Introduction to Matrix and Tensor factorization.

Martin Leginus

Intelligent Web and Information Systems, Aalborg University

February 21th, 2012

Outline

- Motivation
- Basics
- Eigenvectors and eigenvalues
- Singular Value Decomposition
- Matrix Factorization
- SVD versus Regularized KMF
- Tensor factorization
- Higher order Singular Value Decomposition
- Other techniques
- Conclusion

Motivation

Matrix factorization is utilized in different applications:

- Text mining, Recommendation systems.
- Clustering techniques
- Speech denoising
- Image recognition
- Bioinformatics

A matrix factorization estimates A with $\hat{A}$ and in a such way the afore-mentioned tasks can be performed.

Basics

- Matrix operations: addition, substraction and multiplication.
- Determinant - a function that for a square matrix returns a scalar value.
  - If $|M| \neq 0$ - it is an invertible matrix (possible to compute M^{-1}).
  - If $|M| = 0$ - it is a singular matrix.
- Orthogonal matrix is a regular matrix M ($|M| \neq 0$)) such that:

$$M \cdot M^T = I$$

  - $M^T = M^{-1}$
  - product is commutative $M \cdot M^T = M^T \cdot M$
  - $|M| = \pm 1$

Basics

- A trace of a square matrix is the sum of the elements on the main diagonal.
- Frobenius norm (Euclidean norm) of the matrix $M \in R^{m \times n}$:

$$||M_F^2|| = \sum_{i \leq m, j \leq n} |m_{i,j}|^2$$

- Row or column vectors of matrix have Frobenius norm equal to the length of the given vectors.
- Normalized unit vectors are normalized in terms of their Frobenius norm.
- Column (row) rank of a matrix A is the maximum number of linearly independent column (row) vectors of A. Column and row rank of the matrix are always equal.

Eigenvectors and eigenvalues

For the eigenvector c associated with the eigenvalue λ of a matrix A holds:

$$A \cdot c = \lambda \cdot c$$

Example. Consider the matrix

$$A = \begin{pmatrix} 1 & 2 & 1 \\ 6 & -1 & 0 \\ -1 & -2 & -1 \end{pmatrix}.$$

Consider the three column matrices

$$C_1 = \begin{pmatrix} 1 \\ 6 \\ -13 \end{pmatrix}, \; C_2 = \begin{pmatrix} -1 \\ 2 \\ 1 \end{pmatrix}, \text{ and } \; C_3 = \begin{pmatrix} 2 \\ 3 \\ -2 \end{pmatrix}.$$

We have

$$AC_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \; AC_2 = \begin{pmatrix} 4 \\ -8 \\ -4 \end{pmatrix}, \text{ and } \; AC_3 = \begin{pmatrix} 6 \\ 9 \\ -6 \end{pmatrix}.$$

Figure: Example of eigenvalues with associated eigenvectors.

Eigenvectors and eigenvalues

Transformation of a regularized matrix into singular matrix is called eigenvalue problem because:

$$|(A - \lambda I) \cdot c| = 0$$

- Characteristic equation.
- The largest eigenvalue of a matrix is also called the principal eigenvalue.
- Sum of eigenvalues is equal to the trace of a given matrix.
- Multiplication of eigenvalues is equal to the determinant of a given matrix.

Eigenvectors and eigenvalues

- Each eigenvalue produces an infinite number of eigenvectors.
- Eigenvectors for different eigenvalues are linearly independent.

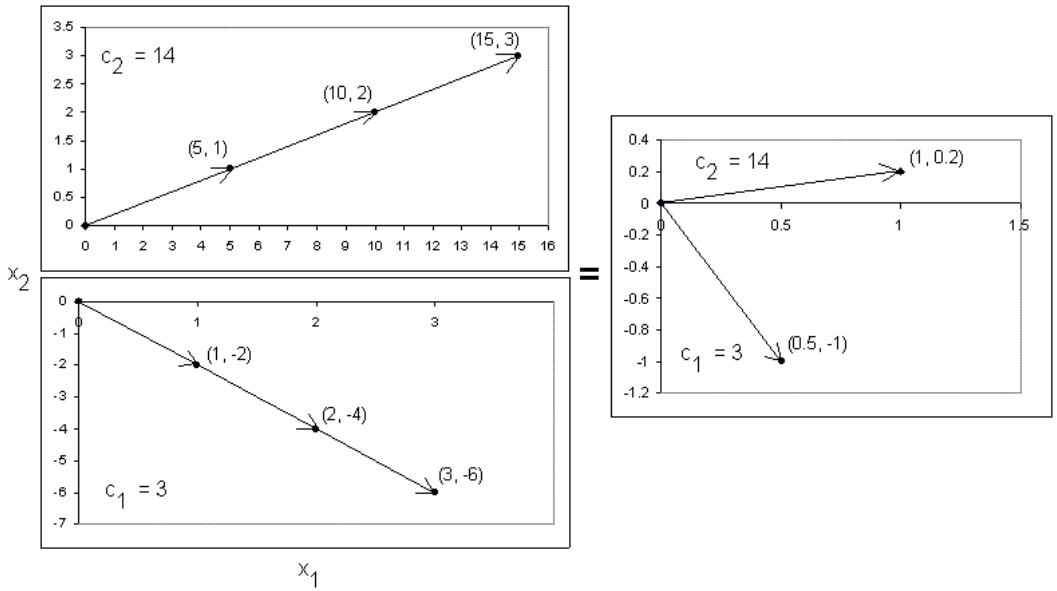


Figure: Eigenvectors for different eigenvalues are linearly independent.

Singular Value Decomposition (SVD)

Singular Value Decomposition is defined as:

$$A = U \times S \times V^T$$

- U is a matrix whose columns are the eigenvectors of the AA^T
- S is a matrix whose diagonal elements are the singular values of A.
- V is a matrix whose columns are the eigenvectors of the $A^T A$

It provides a direct method for the computing the rank of a matrix - a number of non-zero singular values.

Truncated SVD - rank k approximation of matrix.

Considering only top k singular values a noisy data are removed. It provides better approximation of the original matrix.
Rank k Approximation of matrix is dimensionality reduction technique (original vector space is mapped into a low-dimensional space).

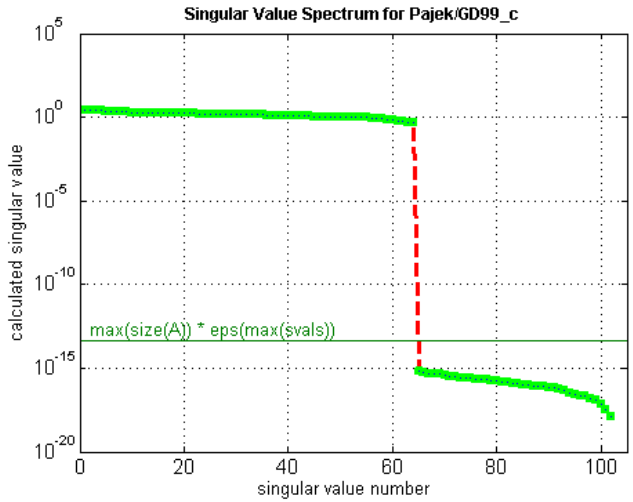


Figure: Singular values of a matrix of rank 64 plus noise.

Latent data structure is masked by noisy data (dimensions).

Truncated SVD - rank k approximation of matrix.

Considering only top k singular values a noisy data are removed. It provides better approximation of the original matrix.
Rank k Approximation of matrix is dimensionality reduction technique (original vector space is mapped into a low-dimensional space).

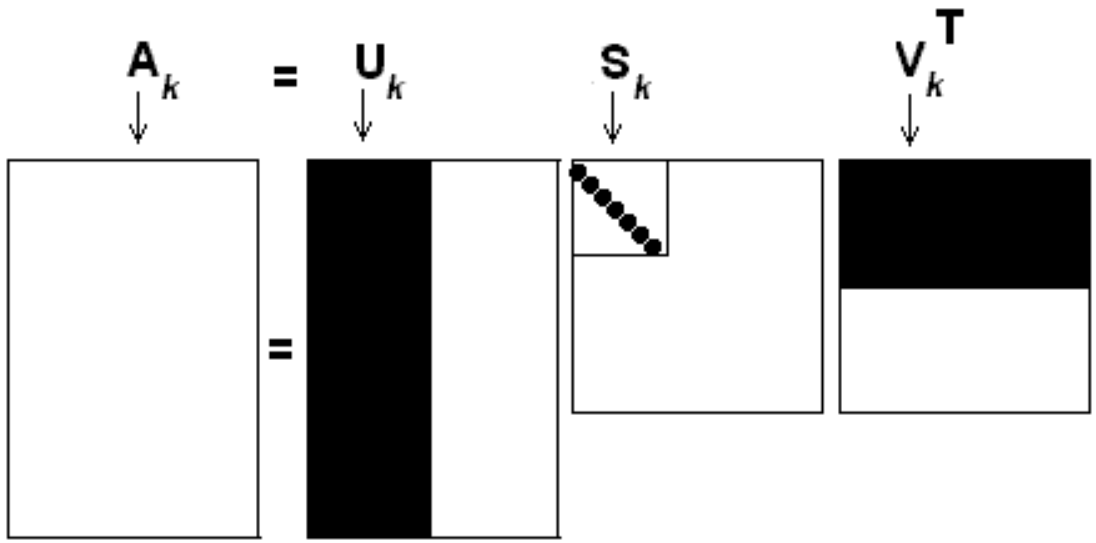


Figure: Truncated SVD - rank k approximation of matrix.

Latent data structure is masked by noisy data (dimensions).

Matrix factorization (MF)

The goal is to approximate the true unobserved ratings-matrix R by $\hat{R} : |U| \times |I|$.

$$\hat{R} = W \cdot H^t$$

where $W : |U| \times k$ and $H : |I| \times k$

- w_u represents k features that describe user u.
- h_i represents k features that describe item i.

$$\hat{r_{u,i}} =< w_u, h_i >= \sum_{f=1}^{k} w_{u,f} \cdot h_{i,f}$$

Often can be added a bias term $b_{u,i}$ which centers the approximation.

Kernel Matrix Factorization

Interactions between the feature vector w_u and h_i are kernelized:

$$\hat{r_{u,i}} =< w_u, h_i >= a + c \cdot K(w_u, h_i)$$

- Terms a, c allow rescaling the approximation.
- Kernel function $K : \mathcal{R}^k \times \mathcal{R}^k \to \mathcal{R}$ can utilize one of the following well-known kernels:

linear:	$K_l(w_u, h_i) =< w_u, h_i >$
polynomial:	$K_p(w_u, h_i) = (1+ < w_u, h_i >)^d$
RBF:	$K_r(w_u, h_i) = exp(-\frac{\|\|w_u, h_i\|\|^2}{2\sigma^2})$
logistic:	$K_s(w_u, h_i) = \phi_s(b_{u,i}+ < w_u, h_i >)$
	where $\phi_s(x) = \frac{1}{1+\exp^{-x}}$

Kernel Matrix Factorization

Benefits of using kernels:

- Using kernel like logistic one results into bounded values of the ratings to the application domain.
- Model non-linear correlations between users and items.
- Kernels lead to different models that can be combined in an ensemble.

Non-negative matrix factorization

Additional constraints on feature matrices W and H such that each entry has to be non-negative.
The motivation is to eliminate interactions between negative correlations (commonly used in CF algorithms).

Learning Matrix Factorization Models.

Minimize an error between an approximated matrix $\hat{R}$ and original matrix R.
The optimization task is defined as: $\underset{W,H}{\operatorname{argmin}} E(S, W, H)$ where:

$$E(S, W, H) := \sum_{r_{u,i} \in S} (r_{u,i} - \hat{r_{u,i}})$$

Overfitting:
Netflix dataset contains 480000 users and 17000 items with 100 million ratings. It leads to estimating 50 million of parameters when $k = 100$ which results into overfitting.
Two strategies:

- Regularization
- Early stopping criterion

Regularization

A regularization term is added to the optimization task.
Tikhonov regularization is used and a parameter λ controls regularization.
The final optimization task is: $\underset{W,H}{\operatorname{argmin}} Opt(S, W, H)$ where:

$$Opt(S, W, H) := E(S, W, H) + \lambda(||W||_F^2 + ||H||_F^2)$$

Optimization by Gradient Descent

Gradient descent is used for MF and also for KMF:

1: **procedure** OPTIMIZE(S, W, H)
2: initialize W, H
3: **repeat**
4: **for** $r_{u,i} \in S$ **do**
5: **for** $f \leftarrow 1, \ldots, k$ **do**
6: $w_{u,f} \leftarrow w_{u,f} - \alpha \frac{\partial}{\partial w_{u,f}} \text{Opt}(\{r_{u,i}\}, W, H)$
7: $h_{i,f} \leftarrow h_{i,f} - \alpha \frac{\partial}{\partial h_{i,f}} \text{Opt}(\{r_{u,i}\}, W, H)$
8: **end for**
9: **end for**
10: **until** Stopping criteria met
11: **return** (W, H)
12: **end procedure**

Figure: Generic learning algorithm for KMF.

SVD versus Regularized KMF

Singular Value Decomposition (SVD) decomposes a matrix into 3 matrices:

$$R = W'\Sigma H', \text{where:} W' : |U| \times |U|, \Sigma : |U| \times |I|, H' : |I| \times |I|$$

SVD is not suitable for recommender systems because of:

- The huge number of missing values that has to be estimated e.g, sparsity rate is 99% for Netflix dataset.
- Lack of regularization leads to overfitting.

	SVD	Regularized MF		
		linear	logistic	lin. non-neg.
Netflix RMSE	0.946 [8]	0.915	0.918	0.914

Figure: RMSE results on Netflix probe for RKMF and k-rank SVD.

Tensors - basics

Tensor of n-th order – is multidimensional array with N indices, denoted as $\mathcal{A} \in R^{I_1 \times I_2 \times ..., \times I_N}$.

Tensor fiber – one dimensional fragment of a tensor (column vector), such that all indices are fixed except for one.

A tensor can be converted into so called mode matrices by arranging particular fibers of a tensor as columns of mode matrices.

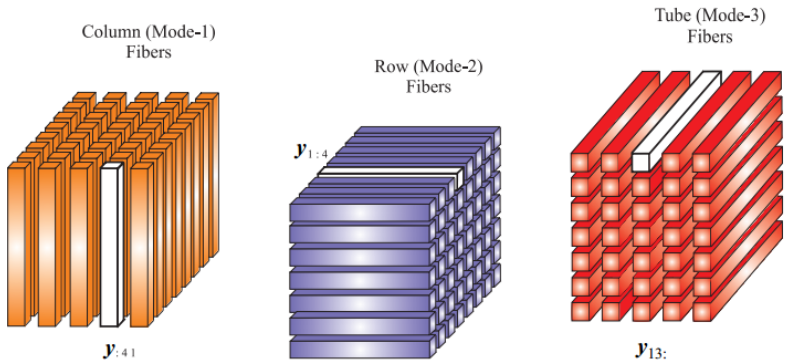
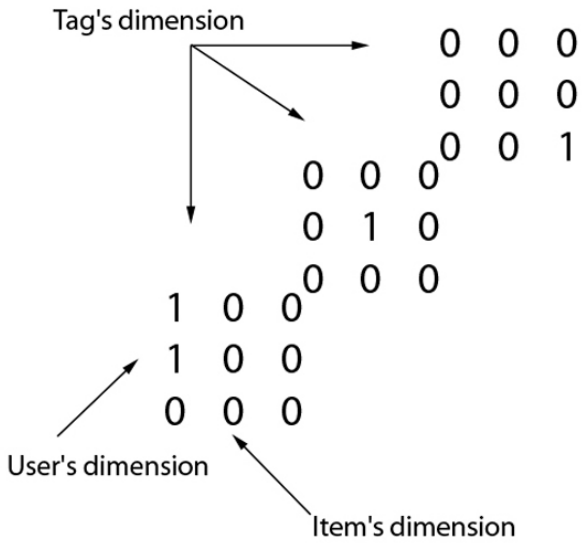


Figure: Column, row and tube fibers of a third-order tensor.

Tensor factorization

- Analyse three dimensional relations e.g., *users*, *tags*, *items*
- Factorization techniques expose patterns and reveal latent relations among *users*, *tags* and *items*

Users	Information items	Tags	Weights
U_1	I_1	T_1	1
U_2	I_1	T_1	1
U_2	I_2	T_2	1
U_3	I_3	T_3	1



- Overcome state-of-the-art tag-based recommenders
- Generate recommendations of items, tags or users using the same tensor.

Higher-order Singular Value Decomposition (HOSVD)

Operations of HOSVD factorization:

1. Mode matrices are created from the initial tensor
2. Singular Value Decomposition is applied to each mode matrix
3. Core tensor S and approximated tensor A' are computed according to:

$$S = A \times_1 (U^{(1)}_{c1})^T \times_2 (U^{(2)}_{c2})^T \times_3 (U^{(3)}_{c3})^T$$

$$A' = S \times_1 U^1_{c1} \times_2 U^1_{c2} \times_3 U^3_{c3}$$

1	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1

1st mode matrix - R_u x R_t R_i

1	1	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1

2nd mode matrix R_t x R_u R_i

1	1	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1

3rd mode matrix R_i x R_t R_u

Higher-order Singular Value Decomposition (HOSVD)

Operations of HOSVD factorization:

1. Mode matrices are created from the initial tensor
2. Singular Value Decomposition is applied to each mode matrix
3. Core tensor S and approximated tensor A' are computed according to:

$$S = A \times_1 (U^{(1)}_{c1})^T \times_2 (U^{(2)}_{c2})^T \times_3 (U^{(3)}_{c3})^T$$

$$A' = S \times_1 U^1_{c1} \times_2 U^1_{c2} \times_3 U^3_{c3}$$

$$F = U \times S \times V^T$$

$$U^1 = \begin{pmatrix} -0.53 & 0 & -0.85 \\ -0.85 & 0 & 0.53 \\ 0 & 1 & 0 \end{pmatrix}$$

$$S^1 = \begin{pmatrix} 1.62 & 0 & 0 \\ 0 & 1.00 & 0 \\ 0 & 0 & 0.62 \end{pmatrix}$$

Figure: Application of the SVD to the 1st mode matrix: U^1 and S^1 matrices

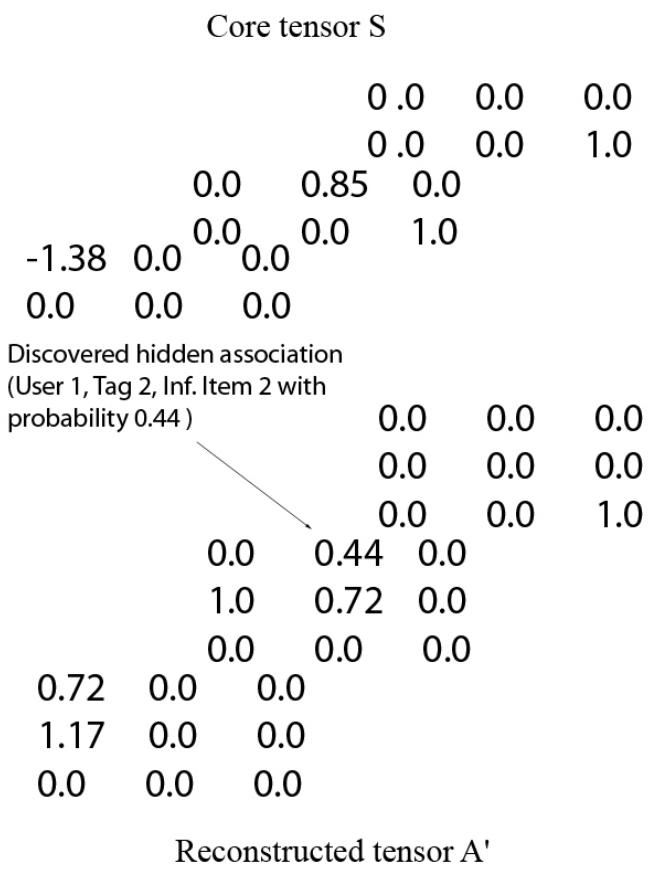
Higher-order Singular Value Decomposition (HOSVD)

Operations of HOSVD factorization:

1. Mode matrices are created from the initial tensor
2. Singular Value Decomposition is applied to each mode matrix
3. Core tensor S and approximated tensor A' are computed according to:

$$S = A \times_1 (U^{(1)}_{c1})^T \times_2 (U^{(2)}_{c2})^T \times_3 (U^{(3)}_{c3})^T$$

$$A' = S \times_1 U^1_{c1} \times_2 U^1_{c2} \times_3 U^3_{c3}$$



Other tensor factorization techniques.

Non-negative tensor factorization (NTF)

$$\text{NTF: } \underset{\mathcal{C},X,Y,Z}{\text{minimize}} \frac{1}{2}\|\mathcal{C} \times_1 X \times_2 Y \times_3 Z - \mathcal{A}\|_F^2 \quad \text{subject to } \mathcal{C}, X, Y, Z \geq 0 \tag{1}$$

where $\mathcal{C}$ is a diagonal core tensor, $\mathcal{A}$ is the initial tensor, X, Y and Z are the factor matrices.

Canonical decomposition (CP or PARAFAC) Similar to the NTF model, the core tensor is diagonal. **Learning tensor factorization models.** Alternating least squares optimization - the original factorization problem is divided into the three sub-problems where is computed the corresponding factor matrix. Different optimization algorithms can be considered e.g., gradient descent etc.

Future work

- Investigate different kernels for tensor factorization.
- Propose a new interpretation schemes for social tagging data e.g., time decay, other attributes of objects.

Conclusion

- Matrix and tensor factorization techniques are widely used within different reasearch areas.
- Computionaly expensive.
- Regularization is needed to avoid a problem of overfitting.