Topic Classification in R

A Tutorial on Using Text Mining and Machine Learning Technologies to Classify Documents

December 2009

johannes.liegl@gmail.com

cc BY

Thanks

- Marco Maier (PhD Student at WU Wien)
  - Answering all of my R-related questions
  - Endless Patience
- Ingo Feinerer (Lecturer at TU Wien)
  - Creator of the **tm-package** for R
  - Answering all of my tm-related questions
  - Answers were always quick and profound although we never met in person

About the Talk

- I want you to **learn something** today
- If you **feel bored** you may **leave at any time** and I will not be mad at you
- Please **ask** at any(!) time if something is unclear
- Please **tell me** if I‘m too fast or speak too low or loud
- Please **contact me** per email if you have any **questions** about the talk, R, Text Mining or Support Vector Machines
- This **presentation** is under a **Creative Commons license**
- The **source code** is under an **LGPL license**
- Both are non-restrictive! (use it, alter it – just refer to me)
- Everything is **online** at www.contextualism.net/talks

Critique

- Please tell me what you think about this presentation
- Just send me an email and tell me one thing you liked and one you didn‘t

Outline

- Why?
- Classification
  - Introduction
  - Support Vector Machines
  - Topic Classification
- Reuters 21578 Dataset
- SVM Training in R
- Evaluation
  - Measures
  - Results

Why? (me)

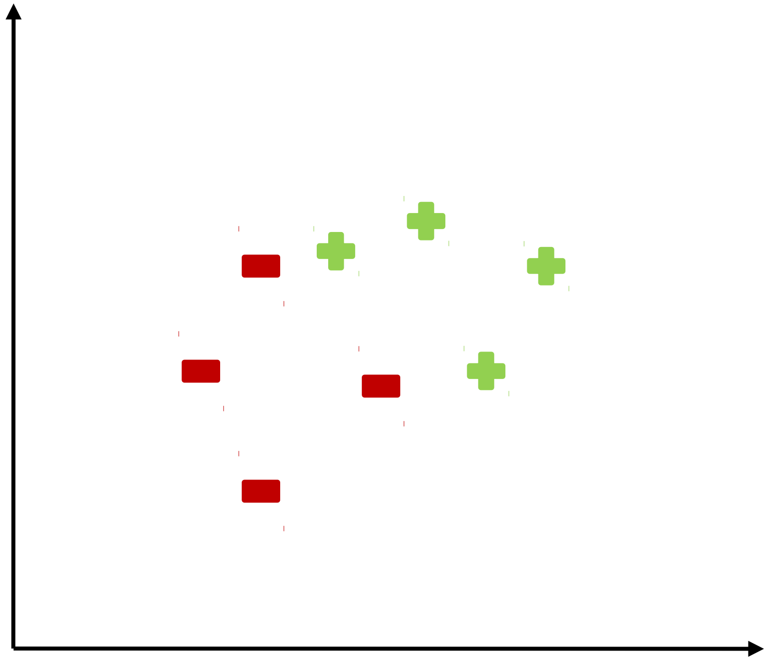
- Wanted to do Text Classification
- Wanted to train a Support Vector Machine
- Wanted to use R
- Wanted to reproduce the results of a scientific article

Why? (you)

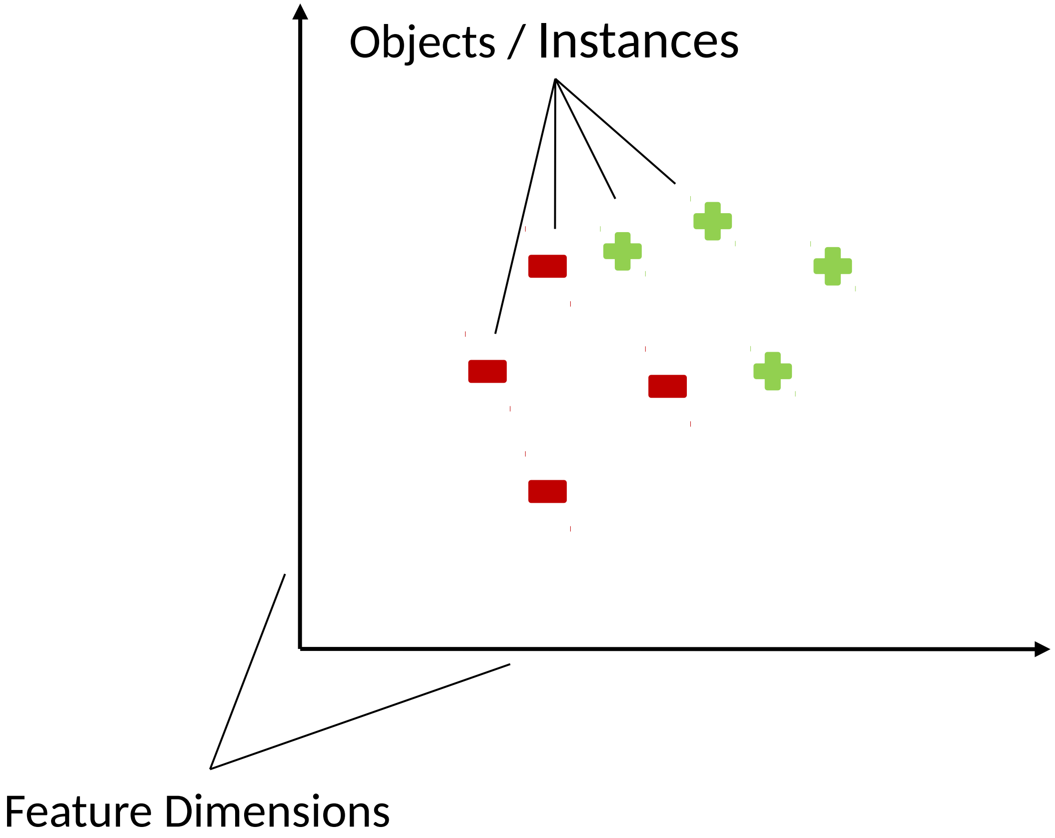
- Suppert Vector Machines are very good classificators
- They are used in a lot of recent publications – they‘re kind of the „new black“
- SVMs are a neat tool in your analytic armory
- If you understand topic classification you know how your SPAM is filtered
- Term-Document-Matrices are a very easy but powerful data-structure
- Usage of Text Mining Techniques could be fruitful for your research

Classification

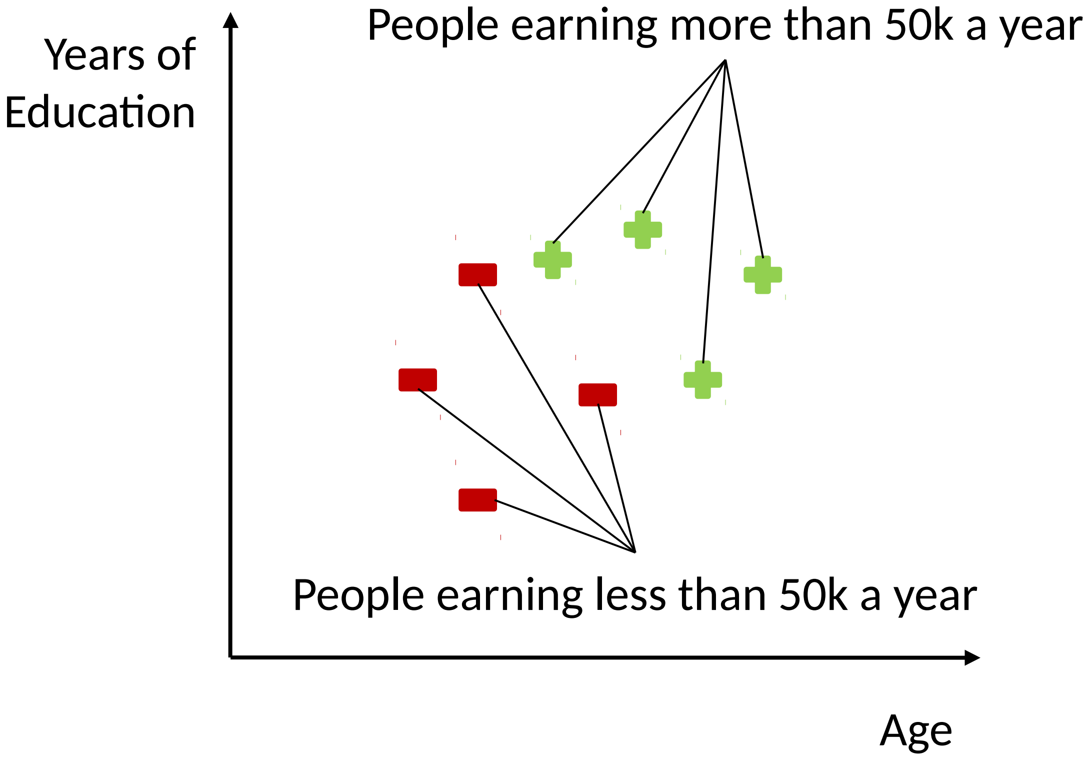
Classification



Classification

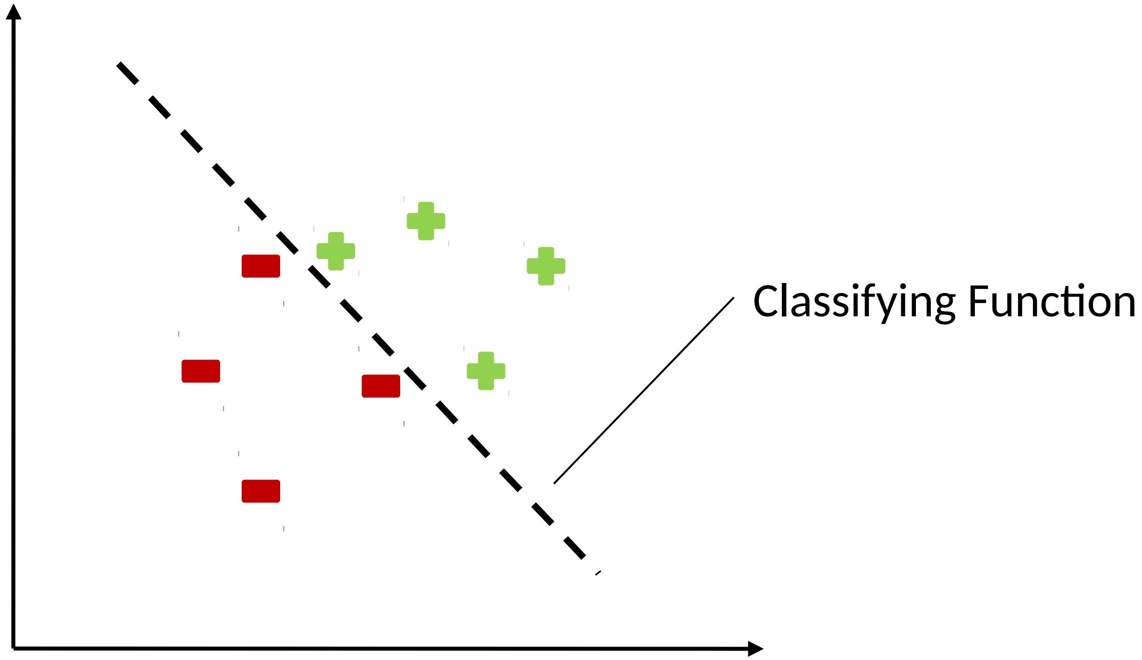


Classification

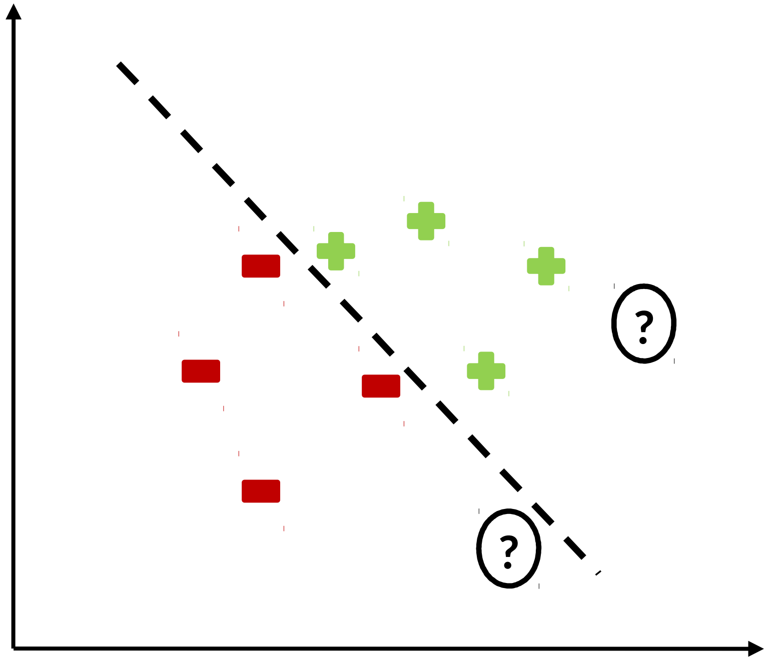


People earning more than 50k a year
Years of Education
People earning less than 50k a year
Age

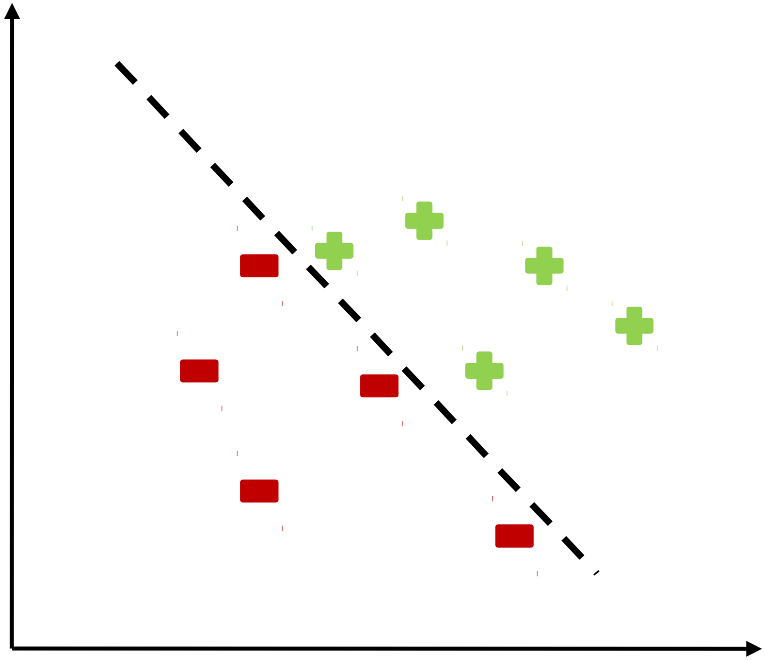
Classification



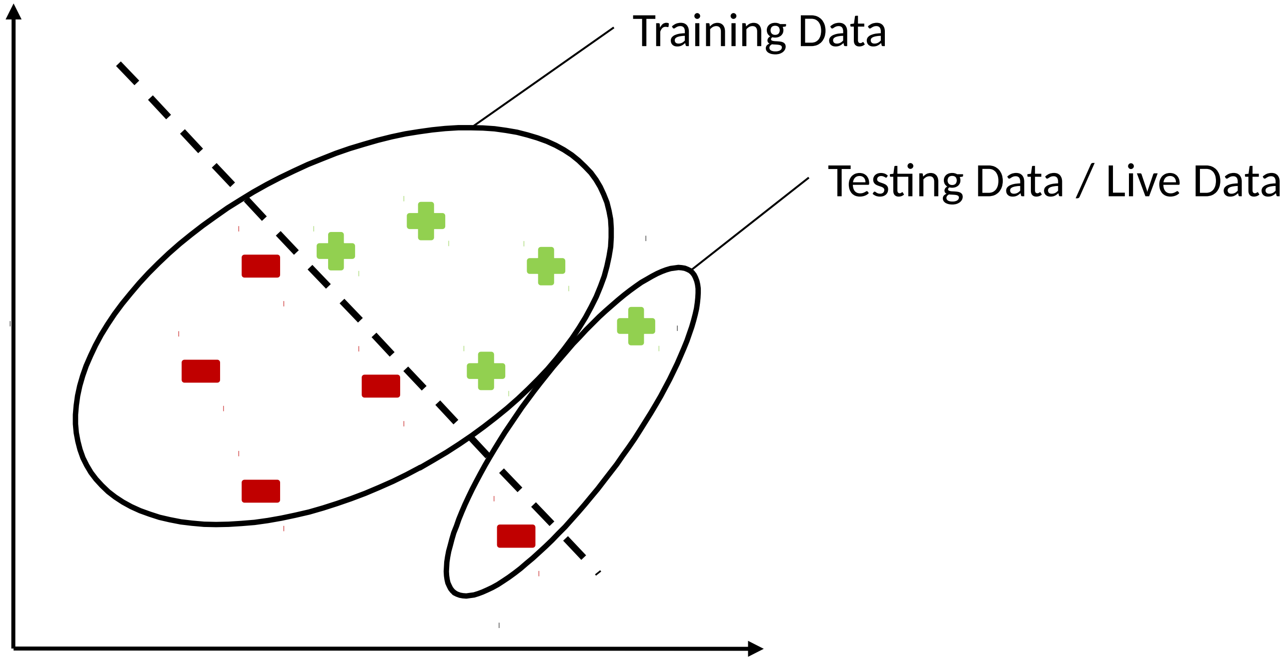
Classification



Classification



Classification



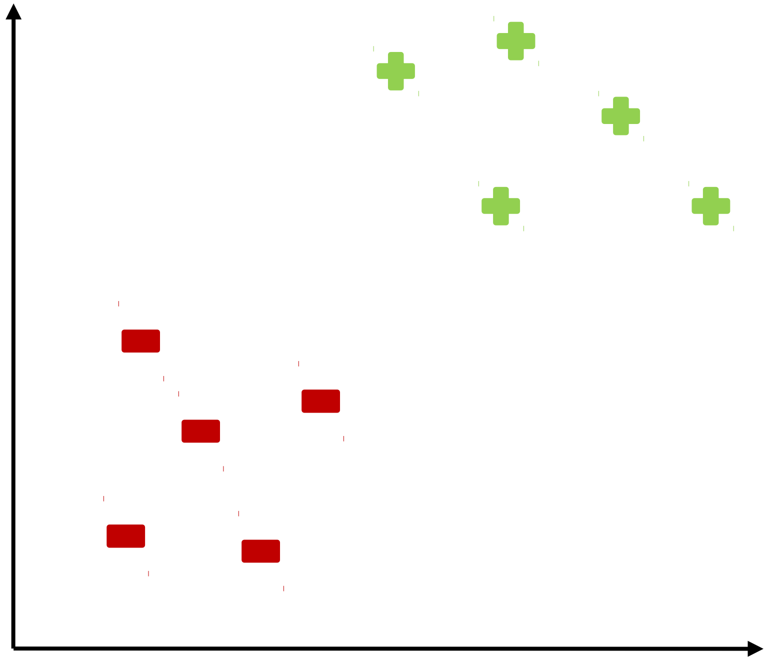
Classification

- Many different Classifiers available
- Strengths and weaknesses
- Decision Tree Classifiers
  - Good in Explaining the Classification Result
- Naive Bayes Classifiers
  - Strong Theory
- K-nn Classifiers
  - Lazy Classifiers
- **Support Vector Machines**
  - **Currently the state of the art**
  - **Perform very well across different domains in practice**

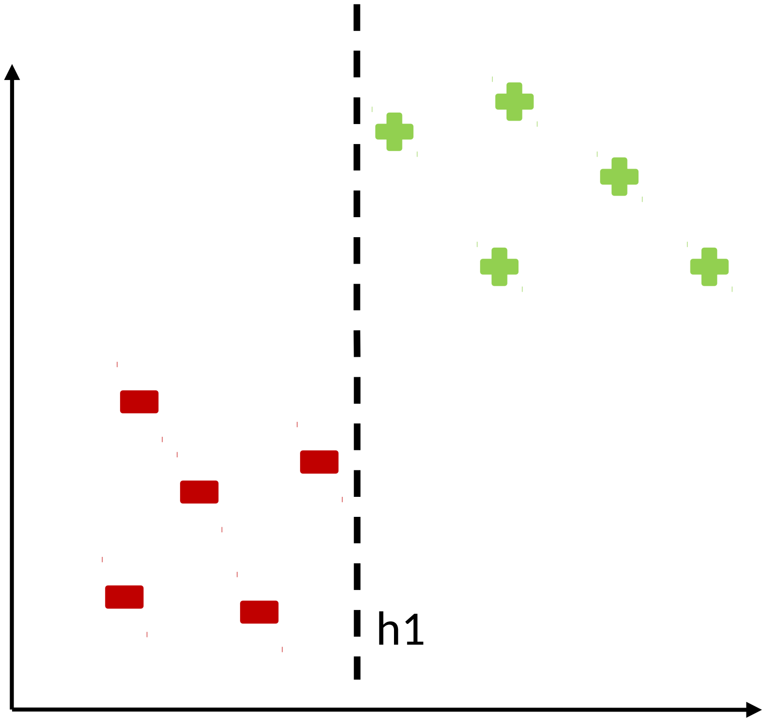
Support Vector Machines

- Solve a linear optimization problem
- Try to find the hyperplane with the largest margin (Large Margine Classifier)
- Use the „Kernel Trick“ to separate instances that are linearly unseparable

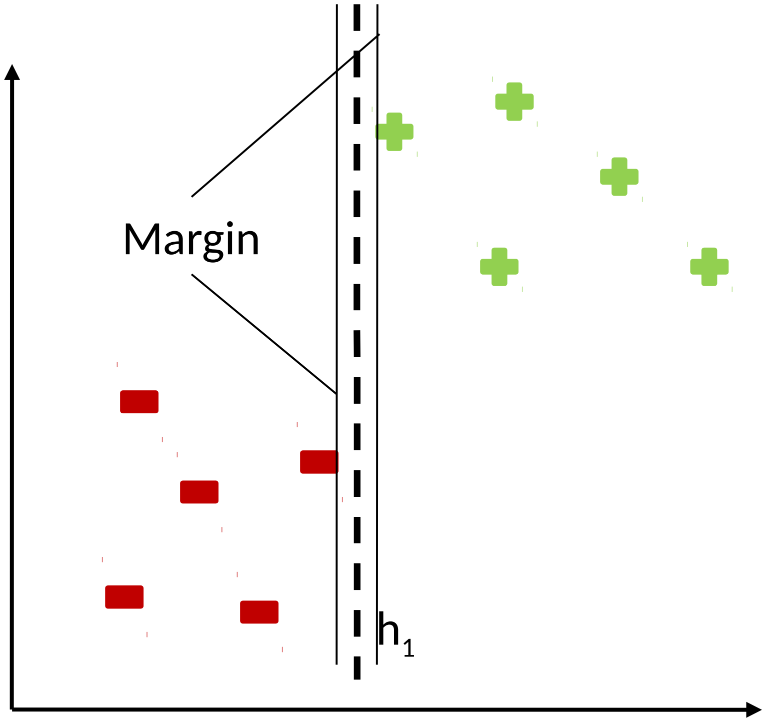
Largest Margin Hyperplane



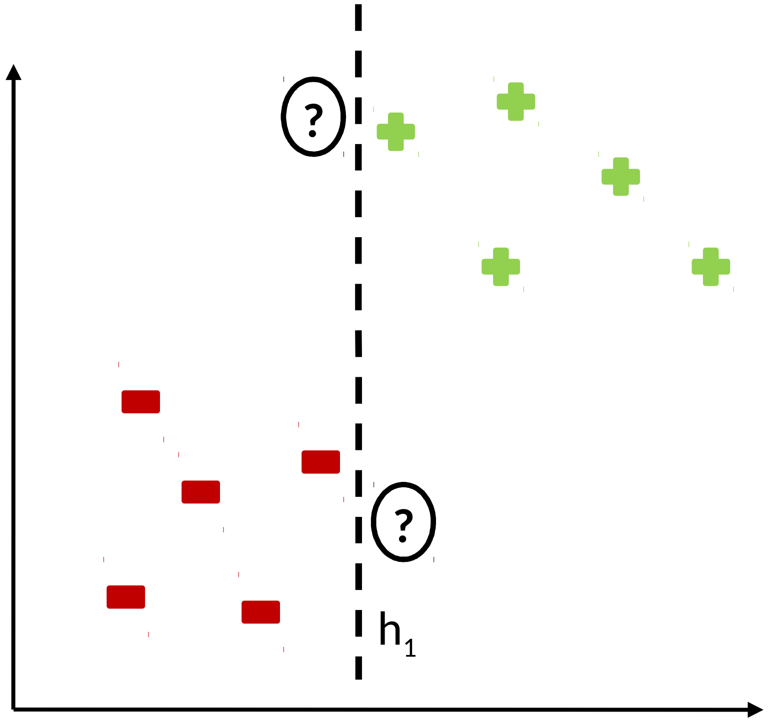
Largest Margin Hyperplane



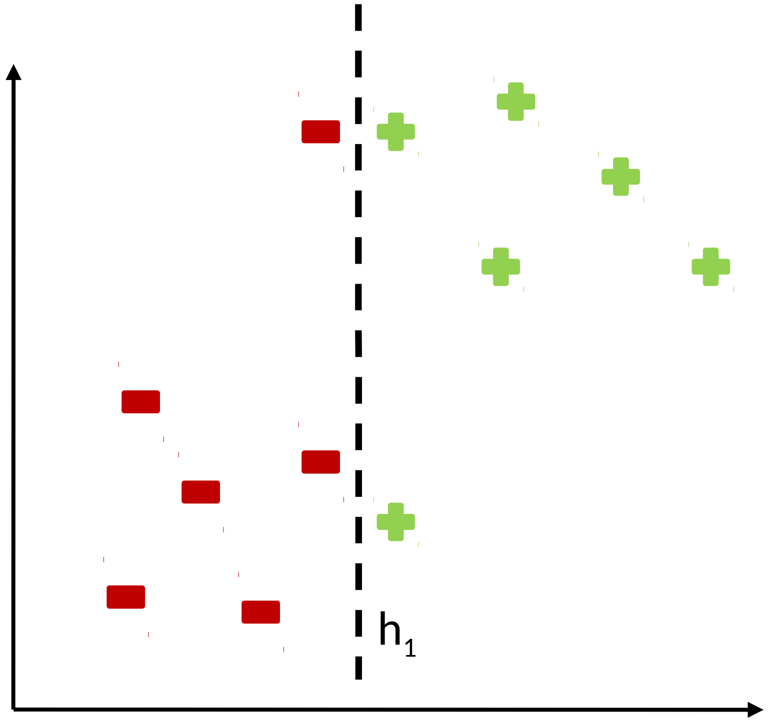
Largest Margin Hyperplane



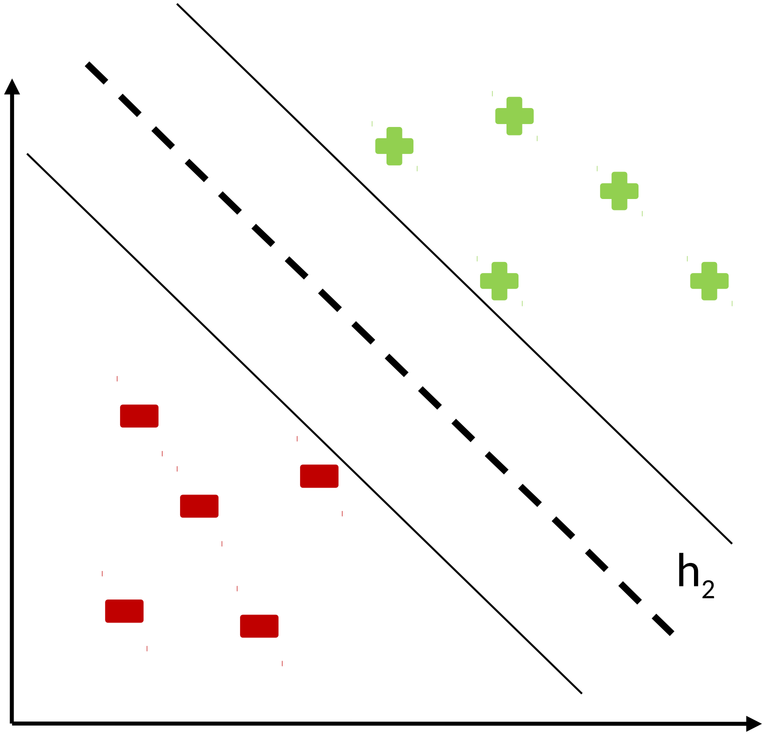
Largest Margin Hyperplane



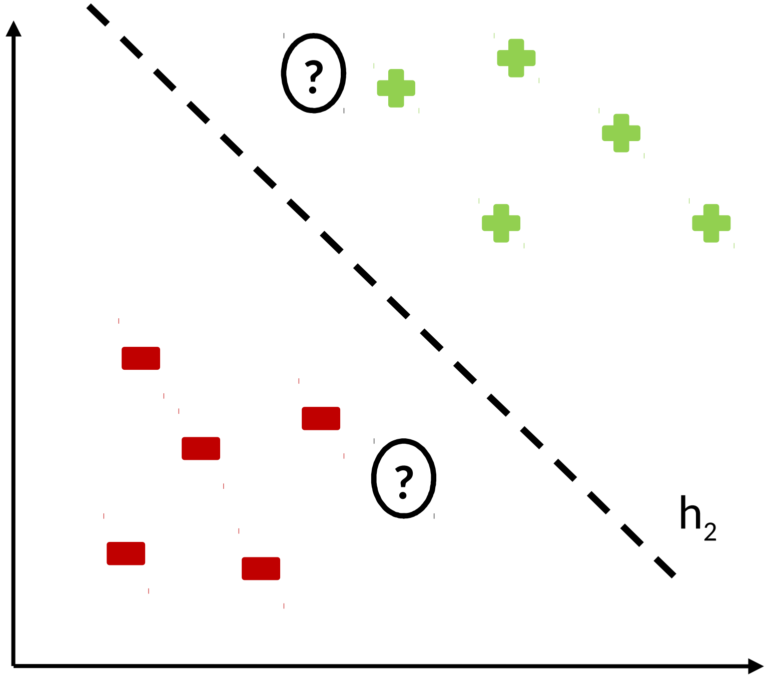
Largest Margin Hyperplane



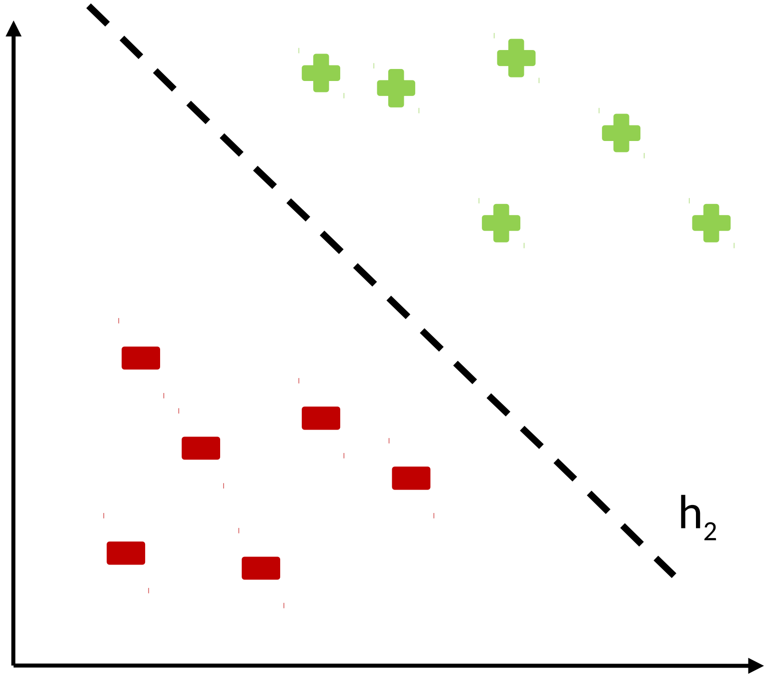
Largest Margin Hyperplane



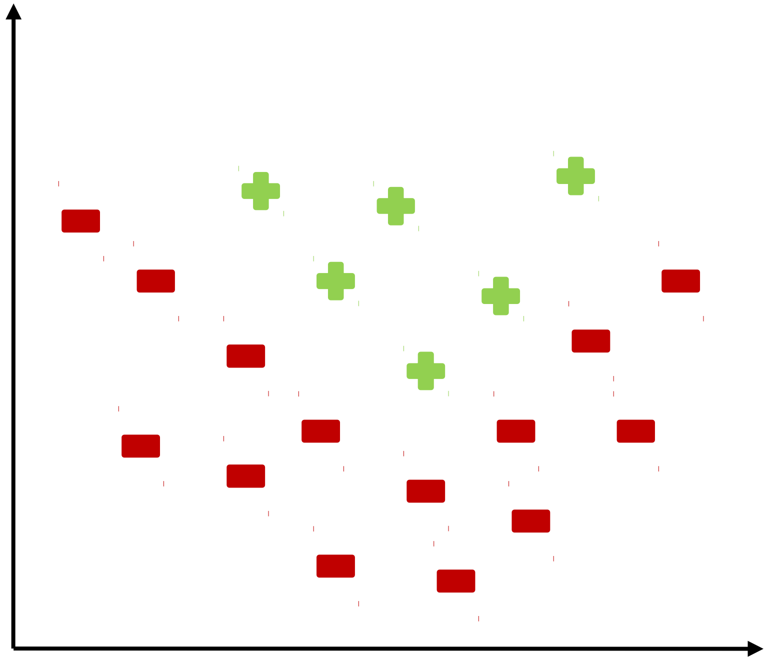
Largest Margin Hyperplane



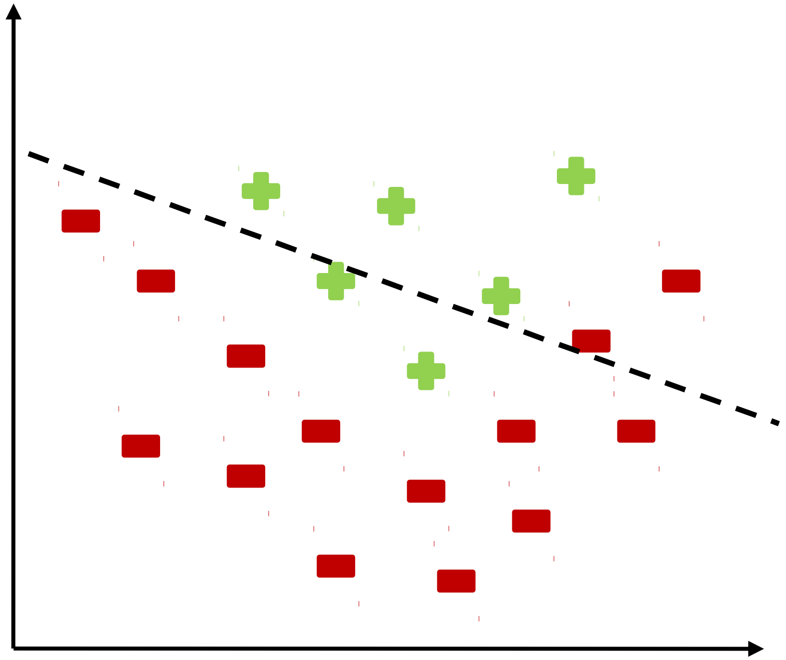
Largest Margin Hyperplane



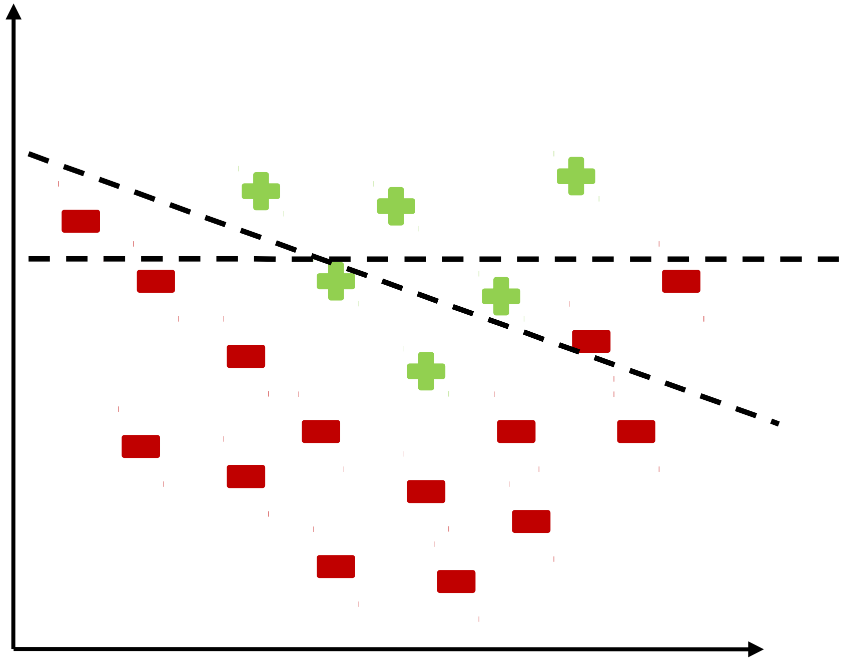
Kernel Trick



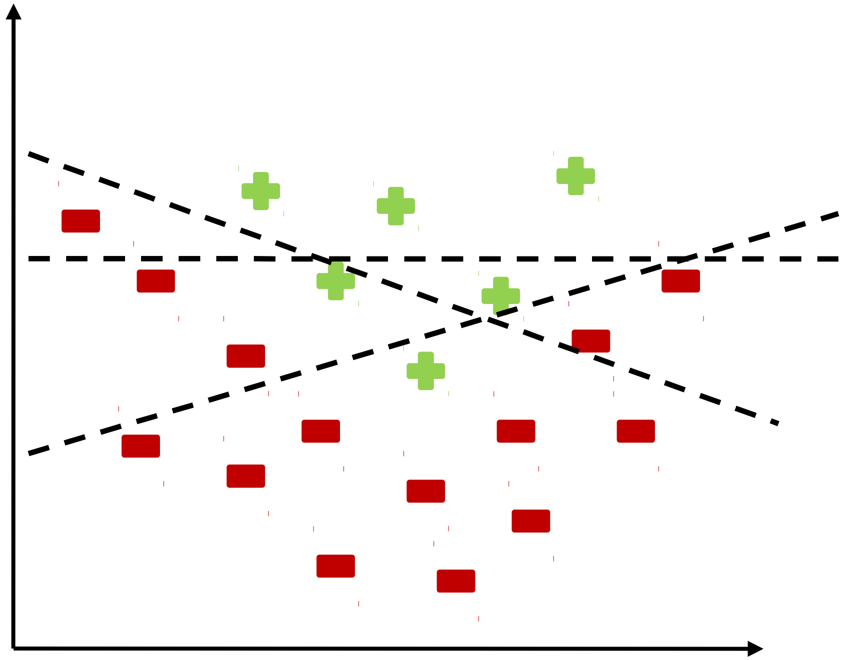
Kernel Trick



Kernel Trick

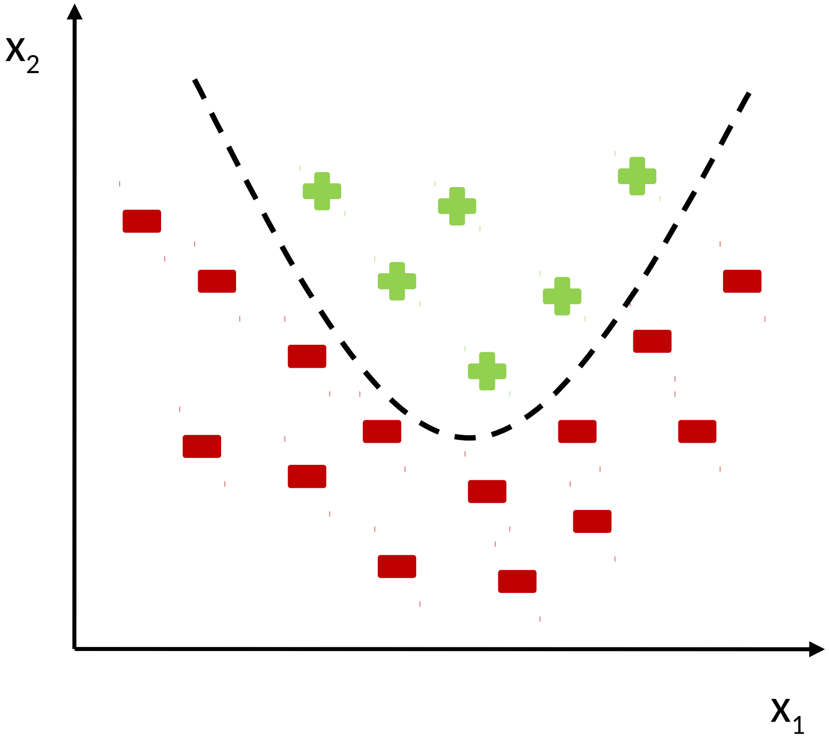


Kernel Trick



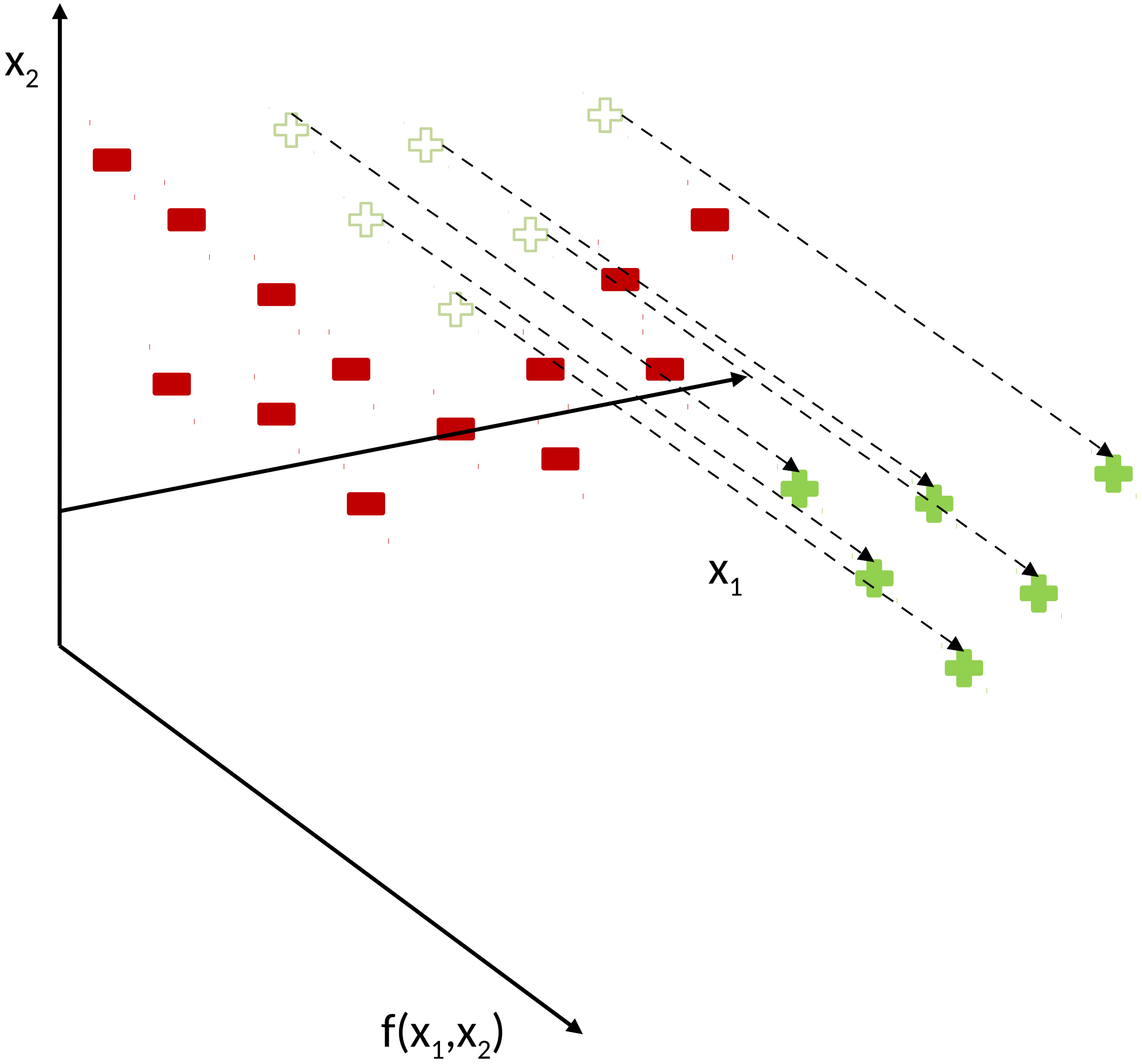
no linear function that separates the data

Kernel Trick

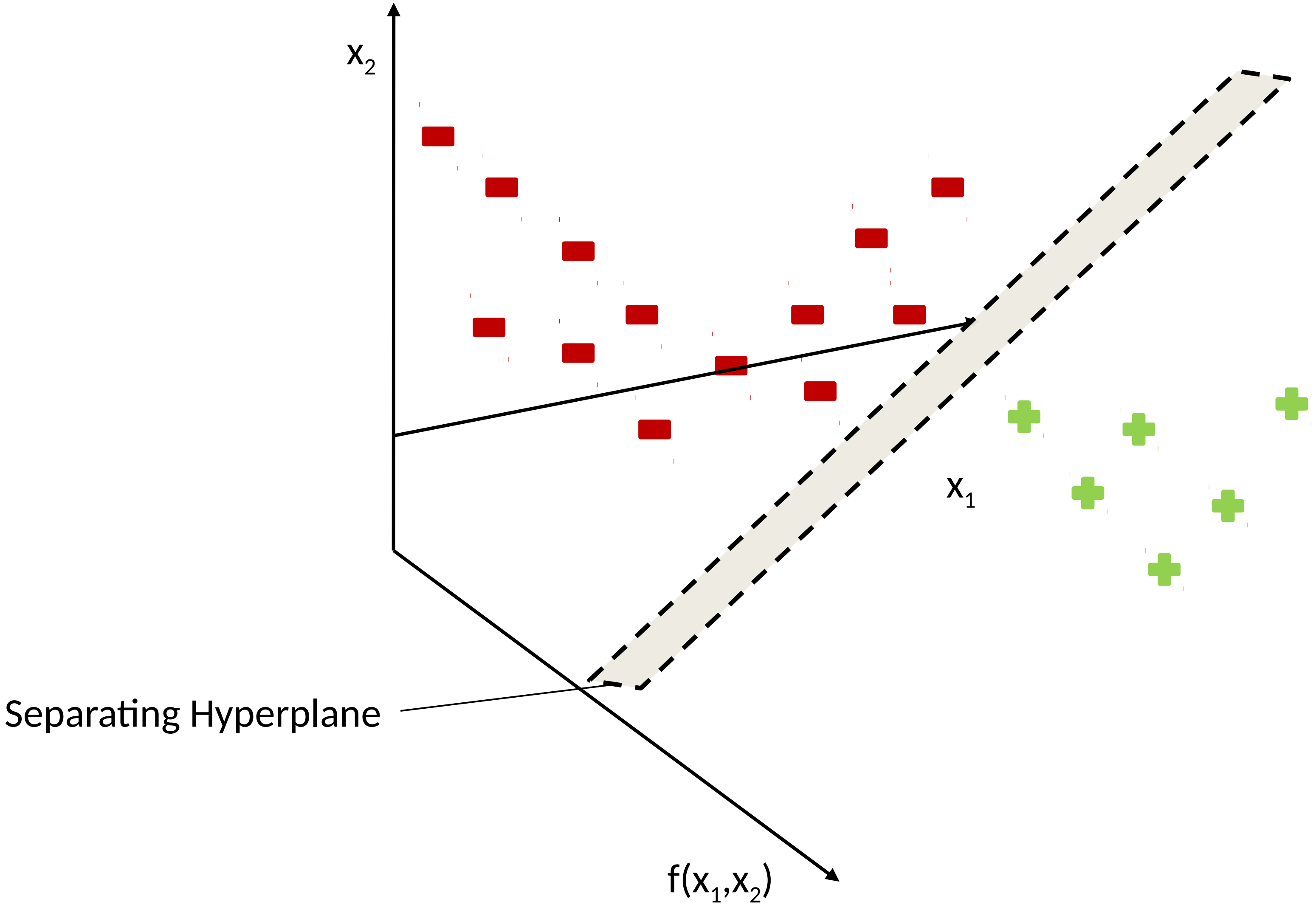


polynomial
function separating
the data

Kernel Trick



Kernel Trick



Kernel Trick

Video

Kernel Trick

- A lot of different Kernels available
  - Linear Kernel
  - Polynomial Kernel
  - RBF Kernel
  - Sigmoid Kernel
- You don‘t have to find your own one!
- Actually it‘s more or less a trial and error approach
- Best Case: Literature tells you the best Kernel for your domain

Topic Classification

- Decide if a given text-document belongs to a given topic or not (e.g. merger, oil, sports)
- Features for learning are term-counts (words)
- Every term becomes a feature dimension
- Feature-Space is high-dimensional
- Good for learning because probability of linear separability rises with the number of dimensions

Reuters 21578 Dataset

- 21.578 text documents (newswire articles)
- 135 topics
- Every document has one or more topics
- ModApte Split creates 3 sets
  - **Training set** (9.603 docs, at least 1 topic per doc, earlier April 7th 1987)
  - **Test set** (3.299 docs, at least 1 topic per doc, April 7th 1987 or later)
  - Unused set (8676 docs)
  - Topic distribution uneven in sets

Reuters 21578 Dataset
Topics with most Documents assigned

Class	Document Count
Earn	3987
Acq	2448
Money-Fx	801
Grain	628
Crude	634
Trade	551
Interest	513
Ship	305
Wheat	306
Corn	254

A Reuters Document

```
<?xml version="1.0"?>
<REUTERS TOPICS="YES" LEWISSPLIT="TRAIN" CGISPLIT="TRAINING-SET" OLDID="5552"
NEWID="9">
  <DATE>26-FEB-1987 15:17:11.20</DATE>
  <TOPICS>
    <D>earn</D>
  </TOPICS>
  <PLACES>
    <D>usa</D>
  </PLACES>
  <PEOPLE/>
  <ORGS/>
  <EXCHANGES/>
  <COMPANIES/>
  <UNKNOWN>Ff0762 reuter f BC-CHAMPION-PRODUCTS-&lt;CH02-26 0067</UNKNOWN>
  <TEXT>
    <TITLE>CHAMPION PRODUCTS &lt;CH&gt; APPROVES STOCK SPLIT</TITLE>
    <DATELINE>ROCHESTER, N.Y., Feb 26 -</DATELINE>
    <BODY>Champion Products Inc said its board of directors approved a two-for-one
stock split of its common shares for shareholders of record as of April 1, 1987.
The company also said its  board voted to recommend to shareholders at the annual
meeting April 23 an increase in the authorized capital stock from five mln to 25
mln shares. Reuter
    </BODY>
  </TEXT>
</REUTERS>
```

A Reuters Document

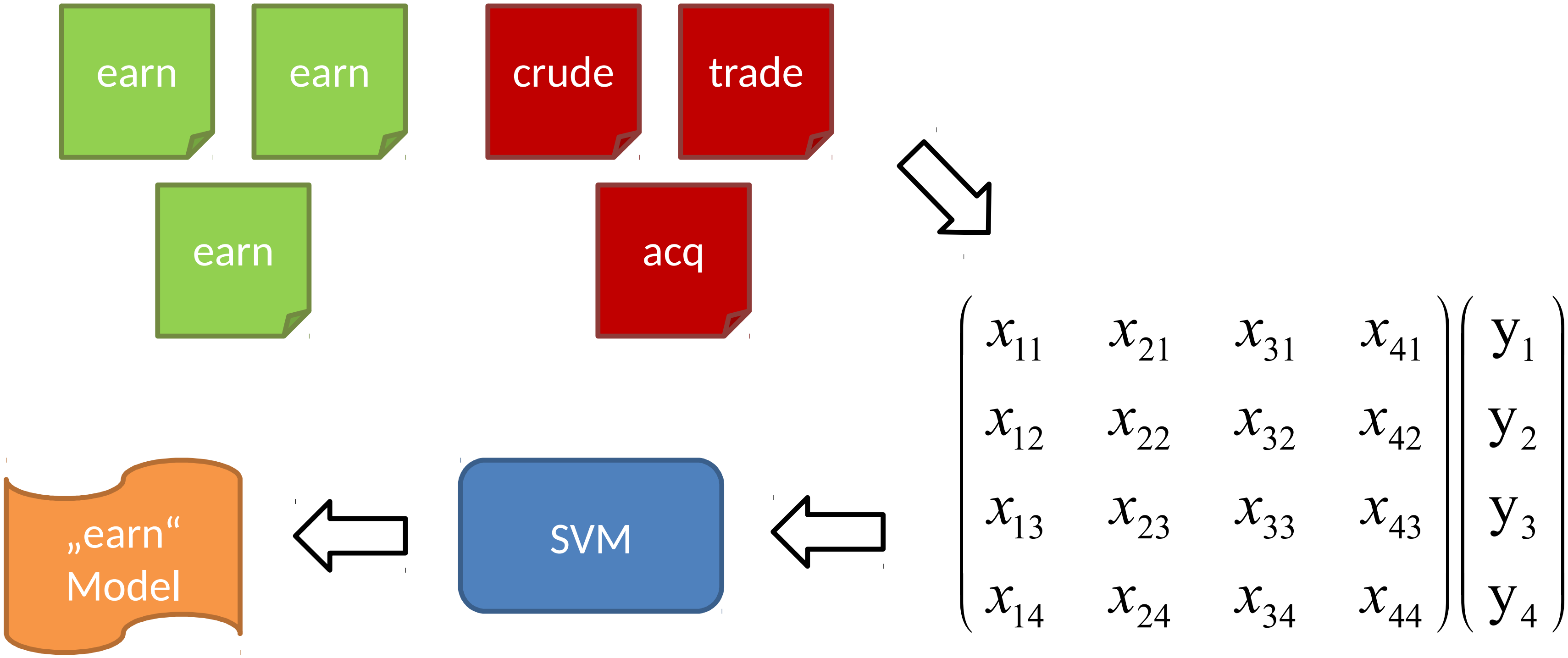
```
<?xml version="1.0"?>
<REUTERS TOPICS="YES" LEWISSPLIT="TRAIN" CGISPLIT="TRAINING-SET" OLDID="5552"
NEWID="9">
  <DATE>26-FEB-1987 15:17:11.20</DATE>
  <TOPICS>
    <D>earn</D>
  </TOPICS>
  <PLACES>
    <D>usa</D>
  </PLACES>
  <PEOPLE/>
  <ORGS/>
  <EXCHANGES/>
  <COMPANIES/>
  <UNKNOWN>Ff0762 reuter f BC-CHAMPION-PRODUCTS-&lt;CH02-26 0067</UNKNOWN>
  <TEXT>
    <TITLE>CHAMPION PRODUCTS &lt;CH&gt; APPROVES STOCK SPLIT</TITLE>
    <DATELINE>ROCHESTER, N.Y., Feb 26 -</DATELINE>
    <BODY>Champion Products Inc said its board of directors approved a two-for-one
stock split of its common shares for shareholders of record as of April 1, 1987.
The company also said its  board voted to recommend to shareholders at the annual
meeting April 23 an increase in the authorized capital stock from five mln to 25
mln shares. Reuter
    </BODY>
  </TEXT>
</REUTERS>
```

SVM Training in R

Our Goal

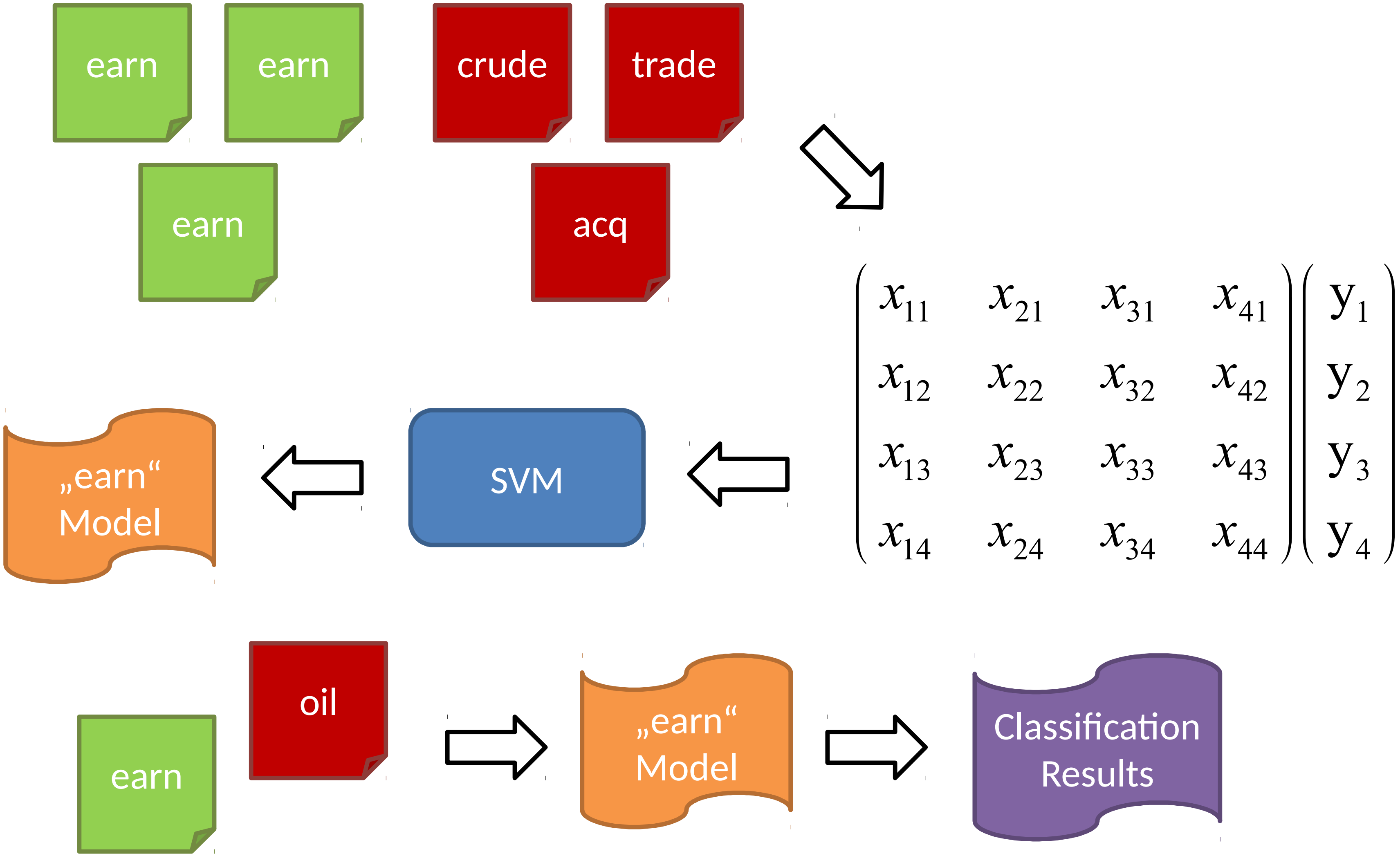
- Take a topic like „**earn**“
- Take *enough* „earn“ and „not earn“ documents from the training set and **create a training corpus**
- Bring training corpus in **adequate format** (data-structure) for SVM training
- Train a **SVM** (model)
- **Predict** test data
- **Compare Results** with those of Joachims

Our Goal



$$\begin{pmatrix} x_{11} & x_{21} & x_{31} & x_{41} \\ x_{12} & x_{22} & x_{32} & x_{42} \\ x_{13} & x_{23} & x_{33} & x_{43} \\ x_{14} & x_{24} & x_{34} & x_{44} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix}$$

Our Goal



earn

earn

earn

crude

trade

acq

$$\begin{pmatrix} x_{11} & x_{21} & x_{31} & x_{41} \\ x_{12} & x_{22} & x_{32} & x_{42} \\ x_{13} & x_{23} & x_{33} & x_{43} \\ x_{14} & x_{24} & x_{34} & x_{44} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix}$$

SVM

„earn“ Model

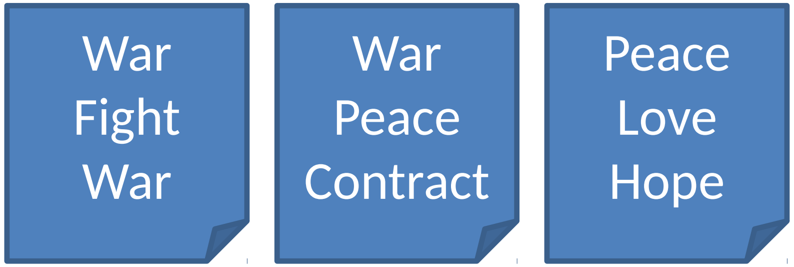
earn

oil

„earn“ Model

Classification Results

Document Term Matrices



War
Fight
War

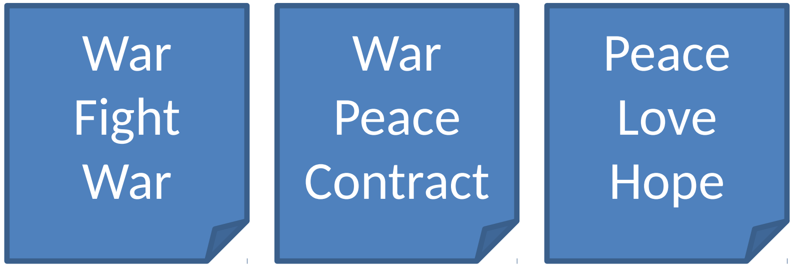
War
Peace
Contract

Peace
Love
Hope

doc$_1$ doc$_2$ doc$_3$

Corpus

Document Term Matrices



War
Fight
War

War
Peace
Contract

Peace
Love
Hope

doc$_1$ doc$_2$ doc$_3$

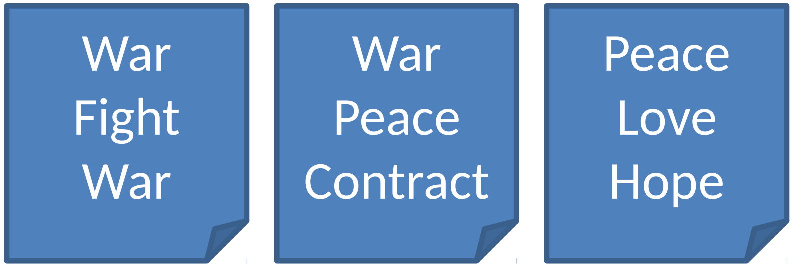
Corpus

Document Term Matrix

	war	fight	peace	contract	love	hope
doc$_1$	2	1	0	0	0	0
doc$_2$	1	0	1	1	0	0
doc$_3$	0	0	1	0	1	1

Term Weight

Document Term Matrices



War
Fight
War

War
Peace
Contract

Peace
Love
Hope

Corpus

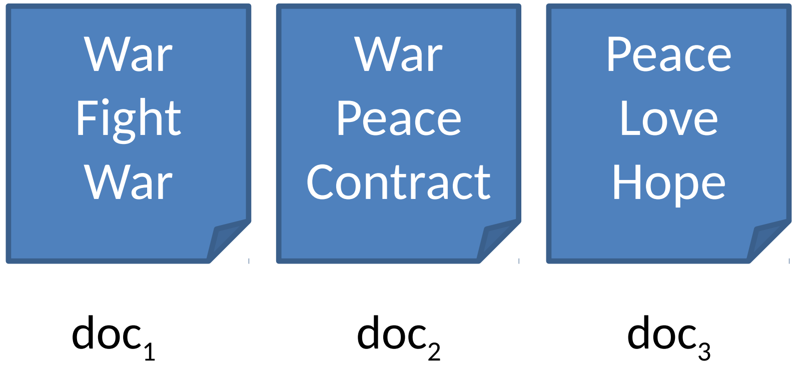
doc_1 doc_2 doc_3

Document Term Matrix

	war	fight	peace	contract	love	hope
doc_1	2	1	0	0	0	0
doc_2	1	0	1	1	0	0
doc_3	0	0	1	0	1	1

Document Vector

Document Term Matrices



War
Fight
War

War
Peace
Contract

Peace
Love
Hope

doc$_1$ doc$_2$ doc$_3$

Corpus

Document Term Matrix

	war	fight	peace	contract	love	hope
doc$_1$	2	1	0	0	0	0
doc$_2$	1	0	1	1	0	0
doc$_3$	0	0	1	0	1	1

Term Vector

Term Weighting

- Term Frequency (TF)
  - Number of times a term occurs in a document
- Term Frequency – Inverse Document Frequency (TF-IDF)
  - $(1+\log(tf_{t,d}))\times\log(N/df_t)$
- Term Frequency – Inverse Document Frequency – Cosine Normalization (TFC)
  - $\dfrac{tf_t \log \dfrac{N}{df_t}}{\sqrt{\sum \left(tf_i \log \dfrac{N}{df_i} \right)^2}}$

Term Weighting

- Weighting strategy is crucial for the classification results
- IDF factor puts heavier weight on terms that occur seldomly (strong discriminative power)
- Normalization dempens the impact of outliers on the model to be built

Questions?

Assumption

- Training and test documents of Reuters ModApte split reside in two separate directories
  - `/reuters-21578-xml-train`
  - `/reuters-21578-xml-test`
- Documents are in XML format
- There‘s a preprocessing script (in R) that accomplishes this task
- The script will be online too and you can ask me anytime if you have problems with it

DirSource

- Abstracts the input location
- Other sources are available for different file formats like CSVSource and GmaneSource
- The source concept eases internal processes in the TM package (standardized interfaces)
- **Our DirSources** will just contain a **list of XML documents** from a given directory

Corpus

- Collection or DB for text documents
- In our case it holds the documents from the training set and test set
- Does some transformation on the XML files
- Strips off XML
- Extracts and stores **meta Information** (topics)
- **XML -> Plain Text**
- Access to documents via Indices
- Some statistics about contained documents

Document Term Matrix & Dictionary

Dictionary: List of all the terms in a Corpus/DTM

	aaa	abbett	⋯	zones	zuckerman
doc$_1$	2	1	⋯	0	1
doc$_2$	1	0	⋯	0	0
⋯					
doc$_{9602}$	0	0	⋯	1	1

tm_filter

- Returns a **filtered corpus**
- Predefined filters available
  - searchFullText
  - sFilter (Meta Data)
- **Custom filters** can be added (e.g. TopicFilter and maxDocNumPerTopicFilter)
- **doclevel** decides if filter is applied on corpus as a whole or each single document
- Theoretically easy – Practically kind of cumbersome (R-internals)

Topic Filter Function

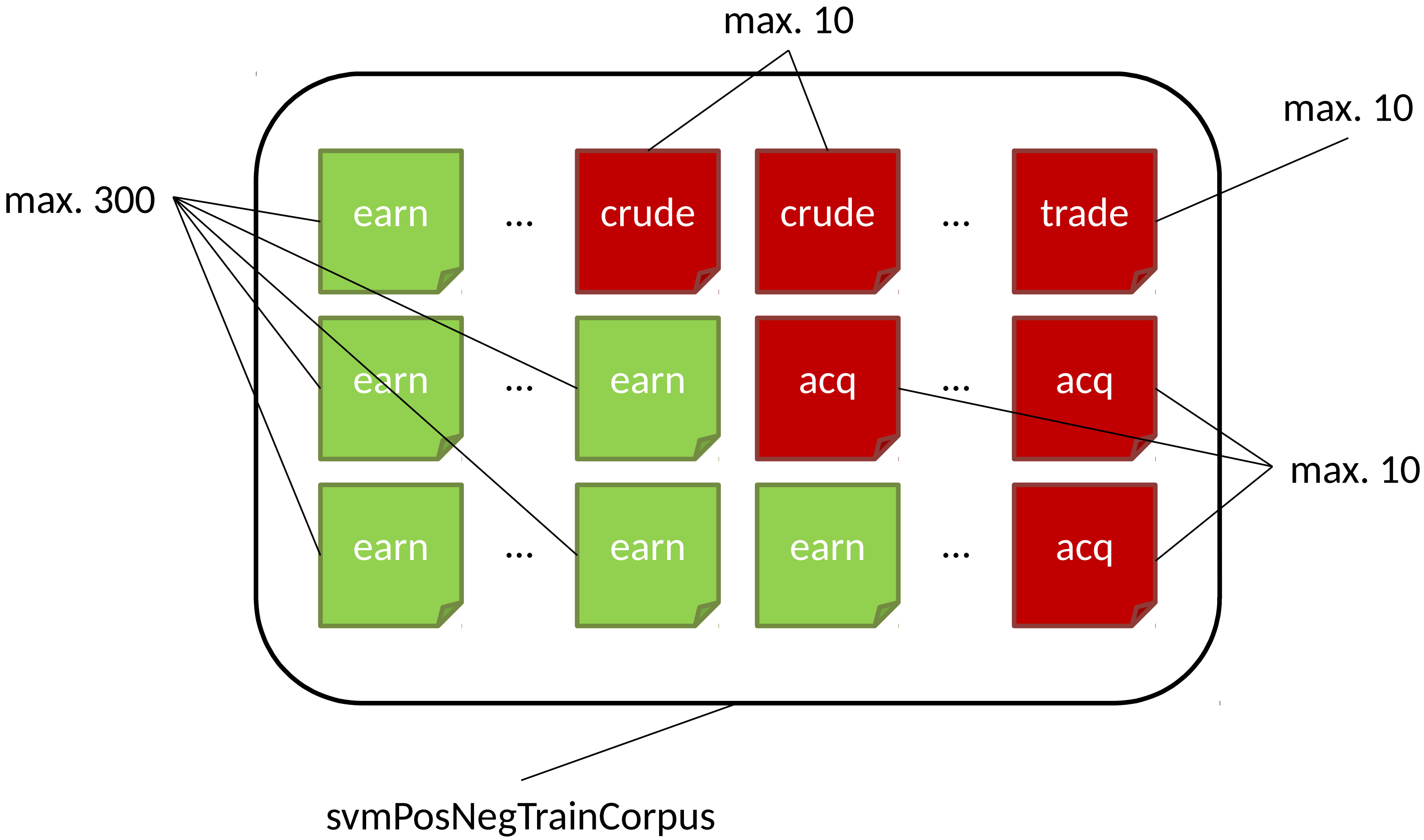
```
topicFilter <- function (object, s, topicOfDoc) {
    query.df <- prescindMeta(object, c("Topics"))
    attach(query.df)
    boolFilter <- c()
    i <- 1

    while (i <= length(Topics)) {
      res <- c(s) %in% Topics[[i]]
      boolFilter <- c(boolFilter, res)
      i <- i + 1
    }
    if (!topicOfDoc)
      boolFilter <- unlist(lapply(boolFilter,`!`))

    try (result <- rownames(query.df) %in% row.names(query.df[boolFilter,]))

    detach(query.df)
    result
}
```

The Training Corpus



Document Term Matrix

- Actually we could construct the DTM now and fed it together with the label vector to the SVM
- But, we have to ensure, that all terms in the dictionary (from ModApte Training Split) are used!
- DocumentTermMatrix removes terms that don‘t occur in the corpus – which would fall back on us during evaluation
- We have to ensure that the matrices always have the same structure using the same terms on the same place (column)

Auxiliary 1-Document Corpus

- Create corpus with just one document
- This document contains all terms from the dictionary
- Merge Auxiliary Corpus with Training Corpus

Sparse Matrices

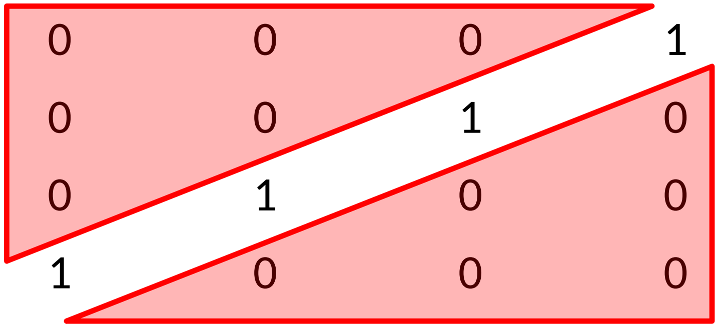
- Most of the values in a DTM are zero (→ sparse)
- Storing the 0s would be waste of space
- Sparse matrices only store non-zero values and some structure
- DTM in tm-package is backed by simple_triplet_matrix
- SVM is backed by CompressedSparseRowMatrix
- Conversion needed

Sparse Matrices

	book	car	peace	war
doc_1	0	0	0	1
doc_2	0	0	1	0
doc_3	0	1	0	0
doc_4	1	0	0	0

Plain Storage
16 numbers

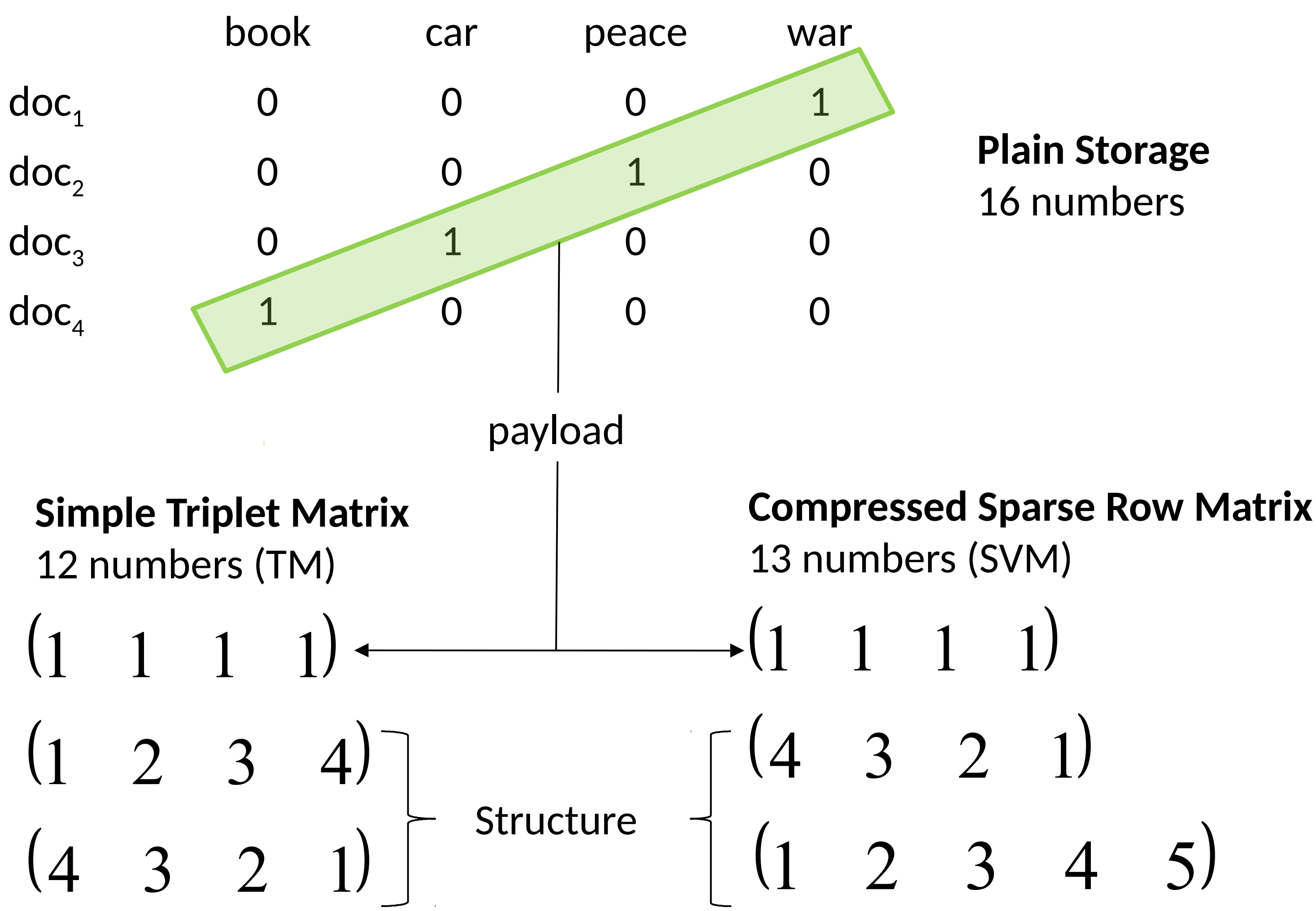
Sparse Matrices



	book	car	peace	war
doc_1	0	0	0	1
doc_2	0	0	1	0
doc_3	0	1	0	0
doc_4	1	0	0	0

Plain Storage
16 numbers

Sparse Matrices



	book	car	peace	war
doc$_1$	0	0	0	1
doc$_2$	0	0	1	0
doc$_3$	0	1	0	0
doc$_4$	1	0	0	0

Plain Storage
16 numbers

payload

Simple Triplet Matrix
12 numbers (TM)

$$(1 \quad 1 \quad 1 \quad 1)$$

$$(1 \quad 2 \quad 3 \quad 4)$$

$$(4 \quad 3 \quad 2 \quad 1)$$

Structure

Compressed Sparse Row Matrix
13 numbers (SVM)

$$(1 \quad 1 \quad 1 \quad 1)$$

$$(4 \quad 3 \quad 2 \quad 1)$$

$$(1 \quad 2 \quad 3 \quad 4 \quad 5)$$

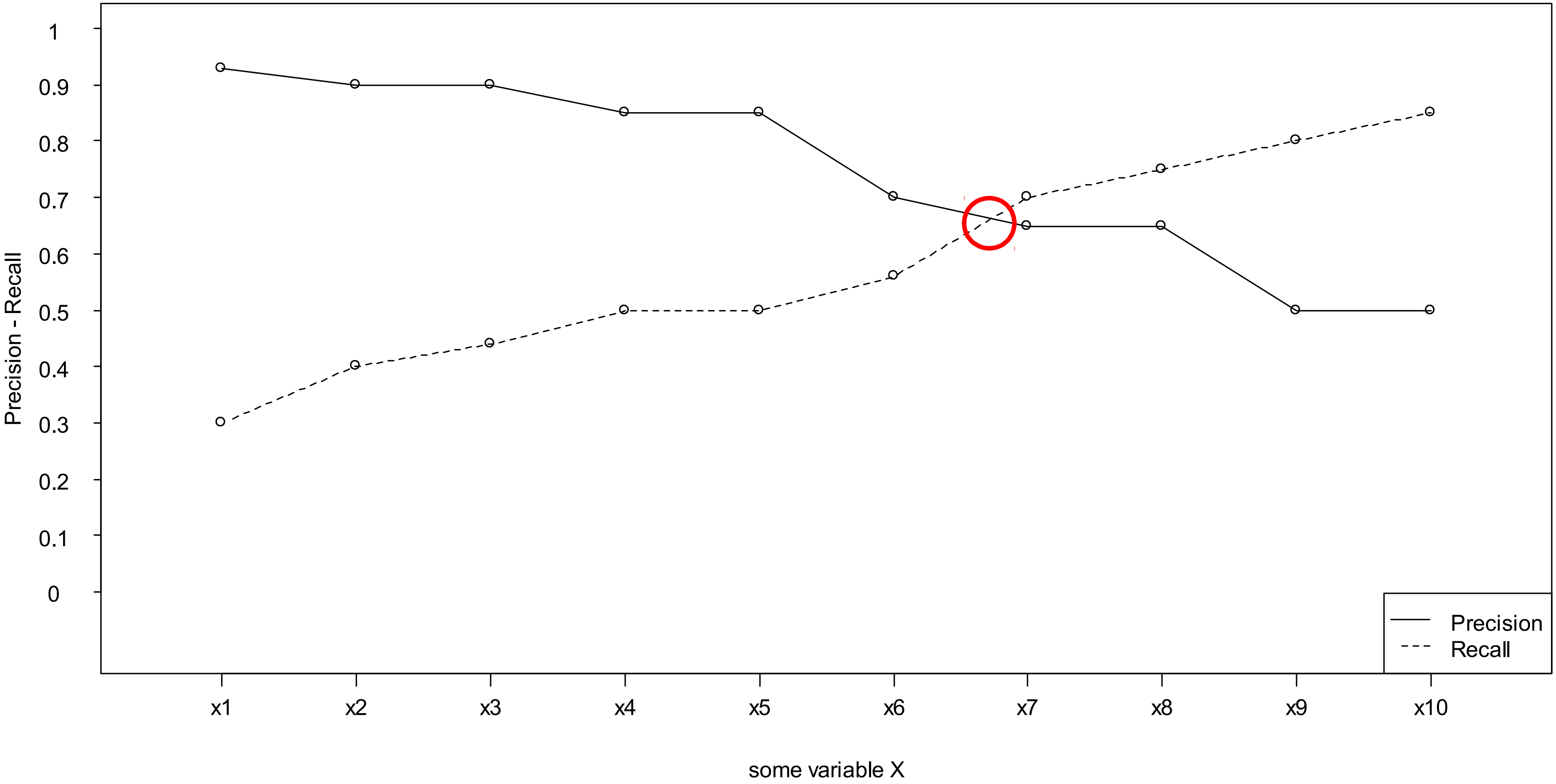
Label Vector

$$yPos = \left.\begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{pmatrix}\right\} 300 \qquad yNeg = \left.\begin{pmatrix} -1 \\ -1 \\ \vdots \\ -1 \\ -1 \end{pmatrix}\right\} 422 \qquad y = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ -1 \\ -1 \end{pmatrix} \begin{matrix} \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} 300 \\ \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} 422 \end{matrix}$$

Evaluation Measures

- Joachiems uses Precision-Recall Breakeven points
- To do so you‘d have to alter a variable in the setting over a range of values
- But he never states which variable he alters
- So I decided to use the Precision and Recall results without their Breakeven point

Precision Recall Breakeven Point



Confusion Matrix

	predicted +	predicted −
true +	a	b
true −	c	d

Confusion Matrix

	predicted +	predicted −
true +	true positive	false negative
true −	false positive	true negative

Confusion Matrix

	predicted +	predicted −
true +	TP	FN
true −	FP	TN

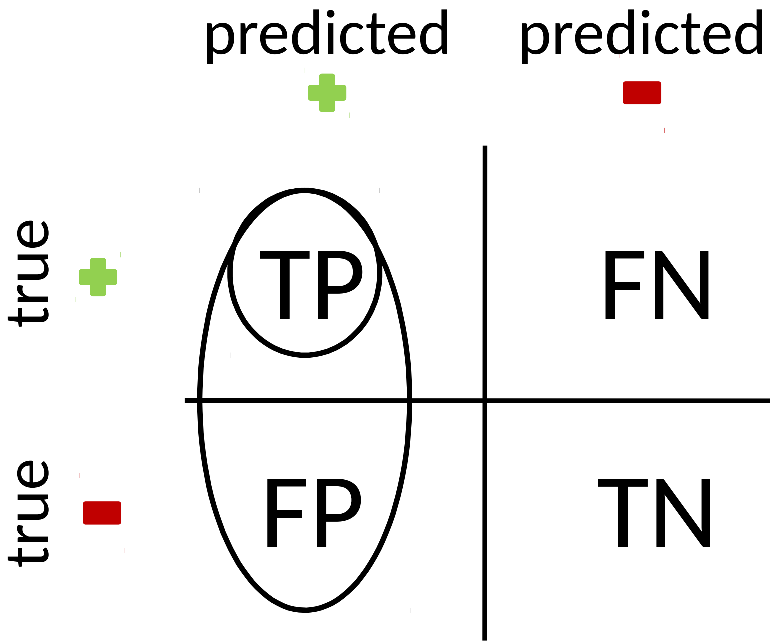
Precision

	predicted +	predicted −
true +	TP	FN
true −	FP	TN

Precision

	predicted +	predicted −
true +	TP	FN
true −	FP	TN

Precision



	predicted +	predicted −
true +	TP	FN
true −	FP	TN

measure of exactness

$$\frac{TP}{TP + FP}$$

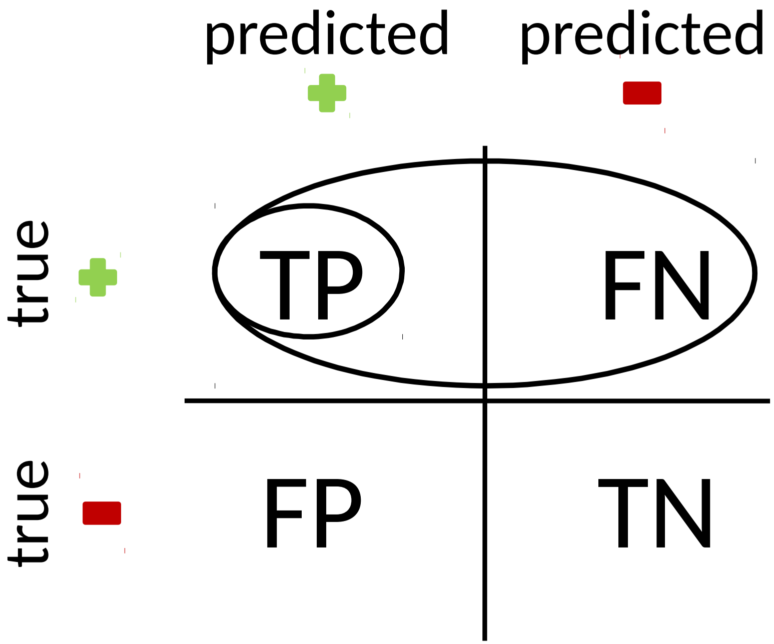
Recall

	predicted +	predicted −
true +	TP	FN
true −	FP	TN

Recall

	predicted +	predicted −
true +	TP	FN
true −	FP	TN

Recall

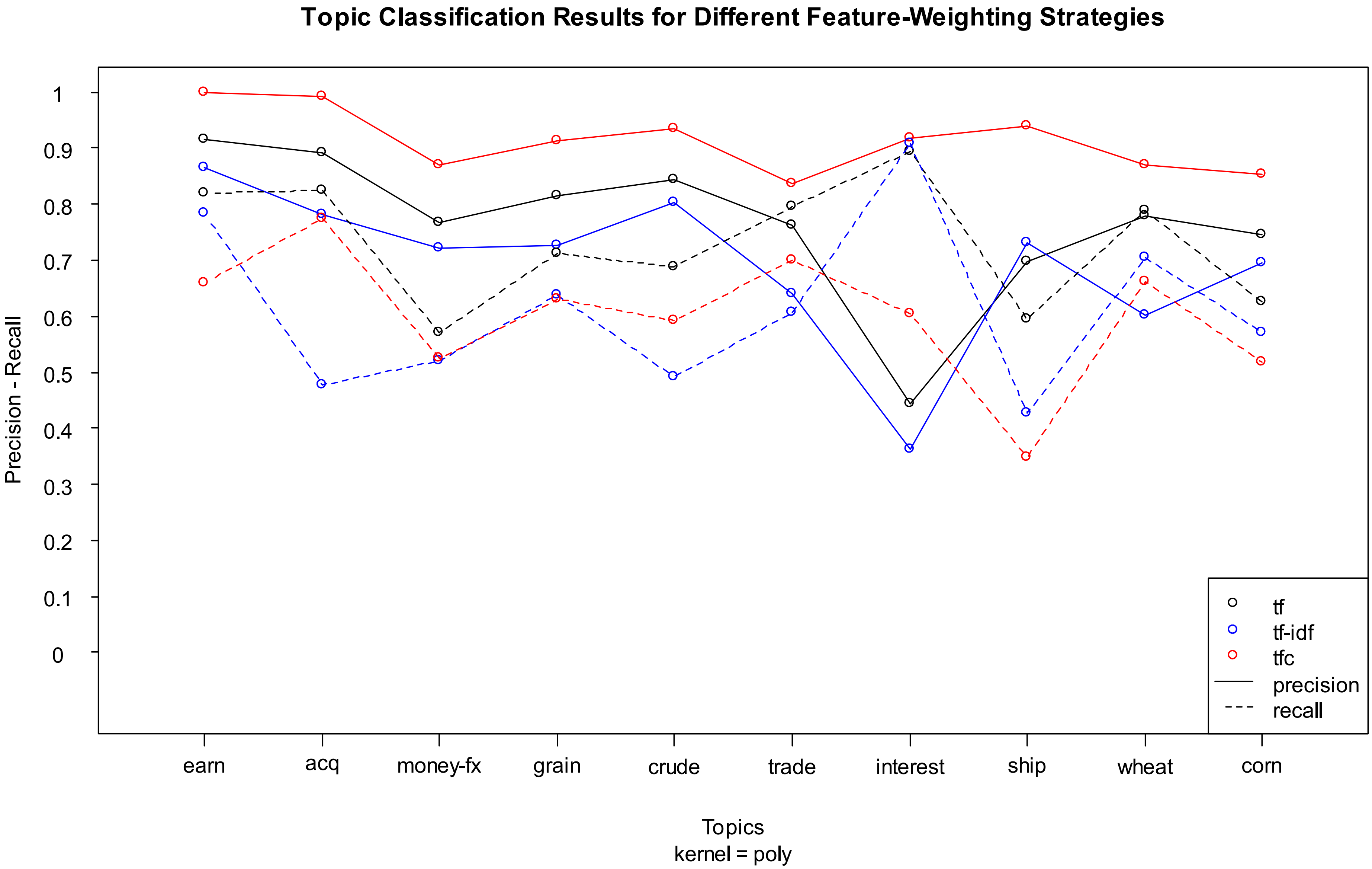


	predicted +	predicted −
true +	TP	FN
true −	FP	TN

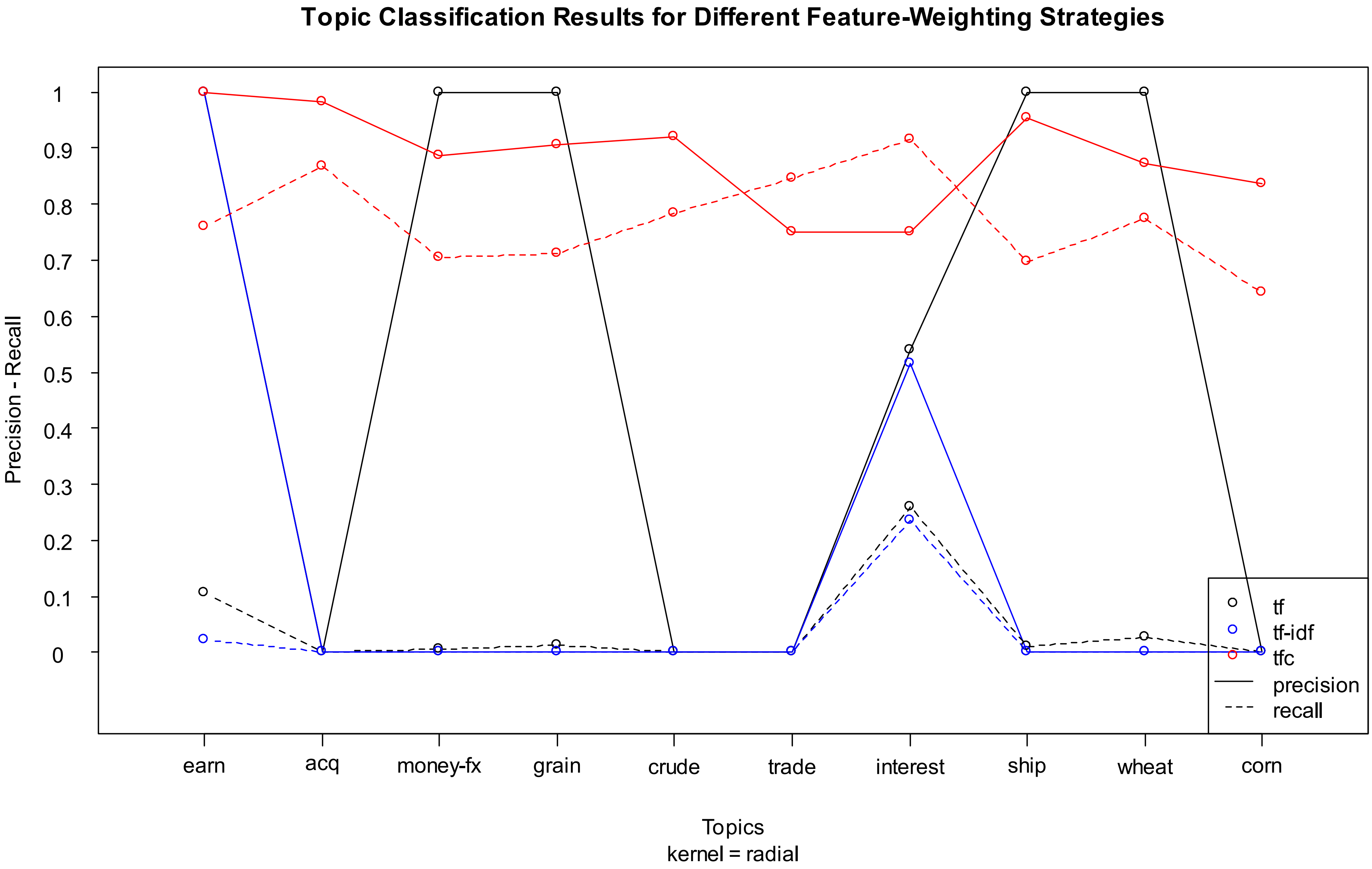
measure of completeness

$$\frac{TP}{TP+FN}$$

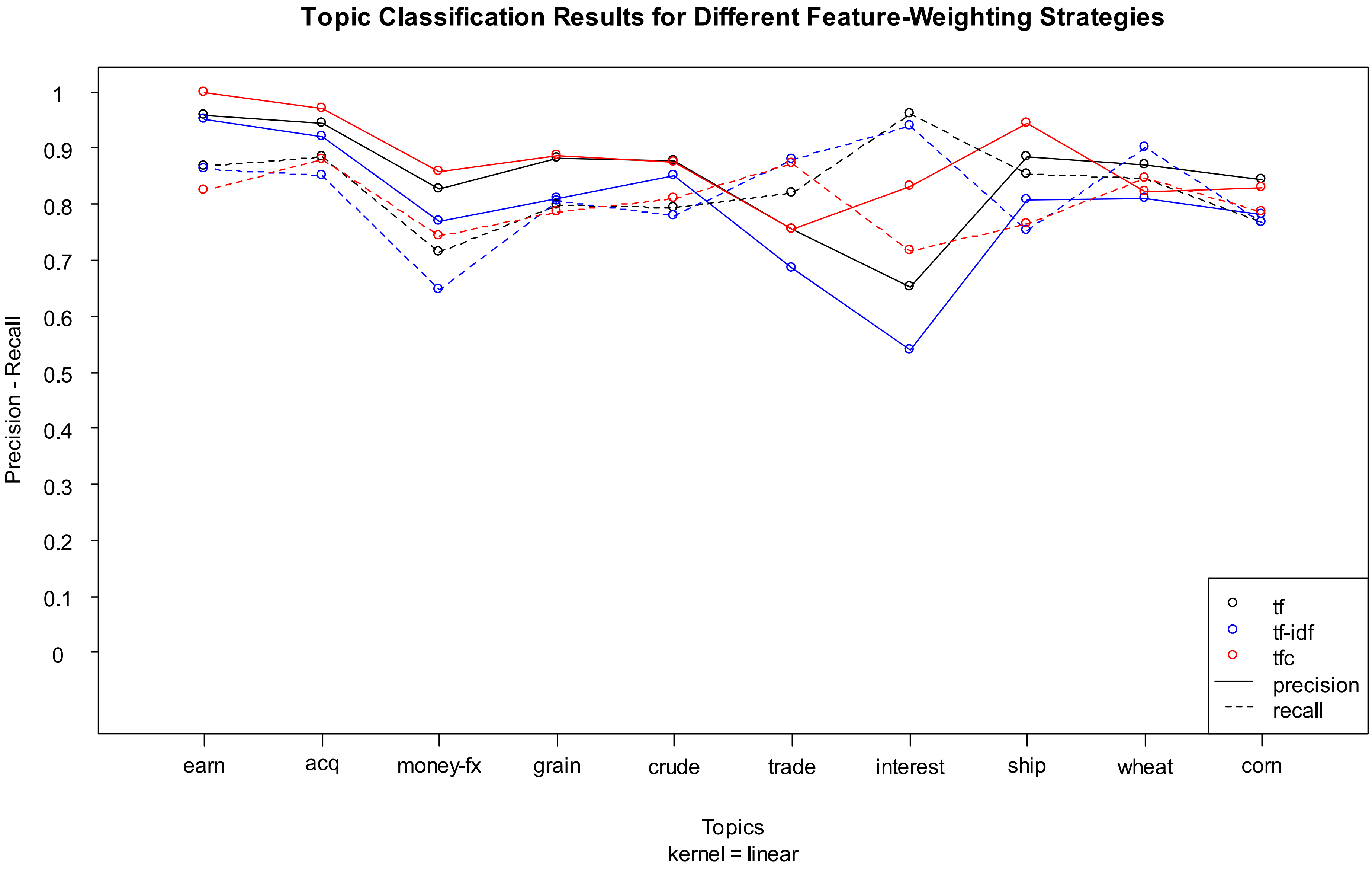
Evaluation Results



Evaluation Results



Evaluation Results



Comparison to Joachims
Kernel=poly, degree=2

	Joachims	Me	
Topic	Prec.-Rec. Breakeven	Precision	Recall
earn	98.4	0.99	0.65
acq	94.6	0.99	0.77
money-fx	72.5	0.87	0.52
grain	93.1	0.91	0.63
crude	87.3	0.93	0.59
trade	75.5	0.83	0.7
interest	63.3	0.91	0.6
ship	85.4	0.93	0.34
wheat	84.5	0.87	0.66
corn	86.5	0.85	0.51

Findings

- Even though the paper was very detailed and well written – some information was missing to fully reproduce the results
- tm-package is very comfortable to use but adding custom functionality is cumbersome because of R-internals
- svm-package is surprisingly easy to use but memory limits are reached very soon
- Classification results are quiet good

Literature

- Feinerer I., Hornik K., Meyer D., Text Mining Infrastructure in R, Journal of Statistical Software, 2008
- Joachims T., Text categorization with Support Vector Machines: Learning with many relevant features, ECML, 1998
- Salton G., Buckley C., Term-weighting approaches in automatic text retrieval, Journal of Inf. Proc. Man., 1988
- Classification Performance Measures, 2006
- XML Encoded Reuters 21578 Dataset

Thank you!

For Attention & Patience

Apendix

Versions of R and R-Packages used

- R 2.9.2
- tm package 0.5 (text mining)
- e1071 1.5-19 (svm)
- slam package 0.1-6 (sparse matrices)
- SparseM package 0.83 (sparse matrices)