Non-negative Matrix Factorization: Algorithms, Extensions and Applications

Emmanouil Benetos

www.soi.city.ac.uk/~sbbj660/

March 2013
Introduction

- **Assumption**: Perception of the whole is based on perception on its parts
- There is evidence for parts-based representations in the brain
- Algorithms for learning holistic representations: principal component analysis, singular value decomposition
- Allowing additive (and not subtractive) combinations leads to parts-based representations
- First work in the ’90s: **positive matrix factorization**
Non-negative Matrix Factorization (NMF): first proposed by Lee and Seung in 1999

Unsupervised algorithm for decomposing multivariate data

Alternatively, a method for dimensionality reduction by factorizing a data matrix into a low rank decomposition

Constraint: non-negativity of data

This allows a parts-based representation because only additive combinations are allowed

Non-negative Matrix Factorization (2)

Model:

- Given a non-negative matrix $V$ find non-negative matrix factors $W$ and $H$ such that:

$$V \approx WH$$  \hfill (1)$$

where:

- $V \in \mathbb{R}^{n \times m}$
- $W \in \mathbb{R}^{n \times r}$
- $H \in \mathbb{R}^{r \times m}$

- The rank $r$ of the factorization is chosen as $(n + m)r < nm$, so that data is compressed
- The columns of $H$ are in one-to-one correspondence with the columns of $V$. Thus $WH$ can be interpreted as weighted sum of each of the basis vectors in $W$, the weights being the corresponding columns of $H$
Non-negative Matrix Factorization (3)

Cost Functions:

- To find an approximate factorization, cost functions that quantify the quality of the approximation need to be defined.
- Euclidean distance:

\[ \| A - B \|^2 = \sum_{ij} (A_{ij} - B_{ij})^2 \]  

(2)

- Kullback-Leibler divergence (aka relative entropy):

\[ D(A||B) = \sum_{ij} \left( A_{ij} \log \frac{A_{ij}}{B_{ij}} - A_{ij} + B_{ij} \right) \]  

(3)

Problem 1: Minimize $||V - WH||^2$ wrt $W$ and $H$, subject to $W, H \geq 0$

Problem 2: Minimize $D(V||WH)$ wrt $W$ and $H$, subject to $W, H \geq 0$

$W, H$ estimated using multiplicative update rules

Alternative solutions: alternating least squares, gradient descent

Other cost functions: Itakura-Saito distance, $\alpha$-divergences, $\beta$-divergences, $\phi$-divergences, Bregman divergences, ...

Algorithms converge to a local minimum
Network representation / generative model:

\[ \langle v \rangle = Wh \]
Applications:

- Detection, dimensionality reduction, clustering, classification, denoising, prediction

Examples:
  - Digital image analysis
  - Text mining
  - Audio signal analysis
  - Bioinformatics
Applications of NMF (2)

Learning parts-based representation of faces:
Discovering semantic features in text:

<table>
<thead>
<tr>
<th>court</th>
<th>president served</th>
</tr>
</thead>
<tbody>
<tr>
<td>government</td>
<td>governor</td>
</tr>
<tr>
<td>council</td>
<td>secretary</td>
</tr>
<tr>
<td>culture</td>
<td>senate</td>
</tr>
<tr>
<td>supreme</td>
<td>congress</td>
</tr>
<tr>
<td>constitutional</td>
<td>presidential</td>
</tr>
<tr>
<td>rights</td>
<td>elected</td>
</tr>
<tr>
<td>justice</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>flowers</th>
<th>disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>leaves</td>
<td>behaviour</td>
</tr>
<tr>
<td>plant</td>
<td>glands</td>
</tr>
<tr>
<td>perennial</td>
<td>contact</td>
</tr>
<tr>
<td>flower</td>
<td>symptoms</td>
</tr>
<tr>
<td>plants</td>
<td>skin</td>
</tr>
<tr>
<td>growing</td>
<td>pain</td>
</tr>
<tr>
<td>annual</td>
<td>infection</td>
</tr>
</tbody>
</table>

Encyclopedia entry: 'Constitution of the United States'

- president (148)
- congress (124)
- power (120)
- united (104)
- constitution (81)
- amendment (71)
- government (57)
- law (49)
Discovering musical notes in a recording:
In 1999, Hofmann proposed a technique for text processing and retrieval, called **Probabilistic Latent Semantic Analysis (PLSA)**

...which is also called **Probabilistic Latent Semantic Indexing (PLSI)**

...and also **Probabilistic Latent Component Analysis (PLCA)**!

In fact, PLSA/PLSI/PLCA are the probabilistic counterparts of NMF using the KL divergence as a cost function

This interpretation offers a framework that is easy to generalise and extend

Model:

- Considering the entries of $V$ as having been generated by a probability distribution $P(x_1, x_2)$, PLSA models $P(x_1, x_2)$ as a mixture of conditionally independent multinomial distributions:

\[
P(x_1, x_2) = \sum_z P(z)P(x_1|z)P(x_2|z) = P(x_1) \sum_z P(x_2|z)P(z|x_1)
\]  

(4)

- The 1st formulation is called symmetric, the 2nd asymmetric
- Parameters can be estimated using the Expectation-Maximization (EM) algorithm
- Essentially: $H$ is $P(x_1|z)$ and $W$ is $P(z)P(x_2|z)$
Probabilistic Latent Semantic Analysis (3)

Example of symmetric PLSA:

- $P(z)$
- $P(x \mid z)$
- $P(z \mid x)$

Graphical representation of probability distributions and relationships between variables.
Convolutive Extensions (1)

- Instead of identifying 1-D components, we might want to detect 2-D structures out of non-negative data.
- The linear model is not good enough - we need a convolutive model.
- Non-negative Matrix Factor Deconvolution (NMFD): Extracting shifted structures from non-negative data.

Model:

\[
V \approx \sum_t W_t \overrightarrow{H}_t
\]  

where: \( V \in \mathbb{R}^{m \times n} \), \( W_t \in \mathbb{R}^{m \times r} \), \( H \in \mathbb{R}^{r \times n} \), and \( \overrightarrow{H}_t \) shifts the columns of \( H \) by \( t \) spots to the right.

Example of NMFD:
Convolutive Extensions (3)

- Probabilistic counterpart of NMFD: Shift-invariant Probabilistic Latent Component Analysis
- Model (shift invariance across 1 dimension):

\[
P(x, y) = \sum_z P(z) \sum_{\tau} P(x, \tau | z) P(y - \tau | z)
\]  

(6)

- Model (shift invariance across 2 dimensions):

\[
P(x, y) = \sum_z P(z) \sum_{\tau_x} \sum_{\tau_y} P(\tau_x, \tau_y | z) P(x - \tau_x, y - \tau_y | z)
\]  

(7)

Example of Shift-invariant PLCA for handwriting recognition:
Example of Shift-invariant PLCA for musical pitch detection:
What if we want to decompose multidimensional data (i.e. tensors)?

NMF extends to Non-negative Tensor Factorization (NTF)

Model:

\[ V \approx \sum_{j=1}^{k} u_j^1 \otimes u_j^2 \otimes \ldots \otimes u_j^n \]  

where \( V \) is an \( n \)-way array. The approximation tensor has rank \( k \).

As in NMF, there are cost functions and update rules for estimating the \( nk \) vectors \( u_j^i \).

Comparison of extracted NMF bases (2nd row) with NTF bases (3rd row):
Tensorial Extensions (3)

- Probabilistic counterpart: Probabilistic Latent Component Analysis (PLCA)

- Model:

\[
P(x) = \sum_z P(z) \prod_{j=1}^{N} P(x_j | z) \tag{9}
\]

where \( P(x) \) is an \( N \)-dimensional distribution of the random variable \( x = x_1, x_2, \cdots, x_N \).

- Unknown parameters estimated using EM algorithm

More Extensions...

- Incorporate sparsity constraints into NMF/PLSA
- Incorporate prior distributions into PLCA models (incorporating information into the problem)
- Keeping fixed bases: supervised algorithms!
- Temporal constraints for modeling time-series: e.g. Non-negative Hidden Markov Model (N-HMM)
- More complex models based on NMF/PLSA: introduce more latent variables
- What if we don’t know the size of \( r \)? Non-parametric techniques (e.g. GaP-NMF)
- For analysing spectra without losing phase: Complex NMF, High-Resolution NMF
Resources

- Matlab Statistics Toolbox
- NMF-CUDA (OpenMP): https://github.com/ebattenberg/nmf-cuda
- NMFN (R): http://cran.r-project.org/web/packages/NMFN/index.html
- nima (Python): http://nimfa.biolab.si/
- libNMF (C): http://www.univie.ac.at/rlcta/software/
- ITL Toolbox (C++): http://www.insight-journal.org/browse/publication/152
- NNMA (C++): http://people.kyb.tuebingen.mpg.de/suvrit/work/progs/nnma.html