
```

$p = 0.333$, $m_0 = 1$, $d_0 = 0.8$, $m_1 = -1$, $d_1 = 0.6$.

A Simple Example

Combination of two Gaussians

```
from random import gauss, random
for i in range(20):
    if random() < 0.333:
        print "%2.4f" % gauss(1,0.8)
    else:
        print "%2.3f" % gauss(-1,0.6)
```

```
-1.150    -1.410     1.0845    -1.451     1.5767
-2.105    -0.339    -2.888      0.248    -1.365
-0.974    -0.019     1.2649    -1.038    -1.0894
 1.7531   -1.570    -1.840      0.205    -1.713
```

$p = 0.333$, $m_0 = 1$, $d_0 = 0.8$, $m_1 = -1$, $d_1 = 0.6$.

Latent variable hacked in as the amount of decimal places!

Approximations: $p = 0.25$, $m_0 = 0.9176$, $m_1 = -1.1606\ldots$

Expectation-Maximization for Gaussian mixtures

Once more a K-Means-like loop

Maximization:

Given a probability for each point and each Gaussian, find the parameters that best explain these probabilities: mean, standard deviation, and weight of each distribution.

Expectation:

Given the parameters of all the Gaussians, and the weight with which each of them participates in the mixture, compute a probability for each point and each Gaussian: it "approximates the true source".

Start with random parameters, and... iterate!

Many loose ends omitted here!

Higher dimensions, covariance matrices for non-spherical Gaussians... Similar schemes apply in other contexts (e.g. estimating transition probabilities of Hidden Markov Models).

Hierarchical Agglomerative Clustering, III

Hints at main variants; divisive variants not covered

Recall:

Initially:	one cluster per point;
Repeat:	merge two "closest" clusters;
Stop when:	[. . .]

Options for "closest":

Single linkage:	closest points are closest (already discussed);
Complete linkage:	farthest points are closest;
UPGMA, WPGMA:	(differently weighted) averages are closest;
Ward method:	(coming. . .)

Many of these methods are $O(n^3)$ if implemented naïvely; most can be reduced to $O(n^2)$ or $O(n^2 \log n)$ using smart data structures and algorithms.

Hierarchical Agglomerative Clustering, IV

`ward.cluster` function in R

Ward method:

Merge clusters so as to greedily minimize the "k-means cost increase" incurred upon merging.

- Keep centroids for constant-time distance evaluation.
- Many equivalent formulations and useful algorithmic approaches: Lance-Williams algorithm, nearest-neighbor-chain algorithm...
- Many generalizations: optimize some other specific, maybe application-oriented objective function...
- Common statement: "Ward method is oriented to decrease variance", intuitively correct (but some claim that it is formally incorrect, cf. a different variant, Podany's MNVAR).

`https://en.wikipedia.org/wiki/Hierarchical_clustering`
`https://en.wikipedia.org/wiki/Nearest-neighbor_chain_algorithm`

Multi-Dimensional, Global-Optimum K-Means, I

NP-hard for almost all nontrivial cases: $k \geq 2$ or $d \geq 2$

- Most simple strategies based on exhaustive search:

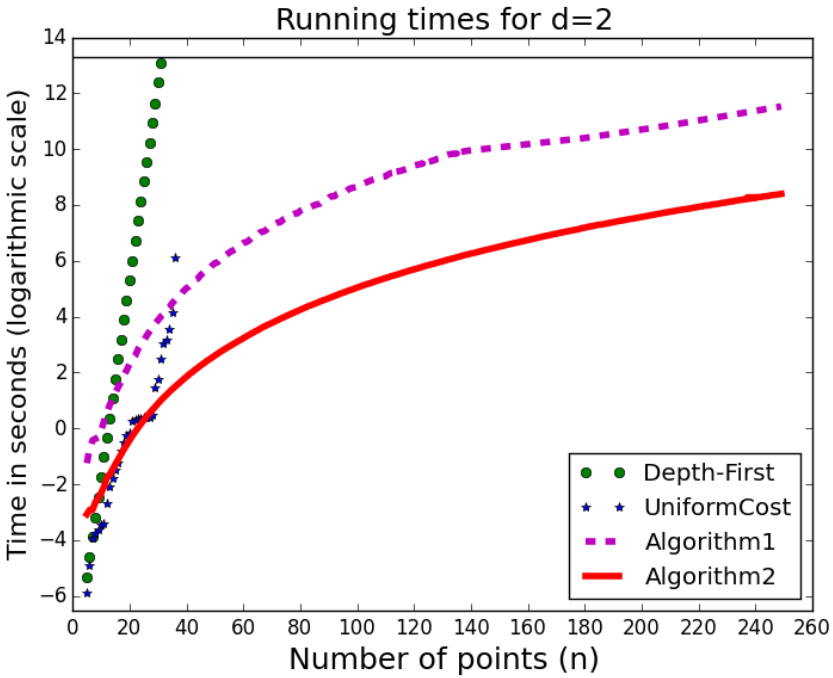
  (including dynamic programming approaches):

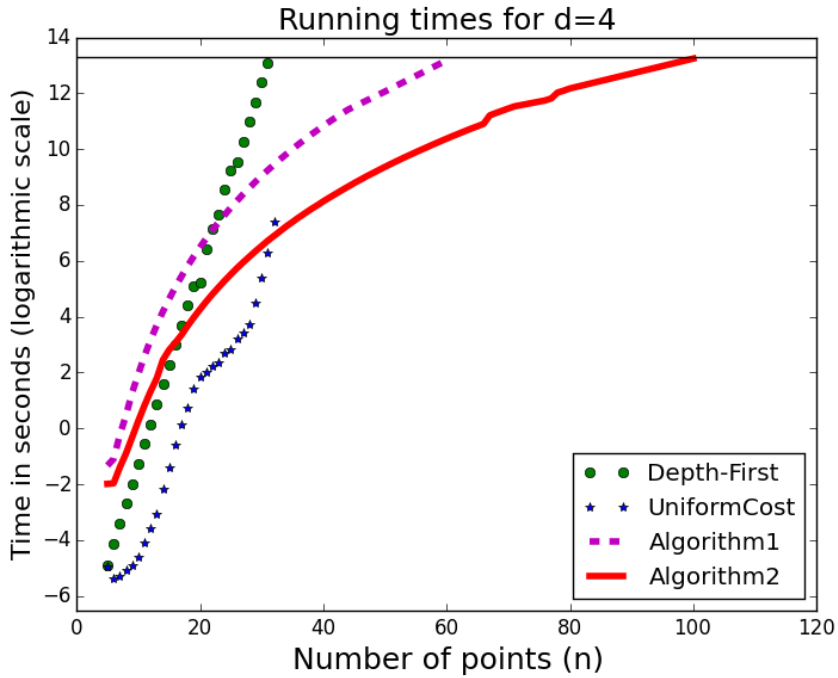
  exponential in the number of points n;
- alternatives that run in time polynomial in n exist!
- Traverse all potential clusterings by a reduction to traversing hyperplane intersection cells and applying computational geometry algorithms;
  - may reach time $O(n^{kd+1})$ or, alternatively,
  - $O(n^{1+k+(k-1)d})$ (better if $k < d$)

  for k clusters of n points in d-dimensional space;
- goes down to $O(n^{d+2})$ for $k = 2$;
- low constants in the $O(\cdot)$!
- Highly parallelizable!

Multi-Dimensional, Global-Optimum K-Means, II

Runtime comparison with Russell-Norvig schemes, $k = 2$





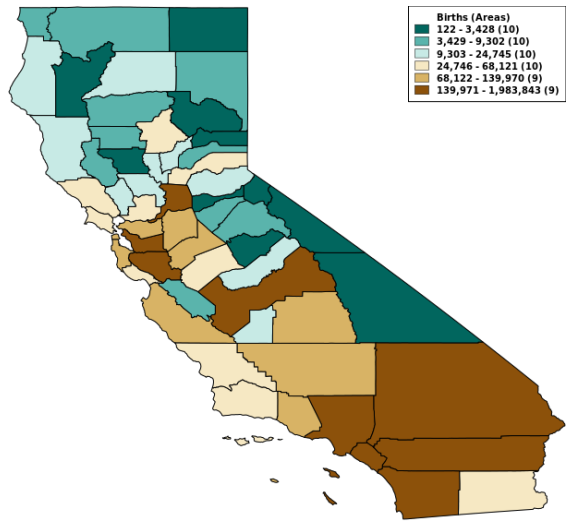
Unsupervised Discretization

Or: one-dimensional clustering

Given list of floats, organize them into just a few "bins" (or "buckets", or "clusters"...)

Separate case of "supervised discretization", not covered here.

Parallel research in so-called `choropleth maps`, a branch of Cartography, where the solution we describe here goes by the name of `Jenks' natural breaks`.



(Source: `Expert Health Data Programming, Inc (EHDP): Vitalnet`)

One-Dimensional, Global-Optimum K-Means, I

Dynamic programming for clustering float values: the `Wang and Song` strategy

Input: desired number of clusters k, and $n \geq k$ floats, x_1 to x_n, assumed given in increasing order (otherwise, do a sort first).

One-Dimensional, Global-Optimum K-Means, I

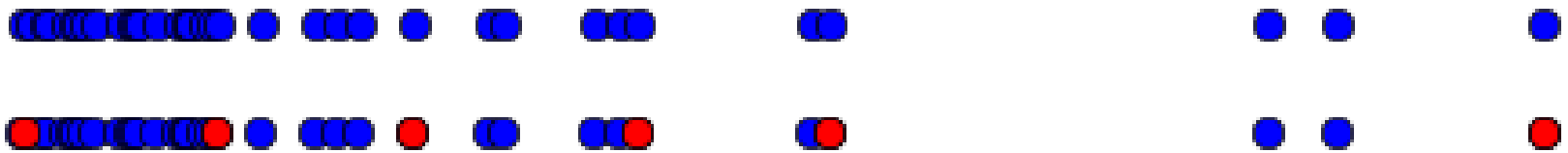
Dynamic programming for clustering float values: the `Wang and Song` strategy

Input: desired number of clusters k, and $n \geq k$ floats, x_1 to x_n, assumed given in increasing order (otherwise, do a sort first).

One-Dimensional, Global-Optimum K-Means, I

Dynamic programming for clustering float values: the `Wang and Song` strategy

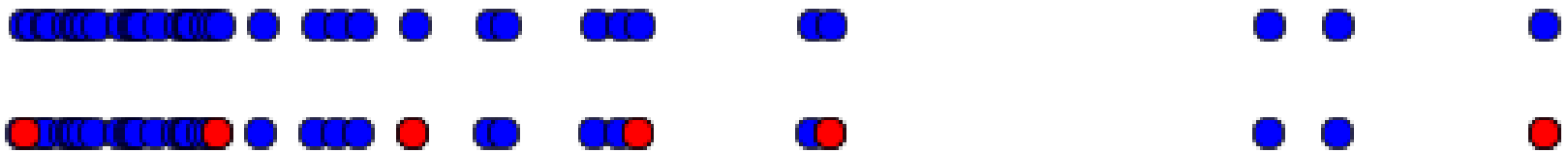
Input: desired number of clusters k, and $n \geq k$ floats, x_1 to x_n, assumed given in increasing order (otherwise, do a sort first).



One-Dimensional, Global-Optimum K-Means, I

Dynamic programming for clustering float values: the `Wang and Song` strategy

Input: desired number of clusters k, and $n \geq k$ floats, x_1 to x_n, assumed given in increasing order (otherwise, do a sort first).

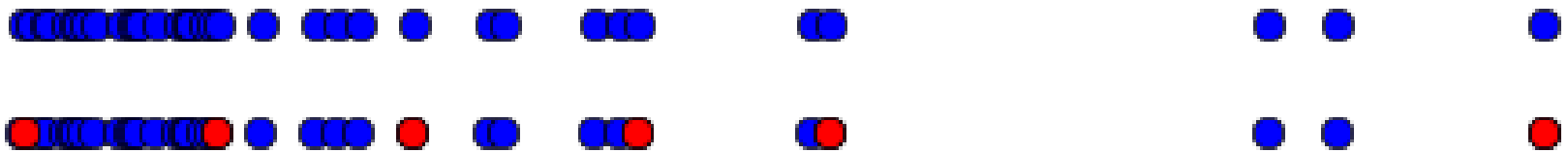


Tabulate: $C[i, m]$, cost of a clustering of x_1 to x_i into m clusters, for $m \leq k$ and $m \leq i$; solution given by $C[n, k]$.

One-Dimensional, Global-Optimum K-Means, I

Dynamic programming for clustering float values: the `Wang and Song` strategy

Input: desired number of clusters k, and $n \geq k$ floats, x_1 to x_n, assumed given in increasing order (otherwise, do a sort first).



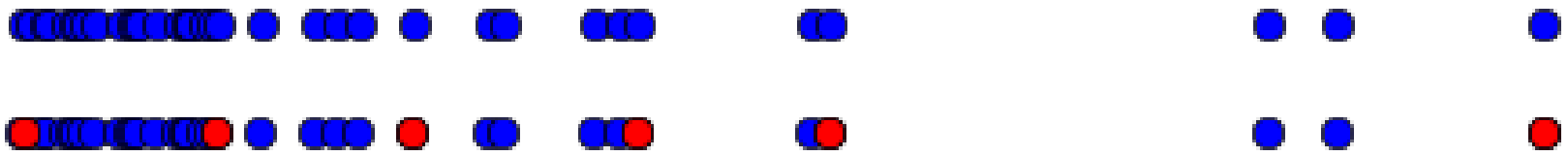
Tabulate: $C[i, m]$, cost of a clustering of x_1 to x_i into m clusters, for $m \leq k$ and $m \leq i$; solution given by $C[n, k]$.

Initialization: $C[i, m] = 0$ if $m = 0$.

One-Dimensional, Global-Optimum K-Means, I

Dynamic programming for clustering float values: the `Wang and Song` strategy

Input: desired number of clusters k, and $n \geq k$ floats, x_1 to x_n, assumed given in increasing order (otherwise, do a sort first).



Tabulate: $C[i, m]$, cost of a clustering of x_1 to x_i into m clusters, for $m \leq k$ and $m \leq i$; solution given by $C[n, k]$.

Initialization: $C[i, m] = 0$ if $m = 0$.

Adding a new point:

Can be related to some solution with "one cluster less" by identifying x_j, the smallest point in the last (m-th) cluster containing the new point.

One-dimensional, Global-Optimum K-Means, II

Demo available

A visual demo of the process for each new point considered has been set up at:

`http://www.cs.upc.edu/~balqui/demoWSJ/`

Alpha stage!

- aesthetics fully postponed to later versions,
- usability at minimal levels...

Needs:

- the number of clusters,
- the points handled so far up to one specific pass,
- and the newcomer point,

Then, shows the computations made in order to account for the new point.

One-dimensional, Global-Optimum K-Means, III

Difference between m clusters and $m-1$ clusters

For appropriately identified values

namely a lower limit h and a candidate to centroid of the m-th cluster $c_{j,i}$,

$$C[i,m] = \min_{h \leq j \leq i}(C[j-1,m-1] + \sum_{j \leq \ell \leq i} d(x_\ell, c_{j,i})^2)$$

One-dimensional, Global-Optimum K-Means, III

Difference between m clusters and $m-1$ clusters

For appropriately identified values

namely a lower limit h and a candidate to centroid of the m-th cluster $c_{j,i}$,

$$C[i,m] = \min_{h \le j \le i}(C[j-1,m-1] + \sum_{j \le \ell \le i} d(x_\ell, c_{j,i})^2)$$

where

$$c_{j,i} = \frac{1}{i-j+1} \sum_{j \le \ell \le i} x_\ell \text{ and}$$
$$h = m.$$

For better understanding, find and run the existing R routine.

One-dimensional, Global-Optimum K-Means, IV

We can do it faster

Strategy leads to an $O(n^3)$ algorithm.

Acceleration: don't compute every $c_{j,i}$ individually but, instead, update $c_{j,i-1}$ to find it.

This spares a linear computation and reduces the cost to $O(n^2)$.

(Jenks' alternative: in Cartography you only need the cutpoints, not the centroids; work out an alternative formula by replacing the centroid by its definition in the minimization scheme.)

Autonomous Learning Topics, V

Proposals to explore on yourself

13. Choose a couple of clustering algorithms to study on your own.
14. Learn about the proof of Kleinberg's Theorem of Impossibility of Clustering — I believe it is really within reach for most of you.
15. Learn about subsequent work along these lines.
16. How to select the number of clusters? There are quite a few proposals (like "knees" and "elbows"), not many of which are principled approaches. Find out the curious solution in R.
17. The x-means algorithm tries to figure out on itself the best value of k. Find out about it.

Autonomous Learning Topics, VI

Proposals to explore on yourself

18. Find out and study in the literature the formulas that we have omitted in our description of Fuzzy C-Means.
19. Compare on KNIME the outcome of Fuzzy C-Means with other available clusterers on your favorite dataset; explore also the effect of the "crispness" parameter.
20. Complete on yourself the procedure to apply EM to the case of two independent one-dimensional Gaussians, as given in the example (good warm-up for the next).
21. Keep analyzing EM: move first to more than two Gaussians, then to multidimensional, spherical ones, then make room for covariance matrices so that they are not spherical anymore...
22. Explore the literature (or Wikipedia) to find out how the EM scheme allows you to infer internal parameters of HMM's (application known as Baum-Welch algorithm, particularly interesting if you happen to be a Viterbi fan).

Autonomous Learning Topics, VII

Proposals to explore on yourself

23. Learn more about hierarchical clustering, the various linkage criteria, and the smart algorithmics behind Ward's method and related ones.
24. Explore the internal workings of globally optimal k-means algorithms.
25. Study Wang and Song, study Jenks, compare them, ask the instructor for a Python implementation where the old Fortran code can be transparently seen, mindlessly translated.
26. Learn further about supervised and/or unsupervised discretization.
27. In semi-supervised clustering, data comes with explicit indications of pairs that must be / must not be in the same cluster. Explore the basics of this topic on your own.

7. Process Mining

8. Pattern Mining and Association Rules

Transactional Data, II

Usual context for pattern mining

Each observation is seen as a data structure on itself.

On the basis of a set of atomic items:

- Simplest (and most common) case: each observation is a set.
- Slight sophistication: multiplicity is relevant (but is likely to need adjustements; analogy: tfidf-like weights...).
- Further sophistications: sequences, rankings, graphs...
- Patterns based often on the same data structure concept (subsets, subsequences, subgraphs...)

Global patterns versus local patterns:

- Patterns are associated to parts of the dataset.
- Compare: predictors are expected to work on the whole of it (and some do not extract any pattern at all).

Transactional Data, III

A real-life example

Logs from a virtual learning platform

Transactions on items:

- one for each "area" of the course.
  - `announcements`, `assessments`, `assignments`, `contents`, `forum`, `organizer`, ...

Transactional Data, III

A real-life example

Logs from a virtual learning platform

Transactions on items:

- one for each "area" of the course.
  - `announcements`, `assessments`, `assignments`, `contents`, `forum`, `organizer`, ...

- Student's sessions are logged:
- for each session, we know whether each "area" was visited in that session;

Transactional Data, III

A real-life example

Logs from a virtual learning platform

Transactions on items:

one for each "area" of the course.

`announcements`, `assessments`, `assignments`, `contents`, `forum`, `organizer`, ...

- Student's sessions are logged:
- for each session, we know whether each "area" was visited in that session;
- therefore each session is a transaction, of type set
- or, equivalently, a propositional model, whereby items are seen as propositional variables.

Computational Difficulty

Exponential menaces lurk at both sides

Two powersets! (At least...)

Even on the simplest set-based transactional data, and for sets as patterns,

- $2^{number\ of\ items}$ potential patterns:

  Boolean hypercube with one dimension per item!
- $2^{number\ of\ transactions}$ potential subsets of the dataset to allow for potentially interesting description.

Options:

- Small enough dataset size and dimensionality.
- Up front, give up ample parts of the pattern space.
  - Most common choice: frequent sets.
  - Come up with yours...

Frequent Set Mining

"Beer and diapers"

Sets that appear often enough as subsets of transactions.

Given a transactional dataset,

- a support threshold is fixed,
- and we are to find all sets of items whose support is (at or) above the threshold.

Frequent Set Mining

"Beer and diapers"

Sets that appear often enough as subsets of transactions.

Given a transactional dataset,

- a support threshold is fixed,
- and we are to find all sets of items whose support is (at or) above the threshold.

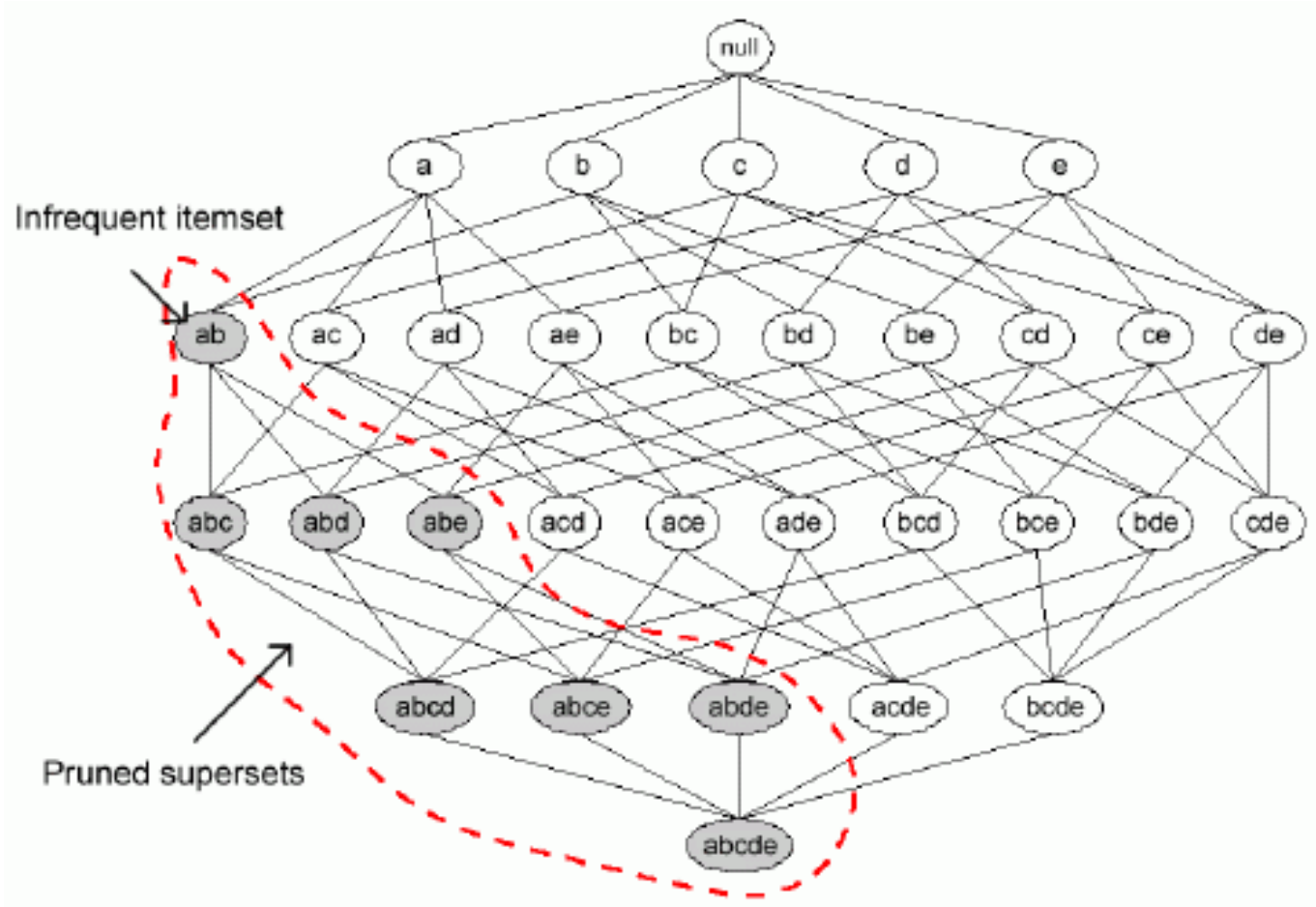
Simple but key idea: supersets of infrequent sets must necessarily be infrequent as well (antimonotonicity).

This allows us to avoid exploration of ample areas of the search space.

However, whether the exponential risk is actually avoided depends on setting correctly the threshold!

Search Space: Powerset of Items

Illustrating antimonotonicity and joins



Source: `http://www.inf.ed.ac.uk/undergraduate/`
`/projects/mathiasengvall/`

Implications

A very close relative of Definite Horn Clauses

Examples from an "e-learning" dataset

$$\texttt{announcements} \wedge \texttt{assignments} \implies \texttt{assessments} \wedge \texttt{organizer}$$

Examples from a "Machine Learning abstracts" dataset

$$\texttt{descent} \implies \texttt{gradient}$$

$$\texttt{hilbert} \implies \texttt{space}$$

$$\texttt{margin support} \implies \texttt{vector}$$

Implications

A very close relative of Definite Horn Clauses

Examples from an "e-learning" dataset

$$\texttt{announcements} \wedge \texttt{assignments} \implies \texttt{assessments} \wedge \texttt{organizer}$$

Examples from a "Machine Learning abstracts" dataset

$$\texttt{descent} \implies \texttt{gradient}$$
$$\texttt{hilbert} \implies \texttt{space}$$
$$\texttt{margin support} \implies \texttt{vector}$$
$$\texttt{carlo} \implies \texttt{monte}$$
$$\texttt{monte} \implies \texttt{carlo}$$

Implications

A very close relative of Definite Horn Clauses

Examples from an "e-learning" dataset

$$\texttt{announcements} \wedge \texttt{assignments} \implies \texttt{assessments} \wedge \texttt{organizer}$$

Examples from a "Machine Learning abstracts" dataset

$$\texttt{descent} \implies \texttt{gradient}$$
$$\texttt{hilbert} \implies \texttt{space}$$
$$\texttt{margin support} \implies \texttt{vector}$$
$$\texttt{carlo} \implies \texttt{monte}$$
$$\texttt{monte} \implies \texttt{carlo}$$

Example from a "census" dataset

$$\texttt{Exec-managerial Husband} \implies \texttt{Married-civ-spouse}$$

Towards Standard Association Rules, I

Confidence-based framework

"Census" dataset:

- Husband $\Longrightarrow$ Male

Towards Standard Association Rules, I

Confidence-based framework

"Census" dataset:

- Husband $\Longrightarrow$ Male... does not hold! (see tuple 7110).

Towards Standard Association Rules, I

Confidence-based framework

"Census" dataset:

- Husband $\Longrightarrow$ Male... does not hold! (see tuple 7110).
- Similarly, Wife $\Longrightarrow$ Female does not hold either: there are two tuples declaring Male and Wife.
- Relaxing implications into partial implications is mandatory.

Towards Standard Association Rules, I

Confidence-based framework

"Census" dataset:

- Husband $\Longrightarrow$ Male... does not hold! (see tuple 7110).
- Similarly, Wife $\Longrightarrow$ Female does not hold either: there are two tuples declaring Male and Wife.
- Relaxing implications into partial implications is mandatory.
- Consequence: over sixty full-confidence rules of the form Husband, SomethingElse $\Longrightarrow$ Male.
- Some appropriate notion of "redundancy" is also necessary.

Towards Standard Association Rules, II

A relaxed notion of "correctness"

There are reasons to be satisfied with an implication even in the presence of counterexamples.

- Transmission or keying errors;
- mistakes in filling up forms;
- mixed populations;
- ...

Partial "approximate" implications that allow for exceptions...

Semantics nontrivial to define!

Most natural option: confidence.

Confidence

Pros and cons

In favor:

- Encompasses frequent sets:
  - A has support over τ if and only if $\emptyset \rightarrow A$ has support and confidence over τ.
- Quite natural.
- Easy to explain to an educated user.

Confidence

Pros and cons

In favor:

- Encompasses frequent sets:
  - A has support over τ if and only if $\emptyset \rightarrow A$ has support and confidence over τ.
- Quite natural.
- Easy to explain to an educated user.

BUT: Handle with care!

- High confidence is compatible with negative correlation.
- Normalization (*lift*) solves the problem but introduces another one: symmetry.

Towards Standard Association Rules, III

Basic setup

Standard Association Mining Process:

1. Read in the dataset;
2. choose your sort of output:
   - association rules,
   - frequent sets,
   - ...
3. set a standard support (say, 10%);
4. for rules, set a standard confidence (say, 80%);
5. run the associator...

Towards Standard Association Rules, III

Basic setup

Standard Association Mining Process:

1. Read in the dataset;
2. choose your sort of output:
   - association rules,
   - frequent sets,
   - ...
3. set a standard support (say, 10%);
4. for rules, set a standard confidence (say, 80%);
5. run the associator...

   Then several things may happen.
   - Hopefully the associator swiftly provides you a few dozen association rules, nearly all of them interesting, and each of them interesting due to different reasons.

Towards Standard Association Rules, III

Basic setup

Standard Association Mining Process:

1. Read in the dataset;
2. choose your sort of output:
   - association rules,
   - frequent sets,
   - ...
3. set a standard support (say, 10%);
4. for rules, set a standard confidence (say, 80%);
5. run the associator...

   Then several things may happen.
   - Hopefully the associator swiftly provides you a few dozen association rules, nearly all of them interesting, and each of them interesting due to different reasons.
   - Unfortunately, this hardly ever happens... if at all.

Towards Standard Association Rules, IV

Threshold fiddling

Usual course of events:

- The associator takes "forever"... exhausts your patience.
  - Kill it,
  - try again with a more strict support threshold.
- It finishes relatively quickly, but with no output or just a handful of uninteresting rules.
  - Try again with more benign support or confidence thresholds.
- It finishes before exhausting your patience, but the output consists of dozens of thousands of association rules.
  - Try again with more strict support or confidence thresholds.

At some point, the quantity of rules is appropriate for human examination...

Towards Standard Association Rules, IV

Threshold fiddling

Usual course of events:

- The associator takes "forever"... exhausts your patience.
  - Kill it,
  - try again with a more strict support threshold.
- It finishes relatively quickly, but with no output or just a handful of uninteresting rules.
  - Try again with more benign support or confidence thresholds.
- It finishes before exhausting your patience, but the output consists of dozens of thousands of association rules.
  - Try again with more strict support or confidence thresholds.

At some point, the quantity of rules is appropriate for human examination... but their interest turns out to be limited.

Towards Standard Association Rules, V

From threshold fiddling to implicational logic

The problem: Automatically selecting a set of association rules of the right size, most of them interesting due to varied reasons

Towards Standard Association Rules, V

From threshold fiddling to implicational logic

The problem: Automatically selecting a set of association rules of the right size, most of them interesting due to varied reasons — and avoiding the issue that high confidence and support do not prevent negative correlation.

Towards Standard Association Rules, V

From threshold fiddling to implicational logic

The problem: Automatically selecting a set of association rules of the right size, most of them interesting due to varied reasons — and avoiding the issue that high confidence and support do not prevent negative correlation.

Alternative "rule quality measures": a hot research topic!

- How to measure the quality in order to select good rules?
- How to set the threshold?

Redundancy?

Three levels:

- redundancy due to the dataset proper;
- redundancy due to the association rule structure (logic-based approach);
- several forms of combined redundancy.

Frequent Closed Sets, I

Dataset-based redundancy

Closure of set X:

Largest set Y that includes X and has the same support.

That is: all transactions that include X, include Y as well.

Frequent, non-closed sets are redundant.

They can be recovered, with their frequencies, from the closed sets: the family of closed sets is sufficient.

(But it might be more expensive to compute!)

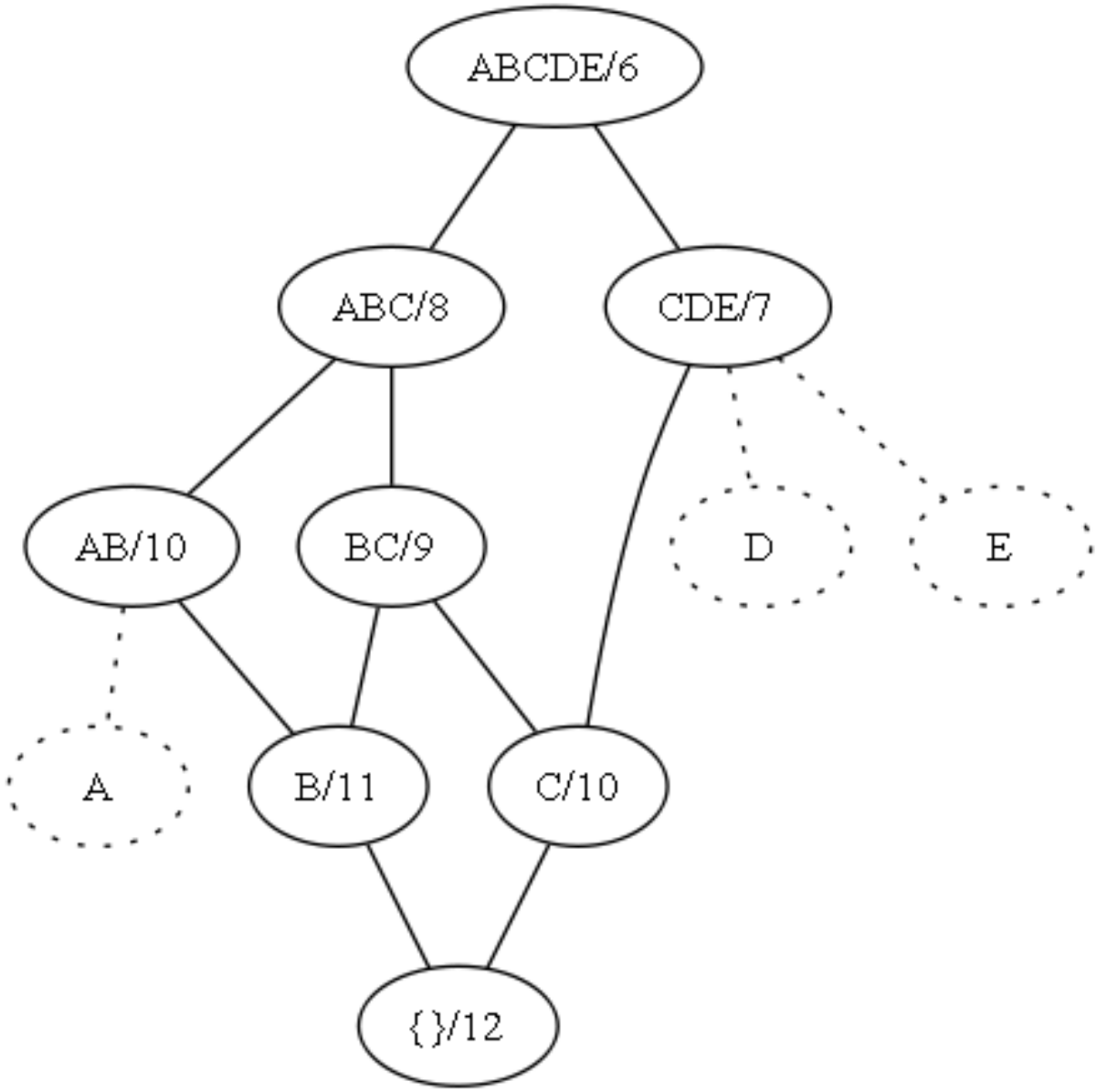
The closures form a lattice

(the intersection of closed sets is again a closed set).

Frequent Closed Sets, II

An example

ABCDE (x6),
ABC (x2),
AB (x2),
CDE (x1),
BC (x1)



Associators in the Wild

Where to find them

- Christian Borgelt's Associators:
  1. `borgelt.net`, tab "Software"; also the alternative algorithms eclat and fpgrowth are available there.
  2. Can be run standalone, or through FrIDA or PyFIM there.
  3. KNIME has an extension with these implementations.
- Many other people offer Frequent Itemsets algorithms and implementations freely on the Web:
  - `Cristofor version` at UMass Boston;
  - `Goethals' ADReM group` at Antwerp
    (even includes a tweetable Python3 version; I keep `a copy here` and `an example of how to call it`: assign the absolute support threshold value to int *m* before calling).

  Try them on
  - `The FIMI datasets` or
  - the relational datasets encoded in transactional compatible form that we used in our first labs:
    `http://www.cs.upc.edu/~balqui/LabADM20200226.zip`

Nonstandard Associators

All these were standard support-and-confidence associators

Standard associators:

Compute frequent sets (support filter), obtain rules, apply a confidence filter, possibly other filters.

Other associators:

- Restrict to closed frequent sets,
- try to avoid redundancy among associations
  - filter by leverage: `Magnum Opus`
  - filter by confidence boost: `yacaree`
  - ...
- try to compute along the way appropriate thresholds...

Algorithmics, I

Search space: powerset of items

Extract frequent sets by either:

- breadth-first search: apriori,
- depth-first search: eclat,
- variants or other schemes, up to many dozens in number.

Essence of the idea of the apriori algorithm:

count the frequency of all items
    (and discard infrequent ones)
for growing sizes $c = 1, 2, 3, \ldots$
    use a join operation on frequent sets of size c
        to construct candidates of size $c + 1$
    check for infrequent subsets of size c, discard if found
    count (at once) the support of the remaining candidates
        and discard infrequent ones
    exit when no candidate remains

Algorithmics, II

Candidate generation

Basic join algorithm for moving on to the next level:

- Identify pairs of frequent sets of size c that only differ in their largest item

  (e.g. $\{2, 3, 5, 7\}$ and $\{2, 3, 5, 11\}$);
- join them into a candidate set of size $c + 1$

  (e.g. $\{2, 3, 5, 7, 11\}$);
- the other $c - 1$ subsets of size c are still to be checked.

Smart data structures ("hash trees") are set up to accelerate this sort of operation.

Key idea is to spare passes through the whole data, assumed in disk and too large for core.

The Logic of Implications

To find inspiration

Given all the implications that hold for a dataset,

- some of them may be redundant (logically entailed by other implications);
- there is an efficient, sound and complete criterion to identify cases of redundancy;
- we can obtain an irredundant basis by removing redundant implications;
- but there may be various ways to reach irredundant bases,
- and they may be of very different sizes.

The Guigues-Duquenne basis

- is a canonical, minimum-size basis for implications;
- equivalent notion exists in functional dependencies;
- there is no equivalent concept for Horn clause syntax.

Redundancy in Association Rules, I

A Logic-based view

Structural reasons of redundancy

For a fixed γ, user-chosen, we focus on statements of the form $\mathrm{conf}(X \to Y) \geq \gamma$.

- Plain redundancy: $X \to Y$ is redundant with respect to $X' \to Y'$ if $\mathrm{conf}(X \to Y) \geq \mathrm{conf}(X' \to Y')$ in every dataset.
- A natural variant, closure-based redundancy, reads the same, but under a condition to share the same closure space

  (gives better summarization by handling full implications separately).

Leads to an efficient, sound and complete criterion to identify redundant rules.

Redundancy in Association Rules, II

Minimum-size bases

Basic antecedents: assume $X \subseteq Y$

Essentially (and omitting some details, like whether redundancy is plain or closure-based), X is a basic antecedent of Y if:

- confidence of $X \to Y$ is at least γ;
- but falls below γ if either we enlarge Y, or we reduce X.

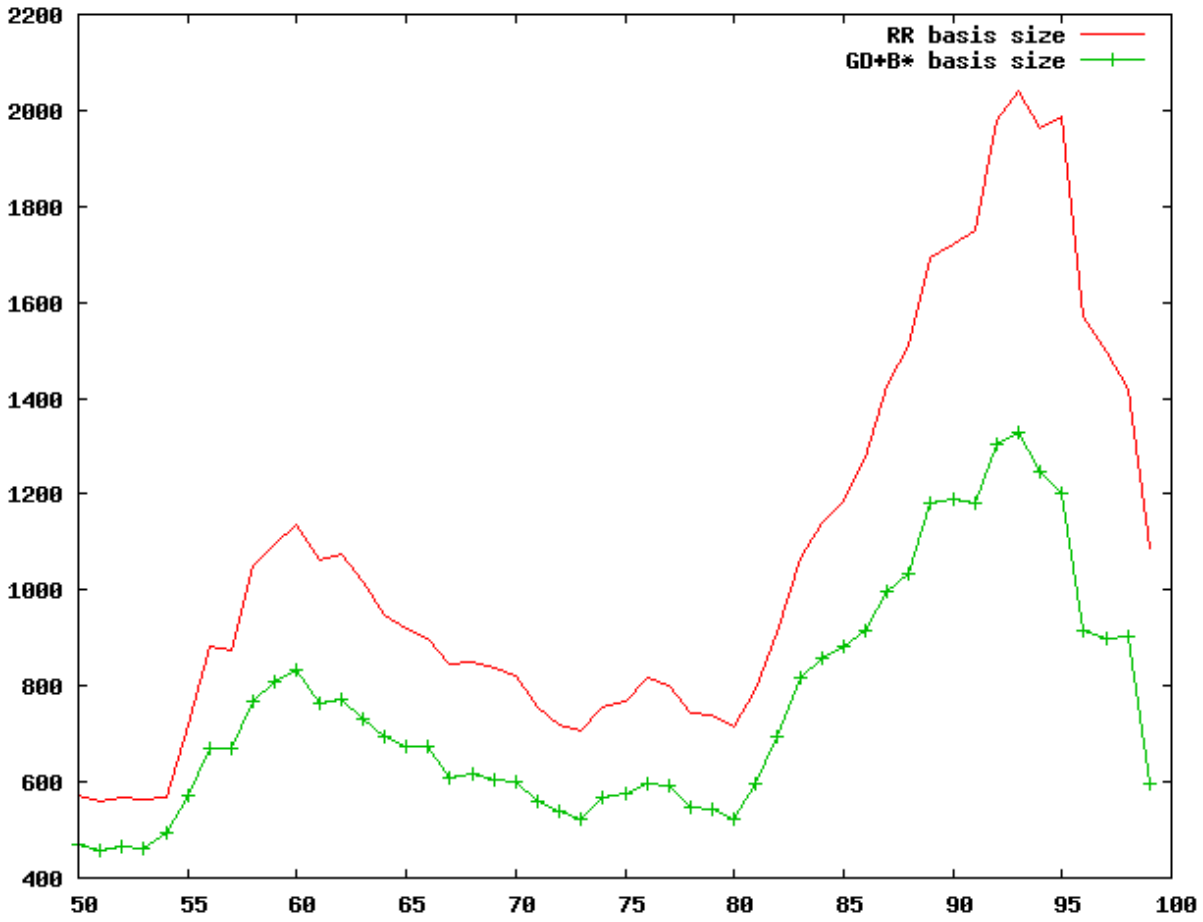
Basis: $X \to Y - X$ for closed Y and basic antecedent X of Y

Facts

1. These rules hold with confidence γ,
2. All the rules that hold with confidence γ can be inferred from these rules,
3. Any alternative set of rules with the same properties has at least as many rules as this one.

Irredundant Rules for FIMI Dataset pumsb-star

An example of basis size with respect to plain and to closure-based redundancies



Quantifying absolute novelty

Intuition of the "confidence boost"

For a given support and confidence thresholds

Assume we have run an "association miner":

- Discard redundant rules: we are left just with the basis.

  (Discarded rules are entailed by the basis:

  each rule left, say R, is not entailed by the others.

  This means that the other rules would not suggest that R passes the confidence threshold, say γ.)

Quantifying absolute novelty

Intuition of the "confidence boost"

For a given support and confidence thresholds

Assume we have run an "association miner":

- Discard redundant rules: we are left just with the basis.

  (Discarded rules are entailed by the basis:

  each rule left, say R, is not entailed by the others.

  This means that the other rules would not suggest that R passes the confidence threshold, say γ.)
- But maybe R becomes redundant at a lower confidence!

Quantifying absolute novelty

Intuition of the "confidence boost"

For a given support and confidence thresholds

Assume we have run an "association miner":

- Discard redundant rules: we are left just with the basis.

  (Discarded rules are entailed by the basis:

  each rule left, say R, is not entailed by the others.

  This means that the other rules would not suggest that R passes the confidence threshold, say γ.)
- But maybe R becomes redundant at a lower confidence!
- Let γ' be the tightest confidence at which R is redundant, and let's consider the quotient γ/γ'.

Low Novelty

Novel, but barely

Suppose:

- The confidence of R is γ.
- Other rules of confidence γ do not entail it.
- Thus, it is irredundant with respect to the rest of the rules found at confidence γ.
- But, if we had run the process at a confidence slightly lower, say $\gamma' < \gamma$, maybe some R' would have been found that entails R.

R only belongs to the basis during the short interval $(\gamma', \gamma]$ of values for the confidence threshold; γ/γ' is low.

At its own confidence, it is novel, but really not too much.

Low Novelty

Novel, but barely

Suppose:

- The confidence of R is γ.
- Other rules of confidence γ do not entail it.
- Thus, it is irredundant with respect to the rest of the rules found at confidence γ.
- But, if we had run the process at a confidence slightly lower, say $\gamma' < \gamma$, maybe some R' would have been found that entails R.

R only belongs to the basis during the short interval $(\gamma', \gamma]$ of values for the confidence threshold; γ/γ' is low.

At its own confidence, it is novel, but really not too much.

Conversely, if any such γ' is considerably lower, R states novel information: we only can make it redundant with rules of much lower confidence, and γ/γ' is high.

Autonomous Learning Topics, VII

Proposals to explore on yourself

28. Explore the rich repertory of rule quality measures and their properties; Geng and Hamilton 2006, "Interestingness measures for data mining: A survey" (ACM Comput. Surv. 38, 3) is a good starting point.
29. Learn more about entailment, calculus, and axiomatization of Horn clauses and their analogy to functional dependencies (searching "Armstrong axioms" on Wikipedia is a good starting point).
30. Find out about the details of the definitions of basic antecedent, basis, and confidence boost, and their corresponding algorithmics. Learn how confidence boost solves the problem of negative correlations.
31. Learn about other alternative ways of defining redundancy among association rules.