CS 1810 Spring 2025 Section 7: PCA and Topic Models

1 Principal Component Analysis

1.1 Motivation

In many supervised learning problems, we try to find rich features that increase the expressivity of our model, e.g., by using basis functions to transform the model input into a higher dimensional space. However, sometimes we want to perform **dimensionality reduction**.

Why might we want to reduce the dimensionality of our data? 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.

Now, 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. **Principal component analysis (PCA)** is a technique to reduce the dimensionality of our data by re-expressing our original data $\boldsymbol{X}$ in terms of a lower-dimensional basis, where the basis vectors are linear combinations of the original features. These basis vectors are the *principal components*. PCA has two goals when finding principal components:

1. Ensure the components are not redundant at all, i.e. that they are orthogonal.
2. Ensure the components capture the directions of greatest variation in the data.

For intuition as to what this looks like, see Figure 1.

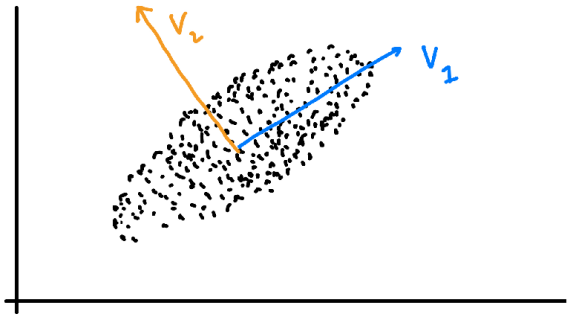


Figure 1: A plot of sample data with two features, x_1 plotted against x_2, along with the principal component directions v_1 and v_2. Note that these are orthogonal. Also, if we consider the projections of all the data points onto v_1, we see there is greater variance in the projections than if we projected onto v_2.

1.2 Mathematical Formulation

Recall that we want to compress our D-dimensional data into $K < D$ dimensions, where we choose K as a hyperparameter. Mathematically, this means we want to find K D-dimensional vectors $\boldsymbol{v}_1, ..., \boldsymbol{v}_K$ such that we can use the lower-dimensional projection $\boldsymbol{X}\boldsymbol{V}_{1:K}$ in place of the raw data $\boldsymbol{X}$. Note that $\boldsymbol{V}_{1:K}$ is $D \times K$ and has $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_K$ as its columns.

PCA imposes that these vectors are orthonormal, meaning they are all unit vectors and mutually orthogonal. Furthermore, PCA crucially assumes that $\boldsymbol{X}$ has been *mean-centered*, meaning that the mean of each column is 0. We will see why this is important soon. Finally, PCA solves for these vectors as such:

1. We want $\boldsymbol{v}_1$ to be the direction of greatest variance in the data. The empirical variance of $\boldsymbol{X}$ in a given direction $\boldsymbol{v}$ is the quantity

   $$\frac{1}{N}(\boldsymbol{X}\boldsymbol{v})^\top \boldsymbol{X}\boldsymbol{v} = \boldsymbol{v}^\top \boldsymbol{S}\boldsymbol{v},$$

   where $\boldsymbol{S} = \frac{1}{N}\boldsymbol{X}^\top \boldsymbol{X}$ is the empirical covariance matrix. Note that $\boldsymbol{X}\boldsymbol{v}$ is the projection of $\boldsymbol{X}$ onto the direction of $\boldsymbol{v}$, and note that there is no mean term in the variance expression since $\boldsymbol{X}$ being mean-centered implies the mean of $\boldsymbol{X}\boldsymbol{v}$ is 0. Hence,

   $$\boldsymbol{v}_1 = \arg\max_{\boldsymbol{v}} \boldsymbol{v}^\top \boldsymbol{S}\boldsymbol{v} \quad \text{s.t. } \boldsymbol{v}^\top \boldsymbol{v} = 1$$

   Using a Langrangian, we see that

   $$\mathcal{L}(\boldsymbol{v}, \lambda) = \boldsymbol{v}^\top \boldsymbol{S}\boldsymbol{v} + \lambda(1 - \boldsymbol{v}^\top \boldsymbol{v}) \Rightarrow \boldsymbol{S}\boldsymbol{v}_1 = \lambda \boldsymbol{v}_1$$

   This shows that $\boldsymbol{v}_1$ is an *eigenvector* of $\boldsymbol{S}$. In fact, it is the eigenvector corresponding with the largest eigenvalue λ_1, since

   $$\boldsymbol{v}_1^\top \boldsymbol{S}\boldsymbol{v}_1 = \lambda \boldsymbol{v}_1^\top \boldsymbol{v}_1 = \lambda,$$

   and $\boldsymbol{v}_1$ maximizes this quantity.

2. For $k = 2, \ldots, K$, we solve

   $$\begin{aligned} \boldsymbol{v}_k &= \arg\max_{\boldsymbol{v}} \boldsymbol{v}^\top \boldsymbol{S}\boldsymbol{v} \\ \text{s.t. } \boldsymbol{v}^\top \boldsymbol{v} &= 1 \\ \boldsymbol{v}^\top \boldsymbol{v}_j &= 0 \quad \forall j < k \end{aligned}$$

   You do not need to know the math here, but it turns out that this results in $\boldsymbol{v}_k$ being the eigenvector of $\boldsymbol{S}$ that corresponds to the k-th largest eigenvalue λ_k, analagous to the case for $\boldsymbol{v}_1$.

Takeaway: Hence, PCA with K components on mean-centered $\boldsymbol{X}$ yields the top K eigenvectors of the *empirical covariance matrix* $\boldsymbol{S}$.

1.3 Alternative Formulations of PCA

We've presented the standard formulation of PCA, which is in line with the initial high-level description we gave of PCA. However, this method can also be formulated in other ways.

PCA as Maximizing Total Variance: In the previous subsection, we formulated PCA as a sequential optimization procedure, where we started off by finding the direction of overall greatest variance, then the direction of greatest variance out of all directions orthogonal to the first one, etc. However, it is not immediately clear that this is equivalent to finding the set of orthonormal directions $\boldsymbol{V}_{1:K}$ that maximize the *total variance* explained by the directions. Note that the total variance is

$$\sum_{k=1}^{K} \boldsymbol{v}_k^\top \boldsymbol{S} \boldsymbol{v}_k = \text{tr}\left(\boldsymbol{V}_{1:K}^\top \boldsymbol{S} \boldsymbol{V}_{1:K}\right)$$

It turns out that there is a linear algebra result which tells us that if we let $\boldsymbol{A}$ be a real symmetric $D \times D$ matrix with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_D$, then

$$\max_{\boldsymbol{V}} \text{tr}\left(\boldsymbol{V}^\top \boldsymbol{A} \boldsymbol{V}\right) = \sum_{k=1}^{K} \lambda_k,$$

where we restrict $\boldsymbol{V}$ to be a $D \times K$ orthogonal matrix. In our case, we know $\boldsymbol{V}_{1:K}$ is orthogonal and that $\boldsymbol{S}$ is symmetric, so it follows that maximum value of the total variance is the sum of the K largest eigenvalues of $\boldsymbol{S}$. This is precisely what the vectors from the previous subsection achieve, as they are the top K eigenvectors of $\boldsymbol{S}$, implying that

$$\sum_{k=1}^{K} \boldsymbol{v}_k^\top \boldsymbol{S} \boldsymbol{v}_k = \sum_{k=1}^{K} \boldsymbol{v}_k^\top \lambda_k \boldsymbol{v}_k = \sum_{k=1}^{K} \lambda_k$$

PCA as Minimizing Reconstruction Loss. We've discussed how we can use our principal component matrix $\boldsymbol{V}_{1:K}$ to project our (mean-centered) data into a lower-dimensional space. For each data point $\boldsymbol{x}^{(n)}$, we can define

$$\boldsymbol{z}^{(n)} = \boldsymbol{V}_{1:K}^\top \boldsymbol{x}^{(n)}$$

as its lower-dimensional representation. If we project this representation back into the original space, we essentially obtain an approximation defined as

$$\hat{\boldsymbol{x}}^{(n)} = \boldsymbol{V}_{1:K} \boldsymbol{z}^{(n)}$$

Now suppose we want to solve for the vectors $\boldsymbol{V}_{1:K}$ that make this approximation as good as possible. We define the *reconstruction loss* as a mean squared error

$$\mathcal{L}\left(\{\boldsymbol{z}^{(n)}\}, \boldsymbol{V}_{1:K}\right) = \frac{1}{N} \sum_{n=1}^{N} ||\boldsymbol{x}^{(n)} - \boldsymbol{V}_{1:K} \boldsymbol{z}^{(n)}||_2^2.$$

In these notes, we will not fully analytically solve the reconstruction loss optimization problem, as it can become technical. We do, however, wish to provide concrete intuitions for the form of principal components. To do so, we express our loss $\mathcal{L}$ in more suggestive ways. In particular,

note that if $K = D$, then we could perfectly reconstruct $\boldsymbol{x}^{(n)}$ since we'd be able to preserve all the features. Thus, $\boldsymbol{x}^{(n)} = \sum_{k=1}^{D} z_k^{(n)} \boldsymbol{v}_k$. Thus, we can simplify our loss as follows:

$$
\begin{aligned}
\mathcal{L}\left(\{\boldsymbol{z}^{(n)}\}, \boldsymbol{V}_{1:K}\right) &= \frac{1}{N}\sum_{n=1}^{N}\left\|\sum_{k=1}^{D} z_k^{(n)}\boldsymbol{v}_k - \sum_{k=1}^{K} z_k^{(n)}\boldsymbol{v}_k\right\|_2^2 \\
&= \frac{1}{N}\sum_{n=1}^{N}\left\|\sum_{k=K+1}^{D} z_k^{(n)}\boldsymbol{v}_k\right\|_2^2 \\
&= \frac{1}{N}\sum_{n=1}^{N}\sum_{k=K+1}^{D}\left(z_k^{(n)}\right)^2 \qquad \text{from orthonormality of the } \boldsymbol{v}_k \\
&= \frac{1}{N}\sum_{k=K+1}^{D}\sum_{n=1}^{N}((\boldsymbol{x}^{(n)})^\top \boldsymbol{v}_k)^\top((\boldsymbol{x}^{(n)})^\top \boldsymbol{v}_k) \qquad \text{from } (\boldsymbol{x}^{(n)})^\top \boldsymbol{v}_k = z_k^{(n)} \\
&= \frac{1}{N}\sum_{k=K+1}^{D}\sum_{n=1}^{N}\boldsymbol{v}_k^\top \boldsymbol{x}^{(n)}(\boldsymbol{x}^{(n)})^\top \boldsymbol{v}_k \\
&= \sum_{k=K+1}^{D}\boldsymbol{v}_k^\top\left[\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}^{(n)}(\boldsymbol{x}^{(n)})^\top\right]\boldsymbol{v}_k \\
&= \sum_{k=K+1}^{D}\boldsymbol{v}_k^\top \boldsymbol{S}\boldsymbol{v}_k.
\end{aligned}
$$

We note that minimizing this sum is equivalent to maximizing the total variance over the directions $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_K$ that we *keep*! This is precisely the previous formulation that we covered.

1.4 Computing Principal Components and Recap

So how do we actually find the principal components? Recall that these are eigenvectors of $\boldsymbol{S}$. It turns out that the answer lies in a **singular value decomposition** of $\boldsymbol{X}$ as $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top$, where $\boldsymbol{U}$ is an orthogonal $N \times N$ matrix, $\boldsymbol{D}$ is a rectangular diagonal $N \times D$ matrix, and $\boldsymbol{V}$ is a $D \times D$ orthogonal matrix. SVD conveniently results in each column $\boldsymbol{v}_k$ of $\boldsymbol{V}$ being an eigenvector of $\boldsymbol{X}^\top\boldsymbol{X}$ with the corresponding eigenvalue equal to d_k^2, where d_k is the k-th diagonal entry of $\boldsymbol{D}$. If we seek to use K principal components, then we simply pick the columns of $\boldsymbol{V}$ corresponding to the top K elements of $\boldsymbol{D}$.

Note: Observe that SVD yields eigenvectors and eigenvalues of $\boldsymbol{X}^\top\boldsymbol{X}$, rather than of $\boldsymbol{S}$. The eigenvectors will be the same because SVD imposes that $\boldsymbol{V}$ is orthogonal. However, the eigenvalues are different. Namely, the eigenvalues of $\boldsymbol{X}^\top\boldsymbol{X}$ are N times those of $\boldsymbol{S}$, since $\boldsymbol{X}^\top\boldsymbol{X}$ is just N times $\boldsymbol{S}$.

As a final summary, here is how we perform PCA:

1. Center the data by subtracting the mean of each feature from each data point. Steps 2 - 5 will be then performed on the centered data $\boldsymbol{X}$, which is still $N \times D$.
2. Calculate the empirical covariance matrix:

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

3. Decide how many dimensions K out of the original D we want to keep in the final representation.

4. Perform an SVD on $\boldsymbol{X}$ to find the K largest eigenvalues of $\boldsymbol{S}$. The K eigenvectors $(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_K)$ corresponding to these eigenvalues will be our lower-dimensional basis. Note that these are D-dimensional vectors.

5. Reduce the dimensionality of a data point $\boldsymbol{x}$ by projecting it onto this basis yielding a new reconstructed vector $\boldsymbol{z}$:
$$\boldsymbol{z} = \boldsymbol{v}^\top \boldsymbol{x}$$

Some important intuitions to understand are:

- The principal components are *orthonormal* directions, so they are all perpendicular to each other. This lets each PC introduce new information compared to all the previous ones.
- The principal components are the directions along whose the sample data has the most variance; that is, we could consider the projections of the sample data onto any particular direction/subspace, and the principal components are the directions which maximize the sample variance of the projections.
- The principal components are thus tied to the sample covariance matrix $\boldsymbol{S}$ of the data; in particular, they are the eigenvectors of this matrix.

1.5 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 are proportional to the amount of variance in the data explained by each principal component. When we retain a subset of D' principal components, we are effectively retaining the corresponding eigenvalues $\lambda_{d'}$ associated with those components. It may be shown that the retained variance is then calculated as the ratio of the sum of retained eigenvalues to the total sum of all eigenvalues:

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

We could also examine the scree plot. A scree plot is a graphical tool used in 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.

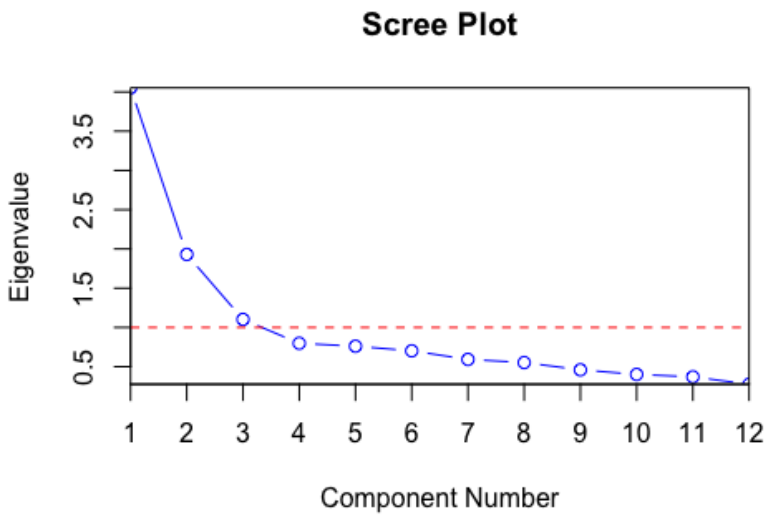


Figure 2: Scree Plot from Wikipedia.

Using a scree plot, we 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 can be chosen as the optimal number of principal components to retain for dimensionality reduction while preserving the most important information in the dataset.

Exercise: PCA by hand

You are given the following data set:

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

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 $\boldsymbol{S}$.
3. You find that $\boldsymbol{S}$ has eigenvector $[-1\ 1]^\top$ with eigenvalue 1 and eigenvector $[1\ 1]^\top$ with eigenvalue 3. What is the (normalized) basis vector $\boldsymbol{v}_1$ of your 1-dimensional representation? Add the basis vector $\boldsymbol{v}_1$ to your plot.
4. Compute the coefficients z_1, z_2, z_3. Add the lower-dimensional representations $z_1\boldsymbol{v}_1, z_2\boldsymbol{v}_1, z_3\boldsymbol{v}_1$ to your plot. Based on your plot, what is the relationship between $z_i\boldsymbol{v}_1$ and $\boldsymbol{x}_i$ 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?

Exercise: Choosing the Number of PCs

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.

2 Topic Models

For this section, we assume that readers are already familiar with mixture models and roughly familiar with expectation maximization (EM). To study those, we refer readers to the textbook and Section 6 notes.

Topic modeling, also referred to as the latent dirichlet allocation (LDA) model, is similar to other latent variable models; in particular, it may be viewed as a mixture of mixture models. As a high-level overview, topic modeling is used for discovering latent topics (themes) in large collections of documents. The goal is to understand the underlying structure and categories in the corpus of text documents. Topic modeling is thus an unsupervised learning method, as we do not have without prior knowledge of the labels or categories. Some of the popular applications of topic modeling include document clustering, text classification, and information retrieval.

Like the mixture models that we study in this course, we will describe topic models generatively: we will assume our corpus is generated by some process, and then use an optimization method (in particular, EM) to train the parameters of our model.

2.1 Probabilistic Latent Semantic Analysis (pLSA)

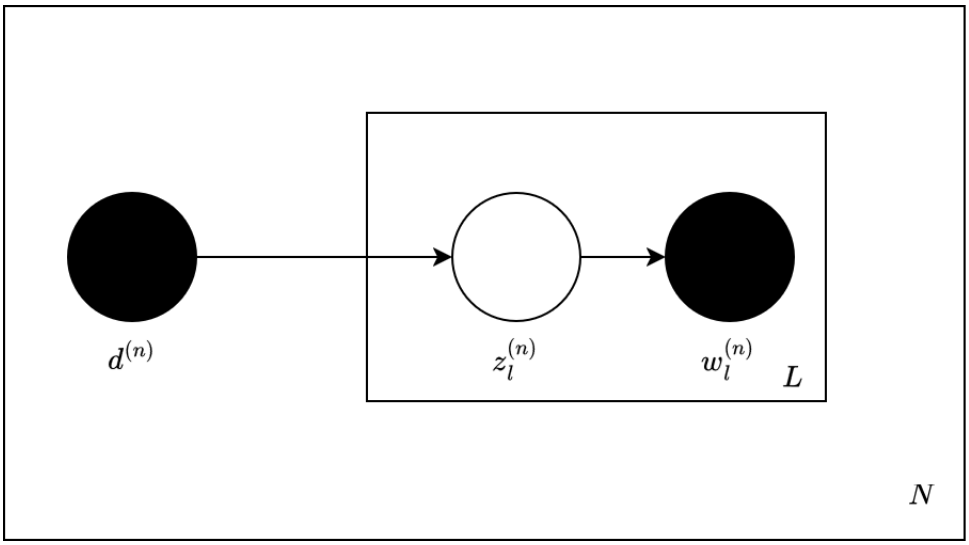
To motivate topic models (LDA), we first introduce the pLSA model and scenario. Consider a collection of N documents, where each document is a mixture of various topics, with K possible topics in total. pLSA is a generative model that assumes each word w in a document d is generated by sampling from a topic, and the topic is sampled from a per-document distribution. We sample L total words. The generative process for a document in pLSA is thus:

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 $p(w \mid z)$.

Our latent variable is thus the topic z, and our goal in pLSA is to learn the conditional probabilities $p(w \mid z)$ and $p(z \mid d)$. We do not impose any initial priors on these probabilities to do this. Instead, 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 $d^{(n)}$ and $w_l^{(n)}$ but define $z(n)_l$ 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:



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 (i.e., a lack of learning the joint $p(w, d)$). LDA assumes a similar generative process as pLSA: we still assume 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 n 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 the topic-level parameter $\boldsymbol{\phi}_k$ 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 parameter $\boldsymbol{\theta}^{(n)}$ which captures the distribution over topics. Hence, $\boldsymbol{\theta}^{(n)} \in [0, 1]^K$. These are the parameters to estimate in LDA.

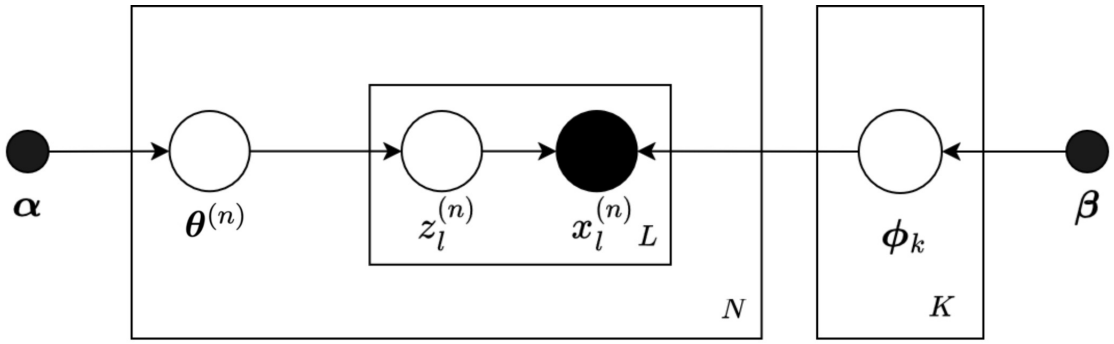
We now describe the data generation process:

1. Let $\boldsymbol{\alpha} \in \mathbb{R}_+^K$ and $\boldsymbol{\beta} \in \mathbb{R}_+^{|\mathcal{W}|}$.

2. For each document $n = 1, \ldots, N$, 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 $x_l^{(n)}$, first sample the topic $z_l^{(n)} \sim Cat(\boldsymbol{\theta}^{(n)})$, then sample the word $x_l^{(n)} \sim Cat\left(\boldsymbol{\phi}_{z_l^{(n)}}\right)$.

For some intuition, the Dirichlet distribution takes in a K-sized vector of values and outputs a probability distribution across K categories. It is the generalization of the Beta. 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, $\boldsymbol{\theta}^{(n)}$ specifies a distribution over topics in that document, and for each possible topic k, $\boldsymbol{\phi}_k$ 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 L and N, 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 main takeaway from LDA is that the introduction of Dirichlet priors for the per-document topic distributions $\boldsymbol{\theta}$ and per-topic word distributions ϕ 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: EM for LDA

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?