CS 181 Spring 2024 Section 7 Solutions
Principal Component Analysis and Topic Models

1 Principal Component Analysis

1.1 Motivation

Textbook Chapter 7.
In many supervised learning problems, we try to find rich features that increase the expressivity of our model. In practice, this often involves using basis functions to transform the model input into a higher dimensional space (eg. given data x, using x and x^2 as features, or using features learned by a neural network). However, sometimes we want to *reduce* the dimensionality of our data.

Exercise 1. *Why would we want to reduce the dimensionality of our data? Can you think of example cases?*

Solution. There can be several reasons:

- Fewer features are easier to interpret: we might want to know why our model outputs a certain diagnosis, and only some of the patient record details will be relevant.
- Models with fewer features are easier to handle computationally.
- Our data might be arbitrarily high-dimensional because of noise, so we would like to access the lower-dimensional signal from the data.

$\square$

When working with high-dimensional data, it is likely that not every feature will give us completely new information, for example, multiple features may be correlated. The idea that we might be able to reduce the dimensionality of our data by eliminating redundant information is a powerful one, and is the essence of *Principal Component Analysis (PCA).*

Recall that our data $\mathbf{X}$ is an N x D dimensional matrix were each row corresponds to a datapoint and each column corresponds to a feature. The goal of PCA is to re-express $\mathbf{X}$ in terms of a new basis such that every column of this transformed matrix now gives us completely new information (ie. the features are *linearly independent*). The columns of the corresponding change-of-basis matrix are called *principal components* and are constructed in such a way that each principal component accounts for more of the variance in our original data than the next. After re-expressing $\mathbf{X}$ in terms of the principal components, we can make a decision about how many columns of the new matrix we want to keep depending on how much of the original variation we want to preserve and what our usecase is. For example, if we wanted to plot the resulting data we might only keep the first two principal components.

1.2 Finding the lower dimensional representation

Let's say we want to compress our D-dimensional data into K dimensions ($K < D$) ie. we want to find K D-dimensional basis vectors $\mathbf{u}_1, ..., \mathbf{u}_K$ such that we can represent $\mathbf{x}_n$ as a linear combination of them:

$$\mathbf{x}_n \approx z_{n,1}\mathbf{u}_1 + ... + z_{n,k}\mathbf{u}_k = \mathbf{U}\mathbf{z}_n$$

where $z_{n,1}, ..., z_{n,k}$ are scalars and $\mathbf{U}$ is a matrix of all the basis vectors. We define the *reconstruction loss* to be the distance from the approximation of each $\mathbf{x}_n$ using the new basis to $\mathbf{x}_n$ itself:

$$\mathcal{L}(\{\mathbf{z}_n\}, \mathbf{U}) = \frac{1}{N}\sum_{n=1}^{N} ||\mathbf{x}_n - \mathbf{U}\mathbf{z}_n||_2^2$$

The solution to the above loss is not unique! To find a unique solution, let's impose the constraint that $\mathbf{U}$ must be *orthonormal*, meaning that for any distinct rows k, k' in $\mathbf{U}$, $\mathbf{u}_k \cdot \mathbf{u}_k = 1$ and $\mathbf{u}_k \cdot \mathbf{u}_{k'} = 0$. Using orthonormal basis vectors yields the nice property that $\mathbf{u}_k^T\mathbf{x}_n = z_{n,k}$.

Let's subtract the mean of our data $\bar{\mathbf{x}}$ from each $\mathbf{x}_n$ so that our bases capture variation from the mean. Our loss function is now

$$\mathcal{L}(\{\mathbf{z}_n\}, \mathbf{U}) = \frac{1}{N}\sum_{n=1}^{N} ||(\mathbf{x}_n - \bar{\mathbf{x}}) - \mathbf{U}\mathbf{z}_n||_2^2 \text{ s.t. } \mathbf{U} \text{ is orthonormal}$$

Note that if $K = D$, then we could perfectly reconstruct $\mathbf{x}_n$ since we'd be able to preserve all the features. Thus, $\mathbf{x}_n - \bar{\mathbf{x}} = \sum_{k=1}^{D} z_{n,k}\mathbf{u}_k$. Thus, we can simplify our loss as follows:

$$\begin{aligned}
\mathcal{L}(\{\mathbf{z}_n\}, \mathbf{U}) &= \frac{1}{N}\sum_{n=1}^{N} ||\sum_{k=1}^{D} z_{n,k}\mathbf{u}_k - \sum_{k=1}^{K} z_{n,k}\mathbf{u}_k||_2^2 \\
&= \frac{1}{N}\sum_{n=1}^{N} ||\sum_{k=K+1}^{D} z_{n,k}\mathbf{u}_k||_2^2 \\
&= \frac{1}{N}\sum_{k=K+1}^{D}\sum_{n=1}^{N} \mathbf{u}_k^T(\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^T\mathbf{u}_k \\
&= \sum_{k=K+1}^{D} \mathbf{u}_k^T \left[\frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^T\right]\mathbf{u}_k
\end{aligned}$$

The expression in the square brackets is the *empirical covariance matrix of* $\mathbf{X}$! Finding the directions of greatest variation correspond to finding the *eigenvectors* of this empirical covariance matrix. Eigenvectors with a higher eigenvalue capture more variance in the data than those with a lower eigenvalue.

To summarize, to perform PCA:

1. Center the data by subtracting the mean of each feature from each data point. Steps 2 - 5 will be performed on the centered data $\mathbf{X}$: $(N \times D)$.
2. Calculate the empirical covariance matrix:

$$\mathbf{S} = \frac{1}{N}(\sum_{n=1}^{N} \mathbf{x}_n \mathbf{x}_n^\top) = \frac{1}{N}\mathbf{X}^\top \mathbf{X}$$

3. Decide how many dimensions K out of the original D we want to keep in the final representation.
4. Find the K largest eigenvalues of $\mathbf{S}$. The $K \times 1$ eigenvectors $(\mathbf{u}_1, \dots, \mathbf{u}_K)$ corresponding to these eigenvalues will be our lower-dimensional basis.
5. Reduce the dimensionality of a data point $\mathbf{x}$ by projecting it onto this basis yielding a new reconstructed vector $\mathbf{z}$:

$$\mathbf{z} = \mathbf{U}^T \mathbf{x}$$

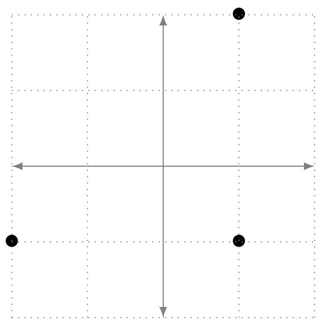
Exercise 2. *You are given the following data set:*

$$\mathbf{x}_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \quad \mathbf{x}_2 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad \mathbf{x}_3 = \begin{bmatrix} -2 \\ -1 \end{bmatrix}$$

You would like to use PCA to find a 1-dimensional representation of the data.

1. *Plot the data set.*
2. *Compute the empirical covariance matrix* $\mathbf{S}$.
3. *You find that* $\mathbf{S}$ *has eigenvector* $[-1\ 1]^\top$ *with eigenvalue* 1 *and eigenvector* $[1\ 1]^\top$ *with eigenvalue* 3*. What is the (normalized) basis vector* $\mathbf{u}_1$ *of your 1-dimensional representation? Add the basis vector* $\mathbf{u}_1$ *to your plot.*
4. *Compute the coefficients* z_1, z_2, z_3*. Add the lower-dimensional representations* $z_1\mathbf{u}_1, z_2\mathbf{u}_1, z_3\mathbf{u}_1$ *to your plot. Based on your plot, what is the relationship between* $z_i\mathbf{u}_1$ *and* $\mathbf{x}_i$ *with respect to the new basis?*
5. *Based on your plot, what would happen if you chose the unused eigenvector to be your basis vector?*

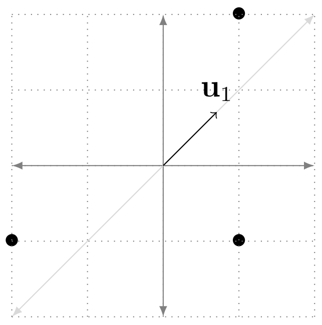
Solution. 1.



2.

$$\mathbf{S} = \frac{1}{3}\mathbf{X}^\top \mathbf{X} = \frac{1}{3}\begin{bmatrix} 1 & -1 \\ 1 & 2 \\ -2 & -1 \end{bmatrix}^\top \begin{bmatrix} 1 & -1 \\ 1 & 2 \\ -2 & -1 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

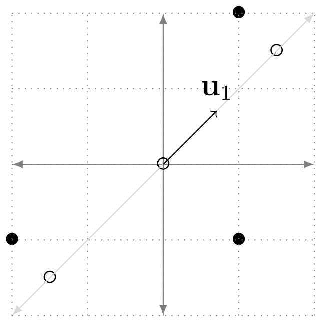
3. We select the eigenvectors with the largest eigenvalues for our basis, so our basis will contain a scalar multiple of $[1\ 1]^\top$. Normalizing $[1\ 1]^\top$ gives us that $\mathbf{u}_1 = [\frac{\sqrt{2}}{2}\ \frac{\sqrt{2}}{2}]^\top$.



4.

$$z_1 = \mathbf{x}_1^\top \mathbf{u}_1 = 0, \quad z_2 = \mathbf{x}_2^\top \mathbf{u}_1 = \frac{3\sqrt{2}}{2}, \quad z_3 = \mathbf{x}_3^\top \mathbf{u}_1 = -\frac{3\sqrt{2}}{2}$$

The open circles in the plot represent the lower-dimensional representation:



$z_i\mathbf{u}_1$ is the projection of $\mathbf{x}_i$ onto the basis vector.

5. If we chose $[-1\ 1]^\top$ to be the basis of our new representation, then the representation would capture less of the variance in the data. For example, $\mathbf{x}_2$ and $\mathbf{x}_3$ would be represented by the same point.

$\square$

Exercise 3. *Suppose that our data are centered (i.e., have sample mean 0). Recall that in lecture, we showed that, when optimizing over (semi)-orthogonal matrices* $\mathbf{U} \in \mathbb{R}^{m \times d}$ *(i.e., where* $\mathbf{U}^T\mathbf{U} = I$*) to minimize the reconstruction loss,*

$$\mathcal{L}(\mathbf{U}) = \frac{1}{n}\sum_{i=1}^{n} ||\mathbf{x_i} - \mathbf{U}\mathbf{U}^T\mathbf{x_i}||_2^2,$$

we found that $\mathbf{U}_d$*, the matrix whose first* d *columns are (in order) the top* d *eigenvectors of the empirical covariance matrix* $\mathbf{\Sigma} = \frac{1}{n}\mathbf{X}^T\mathbf{X}$*, will achieve the minimum (i.e.,* $\mathbf{z} = \mathbf{U}^T\mathbf{x}$ *is the projection of* $\mathbf{x}$ *into* d *dimensions, and* $\mathbf{Uz}$ *is its reconstruction in* $\mathbb{R}^m$*). In class, we showed this for case when* $d = m - 1$ *by using Lagrange multipliers. Show this, in general, for* d*.*

Hint: you may use the following theorem: [Courant-Fischer] Let $\mathbf{A}$ *be a symmetric* $n \times n$ *matrix with eigenvalues* $\lambda_1 \leq \cdots \leq \lambda_n$ *and corresponding eigenvectors* $\mathbf{v_1}, \ldots, \mathbf{v_n}$*. Then*

$$\lambda_1 = \min_{||x||=1} \mathbf{x}^T\mathbf{A}\mathbf{x}$$

$$\lambda_2 = \min_{||\mathbf{x}||=1, \mathbf{x}\perp\mathbf{v}_1} \mathbf{x}^T\mathbf{A}\mathbf{x}$$

$$\vdots$$

$$\lambda_i = \min_{||\mathbf{x}||=1, \mathbf{x}\perp\mathbf{v}_1, \mathbf{x}\perp\mathbf{v}_2, \ldots, \mathbf{x}\perp\mathbf{v}_{i-1}} \mathbf{x}^T\mathbf{A}\mathbf{x}$$

$$\vdots$$

$$\lambda_n = \min_{||\mathbf{x}||=1, \mathbf{x}\perp\mathbf{v}_1, \mathbf{x}\perp\mathbf{v}_2, \ldots, \mathbf{x}\perp\mathbf{v}_{n-1}} \mathbf{x}^T\mathbf{A}\mathbf{x}.$$

Solution. We proceed just as in class. Let $\tilde{\mathbf{U}} \in \mathbb{R}^{m \times m}$ be an orthogonal matrix, and we will let $\mathbf{U}$ be the matrix comprising $\tilde{\mathbf{U}}$'s first d columns. Letting $\tilde{\mathbf{U}}^{(k)}$ denote the kth column of $\tilde{\mathbf{U}}$, the orthogonality of $\tilde{\mathbf{U}}$ implies that the $\tilde{\mathbf{U}}^{(k)}$ form a basis in $\mathbb{R}^m$ so that, for each $\mathbf{x}_i$, we can find $\mathbf{z}_i \in \mathbb{R}^m$ such that

$$\mathbf{x}_i = \sum_{k=1}^{m} \mathbf{z}_{i,k}\tilde{\mathbf{U}}^{(k)}.$$

The orthogonality of $\tilde{\mathbf{U}}$ implies that $\tilde{\mathbf{U}}^T\mathbf{x}_i = (\mathbf{z}_{i,1}, \ldots, \mathbf{z}_{i,m})$ and $\mathbf{U}^T\mathbf{x}_i = (\mathbf{z}_{i,1}, \ldots, \mathbf{z}_{i,d})$. Hence, we can write the reconstruction loss as

$$\mathcal{L}(\mathbf{U}) = \frac{1}{n}\sum_{i=1}^{n} ||\sum_{k=1}^{m} \mathbf{z}_{i,k}\tilde{\mathbf{U}}^{(k)} - \sum_{k=1}^{d} \tilde{\mathbf{U}}^{(k)}\mathbf{z}_{i,k}||_2^2 = \frac{1}{n}\sum_{i=1}^{n} ||\sum_{k=1}^{m} \mathbf{z}_{i,k}\tilde{\mathbf{U}}^{(k)} - \sum_{k=1}^{d} \mathbf{z}_{i,k}\tilde{\mathbf{U}}^{(k)}||_2^2$$

$$= \frac{1}{n}\sum_{i=1}^{n} || \sum_{k=d+1}^{m} \mathbf{z}_{i,k}\tilde{\mathbf{U}}^{(k)}||_2^2 = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=d+1}^{m} \mathbf{z}_{i,k}^2 = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=d+1}^{m} \left(\tilde{\mathbf{U}}^{(k)T}\mathbf{x}_i\right)\left(\mathbf{x}_i^T\tilde{\mathbf{U}}^{(k)}\right)$$

$$= \sum_{k=d+1}^{m} \tilde{\mathbf{U}}^{(k)T}\left(\frac{1}{n}\mathbf{X}^T\mathbf{X}\right)\tilde{\mathbf{U}}^{(k)} = \sum_{k=d+1}^{m} \tilde{\mathbf{U}}^{(k)T}\mathbf{\Sigma}\tilde{\mathbf{U}}^{(k)},$$

where $\mathbf{\Sigma}$ denotes the empirical covariance matrix.

Let $\mathbf{u}_1, \ldots, \mathbf{u}_m$ denote the (unit length) eigenvectors of $\mathbf{\Sigma}$ (ordered by increasing corresponding eigenvalue). When $d + 1 = m$, there's only one element in the sum, and since we know that $\tilde{\mathbf{U}}$ is orthogonal, we must have that $||\tilde{\mathbf{U}}^{(k)}||_2 = 1$ and thus Courant-Fischer says that $\tilde{\mathbf{U}}^{(k)} = \mathbf{u}_1$ will minimize. When $d + 1 = m - 1$, there will be two elements in the sum:

$$\tilde{\mathbf{U}}^{(m-1)T}\mathbf{\Sigma}\tilde{\mathbf{U}}^{(m-1)} + \tilde{\mathbf{U}}^{(m)T}\Sigma\tilde{\mathbf{U}}^{(m)}.$$

Consider solving this optimization problem by first picking $\tilde{\mathbf{U}}^{(m)}$ and then $\tilde{\mathbf{U}}^{(m-1)}$. Now, we either choose $\tilde{\mathbf{U}}^{(m)}$ to be $\mathbf{u}_1$ or we do not. If we choose $\tilde{\mathbf{U}}^{(m)} = \mathbf{u}_1$, then Courant-Fischer implies that picking $\tilde{\mathbf{U}}^{(m-1)} = \mathbf{u}_2$ will minimize. If we do not choose $\tilde{\mathbf{U}}^{(m)}$ to be $\mathbf{u}_1$, then Courant-Fischer implies that picking $\tilde{\mathbf{U}}^{(m-1)} = \mathbf{u}_1$ will minimize and that, if we did not pick $\tilde{\mathbf{U}}^{(m)}$ to be $\mathbf{u}_2$, we could've achieved an even lower value more by picking it to be so. Thus, in either case, we will take $\tilde{\mathbf{U}}^{(m)}$ and $\tilde{\mathbf{U}}^{(m-1)}$ to be $\mathbf{u}_1$ and $\mathbf{u}_2$.

Proceeding iteratively like this shows that we will chose $\tilde{\mathbf{U}}^{(d+1)}, \ldots, \tilde{\mathbf{U}}^{(m)}$ to be $\mathbf{u}_1, \ldots, \mathbf{u}_{m-d}$. Now, note that these are the columns we *exclude* from the matrix $\tilde{\mathbf{U}}$ when we construct $\mathbf{U}$ (which will be the first d column of $\tilde{\mathbf{U}}$). Since $\mathbf{\Sigma}$ is symmetric, the eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_n$ are orthogonal (or can be chosen to be in the case of repeated eigenvalues), taking $\mathbf{U}$'s columns to be $\mathbf{u}_{m-d+1}, \ldots, \mathbf{u}_m$ (i.e., the top d eigenvectors) indeed will minimize the reconstruction loss, as desired. □

1.3 Choosing the Optimal Number of Principal Components

The 'right' number of principal components to use depends on our goals. For example, if we simply wish to visualize our data, then we would project onto a 2D or 3D space. Therefore, we would choose the first 2 or 3 principal components, and project our original data onto the subspace defined by those vectors.

One way to do this is to consider how much variance we wish to preserve in our data. The eigenvalues λ_d represent the amount of variance in the data explained by each principal component. The total variance of the original dataset is the sum of all eigenvalues.

When we retain a subset of D' principal components, we are effectively retaining the corresponding eigenvalues $\lambda_{d'}$ associated with those components.

The retained variance is then calculated as the ratio of the sum of retained eigenvalues to the total sum of all eigenvalues:

$$\text{retained variance} = \frac{\sum_{d'=1}^{D'} \lambda_{d'}}{\sum_{d=1}^{D} \lambda_d} \tag{1}$$

Exercise 4. *Let's consider a simple example where we have calculated the eigenvalues of the covariance matrix Σ after performing PCA on a dataset. Here are the eigenvalues we obtained:*

$$\lambda_1 = 10, \lambda_2 = 6, \lambda_3 = 3, \lambda_4 = 1$$

Now, calculate the retained variance for $D' = 1, 2, 3,$ and 4.

Solution. For $D' = 1$:

$$\text{retained variance} = \frac{\lambda_1}{\sum_{d=1}^{4} \lambda_d} = \frac{10}{10+6+3+1} = \frac{10}{20} = 0.5$$

For $D' = 2$:

$$\text{retained variance} = \frac{\lambda_1 + \lambda_2}{\sum_{d=1}^{4} \lambda_d} = \frac{10+6}{10+6+3+1} = \frac{16}{20} = 0.8$$

For $D' = 3$:

$$\text{retained variance} = \frac{\lambda_1 + \lambda_2}{\sum_{d=1}^{4} \lambda_d} = \frac{10+6+3}{10+6+3+1} = \frac{19}{20} = 0.95$$

For $D' = 4$:

$$\text{retained variance} = \frac{\lambda_1 + \lambda_2}{\sum_{d=1}^{4} \lambda_d} = \frac{10+6+3+1}{10+6+3+1} = \frac{20}{20} = 1$$

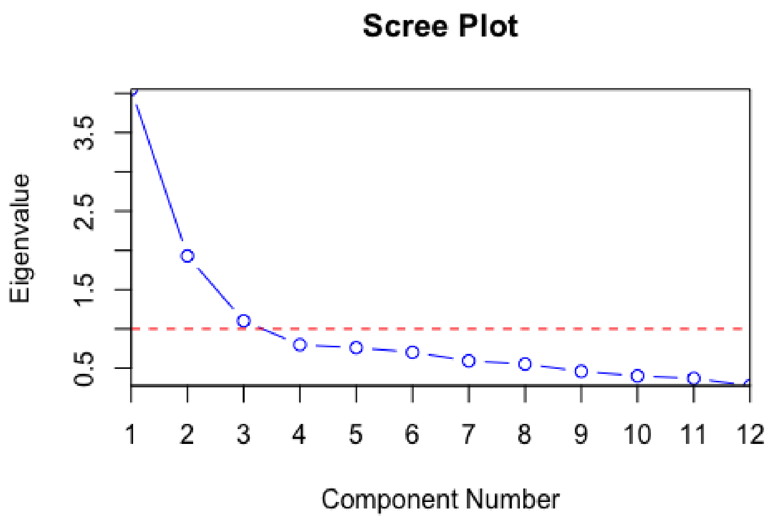
So, we observe that increasing the number of principal components leads to an increased proportion of the variance remaining. □

1.3.1 Scree Plot

We could also examine the scree plot. A scree plot is a graphical tool used in principal component analysis (PCA) to help determine the optimal number of principal components to retain for dimensionality reduction. It displays the eigenvalues of the principal components in decreasing order along the y-axis, while the x-axis represents the number of principal components.

The rationale behind using the scree plot is to retain only the principal components that capture the most significant variation in the data. Principal components with high eigenvalues contribute more to the overall variance of the dataset, while those with lower eigenvalues capture less variance and may represent noise or irrelevant information.

By examining the scree plot, analysts can visually identify the point at which the eigenvalues start to level off, suggesting that additional principal components contribute little to the overall variance. This point corresponds to the optimal number of principal components to retain for dimensionality reduction while preserving the most important information in the dataset.



Scree Plot from Wikipedia.

2 Topic Models

Textbook Section 9.6.

Topic modeling is used for discovering latent topics or themes in large collections of documents. The goal is the underlying structure in a corpus of text documents and categorizing them into different groups based on their content. Topic modeling is a type of unsupervised learning, where the algorithm tries to discover patterns and structures in the data without prior knowledge of the labels or categories. Some of the popular applications of topic modeling include document clustering, text classification, and information retrieval.

Topic modeling is similar to other latent variable models, such as Gaussian mixture models (GMMs). Like the mixture models that we study in this course, we will describe topic models generatively: one in which we assume our corpus is generated by some process. Then, we will seek to use an optimization method, such as EM, to train the parameters of our model.

2.1 Probabilistic Latent Semantic Analysis (pLSA)

Consider a collection of documents, where each document is a mixture of various topics. pLSA is a generative model that assumes each word in a document is generated by sampling from a topic, and the topic is sampled from a per-document distribution. As a generative model, we want to model the joint probability of words and documents, $p(w, d)$. The generative process for a document in pLSA is as follows:

1. Initialize conditional probabilities $p(z \mid d)$ and $p(w \mid z)$ that represent the per-document distribution for topics and per-topic distribution for words.

2. For each document d, choose a topic z with probability $p(z \mid d)$.

3. For the chosen topic z, pick a word w with probability $p(w \mid z)$.

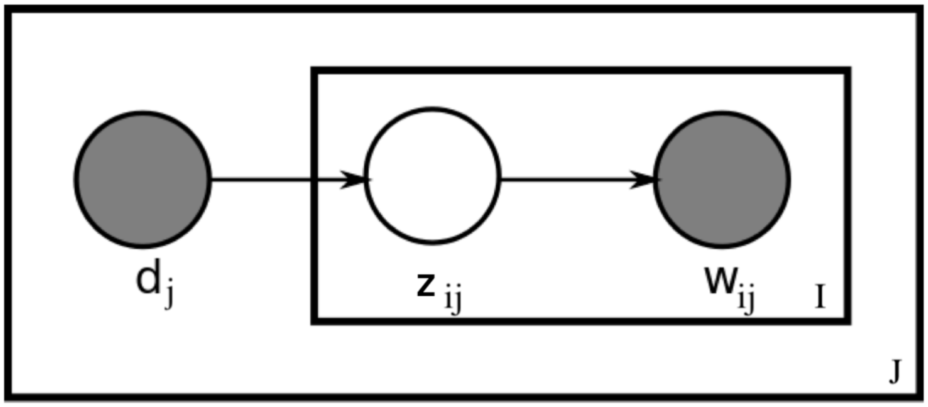
The goal of pLSA is to learn the conditional probabilities $p(w \mid z)$ and $p(z \mid d)$. In pLSA, we do not impose any priors on these probabilities beforehand. Given the observed data $p(w \mid d)$, we can train a latent model to estimate the conditional probabilities $p(w \mid z)$ and $p(z \mid d)$ based on the training data. To do this, we use the Expectation-Maximization algorithm again to maximize the likelihood of the observed data with respect to the latent variables, or the topics. The EM algorithm consists of two steps:

- Expectation (E) step: Compute the posterior probabilities of the topic assignments $p(z \mid w, d)$ using the current estimates of $p(w \mid z)$ and $p(z \mid d)$. To do so, we can use the observed data likelihood function:

$$p(w, d) = \sum_z p(w \mid z) p(z \mid d) p(d)$$

- Maximization (M) step: Update the estimates of $p(w \mid z)$ and $p(z \mid d)$ based on the posterior probabilities computed in the E step.

Note that we only know d_j and w_{ij} but define z_{ij} in order to train the conditional probabilities in such a way that makes intuitive sense in our generative model. The plate diagram for pLSA is below:



The structure behind the observed data:

- start with J number of documents in my corpus; for each document, there is a mixture of topics; for each topic, there is a distribution over words
- for the j-th document, the I number of words is generated as follows:
  - sample a single topic $\mathbf{z}_{ij}$ from the mixture of topics for the j-th document

– sample a single word from the distribution over words representing this topic

In this relatively limited model, the most notable weakness is overfitting. As the number of parameters in the model grows linearly with the number of documents, pLSA is prone to overfitting. The model does not have a prior imposed on the per-document or per-topic distributions, so all of the parameter learning is derived from the training data. In document text, this is particularly bad because the true complexity of possible document features is far more complex than just the documents in the sample corpus.

2.2 Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) is an extension of pLSA that addresses some of its limitations, mainly overfitting and lack of a generative model for new documents. Like pLSA, LDA is a generative probabilistic model for topic modeling that assumes each document is a mixture of topics, and each topic is a distribution over words. However, LDA introduces a Dirichlet prior on the per-document topic distributions and per-topic word distributions, leading to better generalization and the ability to infer topic distributions for new documents. Specifically, we use fixed parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ as an extra "layer" of sampling.

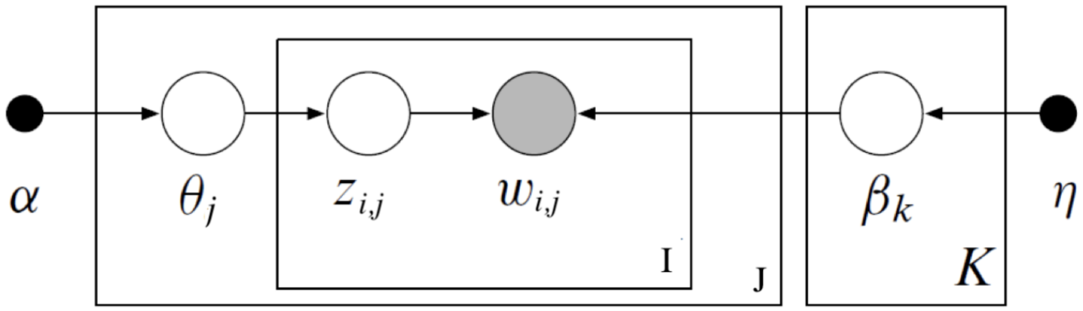
We begin by describing the generative process by which a document i is generated. For topic modeling, similar to K-means, we have to begin by picking the number of topics, K, to look for. We define a topic $\boldsymbol{\phi}_k$ to be a distribution over the words, so $\boldsymbol{\phi}_k \in [0,1]^{|\mathcal{W}|}$, where $\mathcal{W}$ is the set of words. For each document, we have a document-topic distribution $\boldsymbol{\theta}_m \in [0,1]^K$. These are the parameters to estimate in LDA.

We now describe the data generation process:

1. Let $\boldsymbol{\alpha} \in \mathbb{R}_+^K$ and $\boldsymbol{\beta} \in \mathbb{R}_+^{|\mathcal{W}|}$.
2. For each document $m = 1, \ldots, M$, sample a mixture over topics: $\boldsymbol{\theta}_n \sim Dir(\boldsymbol{\alpha})$.
3. For each topic $k = 1, \ldots, K$, sample a mixture over words in that topic: $\boldsymbol{\phi}_k \sim Dir(\boldsymbol{\beta})$.
4. For each word $w_{m,n}$ (for $m = 1, \ldots, M$ and $n = 1, \ldots, N$, the length of the document), first sample the topic $z_{m,n} \sim Cat(\boldsymbol{\theta}_m)$, then sample the word $w_{m,n} \sim Cat(\boldsymbol{\phi}_{z_{m,n}})$.

For some intuition, the Dirichlet distribution takes in a k-sized vector of values and outputs a probability distribution across k categories. Roughly speaking, the Dirichlet parameter is a vector of positive real numbers; the larger the value, the more likely that corresponding component of the sampled vector will have a higher value. At a high level, then, a topic model is a mixture over mixtures: within a single document, $\boldsymbol{\theta}_m$ specifies a distribution over topics in that document, and for each topic, k, in that document, $\boldsymbol{\phi}_k$ specifies a distribution over words.

This process is again summarized in the following plate diagram (where η in the diagram represents the Dirichlet parameter for words per topic):



Comparing this plate diagram to that of pLSA, we can observe that pLSA is just if we consider the plates across I and J, without the fixed inputs of $\boldsymbol{\alpha}$ and η. Like pLSA, we can use a version of EM to optimize the model parameters, but requires an approximation of the posterior distributions (variational inference).

The structure behind the observed data:

- w represents single words
- z represents single topics

Now, instead of supposing that each document has a fixed but unknown mixture of topics, $p(z|d)$, we suppose that each document was randomly assigned a mixture of topics (this is our prior on $p(z|d)$):

- θ represents the per document topic mixture

Since θ is a mixture (the mixture components must sum to 1), it make sense to model θ as a Dirichlet random variable with hyperparameters α.

Instead of supposing that each topic has a fixed but unknown distribution over words, $p(w|z)$, we suppose that each topic was randomly assigned a distribution over words (this is our prior on $p(w|z)$:

- β represents the per topic word distribution

Since θ is a distribution over words (the distribution must sum to 1), it make sense to model θ as a Dirichlet random variable with hyperparameters η.

The main takeaway from LDA is that the introduction of Dirichlet priors for the per-document topic distributions $\boldsymbol{\theta}$ and per-topic word distributions $\boldsymbol{\phi}$ acts as a form of regularization. By imposing these priors, the model incorporates some prior knowledge or assumptions about the distributions, which helps guide the learning process. This regularization effect prevents the model from relying too heavily on the training data, leading to better generalization and robustness against overfitting. As a result, LDA can more effectively estimate the underlying topic structure in the data and produce topic distributions for new, unseen documents. This makes LDA a more robust and widely applicable topic modeling method compared to pLSA, which lacks the regularization provided by the Dirichlet priors.

Exercise 5. *Describe, at a high level, how the EM algorithm for topic models can be viewed as alternating between two optimizations. What are these two optimizations?*

Solution. In EM, we alternate between the E step and M step. In the E step, we write the posterior distribution, q of the latent variable given the data and our current estimates of the parameters, and in the M-step we try to maximize (over our parameters) the expected complete-data log likelihood under q. The two steps alternate between first (E step) finding a lower bound to the true log likelihood (i.e., by finding q and noting that the complete-data log likelihood under q minus q's entropy over the dataset (which is constant w.r.t the parameters) lower bounds the true log likelihood) and then (M step) choosing the parameters which maximize this lower bound.

In the case of topic models, we do the exact same thing. Specifically, for the mth document, the posterior distributions $q_{m,n}$ (which we take the expectation of the complete-data log likelihood with respect to in the M step), will be the posterior distributions of topics given words and our current estimates of the parameters. We calculate these $q_{m,n}$'s in the E step as an optimization towards establishing a lower bound on the true log likelihood, and then, in the M step, we optimize over the parameters $\boldsymbol{\theta}_m, \boldsymbol{\phi}_k$ to maximize this lower bound. Specifically, in the M step, we find parameters that maximize the expected complete-data log likelihood under q_m: $\mathbb{E}_{q_m}[\log p(\mathbf{W}, \mathbf{Z})]$, where $\mathbf{W}$ and $\mathbf{Z}$ denote words and topics, respectively (note that the expectation is taken over the unknown topics, $\mathbf{Z}$).

$\square$