Quantlets, Quantnet, Applications

Lukas Borke
Wolfgang Karl Härdle

Ladislaus von Bortkiewicz Chair of Statistics
C.A.S.E. – Center for Applied Statistics
and Economics
Humboldt–Universität zu Berlin
http://lvb.wiwi.hu-berlin.de
http://www.case.hu-berlin.de

Transparency and Reproducibility

- Required by good scientific practice
- Dormant/dead research materials/contributions
- Knowledge discovery

- Quantnet – open access code-sharing platform
  - Quantlets: program codes (R, MATLAB, SAS), various authors
  - QuantNetXploRer

Example for a search query

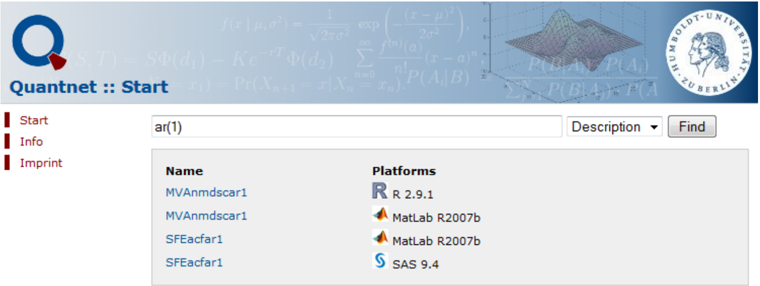


Figure 1: Search results for the search term "ar(1)" in the classical interface

Example for a search query

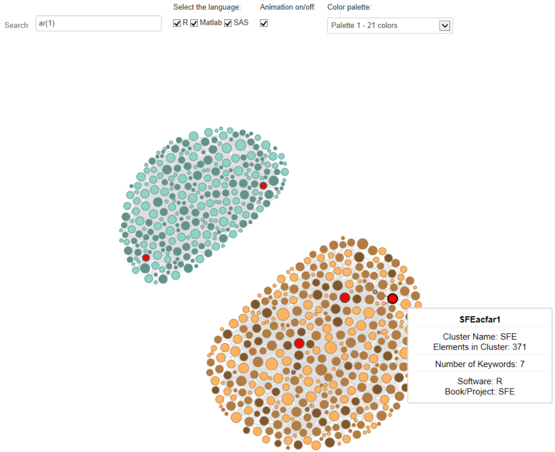


Figure 2: Search results for the search term “ar(1)“ in the graphical interface

Visualization

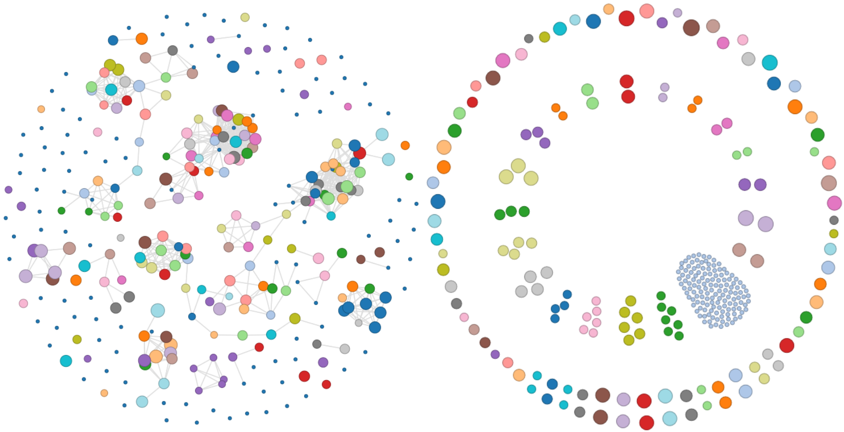


Figure 3: Quantlets from *SFE* (force directed scheme) and *MVA* (clustering scheme)

Most frequent words/terms in QNet

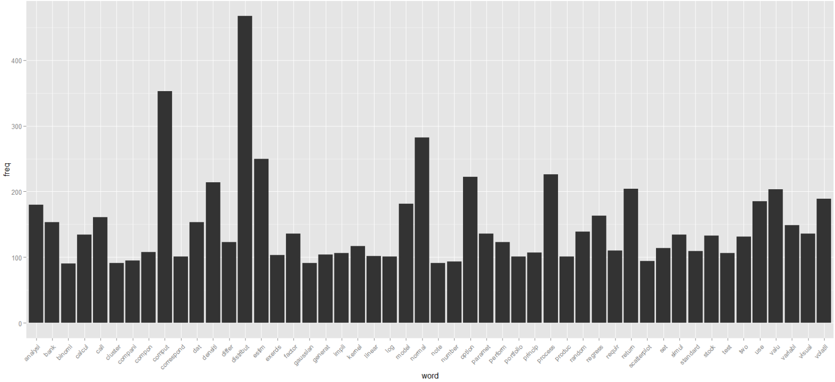
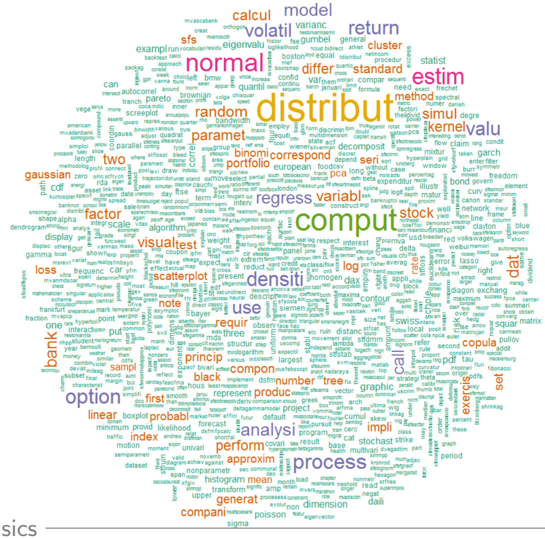


Figure 4: Words with more then 90 occurrences

Wordcloud of the words/terms in QNet



Correlation graph of the QNet terms

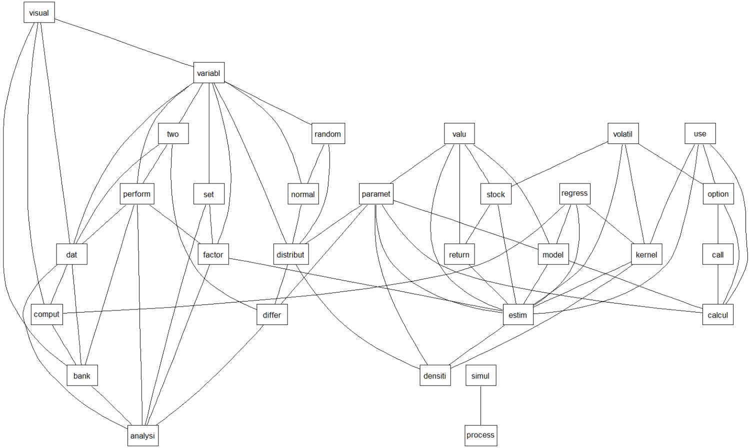


Figure 5: 30 most frequent terms with treshold = 0.1

Correlation graph of the QNet terms

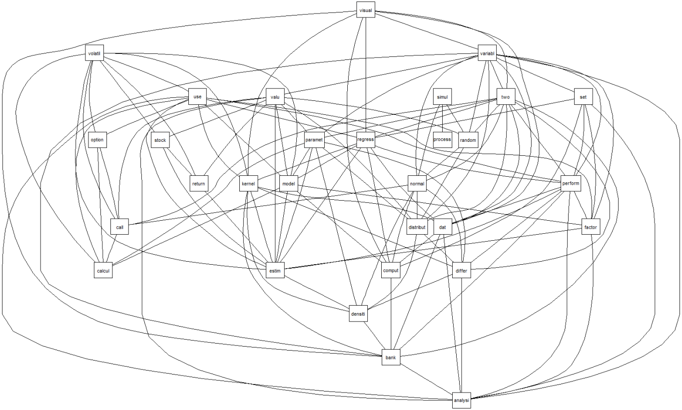


Figure 6: 30 most frequent terms with treshold = 0.05

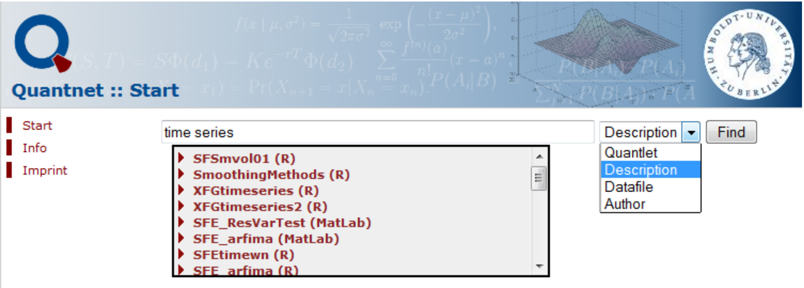
Research Goals

- Text Mining
  - Model calibration
  - Dimension reduction
  - **Semantic based Information Retrieval**
  - **Document Clustering**

- Visualization
  - Optimal projection into 2 dimensions
  - Comparison of MDS, PCA and t-SNE
  - Relationships between document similarity measures and 2D-Geometry

Outline

1. Motivation ✓
2. Interactive GUI
3. Vector Space Model (VSM)
4. Empirical results
5. Conclusion



- Searching parameters: Quantletname, Description, Datafile, Author
- Data types: R, Matlab, SAS

Integrated exploring and navigating

Projects

MSR IBT DSFM BCS SIM QR/LQR FSS TEDAS DP

Keywords: Top 30

normal distribution option
regression VaR returns PCA
call financial volatility
cdf plot kernel DSFM portfolio pdf eigenvalues density visualization
principal components random scatterplot
time series simulation
nonparametric CAT bond binomial Pareto boxplot interest rate

Click here for all Keywords...

Most Recent Quantlets

SFENormalApprox3, SFEsimCIR, SFENormalApprox1, SFENormalApprox3, SFENormalApprox2, SFENormalApprox1, SFENormalApprox4, SFEbsbm, MVAboxbank6,

MVAreturns (R 2.9.1)

Description: MVAreturns shows monthly returns of six US firms from Jan 2000 to Dec 2009.

Download File

Author: Zografia Anastasiadou

Published in: Applied Multivariate Statistical Analysis

See also: MVAportfol_IBM_Ford, MVAportfol_IBM_PanAm

Click the button to demonstrate a graph view. Graph View
Notice: This content requires Java Runtime Environment.
Java Applet and JavaScript should be allowed on your browser.

Keywords: portfolio, returns, time series

Submitted: Fri, August 05 2011 by Awdesch Melzer

Usage: -

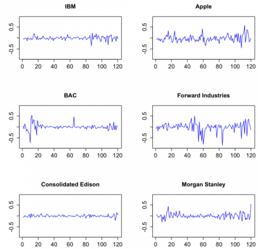
Datafiles: apple.csv, bac.csv, ed.csv, ford.csv, ibm.csv, ms.csv

Input:
- None.
- Please change working directory.

Output:
- Monthly returns of six US firms from Jan 2000 to Dec 2009.

Example:

Description: Returns of six firms from January 1999 to December 2009.



Sourcecode:

```
#clear variables and close windows
rm(list=ls(all=TRUE))
graphics.off()
setwd("C:/...") #Please change working directory
#load data
ibm<-read.csv("ibm.csv")
apple<-read.csv("apple.csv")
bac<-read.csv("bac.csv")
ford<-read.csv("ford.csv")
ed<-read.csv("ed.csv")
ms<-read.csv("ms.csv")
#compute the returns from assets
y1<-ibm[,5]
a=0
i=1
while (i<=120) {
        i=i+1
        a[i]<-(y1[i]-y1[i-1])/y1[i]
        }
#returns for IBM
x1<-a[2:121]
y2<-apple[,5]
b=0
i=1
while (i<=120) {
        i=i+1
        b[i]<-(y2[i]-y2[i-1])/y2[i]
        }
#returns for Apple
x2<-b[2:121]
y3<-bac[,5]
d=0
i=1
while (i<=120) {
        i=i+1
        d[i]<-(y3[i]-y3[i-1])/y3[i]
        }
#returns for Bank of America Corporation
x3<-d[2:121]
y4<-ford[,5]
f=0
i=1
while (i<=120) {
        i=i+1
        f[i]<-(y4[i]-y4[i-1])/y4[i]
        }
#returns for Forward Industries
x4<-f[2:121]
y5<-ed[,5]
g=0
i=1
while (i<=120) {
        i=i+1
        g[i]<-(y5[i]-y5[i-1])/y5[i]
        }
#returns for Consolidated Edison
x5<-g[2:121]
y6<-ms[,5]
```

Figure 7: Quantlet *MVAreturns* containing the search term “time series“

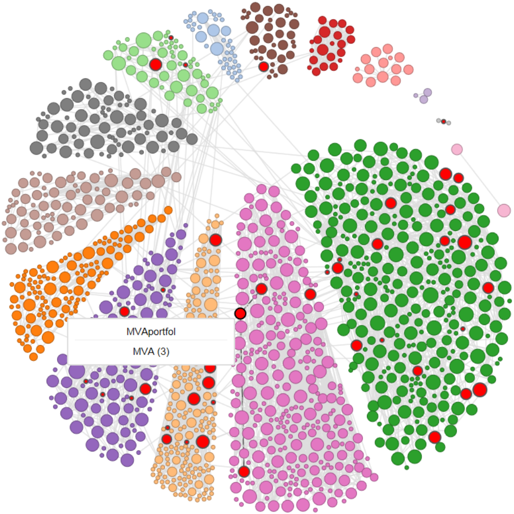
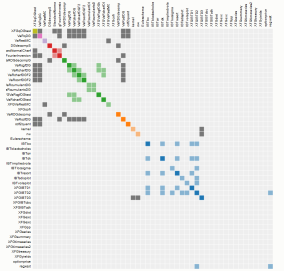


Figure 8: All Quantlets in QuantNetXploRer, search term “time series“

Vector Space Model (VSM)



- Model calibration
  - Text preprocessing
  - Text to Vector: Weighting scheme, Similarity, Distance
  - Basic VSM
  - Generalized VSM
  - LSA – Latent Semantic Analysis

Preprocessing results

	terms	Non-/sparse entries
all terms (raw)	3229	26619/5162384
after preprocessing	2385	19936/3812759
discarding tf = 1	1637	19188/2611471
discarding tf <= 2	1068	18050/1698226
discarding tf <= 3	869	17453/1379030

- Total number of documents: 1607
- Sparsity in every preprocessing step: 99%
- I select the preprocessing configuration "discarding tf <= 2": resulting a "text matrix" with 1068x1607 entries

Text to Vector

- $D = \{d_1, \dots, d_n\}$ – set of documents.
- $T = \{t_1, \dots, t_m\}$ – dictionary, i.e., the set of all different terms occurring in Quantnet.
- $tf(d, t)$ – absolute frequency of term $t \in T$ in document $d \in D$.
- $idf(t) \stackrel{\text{def}}{=} \log(|D|/n_t)$ – inverse document frequency, with $n_t = |\{d \in D | t \in d\}|$.
- $w(d) = \{w(d, t_1), \dots, w(d, t_m)\}, d \in D$ – documents as vectors in a m-dimensional space.
- $w(d, t_i)$ – calculated by a weighting scheme.

Weighting scheme, Similarity, Distance

- Salton et al. (1994): the tf-idf – weighting scheme $w(d,t)$ for $t \in T$ in $d \in D$:

$$w(d,t) = \frac{tf(d,t)idf(t)}{\sqrt{\sum_{j=1}^{m} tf(d,t_j)^2 idf(t_j)^2}}, m = |T|$$

- (normalized tf-idf) Similarity S of two documents

$$S(d_1,d_2) = \sum_{k=1}^{m} w(d_1,t_k) \cdot w(d_2,t_k) = w(d_1)^\top w(d_2)$$

- A frequently used distance measure is the Euclidian distance:

$$dist_d(d_1,d_2) \stackrel{\text{def}}{=} \sqrt{\sum_{k=1}^{m} \{w(d_1,t_k) - w(d_2,t_k)\}^2}$$

Example 1: Shakespeare's tragedies

Let $D = \{d_1, d_2, d_3\}$ be the set of documents/tragedies:

Document 1: Hamlet

Document 2: Julius Caesar

Document 3: Romeo and Juliet

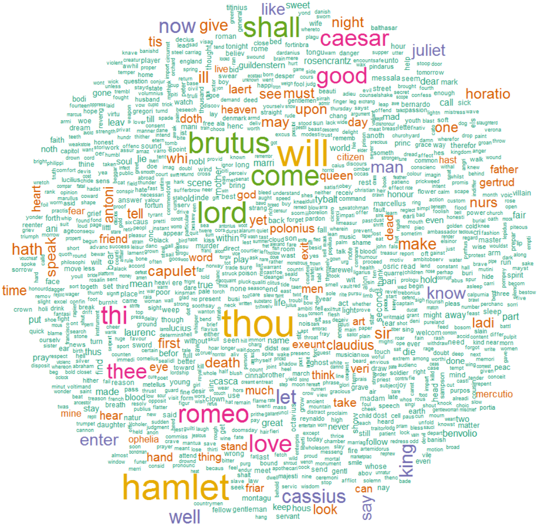


Figure 9: Wordcloud of all words (tf >= 5) in this 3 tragedies

Example 1: Shakespeare's tragedies

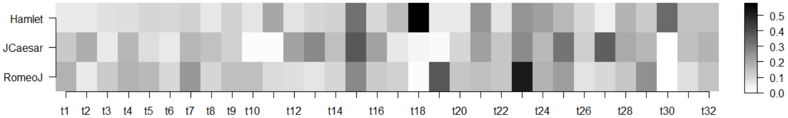


Figure 10: Heatmap of 32 words in this 3 tragedies
(among 100 most frequent)

$$\begin{aligned} T = \{ & \textit{art}, \textit{bear}, \textit{call}, \textit{day}, \textit{dead}, \textit{dear}, \textit{death}, \textit{die}, \textit{eye}, \textit{fair}, \textit{father}, \textit{fear}, \\ & \textit{friend}, \textit{god}, \textit{good}, \textit{heart}, \textit{heaven}, \textit{king}, \textit{ladi}, \textit{lie}, \textit{like}, \textit{live}, \textit{love}, \\ & \textit{make}, \textit{man}, \textit{mean}, \textit{men}, \textit{must}, \textit{night}, \textit{queen}, \textit{think}, \textit{time}\} \\ = \{ & t_1, \ldots, t_{32}\} \end{aligned}$$

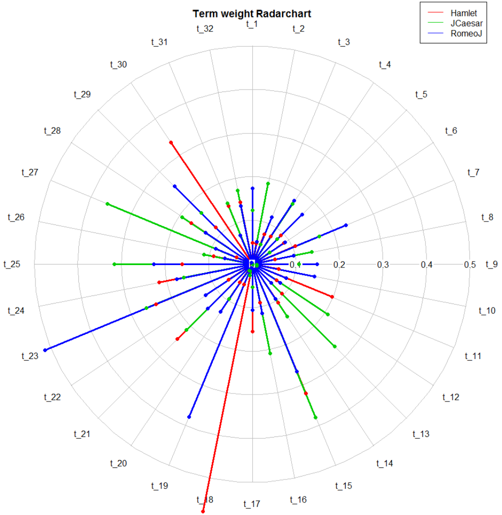


Figure 11: Weighting vectors of the 3 tragedies in a radar chart

Example 1: Shakespeare's tragedies

With the weighting vectors (32 special terms) above we get the similarity matrix:

$$M_S = \begin{pmatrix} 1 & 0.64 & 0.63 \\ 0.64 & 1 & 0.77 \\ 0.63 & 0.77 & 1 \end{pmatrix}$$

And the distance matrix:

$$M_D = \begin{pmatrix} 0 & 0.85 & 0.87 \\ 0.85 & 0 & 0.68 \\ 0.87 & 0.68 & 0 \end{pmatrix}$$

Example 1: Shakespeare's tragedies

With the weighting vectors (of all 5521 terms) in normalized TF-form we get the similarity matrix:

$$M_S = \begin{pmatrix} 1 & 0.39 & 0.46 \\ 0.39 & 1 & 0.42 \\ 0.46 & 0.42 & 1 \end{pmatrix}$$

And the distance matrix:

$$M_D = \begin{pmatrix} 0 & 1.10 & 1.04 \\ 1.10 & 0 & 1.07 \\ 1.04 & 1.07 & 0 \end{pmatrix}$$

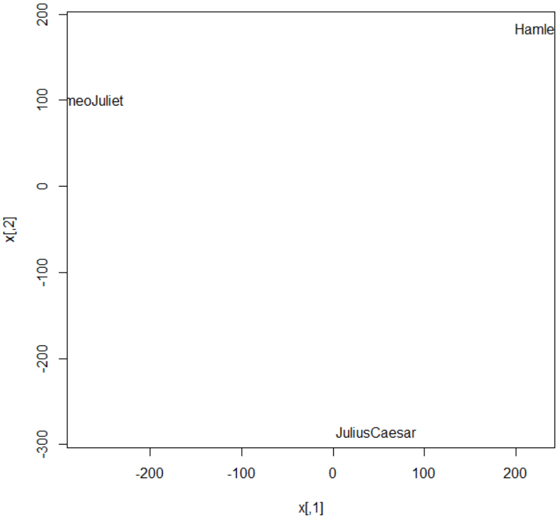


Figure 12: Outlook for the t-SNE projection into 2 dimensions

Basic VSM

- vertical vector d, indexed by terms – Document representation
- matrix $D = [d_1, \dots, d_n]$ – Document corpus representation, also called "term by document" matrix
- considering linear transformations P we get a general similarity $S(d_1, d_2) = (Pd_1)^\top (Pd_2) = d_1^\top P^\top P d_2$
- every mapping P defines another *VSM*
- $M_S = D^\top (P^\top P) D$ – similarity matrix

Example 2: tf and tf-idf similarities in BVSM

- with $P = I_m$ and $d = \{tf(d, t_1), \dots, tf(d, t_m)\}^\top$ we get the classical tf-similarity:
$M_S^{tf} = D^\top D$
- with diagonal $P(i,i)^{idf} = idf(t_i)$ and
$d = \{tf(d, t_1), \dots, tf(d, t_m)\}^\top$ we get the classical tf-idf-similarity:
$M_S^{tf-idf} = D^\top (P^{idf})^\top P^{idf} D$

Drawbacks of BVSM

- Uncorrelated/orthogonal terms in the feature space
- Documents must have common terms to be similar
- Sparseness of document vectors and similarity matrices

Question

- How to incorporate information about semantics?

Solution

- Using statistical information about term-term correlations
- Semantic smoothing

Generalized VSM – term-term correlations

- $S(d_1, d_2) = (D^\top d_1)^\top (D^\top d_2) = d_1^\top D D^\top d_2$ – the GVSM similarity
- $M_S = D^\top (D D^\top) D$ – similarity matrix
- $DD^\top$ – term by term matrix, having a nonzero ij entry if and only if there is a document containing both the i-th and the j-th terms
- terms become semantically related if co-occuring often in the same documents
- also known as a dual space method (Sheridan and Ballerini, 1996)
- when there are less documents than terms – dimensionality reduction

Generalized VSM – Semantic smoothing

- More natural method of incorporating semantics is by directly using a semantic network
- (Miller et al., 1993) used the semantic network WordNet
- Term distance in the hierarchical tree provided by WordNet gives an estimation of their semantic proximity
- (Siolas and d'Alche-Buc, 2000) have included the semantics into the similarity matrix by handcrafting the VSM matrix P
- $M_S = D^\top (P^\top P) D \ = D^\top P^2 D$ – similarity matrix

LSA – Latent Semantic Analysis

- LSA measures semantic information through co-occurrence analysis (Deerwester et al., 1990)
- Technique – singular value decomposition (SVD) of the matrix $D = U\Sigma V^\top$
- $P = U_k^\top = I_k U^\top$ – projection operator onto the first k dimensions
- $M_S = D^\top(UI_kU^\top)D$ – similarity matrix
- It can be shown: $M_S = V\Lambda_k V^\top$, with $D^\top D = V\Sigma^\top U^\top U\Sigma V^\top = V\Lambda V^\top$ and $\Lambda_{ii} = \lambda_i = \sigma_i^2$ eigenvalues of V; Λ_k consisting of the first k eigenvalues and zero-values else.

3 Models for the QuantNet

- Models – BVSM, GVSM and LSA
- Dataset – the whole Quantnet
- Documents – 1607 Quantlets

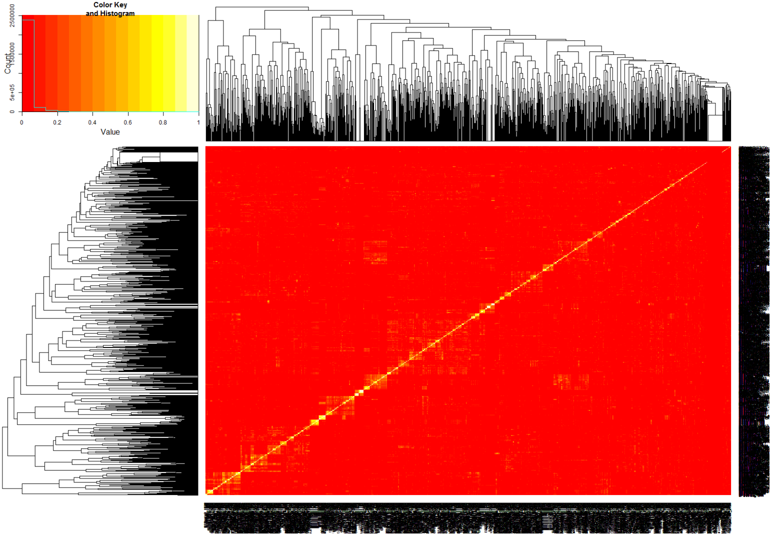


Figure 13: Heat map with 2 Dendrograms of the BVSM SimMatrix

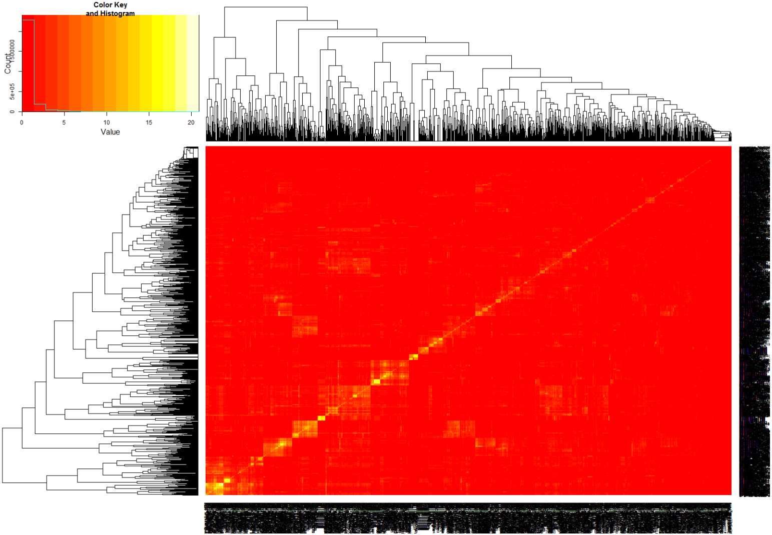


Figure 14: Heat map with 2 Dendrograms of the GVSM SimMatrix

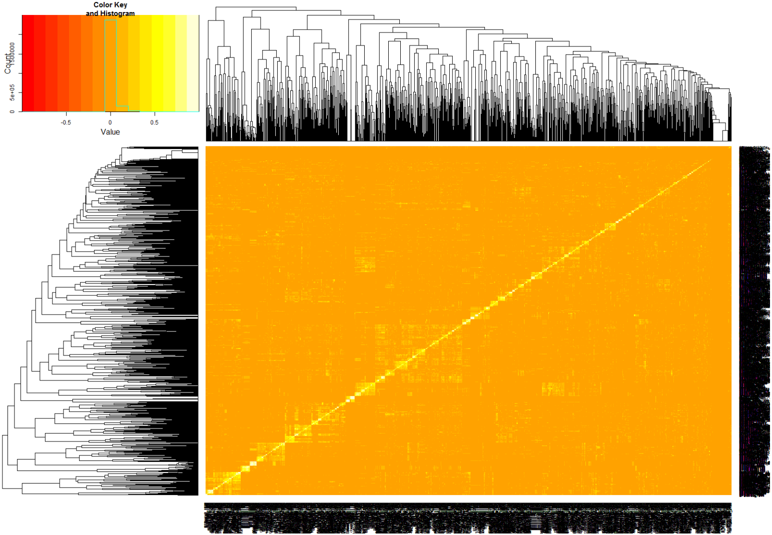


Figure 15: Heat map with 2 Dendrograms of the LSA SimMatrix

Sparseness results

	BVSM	GVSM	LSA
Sparseness TD Matrix	0.99	0.74	0.03
Sparseness Sim Matrix	0.74	0.08	0.05

Table 1: Model Performance regarding the sparseness of the "term by document"-matrix and the similarity matrix in the appropriate models.

Conclusion

- Different weighting scheme approaches and Vector Space Models allow adapted **Similarity based Knowledge Discovery**
- Incorporating **term-term Correlations** and **Semantics** significantly improves the comparison performance
- **Similarity** and **Distance** available for **Clustering** and **extended Visualization**

Quantlets, Quantnet, Applications

Lukas Borke
Wolfgang Karl Härdle

Ladislaus von Bortkiewicz Chair of Statistics
C.A.S.E. – Center for Applied Statistics
and Economics
Humboldt–Universität zu Berlin

http://lvb.wiwi.hu-berlin.de
http://www.case.hu-berlin.de

References

Borgelt, C. and Nürnberger, A.
Experiments in Term Weighting and Keyword Extraction in Document Clustering
LWA, pp. 123-130, Humbold-Universität Berlin, 2004

Bostock, M., Heer, J., Ogievetsky, V. and community
D3: Data-Driven Documents
available on d3js.org, 2014

Chen, C., Härdle, W. and Unwin, A.
Handbook of Data Visualization
Springer, 2008

References

Elsayed, T., Lin, J. and Oard, D. W.
Pairwise Document Similarity in Large Collections with MapReduce
Proceedings of the 46th Annual Meeting of the Association of Computational Linguistics (ACL), pp. 265-268, 2008

Feldman, R. and Dagan, I.
Mining Text Using Keyword Distributions
Journal of Intelligent Information Systems, 10(3), pp. 281-300, DOI: 10.1023/A:1008623632443, 1998

Gentle, J. E., Härdle, W. and Mori, Y.
Handbook of Computational Statistics
Springer, 2nd ed., 2012

References

Hastie, T., Tibshirani, R. and Friedman, J.
The Elements of Statistical Learning: Data Mining, Inference, and Prediction
Springer, 2nd ed., 2009

Härdle, W. and Simar, L.
Applied Multivariate Statistical Analysis
Springer, 3nd ed., 2012

Hotho, A., Nürnberger, A. and Paass, G.
A Brief Survey of Text Mining
LDV Forum, 20(1), pp 19-62, available on www.jlcl.org, 2005

References

Salton, G., Allan, J., Buckley, C. and Singhal, A.
Automatic Analysis, Theme Generation, and Summarization of Machine-Readable Texts
Science, 264(5164), pp. 1421-1426,
DOI: 10.1126/science.264.5164.1421, 1994

Witten, I., Paynter, G., Frank, E., Gutwin, C. and Nevill-Manning, C.
KEA: Practical Automatic Keyphrase Extraction
DL '99 Proceedings of the fourth ACM conference on Digital libraries, pp. 254-255, DOI: 10.1145/313238.313437, 1999

Data Mining: DM

DM is the computational process of discovering/representing patterns in large data sets involving methods at the intersection of **artificial intelligence**, **machine learning**, **statistics**, and **database systems**.

1. Numerical DM
2. Visual DM
3. Text Mining
   (applied on considerably weaker structured text data)

Text Mining

Text Mining or **Knowledge Discovery** from **Text** (KDT) deals with the machine supported analysis of text (Feldman et al., 1995).

It uses techniques from:

- Information Retrieval (IR)
- Information extraction
- Natural Language Processing (NLP)

and connects them with the methods of DM.

Text Mining II

Text Mining offers more models and methods like:

- Classification
- Clustering
- Latent Dirichlet Allocation (LDA) topic model
- TopicTiling

They are worth being researched and applied to the Quantnet.

Index Term Selection I

Goal: decrease the number of words for indexing, so that only the selected keywords describe the documents (Deerwester et al., 1990; Witten et al., 1999)

A simple method for keyword extracting is based on their entropy. $\forall t \in T$ the entropy is defined:

$$W(t) = 1 + \frac{1}{\log_2 |D|} \sum_{d \in D} P(d,t) \log_2 P(d,t),$$

with $P(d,t) = \frac{tf(d,t)}{\sum_{l=1}^{n} tf(d_l,t)}$

Index Term Selection II

The entropy as a measure of the importance of a word in the given domain context:

$W(t)$ is high $\Rightarrow$ prefer this t as index.

An index term selection method (fixed number of index terms) is discussed in "*Experiments in Term Weighting and Keyword Extraction in Document Clustering*" (Borgelt et al., 2004).

Similarity, Distance, Data Mining – Overview

1. Find a **formal representation** of the Quantlets
2. Find a **similarity measure** on the space of Quantlets
3. Afterwards the construction of a **distance measure** is simple:

$$distance(x, y) = \sqrt{sim(x, x) + sim(y, y) - 2 \cdot sim(x, y)}$$

Having similarity and distance $\Rightarrow$ vast amount of Data Mining, Text Mining and Visualization technics.

Distance measure

A frequently used distance measure is the Euclidian distance:

$$dist_d(d_1, d_2) \stackrel{\text{def}}{=} dist\{w(d_1), w(d_2)\} \stackrel{\text{def}}{=} \sqrt{\sum_{k=1}^{m} \{w(d_1, t_k) - w(d_2, t_k)\}^2}$$

It holds for tf-idf:

$$\cos\phi = \frac{x^\top y}{|x| \cdot |y|} = 1 - \frac{1}{2} dist^2\left(\frac{x}{|x|}, \frac{y}{|y|}\right),$$

where $\frac{x}{|x|}$ means $w(d_1)$, $\frac{y}{|y|}$ means $w(d_2)$ and $\cos\phi$ is the angle between x and y.

3 Models on 3 Datasets

- Models – BVSM, GVSM and LSA
- Datasets – 2 books, 1 project from Quantnet
- Project 1 - TEDAS: Tail Event Driven Asset Allocation (micro size - 4 Qlets)
- Book 1 - BCS: Basic Elements of Computational Statistics (low size - 48 Qlets)
- Book 2 - SFE: Statistics of Financial Markets (medium size - 337 Qlets)

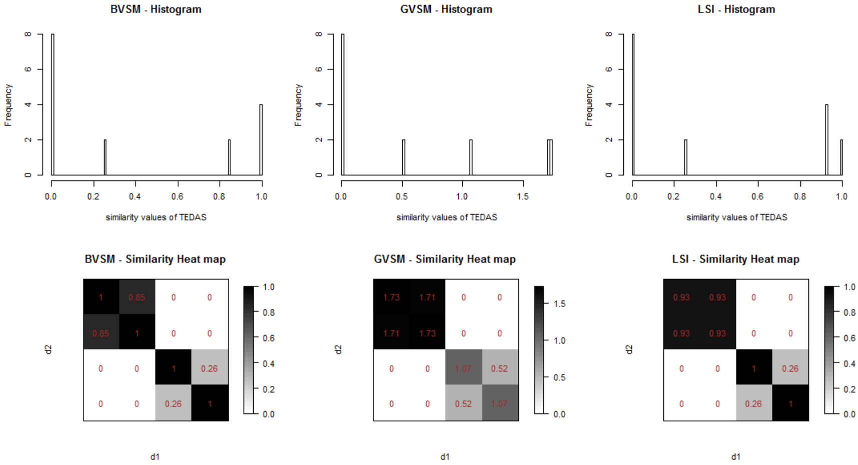


Figure 16: Model characteristics of TEDAS

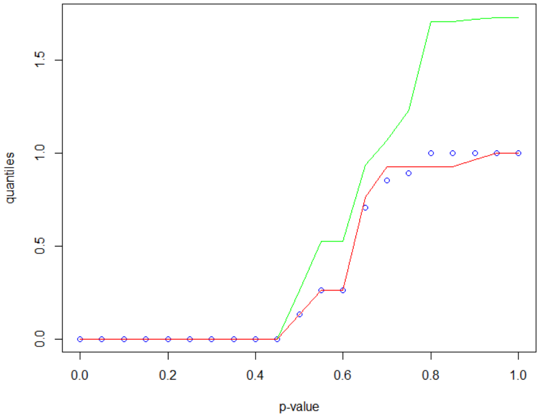


Figure 17: Quantiles of similarity values of 3 models on TEDAS

- Blue dots – BVSM; Green line – GVSM; Red line – LSA

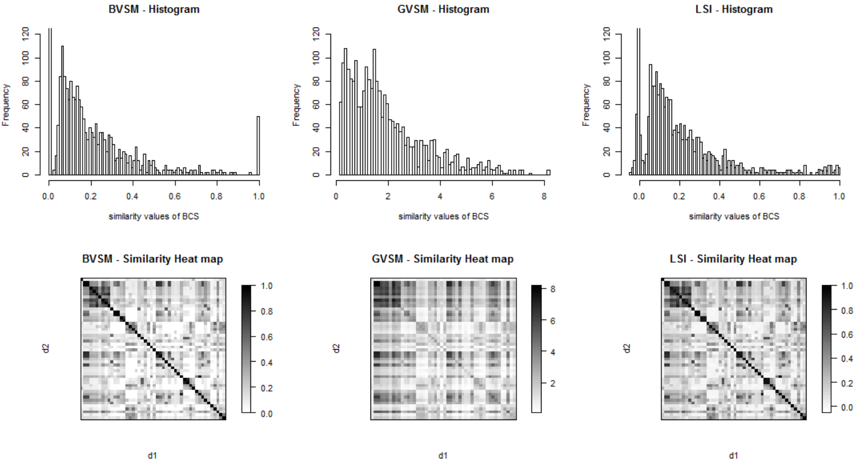


Figure 18: Model characteristics of BCS

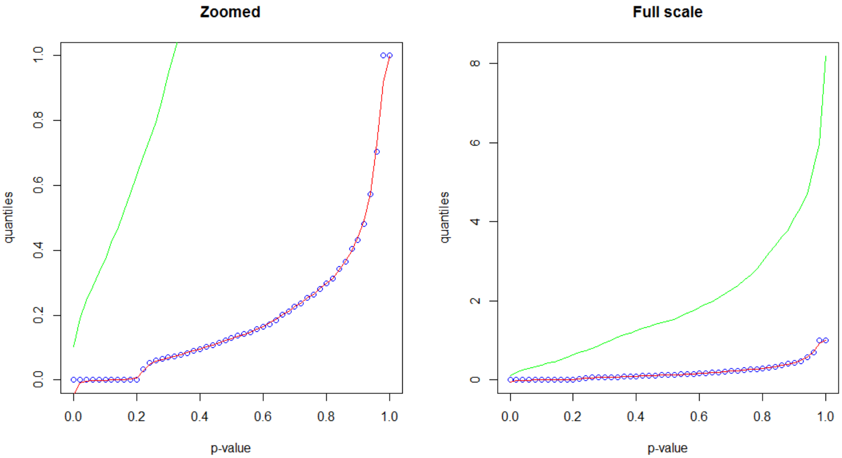


Figure 19: Quantiles of similarity values of 3 models on BCS

- Blue dots – BVSM; Green line – GVSM; Red line – LSA

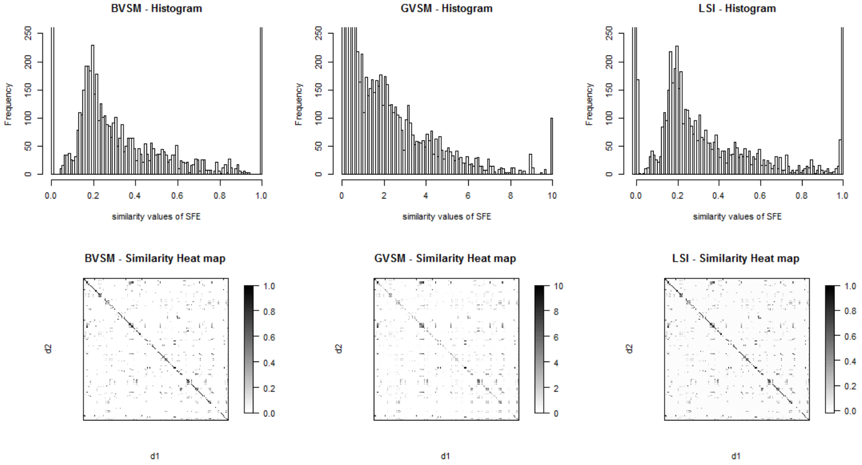


Figure 20: Model characteristics of SFE

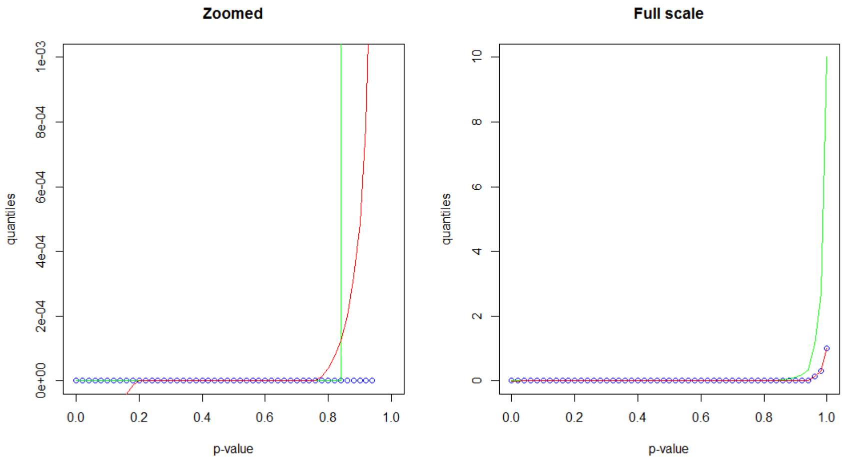


Figure 21: Quantiles of similarity values of 3 models on SFE

- Blue dots – BVSM; Green line – GVSM; Red line – LSA

Sparseness results

	TEDAS	BCS	SFE	MVA*	STF*	SFS*
BVSM	8	504	108668	75424	44576	17146
GVSM	8	0	96940	71464	44204	16612
LSA	8	262	84262	65712	43952	15400
Matrix Dim	16	2304	113569	77841	45369	18225

Table 2: Model Performance regarding the number of zero-values in the similarity matrix. MVA*, STF* and SFS* were additionally examined.

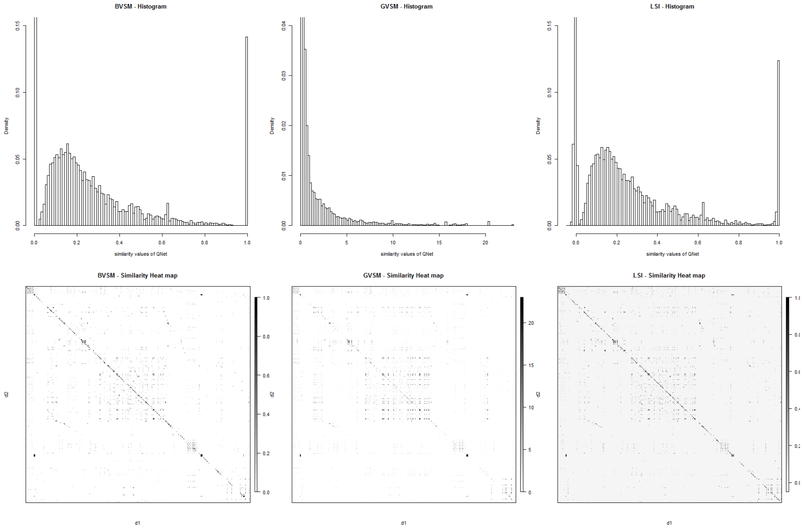


Figure 22: Model characteristics

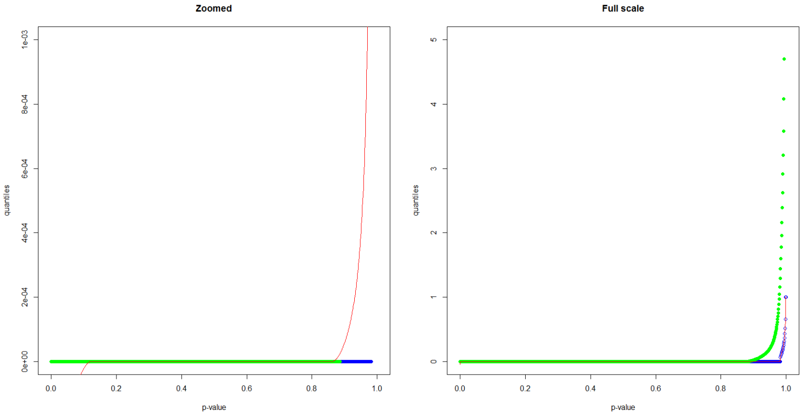


Figure 23: Quantiles of similarity values of 3 models

- Blue dots – BVSM; Green dots – GVSM; Red line – LSA