Dimension reduction

October 13, 2022

Dimension reduction

- Modern technologies often produce “high-dimensional data” with large number of variables.
- In high-dimensional data, most of the variables (features) are irrelevant, redundant, or noisy.
- In “dimension reduction”, we want to use a small number of features to capture most information in data. Major approaches include:
  - **Feature selection**: select a subset of the original variables.
  - **Feature projection**: transforms the data from high to low-dimensional space. The features are not among the original variables, but are transformation of the original variables.

Dimension reduction (cont.)

- Examples of Common Feature projection methods:
  - xCA: principal component analysis (PCA), independent component analysis (ICA).
  - Embedding methods: Locally-linear embedding (LLE), t-distributed stochastic neighbor embedding (tSNE), Uniform Manifold Approximation and Projection (UMAP), etc.
  - Non-negative Matrix Factorization (NMF).
  - Local Dirichlet Allocation (LDA).

Non-negative Matrix Factorization

Problem setup:
Given a non-negative matrix V, find non-negative matrices W and H such that:

$$V \approx WH$$

History:

1. Originally proposed in chemometrics to extract information from chemical systems in the 1970s.
2. Later became well-known after Lee & Seung (Nature, 1999; NIPS, 2001) proposed algorithms to conduct NMF and studied their statistical properties.
3. Gained popularity recently among computer vision, audio signal processing, bioinformatics, etc.

Factorization details

Factorization:

- Factorize $m \times n$ matrix V into an $m \times r$ matrix W and $r \times n$ matrix H.
- Usually r is chosen to be smaller than n, so that W and H have lower dimension than original matrix V.

Interpretation:

- NMF can be rewritten by column as:

$$v \approx Wh$$

  where v and h are the corresponding columns of V and H.

- Each v (one observed data point) is approximated by a linear combination of the columns of W, weighted by the elements of h.
  - W: Basis vectors – columns of W spans the low dimensional space.
  - H: The coordinates of the projection of the original data to the low dimensional space.

Cost functions

- To find an approximate factorization $V \approx WH$, one need to define the cost function to quantify the quality of approximation.
- Commonly, the square of the Euclidean distance between A and B is adopted:

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

  Therefore, we can formulate NMF as the following **optimization problem**:

Problem:

- Minimize $\|V - WH\|^2$ with respect to W and H, subject to the constraints that $W, H \geq 0$

Properties

- Although $\|V - WH\|^2$ is convex in W only or H only, it is not convex in both variables together.
- Therefore, one can not expect an algorithm finding the global minima.
- Gradient descent is the simple technique to find at least the local minima.

General NMF algorithm

Input: Non-negative matrix $V \in \mathbb{R}_+^{m \times n}$ and factorization rank r.

Outputs: W, $H \geq 0$ s.t. $V \approx WH$

Algorithm:

1. **(Starting point)** Generate some initial matrices $W^{(0)} \geq 0$ and $H^{(0)} \geq 0$;
2. **(Iteration)** For $t = 1, 2, ...$ do

$$W_{ik}^{(t)} = W_{ik}^{(t-1)} \frac{(VH^{(t-1)T})_{ik}}{(W^{(t-1)}H^{(t-1)}H^{(t-1)T})_{ik}}$$

$$H_{kj}^{(t)} = H_{kj}^{(t-1)} \frac{(W^{(t-1)T}V)_{kj}}{(W^{(t-1)T}W^{(t-1)}H^{(t-1)})_{kj}}$$

   for all $i \in \{1, 2, ..., m\}$, $j \in \{1, 2, ..., n\}$, $k \in \{1, 2, ..., r\}$.

3. **(Stop criteria)** Stop iteration if W and H are at a stationary point.

Note

- Initialization
  - *K-means* can help to generate initial matrices $W^{(0)}$.
  - Can use *Quadratic Programmming (QP)* to get the initial matrices $H^{(0)}$, by solving $V = W^{(0)}H$.
  - Can use existing knowledge or cross-validation to find r.
- Iterations
  - The Euclidean distance $\|V - WH\|$ is non-increasing under the update rules.
  - The proof for non-increasing was provided in Lee & Seung, NIPS 2001, using an auxiliary function similar to that used in the EM algorithm.

Alternative algorithm

One can also use *Quadratic Programming (QP)* in the updating steps:

- For $t = 1, 2, ...$ do

$$W^{(t)} \leftarrow solve\{V = W^{(t)}H^{(t-1)}; W^{(t)} \geq 0\}$$

$$H^{(t)} \leftarrow solve\{V = W^{(t)}H^{(t)}; H^{(t)} \geq 0\}$$

Each step is a QP problem.

The QP updates are easier to implement and modify if there are additional restrictions you want to put on W and/or H.

NMF in R

The *NMF* package provide functions (*nmf*) to solve NMF problems.

```
R Documentation
nmf {NMF}
Running NMF algorithms

Description

The function nmf is a S4 generic defines the main interface to run
NMF algorithms within the framework defined in package NMF.
It has many methods that facilitates applying, developing and testing
NMF algorithms.

The package vignette vignette('NMF') contains an introduction to the
interface, through a sample data analysis.

Usage

  nmf(x, rank, method, ...)
```

Example: NMF in R


```
> library(NMF)

# random data
x <- rmatrix(20,10)

# run default algorithm with rank 2
res <- nmf(x, 2)

# The result is an object of classNMFfit
> fit(res)
<Object of class:NMFstd>
features: 20
basis/rank: 2
samples: 10
```

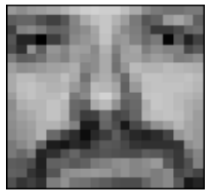
Example: NMF in R (continued)

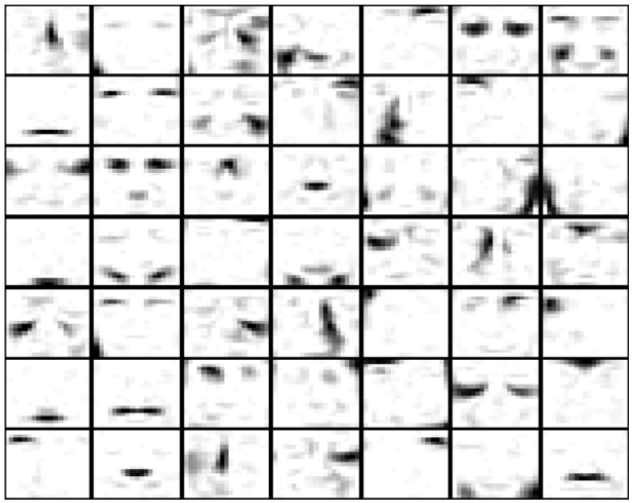
```
# get matrix W using basis() function
w <- basis(res)
dim(w)
[1] 20  2

# get matrix H using coef() function
h <- coef(res)
dim(h)
[1]  2 10

# Additionally, several build-in algorithms are available
# to choose from. Use (method) argument to specify algorithm
> nmfAlgorithm()
 [1] "brunet"    "KL"         "lee"        "Frobenius" "offset"    "nsNMF
 [7] "ls-nmf"    "pe-nmf"    "siNMF"      "snmf/r"    "snmf/l"

# for example
res <- nmf(x, 2, method = "lee")
```

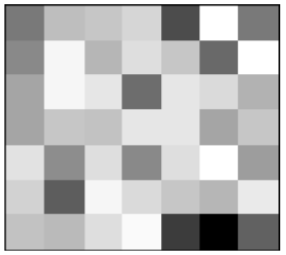
NMF Application 1: Image processing

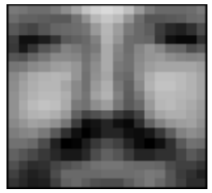
Guillamet et al. (Artificial Intelligence, 2002) conducted facial feature extraction. Each column of data matrix $X \in \mathbb{R}_+^{p \times n}$ is a vectorized gray-level image of a face, with (i, j)-th entry of X being the intensity of the ith pixel in the jth face.

$$\underbrace{X(:,j)}_{j\text{th facial image}} \approx \sum_{k=1}^{r} \underbrace{W(:,k)}_{\text{facial features}} \underbrace{H(k,j)}_{\substack{\text{importance of features} \\ \text{in } j\text{th image}}} = \underbrace{WH(:,j)}_{\substack{\text{approximation} \\ \text{of } j\text{th image}}}$$







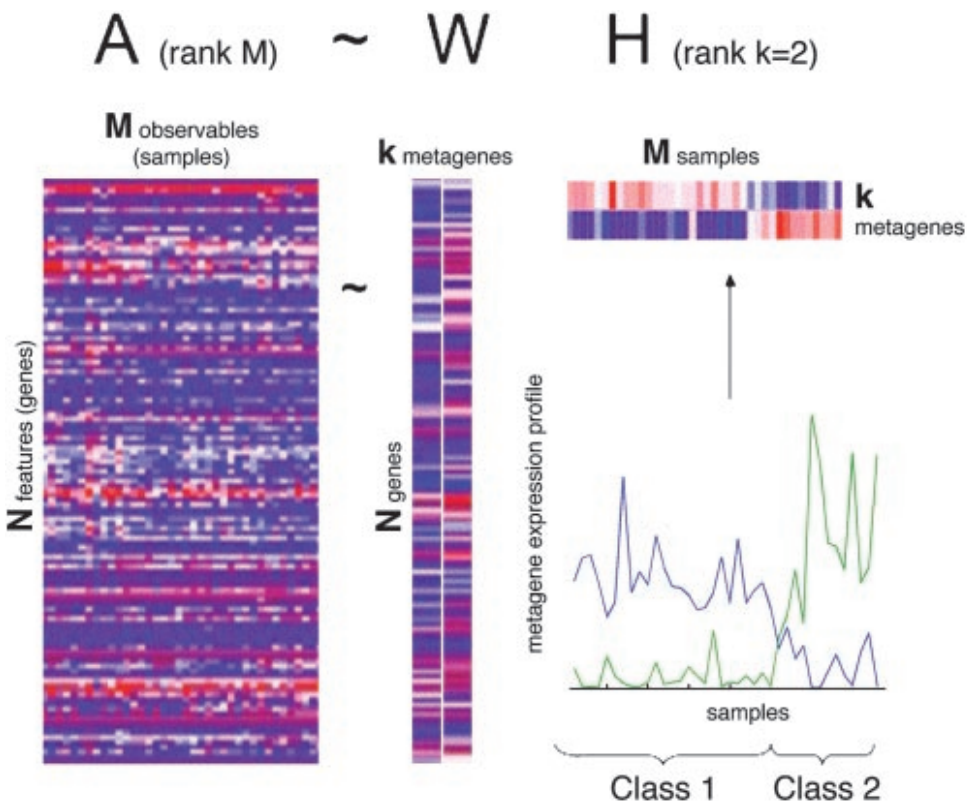


- Columns in W: Basis images (mouth, nose, mustache etc.), after converting back to matrix of the same size as face pictures.
- H: Weights of basis images.

NMF Application 2: Gene expression

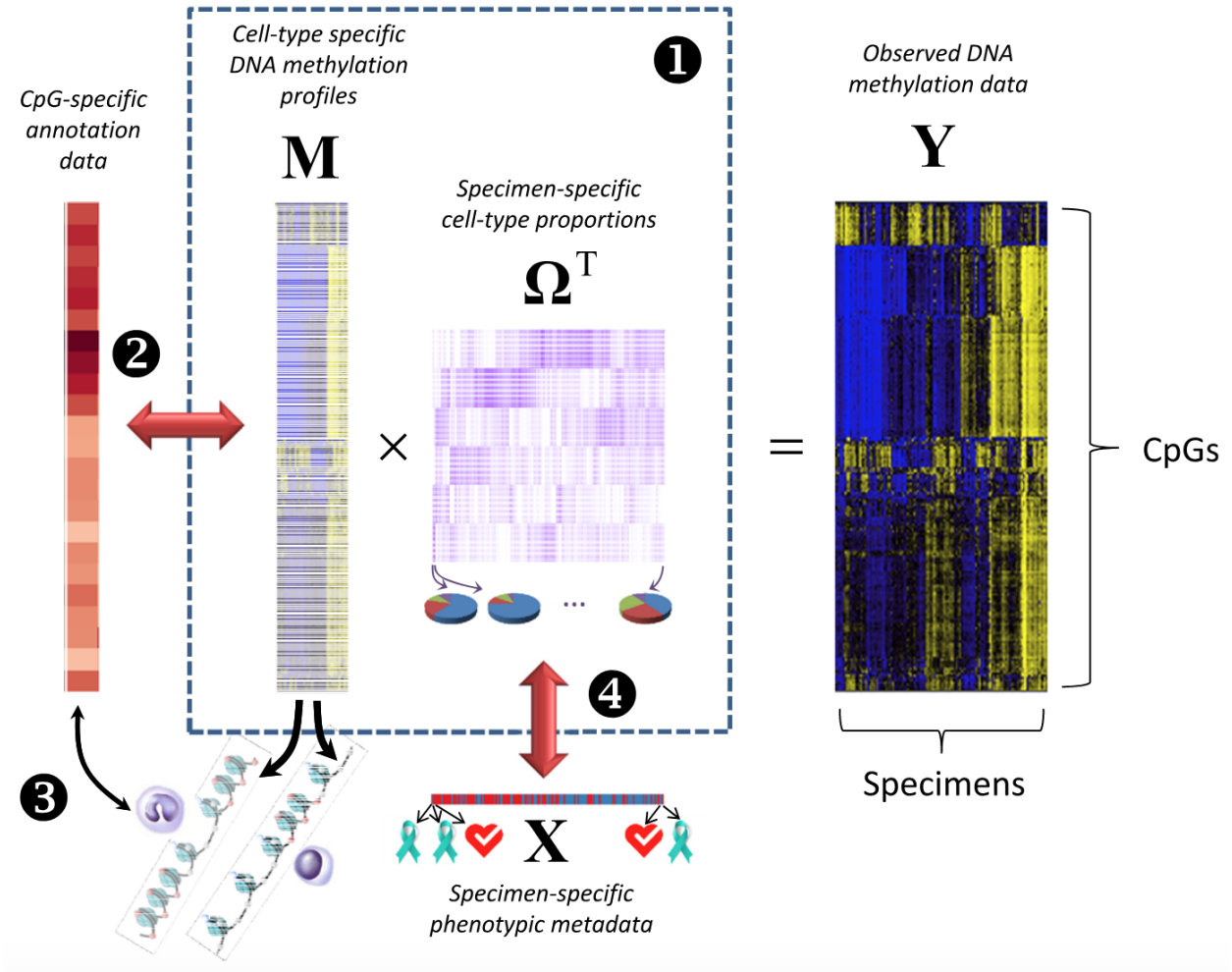
Brunet *et al.* (PNAS, 2004) *Metagenes and molecular pattern discovery using matrix factorization.*

Deconvolute gene expression data (A) into metagene profiles (W) and proportions (H).



NMF Application 3: DNA methylation

Houseman et al. (BMC Bioinformatics, 2016): deconvolutes DNA methylation data (Y) into cell-type specific profile (M) and cell-type proportions (Ω^T).



NMF readings

- Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. **Nature**, 401(6755), 788.
- Lee, D. D., & Seung, H. S. (2001). Algorithms for non-negative matrix factorization. **Advances in neural information processing systems** (pp. 556-562).
- Brunet et al. (2004) Metagenes and molecular pattern discovery using matrix factorization. **PNAS**, 101 (12) 4164-4169; .
- Houseman, E. A. et al. (2016). Reference-free deconvolution of DNA methylation data and mediation by cell composition effects. **BMC bioinformatics**, 17(1), 259.
- Stein-O'Brien G. L. et al. (2018). Enter the matrix: factorization uncovers knowledge from omics. **Trends in Genetics**.

Latent Dirichlet Allocation (LDA)

Modeling text corpora:

- What are the topics in a document?
- Given a document, how to find other documents with similar topics?
- How to classify the documents?

Latent Dirichlet Allocation (LDA):

- A generative probabilistic model for text corpora.
- Proposed by Blei et al. (JMLR, 2003), highly cited.

Notation

Following terms are introduced in modeling text corpora data:

- A **word** is the basic unit of discrete data. In a vocabulary indexed by $\{1, ..., V\}$, a word w in the v^{th} index location of a vocabulary is an unit-basis vector of length V s.t. $w^v = 1$ and $w^\mu = 0$ for $\mu \neq v$
- A **document** is a sequence of N words denoted by $\boldsymbol{w} = (w_1, w_2, ..., w_N)$, where w_n is the n^{th} word in the sequence.
- A **corpus** is a collection of M documents denoted by $\boldsymbol{D} = \{\boldsymbol{w_1}, \boldsymbol{w_2}, ..., \boldsymbol{w_M}\}$

Data generative process

Basic idea:

- Documents are represented as random mixtures over **latent topics**.
- Each topic has its own word usage.

Model: for each document $\boldsymbol{w}$ in a corpus $\boldsymbol{D}$:

1. Choose document length $N \sim Poisson(\xi)$
2. Choose topic allocation $\theta \sim Dir(\alpha)$
3. For each of the N words w_n:
   (a) Choose a topic $z_n \sim Multinomial(\theta)$
   (b) Choose a word w_n from $p(w_n|z_n,\beta)$, a multinomial probability based on the selected topic z_n and the corresponding word usage frequency β.

Notes

1. The observed data are the words in all documents.
2. N: the number of words in a document.
   V: the vocabulary size (total number of different words).
   K: the number of topics.
3. N is an ancillary variable — independent of all other data generating variables (θ and z)
4. Dimensionality K of Dirichlet distribution is assumed known and fixed.
5. α (a K-vector) is the hyper-prior for the proportion of topics.
6. β (a $K \times V$ matrix) is the word usage probabilities for the K topics.
   $\beta_{ij} = p(w^j = 1 | z^i = 1)$. Each row of β sum up to 1.
7. θ (a K-vector) is the proportion of topics for a document. For example, a document can be “70% politics and 30% economy”.

Distributions

The probability density function of θ from Dirichlet distribution:

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{K} \alpha_i))}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \theta_1^{\alpha_1 - 1} \dots \theta_K^{\alpha_K - 1}$$

where $\Gamma()$ is the Gamma function.

Given parameters α, β, for one document, the joint distribution of a topic mixture θ, a set of N topics $\boldsymbol{z}$, and a set of N words $\boldsymbol{w}$ is as follow:

$$p(\theta, \boldsymbol{z}, \boldsymbol{w}|\alpha, \beta) = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta) p(w_n|z_n, \beta)$$

To get the marginal distribution of a document, we integrate over θ and sum over all possible z. The marginal **document** distribution of $\boldsymbol{w}$ is:

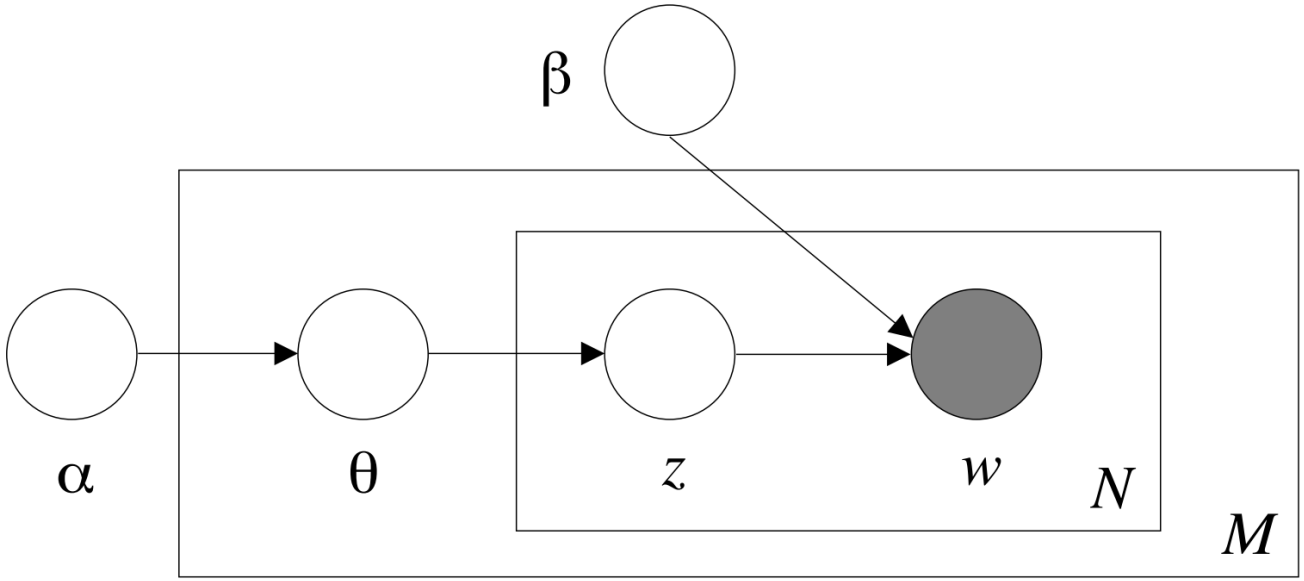
$$p(\boldsymbol{w}|\alpha,\beta) = \int p(\theta|\alpha)\left(\prod_{n=1}^{N}\sum_{z_n} p(z_n|\theta)p(w_n|z_n,\beta)\right)d\theta$$

Taking the product of marginal probabilities of single documents, we obtain the probability of a **corpus**:

$$p(\boldsymbol{D}|\alpha,\beta) = \prod_{d=1}^{M}\int p(\theta_d|\alpha)\left(\prod_{n=1}^{N_d}\sum_{z_{dn}} p(z_{dn}|\theta_d)p(w_{dn}|z_{dn},\beta)\right)d\theta_d$$

M: Number of documents.
θ_d: Topic frequencies for document d.

Graphical model representation



The boxes are “plates” representing replicates.

- Corpus-level parameters: α and β are sampled once per corpus.
- Document-level variables: θ_d are sampled once per document.
- Word-level variables: z_{dn} and w_{dn} are sampled once per word.

Inference

Key problem: the posterior distribution of the hidden variables θ, z, given a document.

By Bayes theorem, we have:

$$p(\theta, \boldsymbol{z}|\boldsymbol{w}, \alpha, \beta) = \frac{p(\theta, \boldsymbol{z}, \boldsymbol{w}|\alpha, \beta)}{p(\boldsymbol{w}|\alpha, \beta)}$$

Challenge: $p(\theta, \boldsymbol{z}|\boldsymbol{w}, \alpha, \beta)$ is intractable because of the coupling between θ and β in $p(\boldsymbol{w}|\alpha, \beta)$:

$$p(\boldsymbol{w}|\alpha, \beta) = \frac{\Gamma(\sum_{i=1}^{K} \alpha_i))}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \int \left(\prod_{i=1}^{K} \theta_i^{\alpha_i - 1} \right) \left(\prod_{n=1}^{N} \sum_{i=1}^{K} \prod_{j=1}^{V} (\theta_i \beta_{ij})^{w_n^j} \right) d\theta$$

Solution: using approximate inference algorithms (**variational inference**).

Variational inference (VI)

Given a distribution $p(z|x)$:

- The main idea is to pick a (variational) distribution $q(z)$ that is "similar" to p.
- A popular method to handle difficult distributions.
- A faster alternative to MCMC.

The choice of q:

- The most important characteristic of q is simplicity: it's easy to work on.
- A popular choice of q: $q(z_1, z_2, \ldots, z_m) = \prod_j q(z_j)$. This is know as the *mean field approximation*.
- The variational distribution is estimated by (kind of) minimizing the KL divergence between p and q.

We have:

$$\begin{aligned}
\log p(\boldsymbol{w}|\alpha,\beta) &= \log \int \sum_{z} p(\theta, \boldsymbol{z}, \boldsymbol{w}|\alpha,\beta) d\theta \\
&= \log \int \sum_{z} \frac{p(\theta, \boldsymbol{z}, \boldsymbol{w}|\alpha,\beta) q(\theta, \boldsymbol{z}|\gamma,\phi)}{q(\theta, \boldsymbol{z}|\gamma,\phi)} d\theta \\
&\geq \int \sum_{z} q(\theta, \boldsymbol{z}|\gamma,\phi) \log p(\theta, \boldsymbol{z}, \boldsymbol{w}|\alpha,\beta) d\theta - \int \sum_{z} q(\theta, \boldsymbol{z}|\gamma,\phi) \log q(\theta, \boldsymbol{z}|\gamma,\phi) d\theta \\
&= E_q[\log p(\theta, \boldsymbol{z}, \boldsymbol{w}|\alpha,\beta)] - E_q[\log q(\theta, \boldsymbol{z}|\gamma,\phi)]
\end{aligned}$$

The inequality is from Jensen's inequality with expectation on $q(\theta, \boldsymbol{z}|\gamma,\phi)$.

This equation holds for any choice of $q(\theta, \boldsymbol{z}|\gamma,\phi)$.

Let RHS

$$L(\gamma, \phi; \alpha, \beta) = E_q[\log p(\theta, \boldsymbol{z}, \boldsymbol{w}|\alpha, \beta)] - E_q[\log q(\theta, \boldsymbol{z}|\gamma, \phi)]$$

Let $\mathscr{D}(.||.)$ be the KL divergence

We have (after some derivation):

$$\log p(\boldsymbol{w}|\alpha, \beta) = L(\gamma, \phi; \alpha, \beta) + \mathscr{D}(q(\theta, \boldsymbol{z}|\gamma, \phi)||p(\theta, \boldsymbol{z}|\boldsymbol{w}, \alpha, \beta))$$

This suggests that minimizing the KL divergence (between the variational posterior $q(\theta, \boldsymbol{z}|\gamma, \phi)$ and true posterior $p(\theta, \boldsymbol{z}|\boldsymbol{w}, \alpha, \beta)$) is equivalent to maximizing $L(\gamma, \phi; \alpha, \beta)$, which is known as the **evidence lower bound** (ELBO).

Variational EM

Using $L(\gamma, \phi; \alpha, \beta)$, approximate empirical Bayes estimates for α and β can be found via an alternating variational EM algorithm by iteratively maximizing L w.r.t. γ and ϕ for fixed α and β, and vice versa.

1. **(E-step)** Find optimal variational parameters $\{\gamma_d^*, \phi_d^* : d \in D\}$ for each document.

   $$(\gamma^*, \phi^*) = argmin_{(\gamma,\phi)} \mathscr{D}(q(\theta, \boldsymbol{z}|\gamma, \phi)||p(\theta, \boldsymbol{z}|\boldsymbol{w}, \alpha, \beta))$$

   which is equivalent to maximizing $L(\gamma, \phi; \alpha, \beta)$ with fixed α, β.

2. **(M-step)** Maximize the document marginal log likelihood lower bound $L(\gamma, \phi; \alpha, \beta)$ with respect to α, β given the optimal γ, ϕ found in E-step.

The discussion so far is for a single document. For a corpus of documents $\boldsymbol{D} = \{\boldsymbol{w_1}, \boldsymbol{w_2}, ..., \boldsymbol{w_M}\}$, the marginal log likelihood is: $l(\alpha, \beta) = \sum_{d=1}^{M} \log p(\boldsymbol{w}_d|\alpha, \beta)$. The expansion is trivial and won't be discussed here.

Choice of variational distributions

Up till now $q(\theta, \boldsymbol{z}|\gamma, \phi)$ can be any distribution. We choose a simple model for variational posterior:

$$q(\theta, \boldsymbol{z}|\gamma, \phi) = q(\theta|\gamma) \prod_{n=1}^{N} q(z_n|\phi_n)$$

γ: Dirichlet parameter of length K
$\phi_1, ..., \phi_n$: multinomial parameters of length V

With $q(\theta, \boldsymbol{z}|\gamma, \phi)$, we now proceed with the parameter estimation. Recall that our goal is to iteratively maximize $L(\gamma, \phi; \alpha, \beta)$ with fixed α, β (E-step) and fixed γ, ϕ (M-step). We start by expanding $L(\gamma, \phi; \alpha, \beta)$:

$$\begin{aligned} L(\gamma, \phi; \alpha, \beta) = E_q[\log p(\theta|\alpha)] + E_q[\log p(\boldsymbol{z}|\theta)] + E_q[\log p(\boldsymbol{w}|\boldsymbol{z}, \beta)] \\ - E_q[\log q(\theta)] - E_q[\log q(\boldsymbol{z})] \end{aligned}$$

γ and ϕ

By computing the derivatives and setting them to zero, we obtain the following pair of update equations (details in Append A.3 from Blei, 2003):

$$\phi_{ni} \propto \beta_{iw_n} exp\{E_q[log(\theta_i)|\gamma]\}$$

$$\gamma_i = \alpha_i + \sum_{n=1}^{N} \phi_{ni}$$

They have intuitive interpretation:

- The Dirichlet update is posterior Dirichlet given expected observations taken under variational distribution, $E[z_n|\phi_n]$
- The multinomial update is akin to using Bayes' theorem, $p(z_n|w_n) \propto p(w_n|z_n)p(z_n)$, where $p(z_n)$ is approximated by the exponential of expected value of its log under variational distribution

α and β

Take $L(\gamma, \phi; \alpha, \beta)$ w.r.t. α_i gives:

$$\frac{\partial L}{\partial \alpha_i} = M(\Psi(\sum_{j=1}^{K} \alpha_j) - \Psi(\alpha_i)) + \sum_{d=1}^{M}(\Psi(\gamma_{di}) - \Psi(\sum_{j=1}^{K} \gamma_{dj}))$$

This derivative depends on α_j, where $j \neq i$. Therefore one can use iterative method to find the maximal α (e.x. Newton-Raphson).

Take $L(\gamma, \phi; \alpha, \beta)$ w.r.t. β, the update of β can be written out analytically:

$$\beta_{ij} \propto \sum_{d=1}^{M} \sum_{n=1}^{N_d} \phi^*_{dni} w^j_{dn}$$

Dimensionality reduction

- Document classification: the choice of feature is challenging.
- Using all words as features in not ideal (noisy).
- From LDA, the estimated topic allocation in each document, i.e., the posterior Dirichlet parameters $\gamma(\boldsymbol{w})$, can be used as feature. This reduces the dimension from N to K.
- The LDA features can be used as predictors for off-the-shelf machine learning tools.

Example

"Arts"	"Budgets"	"Children"	"Education"
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

The William Randolph Hearst Foundation will give $1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. "Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services," Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center's share will be $200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive $400,000 each. The Juilliard School, where music and the performing arts are taught, will get $250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual $100,000 donation, too.

100-topic LDA model on 16,000 documents from TREC AP corpus.

- Top figure: some top-ranked words for a few topics.
- Bottom figure: topic assignment for words in a new testing document.

LDA in R

The *topicmodels* package provide function (*LDA*) to implement LDA in R.

```
> library(topicmodels)
> data("AssociatedPress")

> AssociatedPress
<<DocumentTermMatrix (documents: 2246, terms: 10473)>>
Non-/sparse entries: 302031/23220327
Sparsity           : 99%
Maximal term length: 18
Weighting          : term frequency (tf)

# set a seed so that the output of the model is predictable
> ap_lda <- LDA(AssociatedPress, k = 2, control = list(seed = 1234))

> ap_lda
A LDA_VEM topic model with 2 topics.
```

```
> posterior(ap_lda, AssociatedPress[25:30,])$topics
                1          2
[1,] 0.8627618111 0.13723819
[2,] 0.5162086590 0.48379134
[3,] 0.0009150456 0.99908495
[4,] 0.7011852917 0.29881471
[5,] 0.0062583703 0.99374163
[6,] 0.9582044216 0.04179558
```

LDA reading

- Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. **Journal of machine Learning research**, 3(Jan), 993-1022.