Data Mining and Matrices

06 – Non-Negative Matrix Factorization

Rainer Gemulla, Pauli Miettinen

May 23, 2013
Non-Negative Datasets

Some datasets are intrinsically non-negative:

- Counters (e.g., no. occurrences of each word in a text document)
- Quantities (e.g., amount of each ingredient in a chemical experiment)
- Intensities (e.g., intensity of each color in an image)

The corresponding data matrix $D$ has only non-negative values.

- Decompositions such as SVD and SDD may involve negative values in factors and components
- Negative values describe the absence of something
- Often no natural interpretation

Can we find a decomposition that is more natural to non-negative data?
Example (SVD)

Consider the following “bridge” matrix and its truncated SVD:

\[
\begin{align*}
D &= U \Sigma V^T \\
\end{align*}
\]

Here are the corresponding components:

\[
\begin{align*}
D &= U_{*1} D_{11} V_{*1}^T + U_{*2} D_{22} V_{*2}^T \\
\end{align*}
\]

Negative values make interpretation unnatural or difficult.
Outline

1 Non-Negative Matrix Factorization

2 Algorithms

3 Probabilistic Latent Semantic Analysis

4 Summary
Non-Negative Matrix Factorization (NMF)

Definition (Non-negative matrix factorization, basic form)

Given a non-negative matrix \( D \in \mathbb{R}^{m \times n}_+ \), a non-negative matrix factorization of rank \( k \) is

\[
D \approx LR,
\]

where \( L \in \mathbb{R}^{m \times r}_+ \) and \( R \in \mathbb{R}^{r \times n}_+ \) are both non-negative.

- Additive decomposition: factors and components non-negative
  \( \rightarrow \) No cancellation effects
- Rows of \( R \) can be thought as “parts”
- Row of \( D \) obtained by mixing (or “assembling”) parts in \( L \)
- Smallest \( r \) such that \( D = LR \) exists is called non-negative rank of \( D \)

\[
\text{rank}(D) \leq \text{rank}_+(D) \leq \min \{ m, n \}
\]
Example (NMF)

Consider the following “bridge” matrix and its rank-2 NMF:

\[ D = L \cdot R \]

Here are the corresponding components:

Non-negative matrix decomposition encourage a more natural, part-based representation and (sometimes) sparsity.
Decomposing faces (PCA)

\[
\begin{align*}
\mathbf{D}_{i*} & \text{ (original)} \\
[\mathbf{L}\mathbf{R}]_{i*} & = [\mathbf{U}\Sigma\mathbf{V}^T]_{i*} \\
\mathbf{L}_{i*} & = \mathbf{U}_{i*}\Sigma \\
\mathbf{R} & = \mathbf{V}^T
\end{align*}
\]

PCA factors are hard to interpret.

Lee and Seung, 1999.
Decomposing faces (NMF)

$$D_{i*} \text{ (original)}$$

$$[LR]_{i*} = L_{i*} R$$

NMF factors correspond to parts of faces.

Lee and Seung, 1999.
Decomposing digits (NMF)

NMF factors correspond to parts of digits and “background”.

Cichocki et al., 2009.
Some applications

- Text mining (more later)
- Bioinformatics
- Microarray analysis
- Mineral exploration
- Neuroscience
- Image understanding
- Air pollution research
- Chemometrics
- Spectral data analysis
- Linear sparse coding
- Image classification
- Clustering
- Neural learning process
- Sound recognition
- Remote sensing
- Object characterization

...
**Gaussian NMF**

- Gaussian NMF is the most basic form of non-negative factorizations:

\[
\text{minimize} \quad \| D - LR \|_F^2 \\
\text{s.t.} \quad L \in \mathbb{R}^{m \times r}_+ \\
R \in \mathbb{R}^{r \times n}_+
\]

- Truncated SVD minimizes the same objective (but without non-negativity constraints)

- Many other variants exist
  - Different objective functions (e.g., KL-divergence)
  - Additional regularizations (e.g., \(L_1\)-regularization)
  - Different constraints (e.g., orthogonality of \(R\))
  - Different compositions (e.g., 3 matrices)
  - multi-layer NMF, semi-NMF, sparse NMF, tri-NMF, symmetric NMF, orthogonal NMF, non-smooth NMF (nsNMF), overlapping NMF, convolutive NMF (CNMF), k-Means, ...
k-Means can be seen as a variant of NMF

Additional constraint: $\mathbf{L}$ contains exactly one 1 in each row, rest 0

$\mathbf{D}_{i*}$ (original)

$[\mathbf{LR}]_{i*} = \mathbf{L}_{i*}$

$k$-Means factors correspond to prototypical faces.

Lee and Seung, 1999.
NMF is not unique

- Factors are not “ordered”

\[
\begin{array}{cccccc}
1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 \\
\end{array}
\]

\[
\begin{array}{cccccc}
1 & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 \\
\end{array}
\]

- One way of ordering: decreasing Frobenius norm of components
  (i.e., order by \( \|L_k R_{k*}\|_F \))

- Factors/components are not unique

\[
\begin{array}{cccccc}
1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 \\
\end{array}
\]

\[
\begin{array}{cccccc}
1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

\[
=\]

\[
\begin{array}{cccccc}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 \\
\end{array}
\]

\[
\begin{array}{cccccc}
0.5 & 1 & 0.5 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

\[
=\]

\[
\begin{array}{cccccc}
0.5 & 0 & 0.5 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 \\
\end{array}
\]

\[
\begin{array}{cccccc}
1 & 0 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

\[
=\]

\[
\begin{array}{cccccc}
0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 \\
\end{array}
\]

Additional constraints or regularization can encourage uniqueness.
NMF is not hierarchical

- **Rank-1 NMF**

  \[
  \begin{bmatrix}
  1 & 1 & 1 & 1 & 1 \\
  0 & 1 & 0 & 1 & 0 \\
  0 & 1 & 0 & 1 & 0 
  \end{bmatrix}
  \approx
  \begin{bmatrix}
  0.6 & 1.3 & 0.6 & 1.3 & 0.6 \\
  0.3 & 0.8 & 0.3 & 0.8 & 0.3 \\
  0.3 & 0.8 & 0.3 & 0.8 & 0.3 
  \end{bmatrix}
  =
  \begin{bmatrix}
  0.8 \\
  0.5 \\
  0.5 
  \end{bmatrix}
  \begin{bmatrix}
  0.7 & 1.5 & 0.7 & 1.5 & 0.7 
  \end{bmatrix}
  \]

- **Rank-2 NMF**

  \[
  \begin{bmatrix}
  1 & 1 & 1 & 1 & 1 \\
  0 & 1 & 0 & 1 & 0 \\
  0 & 1 & 0 & 1 & 0 
  \end{bmatrix}
  =
  \begin{bmatrix}
  1 & 1 & 1 & 1 & 1 \\
  0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 
  \end{bmatrix}
  +
  \begin{bmatrix}
  0 & 0 & 0 & 0 & 0 \\
  0 & 1 & 0 & 1 & 0 \\
  0 & 1 & 0 & 1 & 0 
  \end{bmatrix}
  =
  \begin{bmatrix}
  1 & 0 \\
  0 & 1 \\
  0 & 1 
  \end{bmatrix}
  +
  \begin{bmatrix}
  1 & 1 & 1 & 1 & 1 \\
  0 & 1 & 0 & 1 & 0 \\
  0 & 1 & 0 & 1 & 0 
  \end{bmatrix}
  \]

- **Best rank-\(k\) approximation may differ significantly from best rank-\((k-1)\) approximation**
- **Rank influences sparsity, interpretability, and statistical fidelity**
- **Optimum choice of rank is not well-studied (often requires experimentation)**
NMF is difficult

We focus on minimizing \( L(L, R) = \|D - LR\|_F^2 \).

- For varying \( m, n, \) and \( r \), problem is NP-hard
- When \( \text{rank}(D) = 1 \) (or \( r = 1 \)), can be solved in polynomial time
  1. Take first non-zero column of \( D \) as \( L_{m \times 1} \)
  2. Determine \( R_{1 \times n} \) entry by entry (using the fact that \( D_{*j} = LR_{1j} \))
- Problem is not convex
  - Local optimum may not correspond to global optimum
  - Generally little hope to find global optimum
- But: Problem is biconvex
  - For fixed \( R \), \( f(L) = \|D - LR\|_F^2 \) is convex

\[
f(L) = \sum_i \|D_{*i} - L_{i*}R\|_F^2 \quad \text{(chain rule)}
\]
\[
\nabla_{L_{ik}} f(L) = -2(D_{*i} - L_{i*}R)R_{k*}^T \quad \text{(product rule)}
\]
\[
\nabla_{L_{ik}}^2 f(L) = 2R_{k*}R_{k*}^T \geq 0 \quad \text{(does not depend on } L)\]

- For fixed \( L \), \( f(R) = \|D - LR\|_F^2 \) is convex
- Allows for efficient algorithms
General framework

- Gradient descent generally slow
- Stochastic gradient descent inappropriate
- Key approach: alternating minimization
  1. Pick starting point $\mathbf{L}_0$ and $\mathbf{R}_0$
  2. while not converged do
  3. Keep $\mathbf{R}$ fixed, optimize $\mathbf{L}$
  4. Keep $\mathbf{L}$ fixed, optimize $\mathbf{R}$
  5. end while
- Update steps 3 and 4 easier than full problem
- Also called alternating projections or (block) coordinate descent
- Starting point
  - Random
  - Multi-start initialization: try multiple random starting points, run a few epochs, continue with best
  - Based on SVD
  - …
Example

Ignore non-negativity for now. Consider the regularized least-square error:

\[ L(L, R) = \|D - LR\|_F^2 + \lambda (\|L\|_F^2 + \|R\|_F^2) \]

By setting \( m = n = r = 1 \), \( D = (1) \) and \( \lambda = 0.05 \), we obtain

\[ L(l, r) = (1 - lr)^2 + 0.05(l^2 + r^2) \]

\[ \nabla_l f(l) = -2r(1 - lr) + 0.1l \]
\[ \nabla_r f(r) = -2l(1 - lr) + 0.1r \]

Local optima:

\( \left( \sqrt{\frac{19}{20}}, \sqrt{\frac{19}{20}} \right), \left( -\sqrt{\frac{19}{20}}, -\sqrt{\frac{19}{20}} \right) \)

Stationary point: \((0,0)\)
Example (ALS)

- \( f(l, r) = (1 - lr)^2 + 0.05(l^2 + r^2) \)
- \( l \leftarrow \min_l f(l) = \frac{2r}{2r^2+0.1} \)
- \( r \leftarrow \min_r f(r) = \frac{2l}{2l^2+0.1} \)

<table>
<thead>
<tr>
<th>Step</th>
<th>( l )</th>
<th>( r )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>0.49</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>0.49</td>
<td>1.68</td>
</tr>
<tr>
<td>3</td>
<td>0.58</td>
<td>1.68</td>
</tr>
<tr>
<td>4</td>
<td>0.58</td>
<td>1.49</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>100</td>
<td>0.97</td>
<td>0.97</td>
</tr>
</tbody>
</table>

- Converges to local minimum
Example (ALS)

- \( f(l, r) = (1 - lr)^2 + 0.05(l^2 + r^2) \)
- \( l \leftarrow \min_l f(l) = \frac{2r}{2r^2 + 0.1} \)
- \( r \leftarrow \min_r f(r) = \frac{2l}{2l^2 + 0.1} \)

<table>
<thead>
<tr>
<th>Step</th>
<th>( l )</th>
<th>( r )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>:</td>
<td>:</td>
</tr>
</tbody>
</table>

- Converges to stationary point
Alternating non-negative least squares (ANLS)

- Uses non-negative least squares approximation of $L$ and $R$:

\[
\begin{align*}
\text{argmin}_{L \in \mathbb{R}_{+}^{m \times r}} \|D - LR\|_{F}^{2} \quad \text{and} \quad \text{argmin}_{R \in \mathbb{R}_{+}^{r \times n}} \|D - LR\|_{F}^{2}
\end{align*}
\]

- Equivalently: find non-negative least squares solution to $LR = D$

- Common approach: Solve unconstrained least squares problems and “remove” negative values. E.g., when columns (rows) of $L$ ($R$) are linearly independent, set

\[
L = [DR^{\dagger}]_{\epsilon} \quad \text{and} \quad R = [L^{\dagger}D]_{\epsilon}
\]

where

- $R^{\dagger} = R^{T}(RR^{T})^{-1}$ is the right pseudo-inverse of $R$
- $L^{\dagger} = (L^{T}L)^{-1}L^{T}$ is the left pseudo-inverse of $L$
- $[a]_{\epsilon} = \max\{ \epsilon, a \}$ for $\epsilon = 0$ or some small constant (e.g., $\epsilon = 10^{-9}$)

- Difficult to analyze due to non-linear update steps

- Often slow convergence to a “bad” local minimum (better when regularized)
Example (ANLS)

- $f(l, r) = (1 - lr)^2 + 0.05(l^2 + r^2)$ and set $\epsilon = 10^{-9}$
- $l \leftarrow \left[ \frac{2r}{2r^2 + 0.1} \right] \epsilon$
- $r \leftarrow \left[ \frac{2l}{2l^2 + 0.1} \right] \epsilon$

<table>
<thead>
<tr>
<th>Step</th>
<th>$l$</th>
<th>$r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>$1 \cdot 10^{-9}$</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>$1 \cdot 10^{-9}$</td>
<td>$2 \cdot 10^{-8}$</td>
</tr>
<tr>
<td>3</td>
<td>$4 \cdot 10^{-7}$</td>
<td>$2 \cdot 10^{-8}$</td>
</tr>
<tr>
<td>4</td>
<td>$4 \cdot 10^{-7}$</td>
<td>$8 \cdot 10^{-6}$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>100</td>
<td>0.97</td>
<td>0.97</td>
</tr>
</tbody>
</table>

- Converges to local minimum
Hierarchical alternating least squares (HALS)

- Work locally on a single factor, then proceed to next factor, and so on
- Let $D^{(k)}$ be the residual matrix (error) when $k$-th factor is removed:

$$D^{(k)} = D - LR + L^*k R^*k = D - \sum_{k' \neq k} L^*k' R^*k'$$

- HALS minimizes $\|D^{(k)} - L^*k R^*k\|_F^2$ for $k = 1, 2, \ldots, r, 1, \ldots$
  (equivalently: finds best solution for $k$-th factor, fixing the rest)
- In each iteration, set (once or multiple times):

$$L^*k = \frac{1}{\|R^*k\|_F^2} \left[ D^{(k)} R^T_k \right] \epsilon \quad \text{and} \quad R^T_k = \frac{1}{\|L^*k\|_F^2} \left[ (D^{(k)})^T L^*k \right] \epsilon$$

- $D^{(k)}$ can be incrementally maintained $\rightarrow$ fast implementation

$$D^{(k+1)} = D^{(k)} + L^*k R^*k - L^*(k+1) R^{(k+1)*}$$

- Often better performance in practice than ANLS
- Converges to stationary point when initialized with positive matrix and sufficiently small $\epsilon$
Multiplicative updates

- Gradient descent step with step size $\eta_{kj}$
  \[ R_{kj} \leftarrow R_{kj} + \eta_{kj} ([L^T D]_{kj} - [L^T LR]_{kj}) \]

- Setting $\eta_{kj} = \frac{R_{kj}}{(L^T LR)_{kj}}$, we obtain the multiplicative update rules
  \[ L \leftarrow L \odot \frac{DR^T}{LRR^T} \quad \text{and} \quad R \leftarrow R \odot \frac{L^T D}{L^T LR}, \]
  where multiplication ($\odot$) and division are element-wise

- Does not necessarily find optimum $L$ (or $R$), but can be shown to never increase loss

- Faster than ANLS (no computation of pseudo-inverse), easy to implement and parallelize

- Zeros in factors are problematic (divisions become undefined)
  \[ L \leftarrow L \odot \frac{[DR^T]_\epsilon}{LRR^T + \epsilon} \quad \text{and} \quad R \leftarrow R \odot \frac{[L^T D]_\epsilon}{L^T LR + \epsilon} \]

Example (multiplicative updates)

- \( f(l, r) = (1 - lr)^2 + 0.05(l^2 + r^2) \)
- \( l \leftarrow l \frac{l - 0.05l}{lr^2} \)
- \( r \leftarrow r \frac{l - 0.05r}{l^2 r} \)

<table>
<thead>
<tr>
<th>Step</th>
<th>( l )</th>
<th>( r )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>0.48</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>0.48</td>
<td>1.66</td>
</tr>
<tr>
<td>3</td>
<td>0.59</td>
<td>1.66</td>
</tr>
<tr>
<td>4</td>
<td>0.58</td>
<td>1.45</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>100</td>
<td>0.97</td>
<td>0.97</td>
</tr>
</tbody>
</table>

Converges to local minimum
Outline

1. Non-Negative Matrix Factorization
2. Algorithms
3. Probabilistic Latent Semantic Analysis
4. Summary
Topic modeling

- Consider a document-word matrix constructed from some corpus

\[
\tilde{D} = \begin{pmatrix}
\text{air} & \text{water} & \text{pollution} & \text{democrat} & \text{republican} \\
\text{doc 1} & 3 & 2 & 8 & 0 & 0 \\
\text{doc 2} & 1 & 4 & 12 & 0 & 0 \\
\text{doc 3} & 0 & 0 & 0 & 10 & 11 \\
\text{doc 4} & 0 & 0 & 0 & 8 & 5 \\
\text{doc 5} & 1 & 1 & 1 & 1 & 1
\end{pmatrix}
\]

- Documents seem to talk about two “topics”
  1. Environment (with words air, water, and pollution)
  2. Congress (with words democrat and republican)

Can we automatically detect topics in documents?
A probabilistic viewpoint

- Let’s normalize such that the entries sum to unity

\[
D = \begin{pmatrix}
0.04 & 0.03 & 0.12 & 0.00 & 0.00 \\
0.01 & 0.06 & 0.17 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.14 & 0.16 \\
0.00 & 0.00 & 0.00 & 0.12 & 0.07 \\
0.01 & 0.01 & 0.01 & 0.01 & 0.015
\end{pmatrix}
\]

- Put all words in an urn and draw. The probability to draw word \( w \) from document \( d \) is given by

\[
P(d, w) = D_{dw}
\]

- Matrix \( D \) can represent any probability distribution

- pLSA tries to find a distribution that is “close” to \( D \) but exposes information about topics
Probabilistic latent semantic analysis (pLSA)

**Definition (pLSA, NMF formulation)**

Given a rank $r$, find matrices $L$, $\Sigma$, and $R$ such that

$$D \approx L \Sigma R$$

where

- $L_{m \times r}$ is a non-negative, column-stochastic matrix (columns sum to unity),
- $\Sigma_{r \times r}$ is a non-negative, diagonal matrix that sums to unity, and
- $R_{r \times n}$ is a non-negative, row-stochastic matrix (rows sum to unity).

- $\approx$ is usually taken to be the (generalized) KL divergence
- Additional regularization or tempering necessary to avoid overfitting
Example

- pLSA factorization of example matrix

\[
\begin{array}{cccc}
\text{air} & \text{wat} & \text{pol} & \text{dem} \\
0.04 & 0.03 & 0.12 & 0 \\
0.01 & 0.06 & 0.17 & 0 \\
0 & 0 & 0 & 0.14 \\
0 & 0 & 0 & 0.12 \\
0.01 & 0.01 & 0.01 & 0.01 \\
\end{array}
\begin{array}{c}
\approx \\
D \\
\approx \\
L \\
\Sigma \\
R \\
\end{array}
\begin{array}{cccc}
\text{air} & \text{wat} & \text{pol} & \text{dem} \\
0.39 & 0 & 0 & 0.48 \\
0.52 & 0 & 0 & 0.52 \\
0 & 0.58 & 0 & 0 \\
0 & 0.36 & 0 & 0 \\
0.09 & 0.06 & 0 & 0 \\
\end{array}
\]

- Rank \( r \) corresponds to number of topics
- \( \Sigma_{kk} \) corresponds to overall frequency of topic \( k \)
- \( L_{dk} \) corresponds to contribution of document \( d \) to topic \( k \)
- \( R_{kw} \) corresponds to frequency of word \( w \) in topic \( k \)
- pLSA constraints allow for probabilistic interpretation
  \[
P(d, w) \approx [L\Sigma R]_{dw} = \sum_k \Sigma_{kk} L_{dk} R_{kw} = \sum_k P(k) P(d \mid k) P(w \mid k)
\]
- pLSA model imposes conditional independence constraints
  \( \rightarrow \) restricted space of distributions
Another example

Concepts (10 of 128) extracted from Science Magazine articles (12K)

pLSA geometry

- Rewrite probabilistic formulation

\[ P(d, w) = \sum_k P(k)P(d | k)P(w | k) \]

\[ P(w | d) = \sum_z P(w | z)P(z | d) \]

- Generative process of creating a word
  1. Pick a document according to \( P(d) \)
  2. Select a topic acc. to \( P(z | d) \)
  3. Select a word acc. to \( P(w | z) \)

Figure 2. Sketch of the probability simplex and a convex region spanned by class-conditional probabilities in the aspect model.

Kullback-Leibler divergence (1)

- Let $\tilde{D}$ be the unnormalized word-count data and denote by $N$ total number of words.
- Likelihood of seeing $\tilde{D}$ when drawing $N$ words with replacement is proportional to

$$\prod_{d=1}^{m} \prod_{w=1}^{n} P(d, w) \tilde{D}_{dw}$$

- pLSA maximizes the log-likelihood of seeing the data given the model

$$\log P(\tilde{D} | L, \Sigma, R) \propto \sum_{d=1}^{m} \sum_{w=1}^{n} \tilde{D}_{dw} \log P(d, w | L, \Sigma, R)$$

$$\propto - \sum_{d=1}^{m} \sum_{w=1}^{n} D \log \frac{1}{[L \Sigma R]_{dw}}$$

$$= - \sum_{d=1}^{m} \sum_{w=1}^{n} D_{dw} \log \frac{D_{dw}}{[L \Sigma R]_{dw}} + c_D$$

Kullback-Leibler divergence

Gaussier and Goutte, 2005.
Kullback-Leibler divergence (2)

- KL divergence

\[
D_{KL}(P \parallel Q) = \sum_{d=1}^{m} \sum_{w=1}^{n} P_{dw} \log \frac{P_{dw}}{Q_{dw}}
\]

- Interpretation: expected number of extra bits for encoding a value drawn from \(P\) using an optimum code for distribution \(Q\)

- \(D_{KL}(P \parallel Q) \geq 0\)

- \(D_{KL}(P \parallel P) = 0\)

- \(D_{KL}(P \parallel Q) \neq D_{KL}(Q \parallel P)\)

- NMF-based pLSA algorithms minimize the generalized KL divergence

\[
D_{GKL}(\tilde{P} \parallel \tilde{Q}) = \sum_{d=1}^{m} \sum_{w=1}^{n} (\tilde{P}_{dw} \log \frac{\tilde{P}_{dw}}{\tilde{Q}_{dw}} - \tilde{P}_{dw} + \tilde{Q}_{dw}),
\]

where \(\tilde{P} = \tilde{D}\) and \(\tilde{Q} = \text{L} \tilde{\Sigma} \text{R}\)
Multiplicative updates for GKL (w/o tempering)

- We first find a decomposition $\tilde{D} \approx \tilde{L}\tilde{R}$, where $\tilde{L}$ and $\tilde{R}$ are non-negative matrices.
- Update rules

$$
\begin{align*}
\tilde{L} &{} \leftarrow \tilde{L} \circ \frac{\tilde{D}}{\tilde{L}\tilde{R}} \tilde{R}^T \text{diag}(1/\text{rowSums}(\tilde{R})) \\
\tilde{R} &{} \leftarrow \tilde{R} \circ \text{diag}(1/\text{colSums}(\tilde{L})) \tilde{L}^T \frac{\tilde{D}}{\tilde{L}\tilde{R}}
\end{align*}
$$

- GKL is non-increasing under these update rules.
- Normalize by rescaling columns of $\tilde{L}$ and rows of $\tilde{R}$ to obtain

$$
\begin{align*}
L &{} = \tilde{L} \text{diag}(1/\text{colSums}(\tilde{L})) \\
R &{} = \text{diag}(1/\text{rowSums}(\tilde{R}))\tilde{R} \\
\tilde{\Sigma} &{} = \text{diag}(\text{colSums}(\tilde{L}) \circ \text{rowSums}(\tilde{R})) \\
\Sigma &{} = \tilde{\Sigma} / \sum_k \tilde{\Sigma}_{kk}
\end{align*}
$$

Applications of pLSA

- Topic modeling
- Clustering documents
- Clustering terms
- Information retrieval
  - Treat query $q$ as a “new” document (new row in $\tilde{D}$ and $L$)
  - Determine $P(k \mid q)$ by keeping $\Sigma$ and $R$ fixed (“fold in” the query)
  - Retrieve documents with similar topic mixture as query
  - Can deal with synonymy and polysemy

Better generalization performance than LSA (=SVD), esp. with tempering

In practice, outperformed by Latent Dirichlet Allocation (LDA)

Outline

1. Non-Negative Matrix Factorization

2. Algorithms

3. Probabilistic Latent Semantic Analysis

4. Summary
Lessons learned

- Non-negative matrix factorization (NMF) appears natural for non-negative data

- NMF encourages parts-based decomposition, interpretability, and (sometimes) sparseness

- Many variants, many applications

- Usually solved via alternating minimization algorithms
  - Alternating non-negative least squares (ANLS)
  - Projected gradient local hierarchical ALS (HALS)
  - Multiplicative updates

- pLSA is an approach to topic modeling that can be seen as an NMF
Literature

- David Skillicorn
  *Understanding Complex Datasets: Data Mining with Matrix Decompositions* (Chapter 8)
  Chapman and Hall, 2007

- Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan, and Shun-ichi Amari
  *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*
  Wiley, 2009

- Yifeng Li and Alioune Ngom
  *The NMF MATLAB Toolbox*
  http://cs.uwindsor.ca/~li11112c/nmf

- Renaud Gaujoux and Cathal Seoighe
  *NMF R package*
  http://cran.r-project.org/web/packages/NMF/index.html

- References given at bottom of slides