Pattern Recognition Advanced

Topic Models

mcuturi@i.kyoto-u.ac.jp
Today’s Lecture

- Objective: unveil automatically
  - topics in large corpora of histograms,
  - distribution of topics in each text (or more generally object)
- These techniques are called topic models.
- Topic models are related to other algorithms:
  - dictionary learning in computer vision,
  - nonnegative matrix factorization
Today’s Lecture

• A lot of work in the previous decade
  ○ Start with a precursor: **Latent Semantic Indexing** (‘88)
  ○ follow with **probabilistic Latent Semantic Indexing** (‘99)
  ○ continue with **Latent Dirichlet Allocation** (‘03)
  ○ and finish with **Pachinko Allocation** (‘06).

• This field is still very active...
  ○ **non-parametric Bayes** techniques such as
    **Chinese Restaurant Process, Indian Buffet Process**
  ○ new algorithms using **non-negative matrix factorization**

• These ideas can be all seen as a generalization of PCA, where one demands more structure from the principal components.
Reminder: The Naive Bayes Assumption

- From a factorization

\[ P(C, w_1, \ldots, w_n) = \prod_{i=1}^{n} P(w_i|C, w_1, \ldots, w_{i-1}) \]

which handles all the **conditional** structures of text,

- we assume that each word appears **independently conditionally to** \( C \),

\[ P(w_i|C, w_1, \ldots, w_{i-1}) = P(w_i|C, w_1, \ldots, w_{i-1}) \]
\[ = P(w_i|C) \]

- and thus

\[ P(C, w_1, \ldots, w_n) = \prod_{i=1}^{n} P(w_i|C) \]

- The only thing the Bayes classifier considers is **word histograms**
A Few Examples of Learned Topics

Image Source: [Topic Models](http://www.topicmodels.org) Blei Lafferty (2009)
contractual employment female markets criminal
expectation industrial men earnings discretion
gain local women investors justice
promises jobs see sec civil
expectations employees sexual research process
breach relations note structure federal
enforcing unfair employer managers see
supra agreement discrimination firm officer
note economic harassment parole
perform case gender large inmates

**Figure 3.** Five topics from a 50-topic model fit to the *Yale Law Journal* from 1980–2003.

Image Source: [Topic Models](https://www.topicmodels.org) Blei Lafferty (2009)
FIGURE 4. The analysis of a document from Science. Document similarity was computed using Eq. (4); topic words were computed using Eq. (3).
Latent Semantic Indexing

a variation of PCA for normalized word counts...

- Uncover recurring **patterns** in text by considering examples.
- These patterns are **groups of words which tend to appear together**.
- To do so, given a set of $n$ documents, LSI considers a document/word matrix

\[
T = [t_{f_{i,j}}] \in \mathbb{R}^{m \times n}
\]

where $t_{f_{i,j}}$ counts the **term-frequency** of word $j$ in text $i$.
- Using this information, LSI builds a set of influential **groups of words**
- This is similar in spirit to **PCA**:
  - learn **principal components** from data $X \in \mathbb{R}^{d \times N}$ by diagonalizing $XX^T$.
  - represent each datapoint as the **sum of a few principal components** in that basis

\[
x_i = \sum_{j=1}^{d} \langle x_i, e_j \rangle e_j
\]
  - use the **principal coordinates** for denoising or clustering or in supervised tasks.
Renormalizing Frequencies, Preprocessing

Rather than considering only $tf_{ij}$, introduce a term $x_{ij} = l_{ij}g_i$ which incorporates both local and global weights.

- **Local weights (i.e. relative to a term $i$ and document $j$)**
  - **binary weight**: $l_{ij} = \delta_{tf_{ij}>0}$
  - **simple frequency** $l_{ij} = tf_{ij}$,
  - **hellinger** $l_{ij} = \sqrt{tf_{ij}}$
  - **log(1+)** $l_{ij} = \log(tf_{ij} + 1)$
  - **relative to max** $l_{ij} = \frac{tf_{ij}}{2 \max_i(tf_{ij})} + \frac{1}{2}$

- **Global weights (i.e. relative to a term $i$ across all documents)**
  - **equally weighted documents** $g_i = 1$
  - **$l_2$ norm of frequencies** $g_i = \frac{1}{\sqrt{\sum_j tf_{ij}^2}}$
  - $g_i = g_f_i/df_i$, where $g_f_i = \sum_j tf_{ij}$, and $df_i = \sum_j \delta_{tf_{ij}>0}$
  - $g_i = \log_2 \frac{n}{1+df_i}$
  - $g_i = 1 + \sum_j \frac{p_{ij} \log p_{ij}}{\log n}$, where $p_{ij} = \frac{tf_{ij}}{g_{f_i}}$
Word/Document Representation

- Typically, one can define

\[ X = [x_{ij}], \quad x_{ij} = \left( 1 + \sum_j p_{ij} \log p_{ij} \right) \left( \sum_i |l_{ij}| g_i \right) \]

- After preprocessing, consider the *normalized* occurrences of words,

\[
d_j \downarrow \begin{bmatrix}
x_{1,1} & \cdots & x_{1,n} \\
\vdots & \ddots & \vdots \\
x_{m,1} & \cdots & x_{m,n}
\end{bmatrix}
\]

- Represents both term vectors \( t_i \) and document vectors \( d_j \)

- \( \rightarrow \) normalized representation of points (documents) in variables (terms), or vice-versa.
Word/Document Representation

• Each row represents a term, described by its relation to each document:

\[ t_i^T = \begin{bmatrix} x_{i,1} & \ldots & x_{i,n} \end{bmatrix} \]

• Each column represents a document, described by its relation to each word:

\[ d_j = \begin{bmatrix} x_{1,j} \\ \vdots \\ x_{m,j} \end{bmatrix} \]

• \( t_i^T t_i' \) is the correlation between terms \( i, i' \) over all documents.
  ○ \( XX^T \) contains all these dot products.

• \( d_j^T d_j' \) is the correlation between documents \( j, j' \) over all terms.
  ○ \( X^T X \) contains all these dot products
Singular Value Decomposition

- Consider the **singular value decomposition** (SVD) of $X$,

$$X = U \Sigma V^T$$

where $U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{n \times n}$ are orthogonal matrices and $\Sigma \in \mathbb{R}^{m \times n}$ is diagonal.

- The matrix products highlighting term/documents correlations are

$$XX^T = (U \Sigma V^T)(U \Sigma V^T)^T = (U \Sigma V^T)(V^T \Sigma^T U^T) = U \Sigma V^T V^T \Sigma U^T = U \Sigma \Sigma^T U^T$$

$$X^TX = (U \Sigma V^T)^T(U \Sigma V^T) = (V^T \Sigma U^T)(U \Sigma V^T) = V^T U^T U \Sigma V^T = V \Sigma \Sigma^T V^T$$

- $U$ contains the **eigenvectors** of $XX^T$,

- $V$ contains the **eigenvectors** of $X^TX$.

- Both $XX^T$ and $X^TX$ have the same **non-zero** eigenvalues, given by the non-zero entries of $\Sigma \Sigma^T$. 
Singular Value Decomposition

- Let \( l \) be the number of non-zero eigenvalue of \( \Sigma \Sigma^T \). Then

\[
X = \hat{X}(l) \overset{\text{def}}{=} U(l) \Sigma(l) V^T(l)
\]

\[
(t_i^T) \rightarrow \begin{bmatrix}
x_{1,1} & \cdots & x_{1,n} \\
\vdots & \ddots & \vdots \\
x_{m,1} & \cdots & x_{m,n}
\end{bmatrix} = (\tau_i^T) \rightarrow \begin{bmatrix} u_1 \cdots u_l \end{bmatrix} \cdot \begin{bmatrix} \sigma_1 & \cdots & 0 \\
\vdots & \ddots & \vdots \\
0 & \cdots & \sigma_l \end{bmatrix} \cdot \begin{bmatrix} v_1 \\
\vdots \\
v_l \end{bmatrix}
\]

- \( \sigma_1, \ldots, \sigma_l \) are the **singular** values,

- \( u_1, \ldots, u_l \) and \( v_1, \ldots, v_l \) are the **left and right** singular vectors.

- The only part of \( U \) that contributes to \( t_i \) is its \( i \)'th row, written \( \tau_i \).

- The only part of \( V^T \) that contributes to \( d_j \) is the \( j \)'th column, \( \delta_j \).
A property of the SVD is that for $k \leq l$

$$\hat{X}_k = \arg\min_{X \in \mathbb{R}^{m \times n}, \text{Rank}(X) = k} \| X - X_k \|_F$$

$\hat{X}_k$ is an approximation of $X$ with **low rank**.

The term and document vectors can be considered as **concept spaces**

- the $k$ entries of $\tau_i$ provide the occurrence of term $i$ in the $k^{th}$ concept.
- $\delta^T_j$ provides the relation between document $j$ and each concept.
Latent Semantic Indexing Representation of Documents

We can use LSI to

- Quantify the relationship between documents $j$ and $j'$:
  - compare the vectors $\Sigma_k \delta_j^T$ and $\Sigma_k \hat{\delta}_{j'}$
- Compare terms $i$ and $i'$ through $\tau_i^T \Sigma_k$ and $\tau_{i'}^T \Sigma_k$,
  - provides a clustering of the terms in the concept space.
- Project a new document onto the concept space,

$$q \rightarrow \chi = \Sigma_k^{-1} U_k^T q$$
Probabilistic Latent Semantic Indexing
Latent Variable Probabilistic Modeling

- PLSI adds on LSI by considering a **probabilistic** modeling built upon a **latent** class variable.

- Namely, the joint likelihood that word $w$ appears in document $d$ depends on an **unobserved variable** $z \in \mathcal{Z} = \{z_1, \ldots, z_K\}$

which defines a joint probability model over $\mathcal{W} \times \mathcal{D}$ (words × documents) as

$$p(d, w) = P(d)P(w|d), \quad P(w|d) = \sum_{z \in \mathcal{Z}} P(w|z)P(z|d)$$

which thus gives

$$p(d, w) = P(d) \sum_{z \in \mathcal{Z}} P(w|z)P(z|d)$$

we also have that

$$p(d, w) = \sum_{z \in \mathcal{Z}} P(z)P(w|z)P(d|z)$$
The different parameters of the probability below

\[ p(d, w) = P(d) \sum_{z \in \mathbb{Z}} P(w|z)P(z|d) \]

are all multinomial distribution, distributions on the simplex.

\[ P(z), P(w|z)P(d|z) \]

These coefficients can be estimated using maximum likelihood with latent variables.

Typically using the Expectation Maximization algorithm.
Consider again the formula

\[
p(d, w) = \sum_{z \in Z} P(z)P(w|z)P(d|z)
\]

- If we define matrices
  - \(U = [P(w_i|z_k)]_{ik}\)
  - \(V = [P(d_j|z_k)]_{jk}\)
  - \(\Sigma = \text{diag}(P(z_k))\)

we obtain that

\[
P = [P(w_i, d_j)] = U \Sigma V^T
\]

- \(P\) and \(X\) are the same matrices. We have found a different factorization of \(P\) (or \(X\)).

**Difference**

- In LSI, SVD considers the Frobenius norm to penalize for discrepancies.
- In probabilistic LSI, we use a different criterion: likelihood function.
Probabilistic Latent Semantic Indexing

- The probabilistic viewpoint provides a different cost function
- The probabilistic assumption is explicitated by the following graphical model

Here $\theta$ stands for a document $d$, $M$ number of documents, $N$ number of words in a document.

The plates stand for the fact that such dependencies are repeated $M$ and $N$ times.
Latent Dirichlet Allocation
Dirichlet Distribution

- Dirichlet Distribution is a distribution on the **canonical simplex**

\[ \Sigma_d = \{ x \in \mathbb{R}_+^d \mid \sum_{i=1}^{d} x_i = 1 \} \]

- The density is parameterized by a family \( \beta \) of \( d \) real **positive** numbers,

\[ \beta = (\beta_1, \ldots, \beta_d), \]

has the expression

\[ p_\beta(x) = \frac{1}{B(\beta)} \prod_{i=1}^{d} x_i^{\beta_i-1} \]

with normalizing constant \( B(\beta) \) computed using the Gamma function,

\[ B(\beta) = \frac{\prod_{i=1}^{d} \Gamma(\beta_i)}{\Gamma(\sum_{i=1}^{K} \beta_i)} \]
Dirichlet Distribution

- The Dirichlet distribution is **widely used** to model count histograms.
- Here are for instance $\beta = (6, 2, 2), (3, 7, 5), (6, 2, 6), (2, 3, 4)$.

Probabilistic Modeling in Latent Dirichlet Allocation

- LDA assumes that documents are random mixtures over latent topics,
- each topic is characterized by a distribution over words.
- each word is generated following this distribution.
- Consider $K$ topics,
  - a Dirichlet distribution on topics $\alpha \in \mathbb{R}_+^K$ for documents
  - $K$ multinomials on $V$ words described in a Markov matrix (rows sum to 1)

$$\varphi \in \mathbb{R}_+^{K \times V}, \varphi_k \sim \text{Dir}(\beta).$$
Latent Dirichlet Allocation

Assume that all document $d_i = (w_{i1}, \ldots, w_{iN_i})$ has been generated with the following mechanism:

- Choose a distribution of topics $\theta_i \sim \text{Dir}(\alpha), j \in \{1, \ldots, M\}$ for document $d_i$.
- For each of the word locations $(i, j)$, where $j \in \{1, \ldots, N_i\}$
  - Choose a topic $z_{i,j} \sim \text{Multinomial}(\theta_i)$ at each location $j$ in document $d_i$
  - Choose a word $w_{i,j} \sim \text{Multinomial}(\varphi_{z_{i,j}})$. 
The graphical model of LDA can be displayed as
Latent Dirichlet Allocation

- Inferring now all parameters and latent variables
  - set of $K$ topics for $M$ documents,
  - topic mixture $\theta_i$ of each document $d_i$,
  - set of word probabilities for each topic $\phi_k$,
  - topic $z_{ij}$ of each word $w_{ij}$

is a **Bayesian inference** problem.

\[
P(W, Z, \theta, \varphi; \alpha, \beta) = \prod_{i=1}^{K} P(\varphi_i; \beta) \prod_{j=1}^{M} P(\theta_j; \alpha) \prod_{t=1}^{N} P(Z_{j,t|\theta_j}) P(W_{j,t|\varphi Z_{j,t}})
\]
Latent Dirichlet Allocation

- Many different techniques can be used to tackle this issue.
  - **Gibbs sampling**
    Monte carlo techniques designed to sample from the posterior probability of the parameters given the word observations. In that case one can select the most likely parameters/decomposition as the set of parameters maximizing that posterior.
  - **Variational Bayes**
    Optimization based technique which, instead of maximizing directly $P$ as a function of the parameters (which would be intractable), uses a different family of probabilities that considers local parameters for each document. These parameters are optimized so that the resulting probability is close (in Kullback-Leibler divergence sense) to the original probability $P$.

- This is, in practice, the main challenge to use LDA.
Pachinko Allocation
The idea in one image

- From a simple multinomial (per document) to the Pachinko allocation.

Image Source: [Pachinko Allocation: DAG-Structured Mixture Models of Topic Correlations](#) Li Mc-Callum
The idea in one image

- Difference with LDA