Q3-D3-LSA

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

Search query in the classical interface

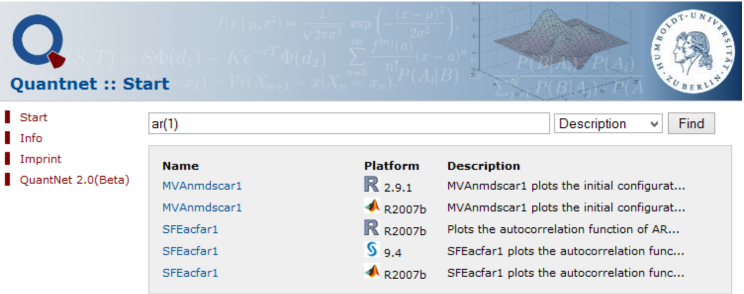


Figure 1: Search results for the search term “ar(1)“ in the traditional Google-style

Search query in the graphical interface

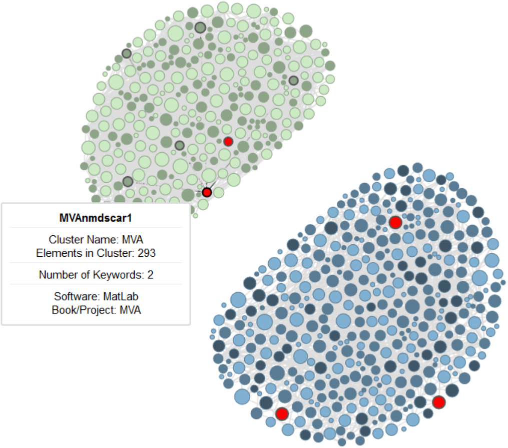


Figure 2: Search term "ar(1)". SFE in blue, MVA in green. R, Matlab, SAS in different brightness levels

Visualization

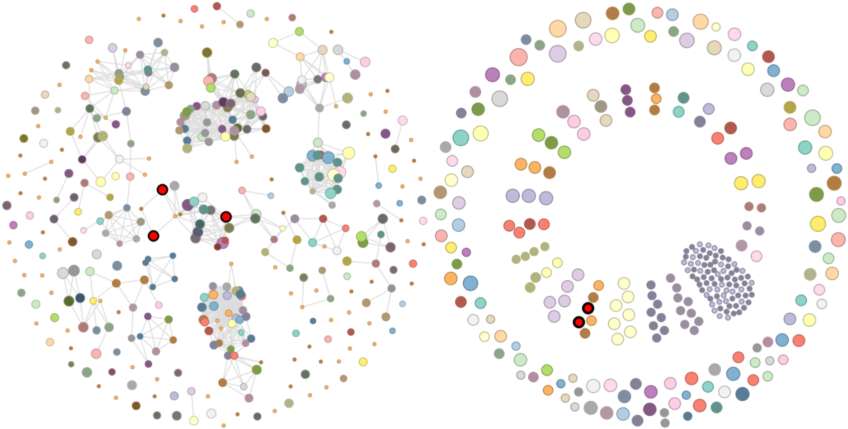


Figure 3: Quantlets from *SFE* (force directed scheme) and *MVA* (orbit clustering scheme). Clusters based on “See-also“ relations and keywords

Network graph of the QNet terms

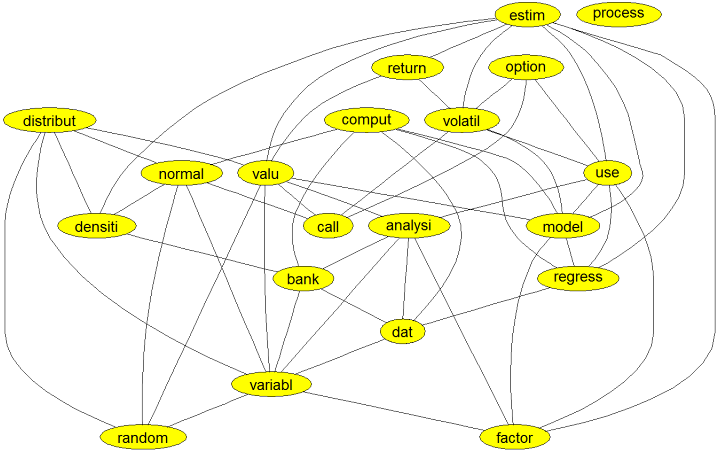


Figure 4: 20 most frequent terms with threshold = 0.05 ▸ More details

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

Objectives

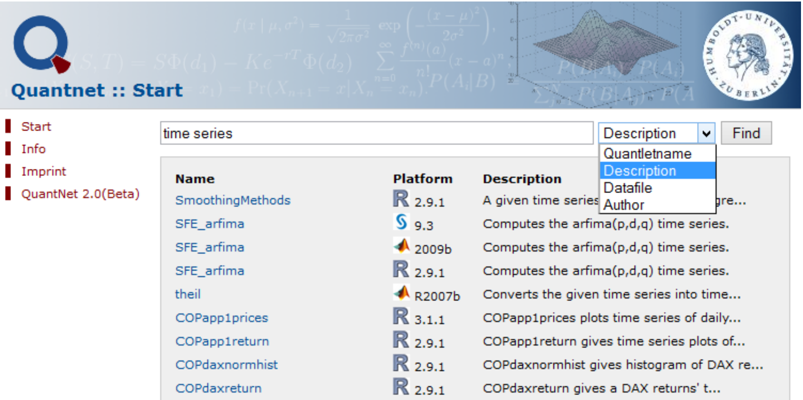
- Q3: Quantlets, Quantnet, Quantmining
  - Relevance based searching
- D3: Data-Driven Documents
  - Knowledge discovery via information visualization
- LSA: Latent Semantic Analysis
  - Semantic Embedding

Statistical Challenges

- Text Mining
  - Model calibration
  - Dimension reduction
  - Semantic based Information Retrieval
  - Document Clustering
- Visualization
  - Projection techniques

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 COP CRIX TERES DP

Keywords: Top 30

option normal visualization
distribution data visualization call
regression graphical representation simulation volatility returns
density scatterplot
PCA financial Black Scholes plot VaR
time series cdf portfolio binomial principal components kernel cluster
analysis eigenvalues
implied volatility Gumbel pdf DSFM

Click here for all Keywords...

Most Recent Quantlets **Current month stats...**

TERES_RollingWindow, CRIXBtcLtcXrp, CRIXmarket, CRIXinmark, TERES_ES_Analytical, CRIXESout, CRIXbic, AsymLaplacedis, TERES_Standardization, PAVAlgo, MSEloglikelihood, blsprice, XFGELESC, SMSfactushealth, MSEedfnormal, LAWS,

Q^2: quantlets about quantlets

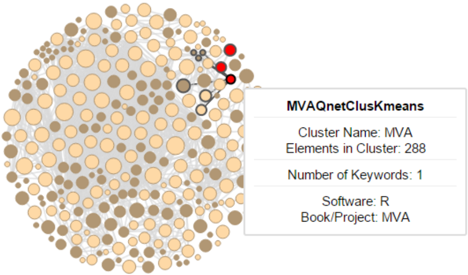


Figure 5: 3 Quantlets from *MVA* doing text mining on Quantnet

MVAQnetClusKmeans, MVAQnetClusKmeansB,
MVAQnetClusKmeansT

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: -

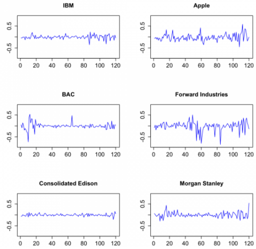
Datafiles: apple.csv, bac.csv, ed.csv, ford.csv, ibm.csv, ms.csv

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 6: Quantlet *MVAreturns* containing the search term “time series“

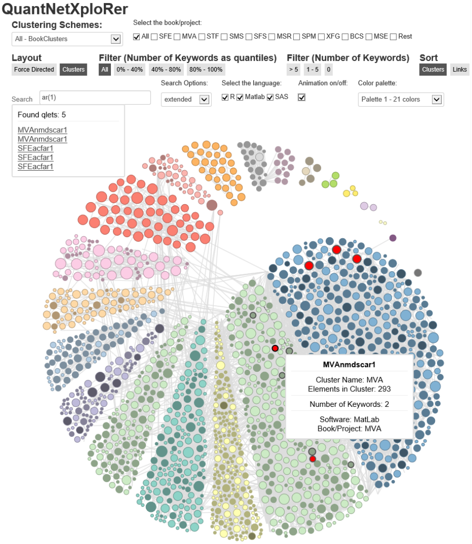
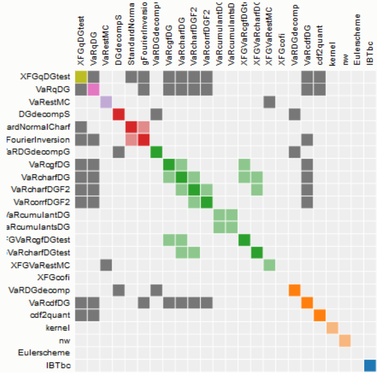


Figure 7: All Quantlets in QuantNetXploRer, search term “ar(1)“

Vector Space Model (VSM)



More details

- Model calibration
  - Text to Vector: Weighting scheme, Similarity, Distance
  - Generalized VSM (GVSM)
    Latent Semantic Analysis

Text to Vector

- $Q = \{d_1, \dots, d_n\}$ set of documents (Quantlets/Gestalten).
- $T = \{t_1, \dots, t_m\}$ dictionary (set of all terms).
- $tf(d, t)$ absolute frequency of term $t \in T$ in $d \in Q$.

	terms	Non-/sparse entries
all terms (after preprocessing)	2007	14583/2225229
discarding tf = 1	1250	13826/1381174
discarding tf <= 2	916	13158/1009098
discarding tf <= 3	735	12615/807645

Table 1: Total number of documents in QNet: 1116; term sparsity: 99%

Text to Vector

- $idf(t) \stackrel{\text{def}}{=} \log(|Q|/n_t)$ inverse document frequency, with $n_t = |\{d \in Q | t \in d\}|$.
- $w(d) = \{w(d, t_1), \ldots, w(d, t_m)\}^\top \in \mathbb{R}^m, d \in Q$, document as vector.
- $w(d, t_i)$ calculated by a weighting scheme.

- $D = [w(d_1), \ldots, w(d_n)] \in \mathbb{R}^{mxn}$, term by document matrix (TDM) .

Weighting scheme, Similarity, Distance

- Salton et al. (1994): the tf-idf – weighting scheme

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

- (normalized tf-idf) Similarity S of two documents d_1 and d_2

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

- Euclidian distance measure:

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

Example 1: German children's rhymes

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

Rhyme 1: Hänschen klein ging allein in die weite Welt hinein.

$d_1 = \{\textit{hänschen}, \textit{klein}, \textit{ging}, \textit{allein}, \textit{in}, \textit{die}, \textit{weite}, \textit{welt}, \textit{hinein}\}$

Rhyme 2: Backe, backe Kuchen, der Bäcker hat gerufen.

$d_2 = \{\textit{backe}, \textit{kuchen}, \textit{der}, \textit{bäcker}, \textit{hat}, \textit{gerufen}\}$

Rhyme 3: Die Affen rasen durch den Wald. Der eine macht den andern kalt.

$d_3 = \{\textit{die}, \textit{affen}, \textit{rasen}, \textit{durch}, \textit{den}, \textit{wald}, \textit{der}, \textit{eine}, \textit{macht}, \textit{andern}, \textit{kalt}\}$

Example 1: German children's rhymes

This implies:

$$\begin{aligned} T = \{&\textit{hänschen}, \textit{klein}, \textit{ging}, \textit{allein}, \textit{in}, \textit{die}, \textit{weite}, \textit{welt}, \textit{hinein},\\ &\textit{backe}, \textit{kuchen}, \textit{der}, \textit{bäcker}, \textit{hat}, \textit{gerufen},\\ &\textit{affen}, \textit{rasen}, \textit{durch}, \textit{den}, \textit{wald}, \textit{eine}, \textit{macht}, \textit{andern}, \textit{kalt}\}\\ = \{&t_1, \ldots, t_{24}\} \end{aligned}$$

Hence, $|D| = 3, |T| = 24$.

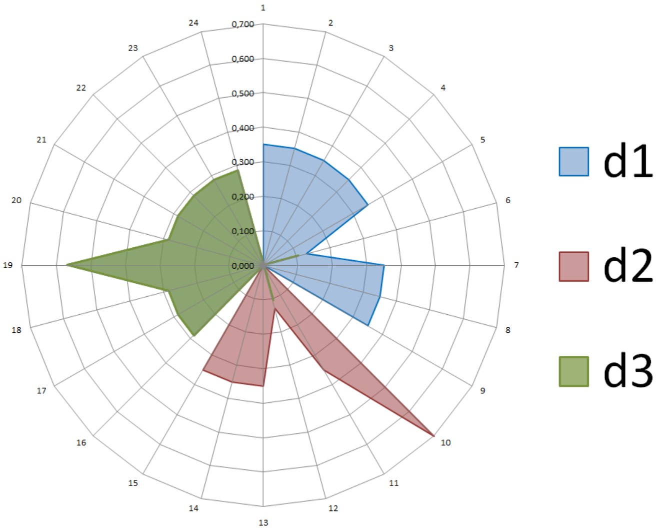


Figure 8: Weighting vectors of the 3 rhymes in a radar chart

Example 1: German children's rhymes

With the weighting vectors above we get the similarity matrix:

$$M_S = \begin{pmatrix} 1 & 0 & 0.014 \\ 0 & 1 & 0.014 \\ 0.014 & 0.014 & 1 \end{pmatrix}$$

And the distance matrix:

$$M_D = \begin{pmatrix} 0 & \sqrt{2} & 1.405 \\ \sqrt{2} & 0 & 1.405 \\ 1.405 & 1.405 & 0 \end{pmatrix}$$

Example 2: Shakespeare's tragedies

Let $Q = \{d_1, d_2, d_3\}$ be the set of documents/tragedies.
The *TDM* is a 5521×3 - matrix.

Document 1: Hamlet (total word number: 16769)

Document 2: Julius Caesar (total word number: 11003)

Document 3: Romeo and Juliet (total word number: 14237)

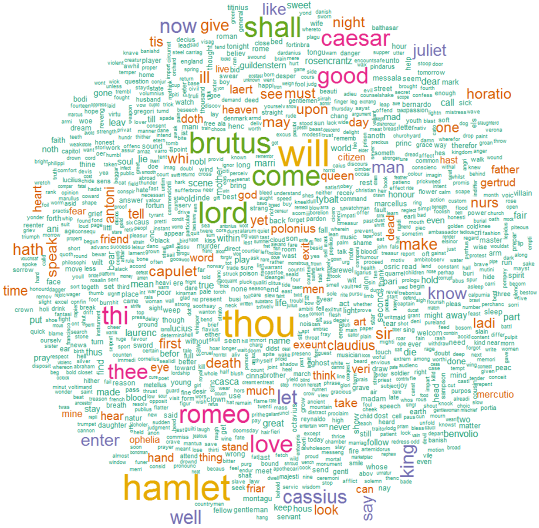


Figure 9: Wordcloud of all words (tf >= 5) in this 3 tragedies

Example 2: Shakespeare's tragedies

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

T – special vocabulary selected among 100 most frequent words.

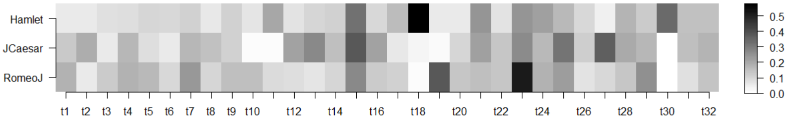


Figure 10: Heatmap of T in 3 tragedies

Radarchart visualization

Similarity matrix M_S and Distance matrix M_D for 32 special terms:

$$M_S = \begin{pmatrix} 1 & 0.64 & 0.63 \\ 0.64 & 1 & 0.77 \\ 0.63 & 0.77 & 1 \end{pmatrix} \qquad M_D = \begin{pmatrix} 0 & 0.85 & 0.87 \\ 0.85 & 0 & 0.68 \\ 0.87 & 0.68 & 0 \end{pmatrix}$$

M_S and M_D for all 5521 terms (in normalized TF-form):

$$M_S = \begin{pmatrix} 1 & 0.39 & 0.46 \\ 0.39 & 1 & 0.42 \\ 0.46 & 0.42 & 1 \end{pmatrix} \qquad M_D = \begin{pmatrix} 0 & 1.10 & 1.04 \\ 1.10 & 0 & 1.07 \\ 1.04 & 1.07 & 0 \end{pmatrix}$$

Practical observations:

- Documents must have common terms to be similar
- Sparsity of document vectors and similarity matrices
- Incorporating term-term correlations and information about semantics necessary

Example 3: NASDAQ Text Data

Let $Q = \{d_1, d_2, d_3\}$ be the set of NASDAQ news.
The TDM is a 1022×3 - matrix.

Document 1: Apple text 1 (total word number: 1729)

Document 2: J. P. Morgan (total word number: 584)

Document 3: Apple text 2 (total word number: 1012)

- NASDAQ articles source
- Data available at RDC
- Sentiment Index (Distillation of News Flow into Analysis of Stock Reactions, Zhang, J., Chen, C., Härdle, W. and Bommes E., 2015)

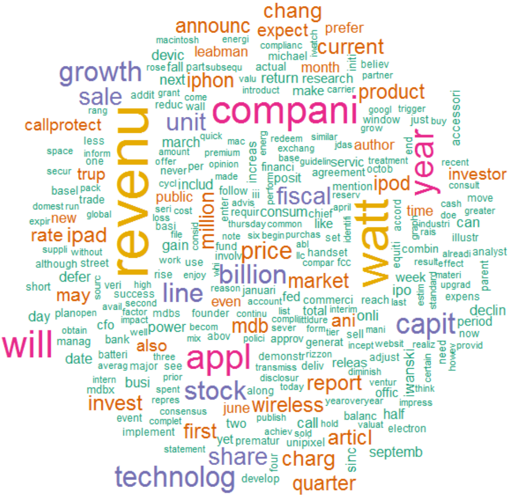


Figure 11: Wordcloud of the top 300 words in NASDAQ Texts

Similarity matrix M_S and Distance matrix M_D for:

all 1022 terms (in normalized TF-form):

$$M_S = \begin{pmatrix} 1 & 0.28 & 0.17 \\ 0.28 & 1 & 0.11 \\ 0.17 & 0.11 & 1 \end{pmatrix} \qquad M_D = \begin{pmatrix} 0 & 1.20 & 1.29 \\ 1.20 & 0 & 1.34 \\ 1.29 & 1.34 & 0 \end{pmatrix}$$

229 special terms (tf $>$ 1, in normalized TF-form):

$$M_S = \begin{pmatrix} 1 & 0.51 & 0.28 \\ 0.51 & 1 & 0.15 \\ 0.28 & 0.15 & 1 \end{pmatrix} \qquad M_D = \begin{pmatrix} 0 & 0.99 & 1.20 \\ 0.99 & 0 & 1.30 \\ 1.20 & 1.30 & 0 \end{pmatrix}$$

41 special terms (tf $>$ 2, in normalized TF-form):

$$M_S = \begin{pmatrix} 1 & 0.52 & 0.53 \\ 0.52 & 1 & 0.69 \\ 0.53 & 0.69 & 1 \end{pmatrix} \qquad M_D = \begin{pmatrix} 0 & 0.98 & 0.96 \\ 0.98 & 0 & 0.79 \\ 0.96 & 0.79 & 0 \end{pmatrix}$$

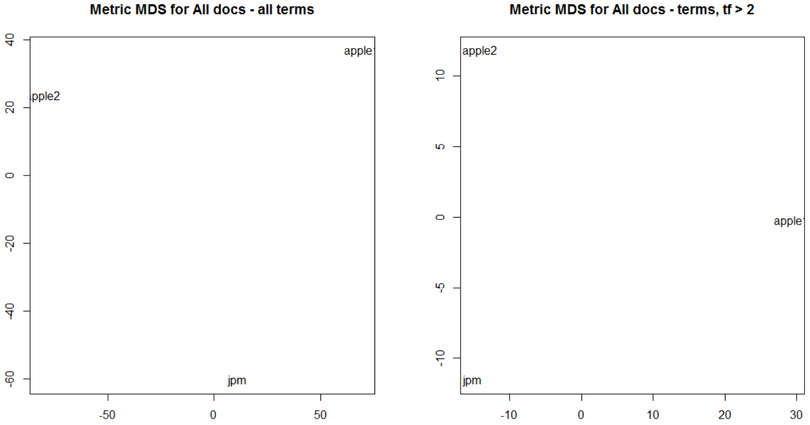


Figure 12: Metric MDS for 3 NASDAQ Texts: all vs. 41 special terms

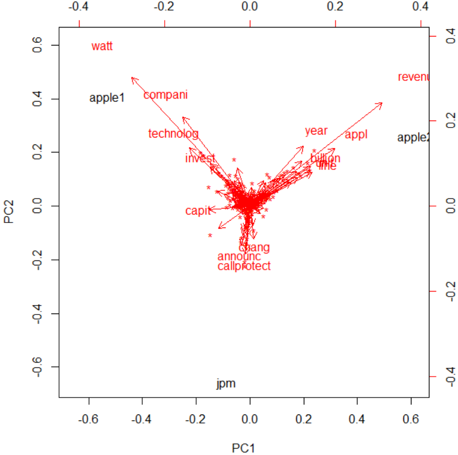


Figure 13: PCA projection of NASDAQ Texts on PC1 and PC2 (all terms)

PC1 (top 5 words): revenu, appl, line, billion, fiscal
PC2 (top 5 words): watt, revenu, compani, year, technolog

The Apple texts are well separated from J.P.M. by PC2 with words like watt, company and technology.

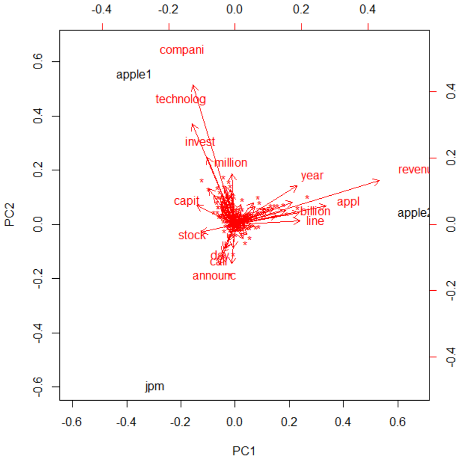


Figure 14: PCA projection of NASDAQ Texts on PC1 and PC2 (229 terms)

PC1 (top 5 words): revenu, appl, line, billion, year
PC2 (top 5 words): compani, technolog, invest, million, revenu

The Apple texts are well separated from J.P.M. by PC2 with words like company, technology and invest.

Generalized VSM (GVSM)

Generalize similarity S with a linear mapping P:

$$S(d_1, d_2) = (Pd_1)^\top (Pd_2) = d_1^\top P^\top P d_2$$

Every P defines another VSM:

$$M_S^{(P)} = D^\top (P^\top P) D$$

M_S similarity matrix, D term by document matrix

GVSM

Basic VSM (BVSM)

- $P = I_m$ and $w(d) = \{tf(d, t_1), \ldots, tf(d, t_m)\}^\top$
  classical tf-similarity: $M_S^{tf} = D^\top D$
- diagonal $P(i, i)^{idf} = idf(t_i)$ and
  $w(d) = \{tf(d, t_1), \ldots, tf(d, t_m)\}^\top$
  classical tf-idf-similarity: $M_S^{tfidf} = D^\top (P^{idf})^\top P^{idf} D$
- starting with
  $w(d) = \{tf(d, t_1) idf(t_1), \ldots, tf(d, t_m) idf(t_m)\}^\top$
  and letting $P = I_m$:
  $M_S^{tfidf} = D^\top I_m D = D^\top D$

GVSM

- Term-Term correlations: ▸ GVSM(TT)
  - $P = D^\top$, $M_S^{TT} = D^\top (DD^\top) D$
  - $DD^\top$: term by term correlation matrix
- Latent Semantic Analysis ▸ LSA
  - $D = U \Sigma V^\top$: singular value decomposition (SVD)
  - $P = U_k^\top = I_k U^\top$: projection onto the first k dimensions
  - $M_S^{LSA} = D^\top (U I_k U^\top) D$
  - The k dimensions as the main semantic components and $U_k U_k^\top = U I_k U^\top$ their correlation.

Power of LSA

- Highest-performing variants of LSA-based search algorithms perform as well as PageRank-based Google search engine (Miller et al., 2009)
- In half of the studies with 30 sets LSA performance equal to or better than that of humans (Bradford, 2009)
- Positive correlation of LSA comparable with the more sophisticated WordNet based methods and also human ratings ($r = 0.88$), in Mohamed, M. and Oussalah, M., 2014

Latent Semantic Space

1. Create directly by using the quantlets, matrix $D =$ the set of quantlets
2. First train by domain-specific and generic background documents
   - Fold in Quantlets into the semantic space after the previous SVD process
   - Gain of higher retrieval performance (bigger vocabulary set, more semantic structure)
   - Chapters or sub-chapters from our e-books well suited

3 Models for the QuantNet

- Models: BVSM, GVSM(TT) and GVSM(LSA)
  - 3 configurations in LSA with dimension parameter k equal to 300, 155 (50% of the weight of all singular values) and 50
- Dataset: the whole Quantnet
- Documents: 1116 Gestalten (from 1627 individual Quantlets)
- Clustering methods: k-Means, k-Medoids, HCA
- Cluster validation: Calinski, Silhouette criterion and topic labeling
- Information retrieval: Recall vs. Precision (5 sample queries)

Sparsity results

	BVSM	TT	LSA:300	LSA:155(50%)	LSA:50
TDM	0.99	0.71	0.51	0.51	0.48
M_S	0.71	0.08	0.38	0.40	0.38

Table 2: Model Performance regarding the sparsity of the term by document matrix TDM and the similarity matrix M_S in the appropriate models (weighting scheme: tf-idf normed).

Sparsity: the ratio of the number of zero entries to the total number of entries of a matrix. In general: the lower the sparsity, the better.

More details about sparsity and similarity structure in
▸ BVSM ▸ GVSM(TT) ▸ GVSM(LSA:300) ▸ GVSM(LSA:155(50%)) ▸ GVSM(LSA:50)

Optimal number of clusters - k-Means

	BVSM	TT	LSA:300	LSA:155	LSA:50
NC: Best	3	2	2	2	3
NC: 2nd-Best	5	4	7	7	7
NC: 3rd-Best	12	7	11	11	10

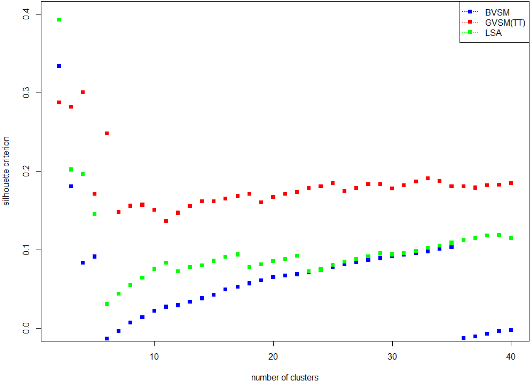
Table 3: NC: number of clusters, algorithm: k-Means, criterion: Calinski, tested size range: 2 to 25, iterations: 100 per cluster size

More details about k-Means

More details about the Calinski criterion in

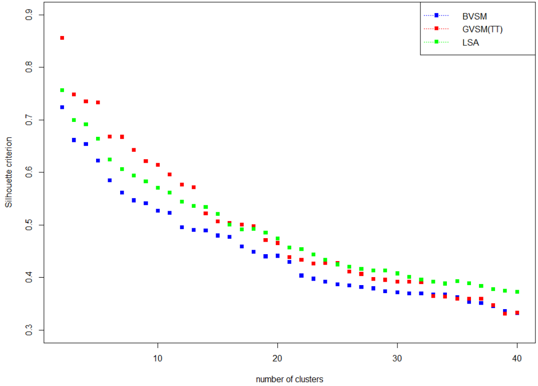
BVSM GVSM(TT) LSA(300) LSA LSA(50)

Optimal model and number of clusters - k-Medoids

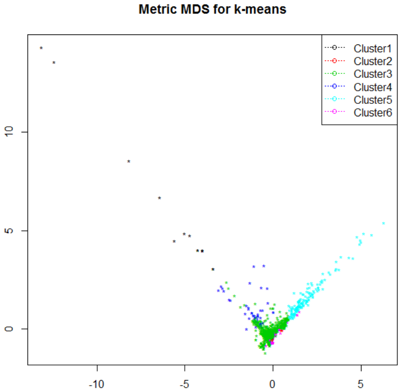


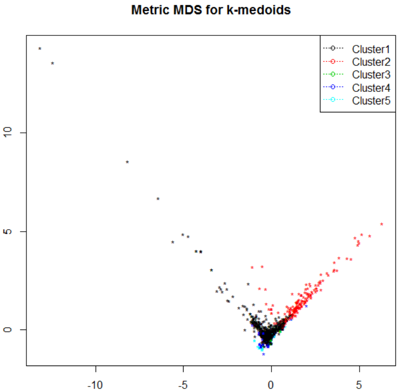
Algorithm: k-Medoids in 3 models, criterion: Silhouette (higher values are better), tested size range: 2 to 40.

Optimal model and number of clusters - hierarchical clustering



Algorithm: hierarchical clustering(HCA) in 3 models, criterion: Silhouette (higher values are better), tested size range: 2 to 40. More ...





LSA

K-Means-Clusters: 1: factor analysi load 2: bond cat homogen 3: comput option estim
4: compon princip pca 5: distribut normal densiti 6: process simul stochast

K-Medoids-Clusters: 1: absolut accord acf 2: distribut normal empir 3: bond cat homogen
4: call option black 5: stock index dax

More clustering and models

Dendrogram (all Qlets) cut in 20 clusters

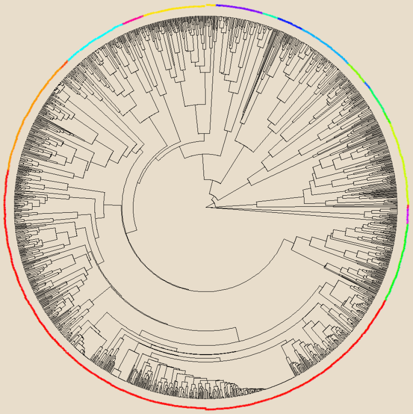


Figure 15: Created by hierarchical clustering (ward-method) in LSA model

Search queries in 3 models

Queries: q_1 =„linear regression“, q_2 = „series“, q_3 = „auto regressive“, q_4 = „spectral clustering“, q_5 = „black scholes“.

Term by document matrix of the queries in TF-form:

	q1	q2	q3	q4	q5
auto	0	0	1	0	0
black	0	0	0	0	1
cluster	0	0	0	1	0
linear	1	0	0	0	0
regress	1	0	1	0	0
schole	0	0	0	0	1
seri	0	1	0	0	0
spectral	0	0	0	1	0

Search queries - First performance results wrt. Recall

	BVSM	TT	LSA
q1: linear regression	0	12	4
q2: series	0	4	4
q3: auto regressive	0	11	1
q4: spectral clustering	0	16	1
q5: black scholes	3	6	4

Table 4: Number of Qlet-names retrieved/recalled by 3 models; weighting scheme: tf-idf normed; measure: cosine similarity; similarity threshold for recall: 0.7

Search queries - Recall vs. Precision

q2 = „series“

BVSM: no hits

GVSM(TT):
manh (0.89), theil (0.83), ultra (0.82), legendre (0.76)

LSA:
manh (0.93), theil (0.86), legendre (0.85), ultra (0.83)

Conclusion:

- GVSM(TT) and LSA provide the same hits
- LSA uniformly better than GVSM(TT) in the degree of similarity

Search queries - Recall vs. Precision

q3 = „auto regressive"

BVSM: no hits

GVSM(TT):

MSEanovapull (0.77), SPMsplineregression (0.77), SPMspline (0.76), MSEivgss (0.74), MSEglmest (0.73), MSElogit (0.73), SPMengelcurve1 (0.73), SPMknnreg (0.72), SPMcps85lin (0.71), SPMengelcurve (0.71), SPMkernelregression (0.71)

LSA:

MSEanovapull (0.83)

Conclusion:

- Quantity is not quality, most hits of GVSM(TT) deal with „linear regression"

Search queries - Recall vs. Precision

q5 = „black scholes“

BVSM:
blsprice_1745 (1.00), blsprice_1746 (1.00), blsprice_1747 (1.00)

GVSM(TT):
blsprice_1745 (1.00), blsprice_1746 (1.00), blsprice_1747 (1.00), blspricevec (0.86), IBTblackscholes (0.74), blackscholes (0.72)

LSA:
blsprice_1745 (1.00), blsprice_1746 (1.00), blsprice_1747 (1.00), blspricevec (0.79)

Conclusion:

- GVSM(TT) $>_{recall}$ LSA $>_{recall}$ BVSM
- Very high Precision in all models, but not the Recall

Similarities of Qlet samples in 3 models

Models from left to right: BVSM, GVSM(TT), GVSM(LSA).

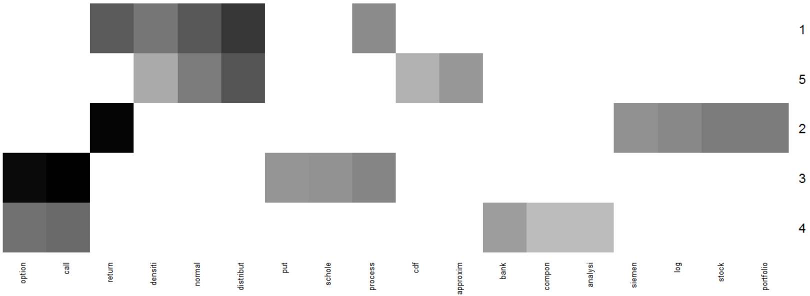
Sample of Qlets: STFloss, MVApcp2, adfreg.

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0.06 & 0 \\ 0.06 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0.24 \\ 0 & 0.24 & 1 \end{pmatrix}$$

Sample of Qlets: LOB visual, VaRcumulantsDG, BCS_MLRleaps.

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0.06 & 0.1 \\ 0.06 & 1 & 0.1 \\ 0.1 & 0.1 & 1 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0.01 & 0.07 \\ 0.01 & 1 & 0.02 \\ 0.07 & 0.02 & 1 \end{pmatrix}$$

LSA - A first insight into the interpretation



The first 5 PC's of the semantic space. Top 5 words of every PC colored

PC1 (6.1): distribut normal return densiti process
PC2 (5.2): return stock portfolio log siemen
PC3 (5.1): call option process schole put
PC4 (5.0): call option bank compon analysi
PC5 (4.8): distribut normal approxim densiti cdf

Conclusion

- **Similarity** and **Distance** available for Clustering, Information Retrieval and extended Visualization

- Different model configurations allow adapted Similarity based Knowledge Discovery

- Incorporating **term-term Correlations** and **Semantics**:
  - Sparsity reduction
  - more recall/precision (IR)
  - finding semantic topics and labels (clusters)

Future Perspectives

- Comparison and Visualization of GVSM techniques (in particular GVSM(TT) and LSA)
- Relevance based search by cluster analysis (fitting the optimal model and clustering method)
- Implementation of the „optimal“ method into QNet

Q3-D3-LSA

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

Network graph

- Rgraphviz (Gentry et al., 2014) from the BioConductor repository for R (bioconductor.org) is used to plot the network graph that displays the correlation between chosen words in the corpus. Here we choose 20 of the most frequent words as the nodes and include links between words when they have at least a correlation of 0.05.

Back to the Network Graph

Matrix diagram

- This matrix diagram visualizes connections between Qlets wrt. kategory "See also" in the book XFG in the QNet. Each colored cell represents two Qlets that are connected via "See also"; darker cells indicate Qlets that have connections to other QLets more frequently. Additionally, the colors are chosen corresponding to similar keywords in the Qlets. Use the drop-down menu to reorder the matrix and explore the data.

Back to the Matrix diagram

Partitional Clustering methods

- K-Means clustering aims to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster.
- K-medoids clustering is related to the k-means. Both attempt to minimize the distance between points labeled to be in a cluster and a point designated as the center of that cluster. In contrast to the k-means, k-medoids chooses datapoints as centers (medoids) and works with an arbitrary matrix of distances.

▸ Back to k-Means results ▸ Back to K-Medoids results

Hierarchical Clustering methods

- Hierarchical cluster analysis (HCA) is a method which seeks to build a hierarchy of clusters using a set of dissimilarities for the *n* objects being clustered. It uses agglomeration methods like "ward.D", "ward.D2", "single", "complete", "average".
- Choosing k using the Silhouette. The silhouette of a datum is a measure of how closely it is matched to data within its cluster and how loosely it is matched to data of the neighbouring cluster, i.e. the cluster whose average distance from the datum is lowest. A silhouette close to 1 implies the datum is in an appropriate cluster, while a silhouette close to -1 implies the datum is in the wrong cluster.

Back to K-Medoids results Back to HCA results

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.

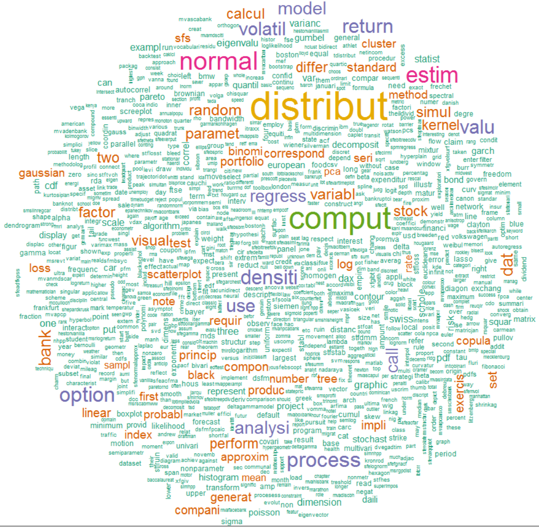
Text Mining II

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.

Wordcloud of the words/terms in QNet



Most frequent words/terms in QNet

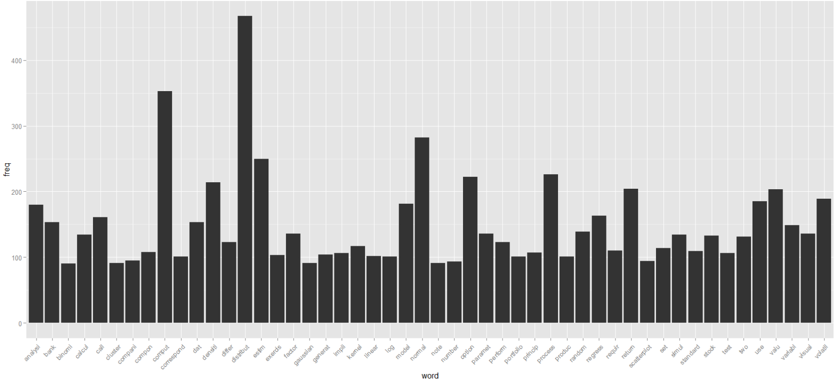


Figure 16: Words with more then 90 occurrences

Correlation graph of the QNet terms

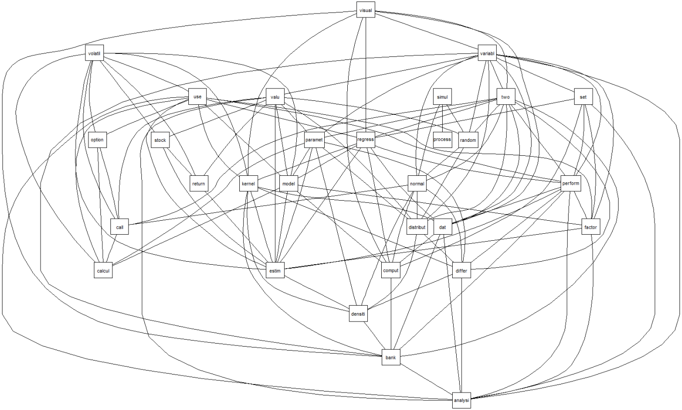


Figure 17: 30 most frequent terms with threshold = 0.05

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.

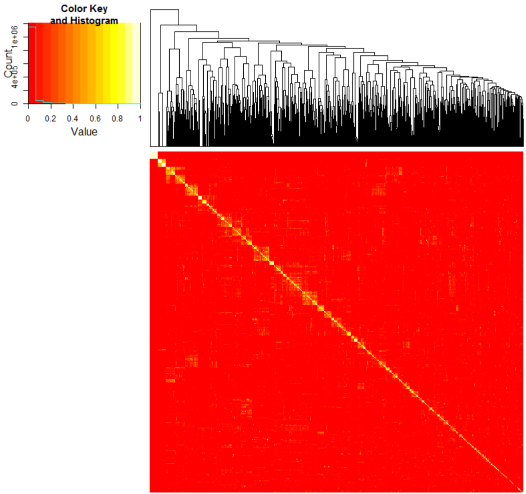


Figure 18: Heat map with Dendrogram - BVSM SimMatrix

▸ Back to sparsity results

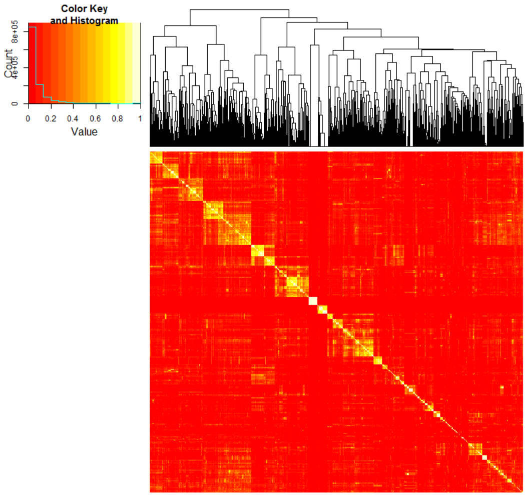


Figure 19: Heat map with Dendrogram - GVSM(TT) SimMatrix

Back to sparsity results

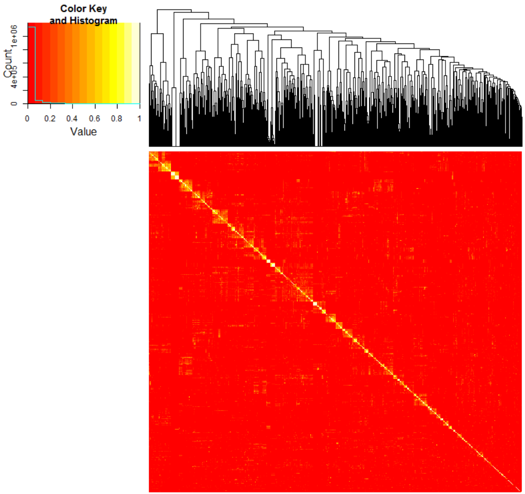


Figure 20: Heat map with Dendrogram - LSA:300 SimMatrix

Back to sparsity results

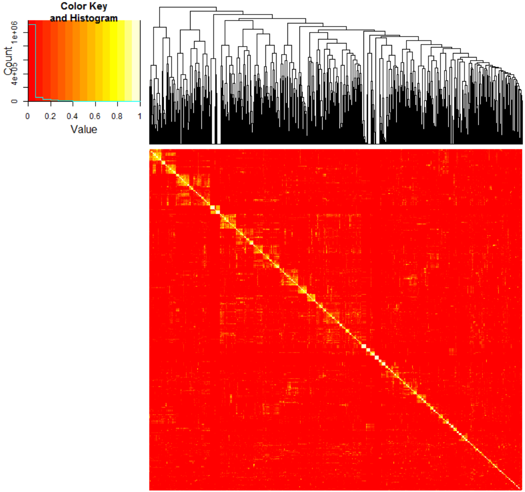


Figure 21: Heat map with Dendrogram - LSA:155(50%) SimMatrix

Back to sparsity results

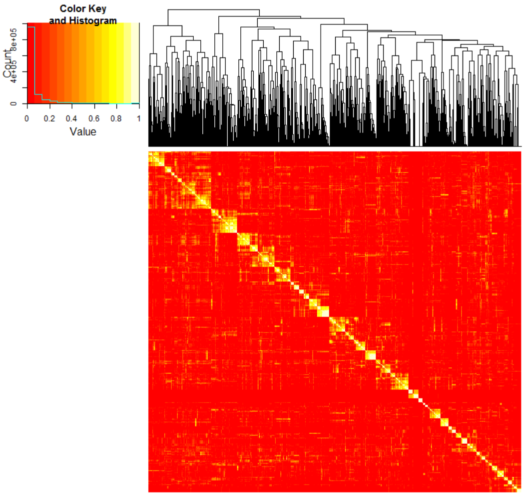


Figure 22: Heat map with Dendrogram - LSA:50 SimMatrix

Back to sparsity results

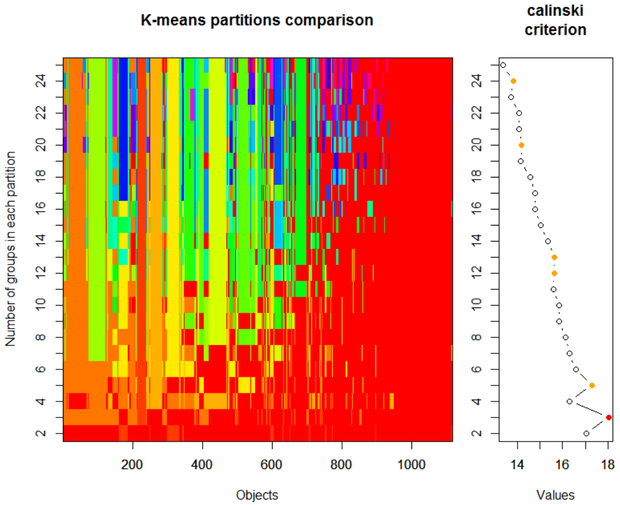


Figure 23: Cascading from a small to a large number of groups: BVSM

Back to Calinski results

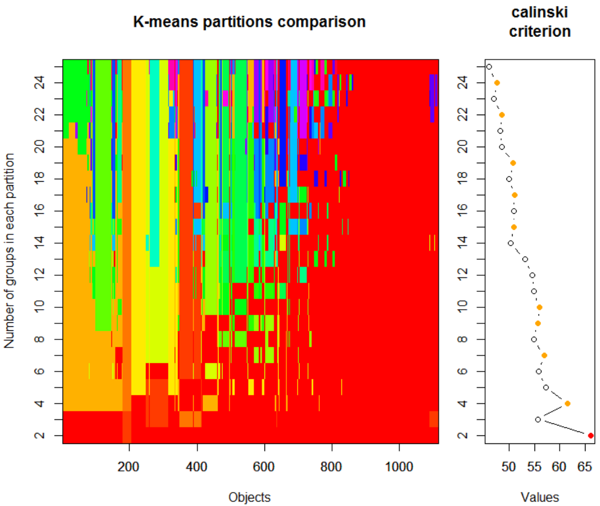


Figure 24: Cascading from a small to a large number of groups: GVSM(TT)

▸ Back to Calinski results

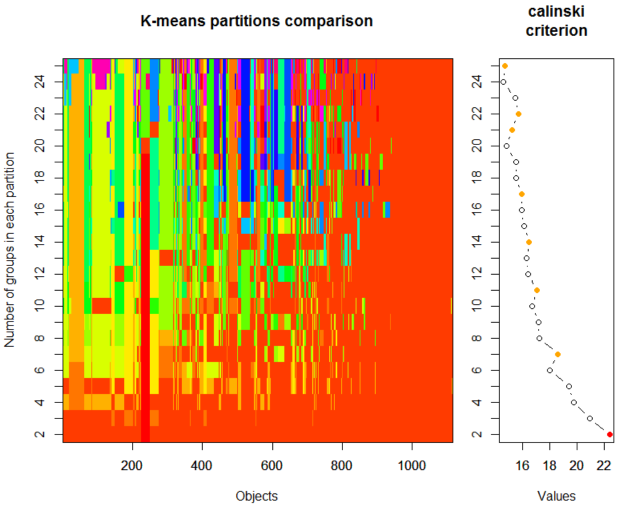


Figure 25: Cascading from a small to a large number of groups: LSA(300)

▸ Back to Calinski results

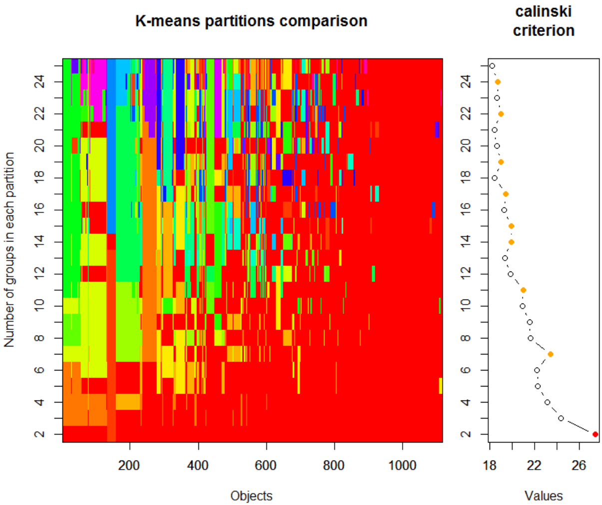


Figure 26: Cascading from a small to a large number of groups: LSA

Back to Calinski results

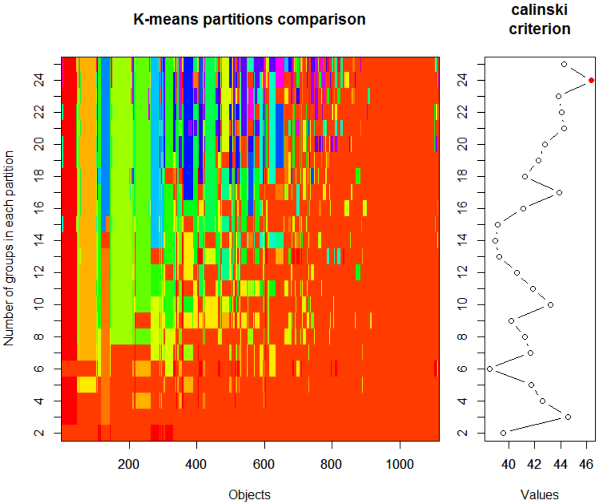


Figure 27: Cascading from a small to a large number of groups: LSA(50)

Back to Calinski results

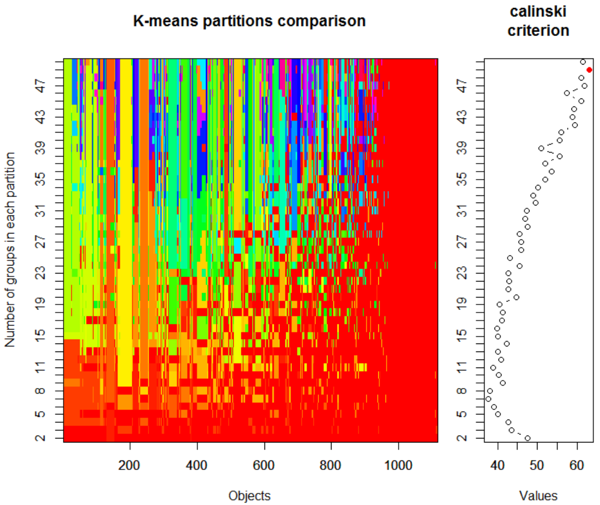
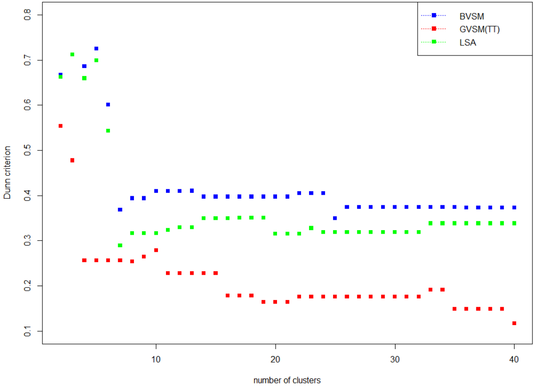


Figure 28: Cascading from a small to a large number of groups: LSA(50)

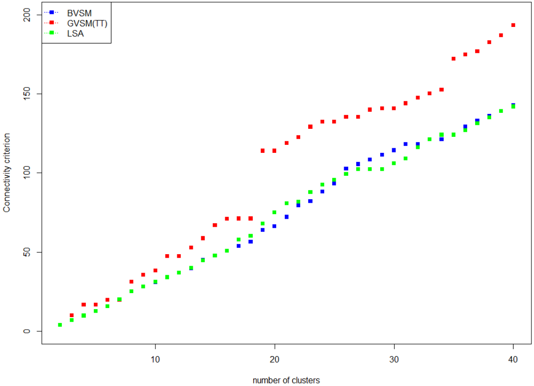
Back to Calinski results

Optimal model and number of clusters - hierarchical clustering



Algorithm: hierarchical clustering in 3 models, criterion: Dunn (higher values are better), tested size range: 2 to 40. Back

Optimal model and number of clusters - hierarchical clustering



Algorithm: hierarchical clustering in 3 models, criterion: Connectivity (lower values are better), tested size range: 2 - 40

A first insight into the Cluster Validation

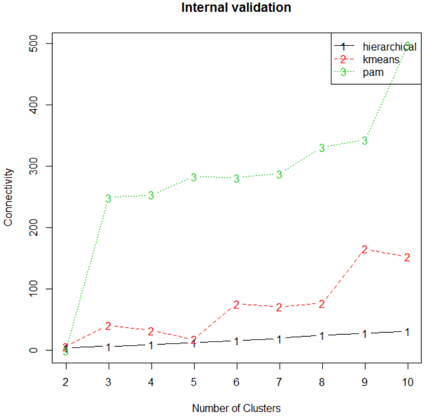


Figure 29: BVSM: Connectivity measure - lower values are better

A first insight into the Cluster Validation

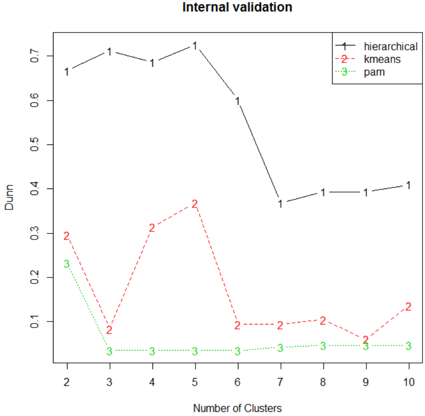


Figure 30: BVSM: Dunn measure - higher values are better

A first insight into the Cluster Validation

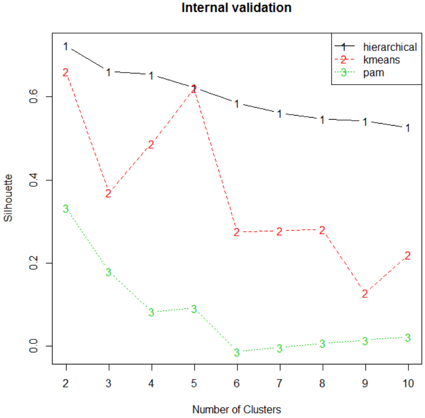


Figure 31: BVSM: Silhouette - higher values are better

Back

Drawbacks of the classical tf-idf approach

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

Solution

- Using statistical information about term-term correlations
- Incorporating information about semantics (Semantic smoothing)

GVSM – term-term correlations

- $P = D^\top$
- $S(d_1, d_2) = (D^\top d_1)^\top (D^\top d_2) = d_1^\top D D^\top d_2$
- $M_S^{TT} = D^\top (D D^\top) D$
- $DD^\top$ – 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

Back to GVSM(TT)

GVSM – Latent Semantic Analysis (LSA)

- 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
- $M_S = D^\top (U I_k U^\top) D$ – 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.

Back to GVSM(LSA)

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

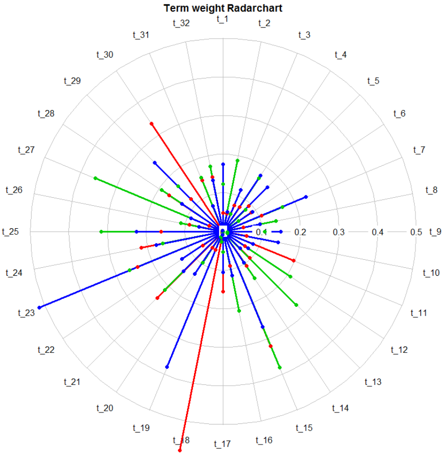
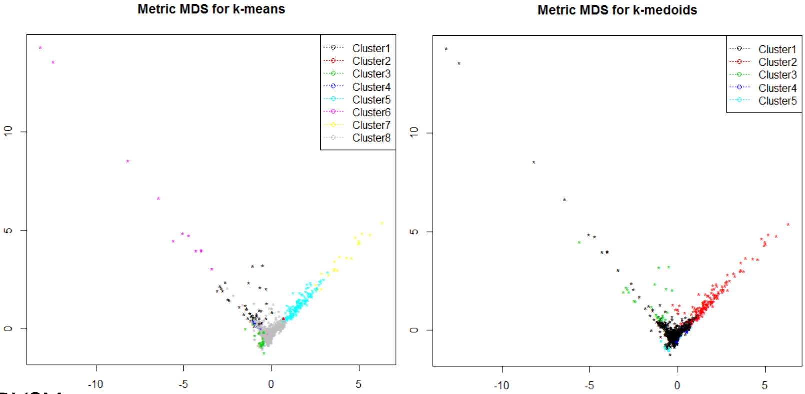


Figure 32: Weighting vectors of the tragedies (Hamlet, Julius Caesar, Romeo and Juliet) in a radar chart. Highest values: “king“ (t_{18}), “queen“ (t_{30}), “good“ (t_{15}), “men“ (t_{27}), “love“ (t_{23}), “ladi“ (t_{19}),

Back to Heatmap

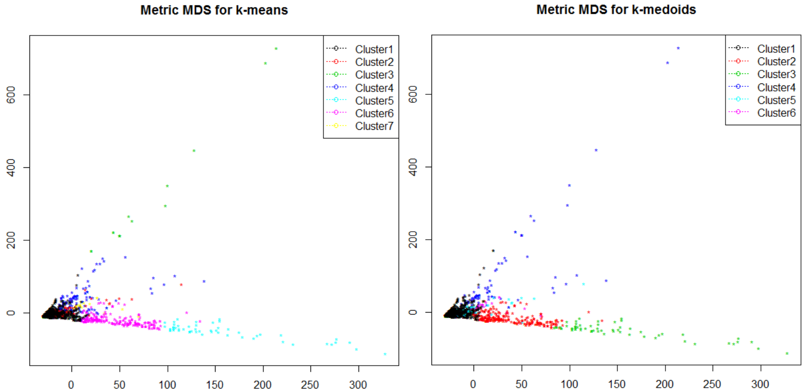


BVSM

K-Means-Clusters: 1: compon princip pca 2: figur panel left 3: volatil option impli 4: decomposit correspond factori 5: distribut normal densiti 6: factor analysi load 7: distribut normal pdf 8: comput process estim

K-Medoids-Clusters: 1: absolut accord acf 2: distribut empir normal 3: bank compon eigenvalu 4: bond cat amount burr 5: stock compani dax

Back



GVSM(TT)

K-Means-Clusters: 1: SIMqrL1 XFGLSK SFEVaRcopulaSIM2ptv 2: XFGiv03 XFGLSK XFGiv00 3: SMSfacthletic SMSfactbank SMSfactsigma 4: SMSclusbank3 SMSclusbank2 SMScluscomp 5: BCS_tQQplots BCS_Binnorm BCS_StablePdfCdfSpecial 6: BCS_tQQplots BCS_StablePdfCdfSpecial BCS_HAC 7: SFEmvol02 SFEmvol03 SFEgarchest

K-Medoids-Clusters: 1: acf ADcritBurr ADcritln 2: BCS_Binnorm BCS_ChiNormApprox BCS_tQQplots 3: BCS_tQQplots MVAedfnormal BCS_Binnorm 4: MVAnpcatime SMSnpcageopol SMSpcacarm 5: XFGiv00 XFGiv03 SFEBSCopt1 6: SFEvolnonparest SFEmvol02 SFEmvol03

Back

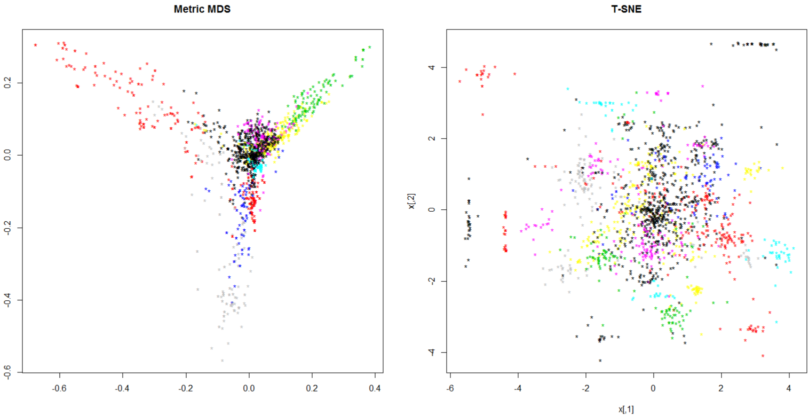


Figure 33: BVSM - k-Means clustering with MDS and T-SNE Visualization

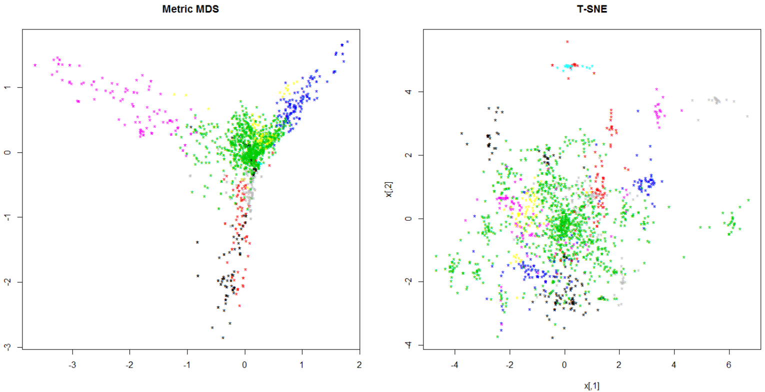


Figure 34: GVSM - k-Means clustering with MDS and T-SNE Visualization

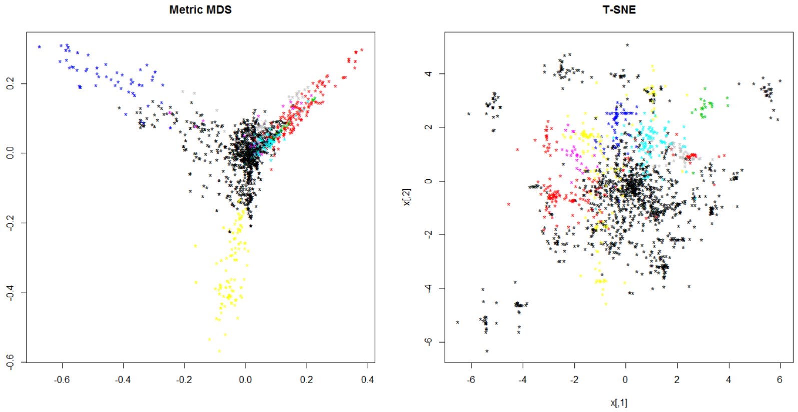


Figure 35: LSA - k-Means clustering with MDS and T-SNE Visualization