Text to Matrix Generator Toolbox

A Brief Introduction

Eugenia Maria Kontopoulou, Dimitrios Zeimpekis
and Efstratios Gallopoulos

Department of
Computer Engineering and Informatics
University of Patras

Patras, 09/05/2014

Outline

1. Introduction to Text Data

2. The Text-to-Matrix Generator

Text Data

From document collections . . .

Documents

Labels	Titles
B1	Identifying users of social networks from their data footprint: An application of large-scale matrix factorizations
B2	Data fusion based on coupled matrix and tensor factorizations
B3	On incremental deterministic methods for dominant space estimation for large data sets
B4	Fast projection methods for robust separable nonnegative matrix factorization
B5	Experiments with randomized algorithms in the text to matrix generator toolbox

Text Data

. . . to Term-Document structures . . .

Term-Document Matrix (TDM)

33×5

	Documents				
terms	B1	B2	B3	B4	B5
algorithm	0	0	0	0	2.3219
applic	2.3219	0	0	0	0
base	0	2.3219	0	0	0
coupl	0	2.3219	0	0	0
data	0.7370	0.7370	0.7370	0	0
determinist	0	0	2.3219	0	0
domin	0	0	2.3219	0	0
estim	0	0	2.3219	0	0
experi	0	0	0	0	2.3219
factor	0.7370	0.7370	0	0.7370	0
fast	0	0	0	2.3219	0
footprint	2.3219	0	0	0	0
fusion	0	2.3219	0	0	0
gener	0	0	0	0	2.3219
identifi	2.3219	0	0	0	0
increment	0	0	2.3219	0	0
larg	1.3219	0	1.3219	0	0
matrix	0.3219	0.3219	0	0.3219	0.3219
method	0	0	1.3219	1.3219	0
network	2.3219	0	0	0	0
nonneg	0	0	0	2.3219	0
project	0	0	0	2.3219	0
random	0	0	0	0	2.3219
robust	0	0	0	2.3219	0
scale	2.3219	0	0	0	0
separ	0	0	0	2.3219	0
set	0	0	2.3219	0	0
social	2.3219	0	0	0	0
space	0	0	2.3219	0	0
tensor	0	2.3219	0	0	0
text	0	0	0	0	2.3219
toolbox	0	0	0	0	2.3219
user	2.3219	0	0	0	0

✓ tf-idf

✓ Stemming

Text Data

. . . for text mining tasks

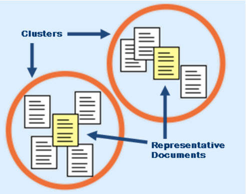
Retrieval

Text Data

. . . for text mining tasks

Retrieval

Clustering

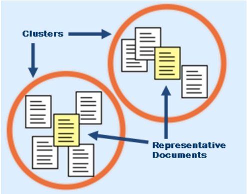


Text Data

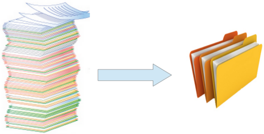
. . . for text mining tasks

Retrieval

Clustering



Classification



Outline

1. Introduction to Text Data

2. The Text-to-Matrix Generator

Text to Matrix Generator

What is TMG:

- Toolbox developed in University of Patras for text mining tasks over document collections
- Educational and Research tool

TMG: A MATLAB Toolbox for Generating Term-Document Matrices from Text Collections (ZG06b)

Grouping Multidimensional Data

Recent Advances in Clustering
Kogan, Jacob; Nicholas, Charles; Teboulle, Marc (Eds.)

2006, XII, 268 p.

Grouping Multidimensional Data
2006, pp 187-210

TMG: A MATLAB Toolbox for Generating Term-Document Matrices from Text Collections

D. Zeimpekis, E. Gallopoulos

Text to Matrix Generator

What is TMG:

- Toolbox developed in University of Patras for text mining tasks over document collections
- Educational and Research tool

Implementation:

- over 17.000 lines of `matlab` and `perl`
- takes advantage from sparse technology provided by MATLAB
- first version by Zeimpekis (´06)

TMG: A MATLAB Toolbox for Generating Term-Document Matrices from Text Collections (ZG06b)

Grouping Multidimensional Data

Recent Advances in Clustering
Kogan, Jacob; Nicholas, Charles; Teboulle, Marc (Eds.)

2006, XII, 268 p.

Grouping Multidimensional Data
2006, pp 187-210

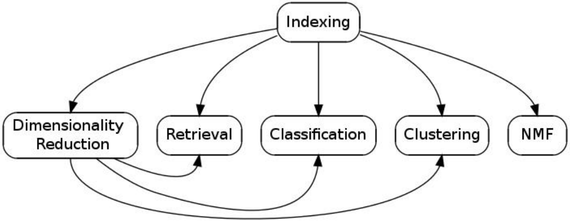
TMG: A MATLAB Toolbox for Generating Term-Document Matrices from Text Collections

D. Zeimpekis, E. Gallopoulos

Structure

Six basic modules:

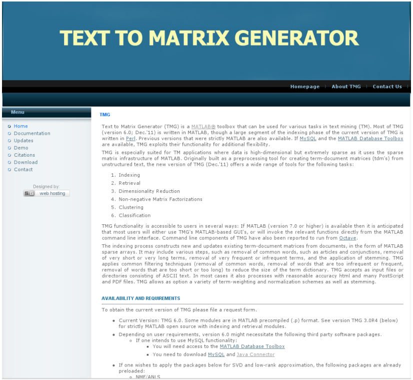
1. Indexing
2. Dimensionality Reduction
3. Non-Negative Matrix Factorizations
4. Retrieval
5. Clustering
6. Classification



How can I find TMG?

Free under request from:

http://scgroup20.ceid.upatras.gr:8000/tmg/



TMG

Text to Matrix Generator (TMG) is a MATLAB® toolbox that can be used for various tasks in text mining (TM). Most of TMG (version 6.0; Dec.'11) is written in MATLAB, though a large segment of the indexing phase of the current version of TMG is written in Perl. Previous versions that were strictly MATLAB are also available. If MySQL and the MATLAB Database Toolbox are available, TMG exploits their functionality for additional flexibility.

TMG is especially suited for TM applications where data is high-dimensional but extremely sparse as it uses the sparse matrix infrastructure of MATLAB. Originally built as a preprocessing tool for creating term-document matrices (tdm's) from unstructured text, the new version of TMG (Dec.'11) offers a wide range of tools for the following tasks:

1. Indexing
2. Retrieval
3. Dimensionality Reduction
4. Non-negative Matrix Factorizations
5. Clustering
6. Classification

TMG functionality is accessible to users in several ways: If MATLAB (version 7.0 or higher) is available then it is anticipated that most users will either use TMG's MATLAB-based GUI's, or will invoke the relevant functions directly from the MATLAB command line interface. Command line components of TMG have also been reported to run from Octave.

The indexing process constructs new and updates existing term-document matrices from documents, in the form of MATLAB sparse arrays. It may include various steps, such as removal of common words, such as articles and conjunctions, removal of very short or very long terms, removal of very frequent or infrequent terms, and the application of stemming. TMG applies common filtering techniques (removal of common words, removal of words that are too infrequent or frequent, removal of words that are too short or too long) to reduce the size of the term dictionary. TMG accepts as input files or directories consisting of ASCII text. In most cases it also processes with reasonable accuracy html and many PostScript and PDF files. TMG allows as option a variety of term-weighting and normalization schemes as well as stemming.

AVAILABILITY AND REQUIREMENTS

To obtain the current version of TMG please file a request form.

- Current Version: TMG 6.0. Some modules are in MATLAB precompiled (.p) format. See version TMG 3.0R4 (below) for strictly MATLAB open source with indexing and retrieval modules.
- Depending on user requirements, version 6.0 might necessitate the following third party software packages.
  - If one intends to use MySQL functionality:
    - You will need access to the MATLAB Database Toolbox
    - You need to download MySQL and Java Connector
- If one wishes to apply the packages below for SVD and low-rank approximation, the following packages are already preloaded:
  - NMF/ANLS

More than 4000 requests worldwide . . .

Caltech, Maryland, Purdue, Carnegie Mellon, Tennessee, Berkeley, Texas, Minnesota, Stanford, MIT, Columbia Renault, Leuven, Max-Planck, Michigan, Oxford, Philips, Princeton, Illinois, ETH, RPI, Los Alamos, Toronto, Queen Mary, St Andrews, Colorado, Texas, Livermore, Mathworks, Yahoo!, . . .

Part I

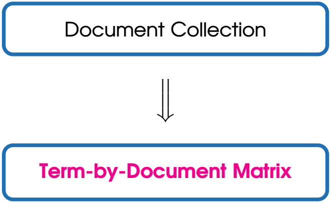
Introduction in version 6.0R7

Outline

1. Indexing Module
2. Dimensionality Reduction and Nonnegative Matrix Factorizations Modules
3. Retrieval Module
4. Clustering Module
5. Classification Module
6. Conclusions

Generate, Update and Downdate Term-by-Document Matrices I

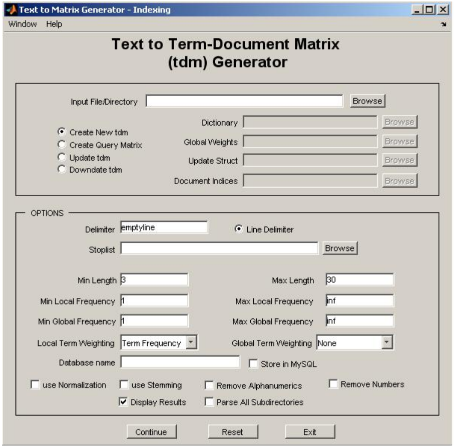
Purpose


Document Collection

⇓

Term-by-Document Matrix

Graphical User Interface



Text to Term-Document Matrix (tdm) Generator

Input File/Directory

● Create New tdm
○ Create Query Matrix
○ Update tdm
○ Downdate tdm

Dictionary

Global Weights

Update Struct

Document Indices

OPTIONS

Delimiter: emptyline

● Line Delimiter

Stoplist

Min Length: 3

Max Length: 30

Min Local Frequency: 1

Max Local Frequency: inf

Min Global Frequency: 1

Max Global Frequency: inf

Local Term Weighting: Term Frequency

Global Term Weighting: None

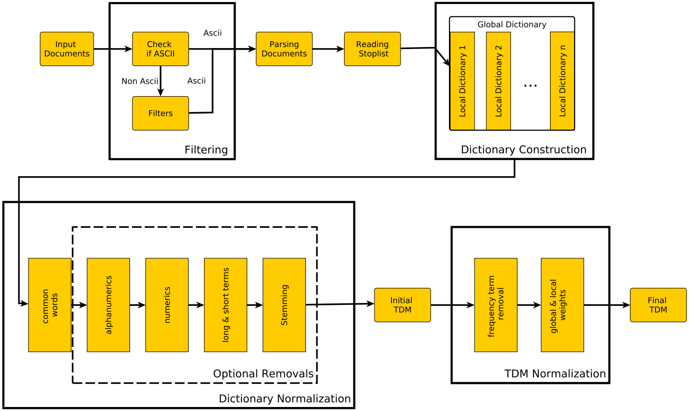
Database name

□ Store in MySQL

□ use Normalization
□ use Stemming
□ Remove Alphanumerics
□ Remove Numbers
☑ Display Results
□ Parse All Subdirectories

Generate, Update and Downdate Term-by-Document Matrices II

Procedure



Input Documents

Check if ASCII

Ascii

Non Ascii

Filters

Ascii

Filtering

Parsing Documents

Reading Stoplist

Global Dictionary

Local Dictionary 1

Local Dictionary 2

...

Local Dictionary n

Dictionary Construction

common words

alphanumerics

numerics

long & short terms

Stemming

Optional Removals

Dictionary Normalization

Initial TDM

frequency term removal

global & local weights

TDM Normalization

Final TDM

Generate, Update and Downdate Term-by-Document Matrices III

Supported non-ASCII formats

	ver.5.0R6	Filter ver.5.0R6	ver. 6.0R7	Filter ver. 6.0R7
doc	×	×	√	TIKA
docx	×	×	√	TIKA
htm	√	`strip_html`	√	`strip_html`
html	√	`strip_html`	√	TIKA
odt	×	×	√	TIKA
pdf	√	`ps2ascii`	√	`ps2ascii`
ps	√	`ps2ascii`	√	`ps2ascii`
rtf	×	×	√	TIKA
tex	×	×	√	Untex

Update

Update the `TDM` by inserting new documents

Downdate

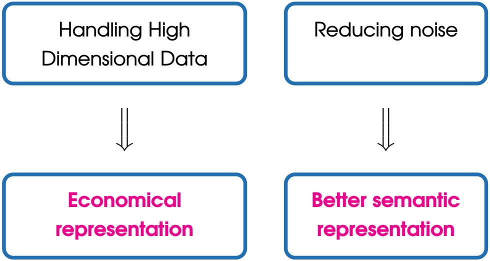
Downdate the `TDM` by extracting useless documents

Outline

1. Indexing Module
2. Dimensionality Reduction and Nonnegative Matrix Factorizations Modules
3. Retrieval Module
4. Clustering Module
5. Classification Module
6. Conclusions

Dimensionality Reduction I

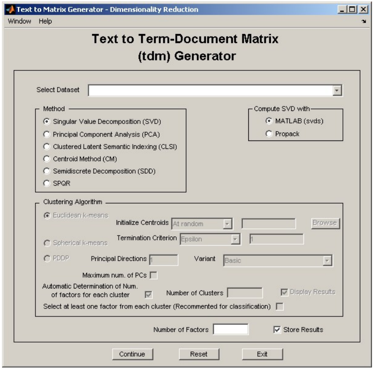
Purpose



Handling High Dimensional Data ⇓ **Economical representation**

Reducing noise ⇓ **Better semantic representation**

Graphical User Interface



Text to Term-Document Matrix (tdm) Generator

Select Dataset

Method
- ● Singular Value Decomposition (SVD)
- ○ Principal Component Analysis (PCA)
- ○ Clustered Latent Semantic Indexing (CLSI)
- ○ Centroid Method (CM)
- ○ Semidiscrete Decomposition (SDD)
- ○ SPQR

Compute SVD with
- ● MATLAB (svds)
- ○ Propack

Clustering Algorithm
- ● Euclidean k-means
- ○ Spherical k-means
- ○ PDDP

Initialize Centroids: At random

Termination Criterion: Epsilon 1

Principal Directions: 1

Variant: Basic

Maximum num. of PCs □

Automatic Determination of Num. of factors for each cluster ☑

Number of Clusters

☑ Display Results

Select at least one factor from each cluster (Recommented for classification) □

Number of Factors

☑ Store Results

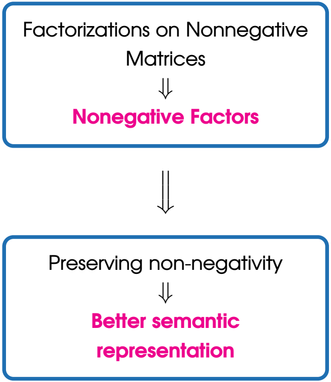
Available Methods

1. Singular Value Decomposition (`SVD`)
   - ✓ MATLAB `svds`
   - ✓ PROPACK `svd` (Larsen (Lar))
2. Centroids Method (`CM`) (Park, Jeon & Rosen (PJR03))
3. Semidiscrete Decomposition (`SDD`) (Kolda & O'Leary (KO00))
4. **Clustered LSI (`CLSI`)** (Zeimpekis & Gallopoulos (ZG05; ZG06a))
5. Sparse Pivoted QR Decomposition (`SPQR`) (Berry, Pulatova & Stewart (BPS05))
6. Principal Component Analysis (`PCA`)

`SDD` and `SPQR` call routines available from `Netlib(TOMS)`

Nonnegative Matrix Factorizations (NMF) I

Purpose



Factorizations on Nonnegative Matrices

⇓

Nonegative Factors

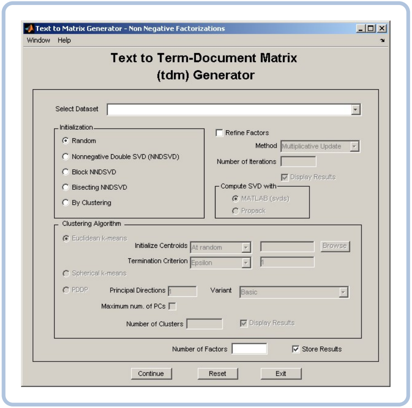
⇓

Preserving non-negativity

⇓

Better semantic representation

Graphical User Interface



Text to Matrix Generator - Non Negative Factorizations

Text to Term-Document Matrix (tdm) Generator

Select Dataset

Initialization
- ● Random
- ○ Nonnegative Double SVD (NNDSVD)
- ○ Block NNDSVD
- ○ Bisecting NNDSVD
- ○ By Clustering

□ Refine Factors

Method: Multiplicative Update

Number of Iterations

☑ Display Results

Compute SVD with
- ● MATLAB (svds)
- ○ Propack

Clustering Algorithm
- ● Euclidean k-means
  - Initialize Centroids: At random
  - Termination Criterion: Epsilon, 1
- ○ Spherical k-means
- ○ PDDP
  - Principal Directions: 1
  - Variant: Basic
  - Maximum num. of PCs □

Number of Clusters

☑ Display Results

Number of Factors

☑ Store Results

✓ Final results depend on initialization

✓ Resulting factors can be refined

Nonnegative Matrix Factorizations (NMF) II

Initialization Techniques

1. Random Initialization
2. **Nonnegative Double SVD NNDSVD** (Boutsidis & Gallopoulos (BG08))
3. **Block Nonnegative Double SVD** (Zeimpekis & Gallopoulos (ZG08))
4. **Bisecting Nonnegative Double SVD** (Zeimpekis & Gallopoulos (ZG08))
5. By Clustering (Wild, Curry, Dougherty (WCD04))

NNDSVD uses prepared implementation

Factors Refinement

1. Multiplicative Update Algorithm (Lee & Seung (LS01))
2. Alternating Non-Negative-Constrained Least Squares (NMF/ANLS) (Kim & Park (KH08))

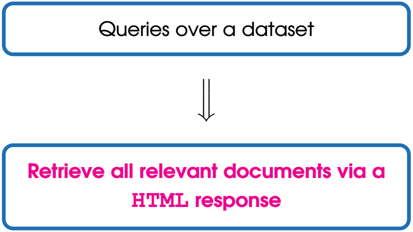
NMF/ANLS uses prepared implementation

Outline

1. Indexing Module
2. Dimensionality Reduction and Nonnegative Matrix Factorizations Modules
3. Retrieval Module
4. Clustering Module
5. Classification Module
6. Conclusions

Retrieval I

Purpose

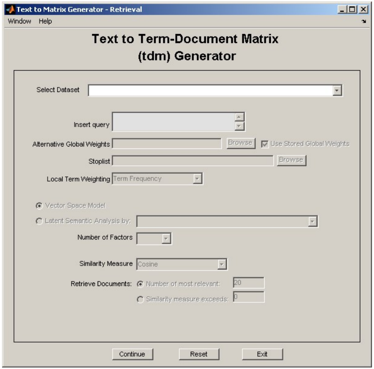


Queries over a dataset

⇓

Retrieve all relevant documents via a `HTML` response

Graphical User Interface



Text to Term-Document Matrix (tdm) Generator

Select Dataset

Insert query

Alternative Global Weights ☑ Use Stored Global Weights

Stoplist

Local Term Weighting: Term Frequency

● Vector Space Model

○ Latent Semantic Analysis by:

Number of Factors

Similarity Measure: Cosine

Retrieve Documents: ● Number of most relevant: 20

○ Similarity measure exceeds: 0

Available Methods

1. Vector Space Model (VSM) (Salton, Wong, & Yang (SWY75))
2. Latent Semantic Analysis (LSA) (Berry et al. (BDJ99; Dee+90))

LSA can be combined with any DR or NMF technique

Outline

1. Indexing Module

2. Dimensionality Reduction and Nonnegative Matrix Factorizations Modules

3. Retrieval Module

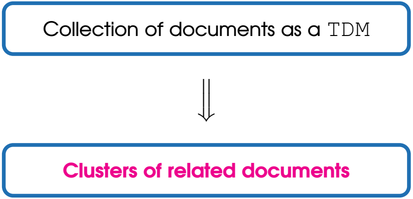
4. **Clustering Module**

5. Classification Module

6. Conclusions

Clustering I

Purpose

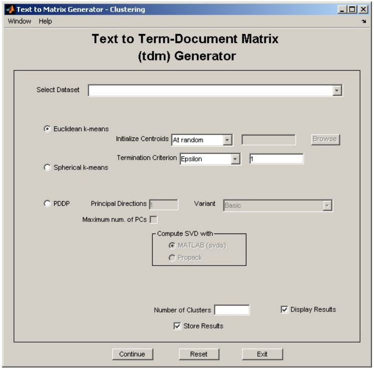


Collection of documents as a `TDM`

⇓

Clusters of related documents

Graphical User Interface



Text to Term-Document Matrix (tdm) Generator

Select Dataset

● Euclidean k-means

Initialize Centroids: At random

Termination Criterion: Epsilon 1

○ Spherical k-means

○ PDDP

Principal Directions: 1

Variant: Basic

Maximum num. of PCs □

Compute SVD with

● MATLAB (svds)

○ Propack

Number of Clusters

☑ Display Results

☑ Store Results

Available Methods

1. Euclidean k-means
2. Spherical k-means(DM01)
3. Principal Direction Divisive Partitioning (`PDDP`) (Boley (Bol97))
4. **`PDDP(l)`** (Zeimpekis & Gallopoulos (ZG03))
5. **`PDDP(l)`** with some hybrid variants of `PDDP` and kmeans (Zeimpekis & Gallopoulos (ZG03))

PDDP(l) Variants

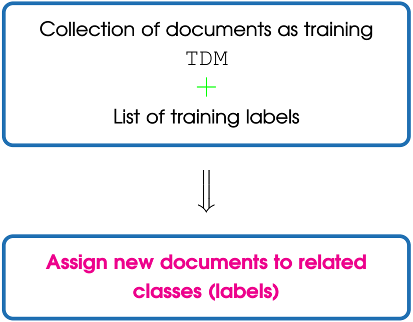
- ✓ Split with k-means
- ✓ Optimal Split
- ✓ Optimal Split with k-means
- ✓ Optimal Split on Projections

Outline

1. Indexing Module
2. Dimensionality Reduction and Nonnegative Matrix Factorizations Modules
3. Retrieval Module
4. Clustering Module
5. **Classification Module**
6. Conclusions

Classification I

Purpose

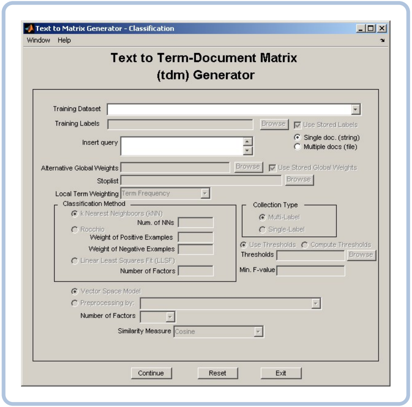


Collection of documents as training TDM + List of training labels

⇓

Assign new documents to related classes (labels)

Graphical User Interface



Text to Matrix Generator - Classification

Text to Term-Document Matrix (tdm) Generator

Training Dataset

Training Labels ☑ Use Stored Labels

Insert query ● Single doc. (string) ○ Multiple docs (file)

Alternative Global Weights ☑ Use Stored Global Weights

Stoplist

Local Term Weighting: Term Frequency

Classification Method

- ● k Nearest Neighboors (kNN) — Num. of NNs
- ○ Rocchio — Weight of Positive Examples; Weight of Negative Examples
- ○ Linear Least Squares Fit (LLSF) — Number of Factors

Collection Type

- ● Multi-Label
- ○ Single-Label

● Use Thresholds ○ Compute Thresholds

Thresholds

Min. F-value

● Vector Space Model

○ Preprocessing by:

Number of Factors

Similarity Measure: Cosine

Classification II

Available Methods

1. k Nearest Neighboors (knn)
2. Rocchio
3. Linear Least Squares Fit (LLSF) (Yang & Chute (YC92))

✓ Combination with CLSI, CM and SVD DR techniques

✓ Implementations for multilabel and singlelabel collections

Outline

1. Indexing Module
2. Dimensionality Reduction and Nonnegative Matrix Factorizations Modules
3. Retrieval Module
4. Clustering Module
5. Classification Module
6. Conclusions

Summary

Goal:

☺ Make TMG more user friendly

Summary

Goal:

☺ Make TMG more user friendly

Work in Progress:

- Smarter parsing → boost parsing time
- Increase the degrees of freedom during parsing phase (e.g. stoplist, incorporation of new filters)
- Manual writing using MATLAB `publish`
- New stemming algorithms (e.g. greek stemmer)
- GUIs makeover
- Incorporation of new capabilities (e.g. WordNet, Wordle)

Thank you!

Questions ?

Bibliography I

(BDJ99) M.W. Berry, Z. Drmač, and E. R. Jessup. ``Matrices, vector spaces, and Information Retrieval''. In: *SIAM Rev.* 41 (1999), pp. 335-362.

(BG08) C. Boutsidis and E. Gallopoulos. ``SVD based initialization: A head start for nonnegative matrix factorization''. In: *Pattern Recognition* 41 (Apr. 2008), pp. 1350-1362.

(Bol97) D. Boley. ``Principal Direction Divisive Partitioning''. In: *Data Mining and Knowledge Discovery* 2 (1997), pp. 325-344.

(BPS05) M. W. Berry, S. A. Pulatova, and G. W. Stewart. ``Algorithm 844: Computing sparse reduced-rank approximations to sparse matrices''. In: *ACM TOMS* 31.2 (June 2005), 252—269.

(Dee+90) S. Deerwester et al. ``Indexing by Latent Semantic Analysis''. In: *Journal of the American Society for Information Science* 41.6 (1990), pp. 391-407.

(DM01) I. S. Dhillon and D. S. Modha. ``Concept decompositions for large sparse text data using clustering''. In: *Machine Learning* 42.1 (2001), pp. 143-175.

(KH08) H. Kim and H.Park. In: *SIAM J. M. Anal. and Appl.* (2008).

Bibliography II

(KO00) T. G. Kolda and D. P. O'Leary. ``Algorithm 805: computation and uses of the semidiscrete matrix decomposition''. In: *ACM TOMS* 26.3 (2000), pp. 415-435.

(Lar) R.M. Larsen. *PROPACK: A software package for the symmetric eigenvalue problem and singular value problems on Lanczos and Lanczos bidiagonalization with partial reorthogonalization*.

(LS01) D. D. Lee and H. S. Seung. ``Algorithms for Non-negative Matrix Factorization''. In: *NIPS*. MIT Press, 2001, pp. 556-562.

(PJR03) H. Park, M. Jeon, and J.B. Rosen. ``Lower Dimensional Representation of Text Data Based on Centroids and Least Squares''. In: *BIT Numerical Mathematics* 43.2 (2003), pp. 427-448.

(SWY75) G. Salton, A. Wong, and C. S. Yang. ``A vector space model for automatic indexing''. In: *Comm. ACM* 18.11 (1975), pp. 613-620.

(WCD04) S. Wild, J. Curry, and A. Dougherty. ``Improving non-negative matrix factorizations through structured initialization.'' In: *Pattern Recognition* 37.11 (2004), pp. 2217-2232.

(YC92) Y.Yang and C. G. Chute. ``A linear least squares fit mapping method for information retrieval from natural language texts''. In: (1992), pp. 447-453.

Bibliography III

(ZG03) D. Zeimpekis and E. Gallopoulos. ``PDDP(*l*): Towards a flexible principal direction divisive partitioning clustering algorithm''. In: *Proc. Workshop on Clustering Large Data Sets (held in conjunction with the Third IEEE Int'l. Conf. Data Min.)* Ed. by D. Boley et al. Melbourne, FL, 2003, pp. 26-35.

(ZG05) D. Zeimpekis and E. Gallopoulos. ``CLSI: A Flexible Approximation Scheme from Clustered Term-Document Matrices''. In: *Proc. 5th SIAM Int'l Conf. Data Mining*. Ed. by H. Kargupta *et al.* SIAM. Philadelphia, 2005, pp. 631-635.

(ZG06a) D. Zeimpekis and E. Gallopoulos. ``Linear and Non-Linear Dimensional Reduction via Class Representatives for Text Classification''. In: *6th Int'l. Conf. Data Mining (ICDM'06)*. Los Alamitos, CA, USA: IEEE Computer Society, 2006, pp. 1172-1177.

(ZG06b) D. Zeimpekis and E. Gallopoulos. ``TMG: A MATLAB toolbox for generating term document matrices from text collections''. In: *Grouping Multidimensional Data: Recent Advances in Clustering*. Ed. by J. Kogan, C. Nicholas, and M. Teboulle. Berlin: Springer, 2006, 187—210.

(ZG08) D. Zeimpekis and E. Gallopoulos. ``Document clustering using nmf based on spectral information''. In: *Text Mining Workshop (Atlanta)*. 2008.