

Interpretable Nonnegative Matrix Decompositions

Saara Hyvönen, **Pauli Miettinen**, and Evimaria Terzi

27 August 2008



Outline

- 1 Introduction
- 2 Definitions
- 3 Algorithms and Complexity
- 4 Experiments
 - Synthetic Data
 - Real Data
- 5 Conclusions



Outline

- 1 Introduction
- 2 Definitions
- 3 Algorithms and Complexity
- 4 Experiments
 - Synthetic Data
 - Real Data
- 5 Conclusions



A Motivating Problem

A dialectologist has some dialectal information in a matrix

$$A = (a_{ij})$$

- rows correspond to dialectal features
- columns correspond to areas (e.g., municipalities)
- $a_{ij} = 1$ if feature is present in the dialect spoken in the area.

Dialectologist wants to solve the following two problems:

- 1 What are the k main characteristic features of dialects?
- 2 What are the k characteristic areas for dialects?
 - To make more studies on few selected areas.

Some type of matrix decomposition is sought.



First Idea: NMF

Dialectologist don't want to see negative values in the decomposition.

- “Dialect spoken in area A contains 1.2 of feature X and -0.2 of feature Y ” vs. “Dialect spoken in area A contains 0.7 of feature Z and 0.3 of feature V .”
 - Negative values can yield negative features
- She considers **Nonnegative Matrix Factorization**.
 - A is represented as $A \approx WH$ where W and H are nonnegative and their inner dimension is k .

But the columns of W and rows of H are just some nonnegative vectors

- ⇒ They don't give the Dialectologist her characteristic areas and features.

Second Idea: CX and CUR Decompositions

Dialectologist could use **Column (CX)** and **Column-Row (CUR)** decompositions.

CX Matrix A is represented as $A \approx CX$ with C containing k columns of A (while X is arbitrary).

CUR Matrix A is represented as $A \approx CUR$ with C as above and R containing r rows of A (while U is arbitrary).

Columns of C and rows of R now give the desired characteristic areas and features.

But now X and U can have negative values.

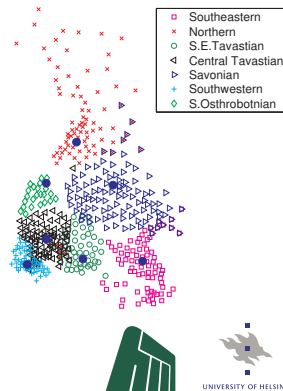


Solution: Nonnegative CX and CUR Decompositions

Dialectologist's solution is to force also X and U be nonnegative.

Thus

- Characteristic areas are given by columns of C .
- Characteristic features are given by rows of R (or, by columns of C when CX decomposition is done to A^T).
- Other features and areas are represented using only nonnegative linear combinations.



UNIVERSITY OF HELSINKI

Outline

- 1 Introduction
- 2 Definitions
- 3 Algorithms and Complexity
- 4 Experiments
 - Synthetic Data
 - Real Data
- 5 Conclusions



The Nonnegative CX Decomposition

Problem (Nonnegative CX Decomposition, NNCX)

Given a matrix $A \in \mathbb{R}_+^{m \times n}$ and an integer k , find an $m \times k$ matrix C of k columns of A and a matrix $X \in \mathbb{R}_+^{k \times n}$ minimizing

$$\|A - CX\|_F.$$

Example:

$$A = \begin{pmatrix} 0.6 & 0.9 & 0.6 & 0.4 & 0.7 \\ 1.0 & 0.7 & 0.9 & 1.0 & 0.9 \\ 0.6 & 0.5 & 0.2 & 0.4 & 1.0 \end{pmatrix}$$

$$C = \begin{pmatrix} 0.6 & 0.9 & 0.6 \\ 1.0 & 0.7 & 0.9 \\ 0.6 & 0.5 & 0.2 \end{pmatrix} X = \begin{pmatrix} 1.0 & 0.0 & 0.0 & 0.9 & 1.7 \\ 0.0 & 1.0 & 0.0 & 0.0 & 0.5 \\ 0.0 & 0.0 & 1.0 & 0.5 & 0.0 \end{pmatrix}$$



The Nonnegative CUR Decomposition

Problem (Nonnegative CUR Decomposition, NNCUR)

Given a matrix $A \in \mathbb{R}_+^{m \times n}$ and integers k and r , find an $m \times k$ matrix C of k columns of A , an $r \times n$ matrix R of r rows of A , and a matrix $U \in \mathbb{R}_+^{k \times r}$ minimizing $\|A - CUR\|_F$.

Example:

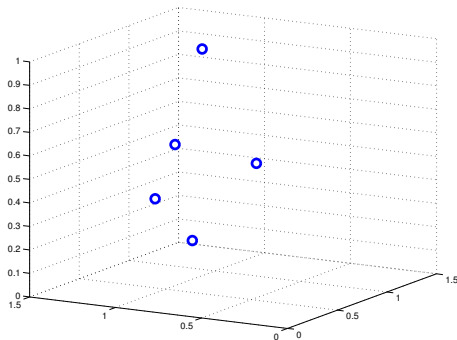
$$A = \begin{pmatrix} 0.6 & 0.9 & 0.6 & 0.4 & 0.7 \\ 1.0 & 0.7 & 0.9 & 1.0 & 0.9 \\ 0.6 & 0.5 & 0.2 & 0.4 & 1.0 \end{pmatrix}$$

$$C = \begin{pmatrix} 0.6 & 0.9 & 0.6 \\ 1.0 & 0.7 & 0.9 \\ 0.6 & 0.5 & 0.2 \end{pmatrix} \quad U = \begin{pmatrix} 0.0 & 1.3 \\ 2.2 & 0.0 \\ 0.0 & 0.7 \end{pmatrix} \quad R = \begin{pmatrix} 0.6 & 0.9 & 0.6 & 0.4 & 0.7 \\ 1.0 & 0.7 & 0.9 & 1.0 & 0.9 \end{pmatrix}$$

NNCX as a Convex Cone

- Columns of A represent points in space.

$$\begin{pmatrix} 0.6 & 0.9 & 0.6 & 0.4 & 0.7 \\ 1.0 & 0.7 & 0.9 & 1.0 & 0.9 \\ 0.6 & 0.5 & 0.2 & 0.4 & 1.0 \end{pmatrix}$$

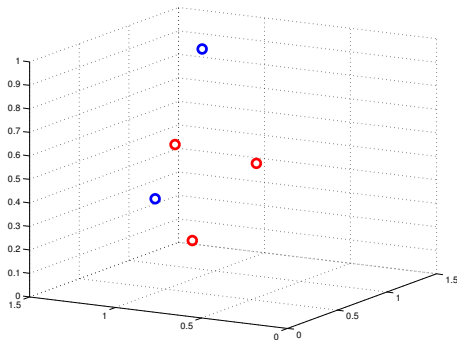


UNIVERSITY OF HELSINKI

NNCX as a Convex Cone

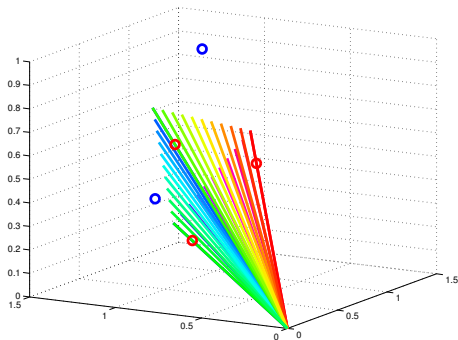
- C selects some of these points.

$$\begin{pmatrix} 0.6 & 0.9 & 0.6 & 0.4 & 0.7 \\ 1.0 & 0.7 & 0.9 & 1.0 & 0.9 \\ 0.6 & 0.5 & 0.2 & 0.4 & 1.0 \end{pmatrix}$$



NNCX as a Convex Cone

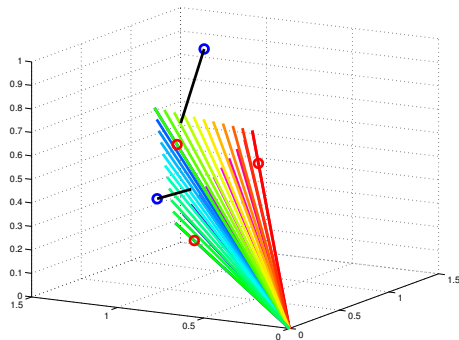
- Points in C generate some **convex cone** \mathcal{C} .
 - $v \in \mathcal{C}$ if there is $x \in \mathbb{R}_+^k$ s.t. $v = Cx$.



NNCX as a Convex Cone

- $\|A - CX\|_F^2$ equals to the sum of squared shortest distances from A 's columns to cone's points.

$$\left\| \begin{pmatrix} 0.6 & 0.9 & 0.6 \\ 1.0 & 0.7 & 0.9 \\ 0.6 & 0.5 & 0.2 \end{pmatrix} \begin{pmatrix} 0.6 \\ 0 \\ 0.3 \end{pmatrix} - \begin{pmatrix} 0.4 \\ 1.0 \\ 0.4 \end{pmatrix} \right\|_2^2 = 0.0369$$



Outline

- 1 Introduction
- 2 Definitions
- 3 Algorithms and Complexity**
- 4 Experiments
 - Synthetic Data
 - Real Data
- 5 Conclusions



UNIVERSITY OF HELSINKI

The Two Subproblems of $[NN]CX$

Finding matrix C (aka Column Subset Selection problem)

Finding matrix X when some matrix C is given

The Two Subproblems of [NN]CX

Finding matrix C (aka Column Subset Selection problem)

- more combinatorial on its nature
- nonnegativity constraint, in general, does not have any effects
- computational complexity is unknown (assumed to be NP-hard)

Finding matrix X when some matrix C is given

The Two Subproblems of [NN]CX

Finding matrix C (aka Column Subset Selection problem)

- more combinatorial on its nature
- nonnegativity constraint, in general, does not have any effects
- computational complexity is unknown (assumed to be NP-hard)

Finding matrix X when some matrix C is given

- constrained (in NNCX) least squares fitting problem
- well-known methods to solve the problem in polynomial time
 - for CX one can use Moore–Penrose pseudo-inverse for $X = C^\dagger A$
 - for NNCX the problem is a convex quadratic program (solved using, e.g., quasi-Newtonian methods).

The Local Algorithm for NNCX

Assume we can find X when C is given. Local performs a standard greedy local search to select C .

Local

- 1 initialize C randomly and compute X
- 2 **while** reconstruction error decreases
 - 1 select c , a column of C , and α , a column of A not in C such that if c is replaced with α the reconstruction error decreases most
 - 2 replace c with α
- 3 compute X and **return** C and X

N.B. We need to solve X in step 2.1.



The ALS Algorithm

The ALS algorithm uses the alternating least squares method often employed in NMF algorithms.

ALS

- 1 initialize \tilde{C} randomly
 - 2 **while** reconstruction error decreases
 - 1 find nonnegative X to minimize $\|A - \tilde{C}X\|_F$
 - 2 find nonnegative \tilde{C} to minimize $\|A - \tilde{C}X\|_F$
 - 3 match columns of \tilde{C} to their nearest columns in A
 - 4 let C be those columns, compute X and **return** C and X
- \tilde{C} does not contain A 's columns.
 - Matching can be done in polynomial time.



How to Use Columns: Convex Quadratic Programming

Given C , we can find nonnegative X minimizing $\|A - CX\|_F$ in polynomial time

- convex quadratic programming
- quasi-Newton methods (L-BFGS)
- also convex optimization methods are possible

But these methods can take quite some time.

- `Local` needs to solve X $k(n - k)$ times for a single local swap.
- When final C is selected, they can be used as a post-processing step.



How to Use Columns: Projection Method

We employ a simple **projection method**:

- 1 let $X = C^\dagger A$ (Moore–Penrose pseudo-inverse)
- 2 for $x_{ij} < 0$ let $x_{ij} = 0$

This method is fast in practice and is often used in NMF algorithms. However, no guarantees on its performance can be given.

In experiments, we used only this method for a fair comparison.



Algorithms for NNCUR Decomposition

Let Alg be an algorithm for NNCX.

Algorithm for NNCUR:

- 1 $C = \text{Alg}(A)$
- 2 $R = \text{Alg}(A^T)$
- 3 find nonnegative U s.t. $\|A - CUR\|_F$ is minimized

For 3 we can use $U = C^\dagger A R^\dagger$ and use the projection method.
Same method is used also for standard CUR.



Outline

- 1 Introduction
- 2 Definitions
- 3 Algorithms and Complexity
- 4 Experiments**
 - Synthetic Data
 - Real Data
- 5 Conclusions



Algorithms Used

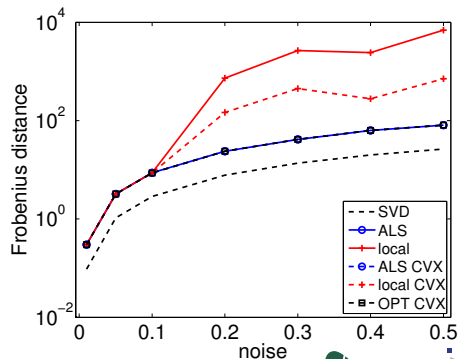
- Local
- ALS
- 844 by Berry, Pulatova, and Stewart (ACM Trans. Math. Softw. 2005)
- DMM by Drineas, Mahoney, and Muthukrishnan (ESA, APPROX, and arXiv 2006–07)
 - based on sampling, approximates SVD within $1 + \varepsilon$ w.h.p., but needs lots of columns in C .
- K-means, which selects C using k-means
- NMF
 - theoretical lower bound for NNCX and NNCUR
- SVD
 - lower bound for all methods



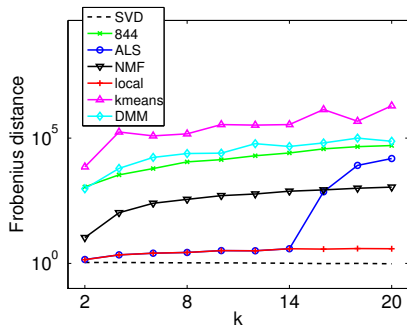
To Find Optimal X or Not

We used convex optimization (CVX) to solve optimal X.

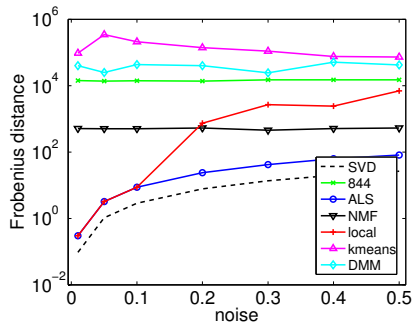
- SVD's distance to optimal CX decomposition (OPT CVX)
- ALS is optimal even without CVX (ALS, ALS CVX, and OPT CVX coincide everywhere)
- Local benefits somewhat from convex optimization post-processing



Noise and Decomposition Size



Left: Local is the best (ex. SVD).



Right: ALS is the best (ex. SVD).



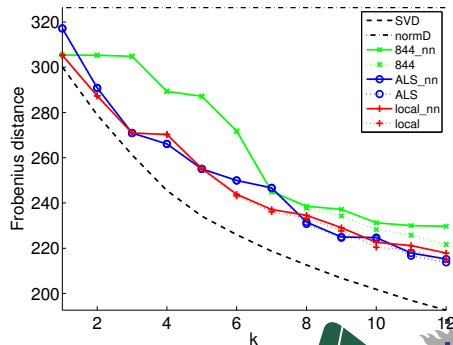
CUR and NNCUR Decompositions of the Newsgroup Data

Newsgroup data with CUR and NNCUR decompositions. Local and ALS are the two best methods.

Only very small increase in reconstruction error when nonnegativity is required

- ∴ data has latent NNCUR structure.

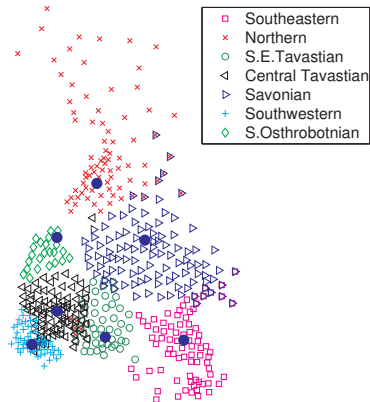
DMM is not included due to its bad performance.



The Dialect Data

Dialect data with NNCUR decomposition using ALS.

- Symbols show the spread of the features (rows) selected.
- Solid dots mark the representative municipalities (columns) selected.
- Spread of features coincides well with current understanding of Finland's dialects.



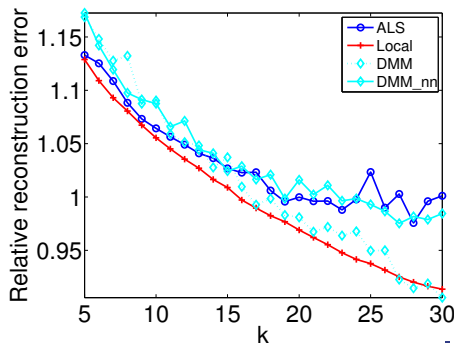
How Many Columns Are Needed to Beat SVD?

Relative error against SVD:

$$\text{error}/\text{SVD}(5)$$

Jester joke dataset, similar experiment done in Drineas et al. (arXiv), [NN]CX decomposition

- `Local` is mostly best – better than `DMM` without nonnegativity
- It takes $k = 16$ for `Local` to be better than SVD with $k = 5$.



Outline

- 1 Introduction
- 2 Definitions
- 3 Algorithms and Complexity
- 4 Experiments
 - Synthetic Data
 - Real Data
- 5 Conclusions



UNIVERSITY OF HELSINKI

Conclusions

- We studied nonnegative variants of CX and CUR decompositions.
- Several real-world datasets seem to have such structure.
- Very simple algorithms were able to find good decompositions.
 - Our algorithms can be better than general CX and CUR algorithms.
- Better algorithms are sought.
 - Perhaps the convex cone interpretation helps.
- Model-selection issue: how big C and R should be?
- CX and CUR decompositions are still relatively little studied in CS (esp. data mining) community.

Thank You!

