CS 181 Spring 2024 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 *reduce* the dimensionality of our data.

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 eliminating redundant information.

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 (i.e., the features are *linearly independent*). The columns of the corresponding change-of-basis matrix are called *principal components*, and are constructed so that each principal component (from left to right) 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 use case is. For example, if we wanted to plot the resulting data we might only keep the first two principal components.

1.2 PCA Intuition: Minimizing the Reconstruction Loss

To derive PCA, suppose we want to compress our D-dimensional data into $K < D$ dimensions, i.e., we want to find K D-dimensional vectors $\mathbf{v}_1, ..., \mathbf{v}_K$ such that we can represent each $\mathbf{x}_n$ as a linear combination of them:

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

where $z_{n,1}, ..., z_{n,k}$ are scalars and $\mathbf{v}$ is a matrix of all the basis vectors. In particular, we want to choose the $\mathbf{v}_i$ to be the vectors capturing the most variance in our data; for intuition as to why

this is the case, see the below figure. In the remainder of this section (Section 1.2), we will try to give some mathematical intuitions for why this is the case. Then in Section 1.3, we will provide a summary of the PCA procedure; this is more important to remember than the exact details of the derivations in Section 1.2.

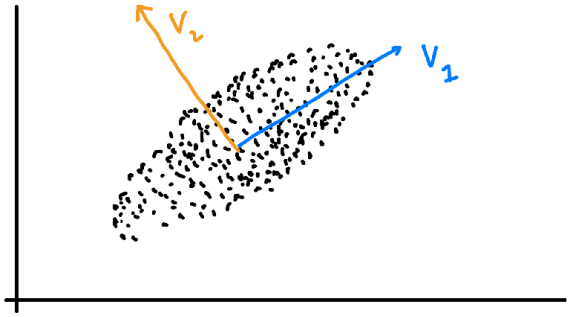


Figure 1: A plot of sample data with two features, x_1 plotted against x_2, along with the principal component directions v_1 and v_2. If we consider the projections of all the data points onto v_1, we see there is greater variance in the projections than if we projected onto v_2.

Mean-centering the data. First, suppose that the $N \times D$ data matrix $\mathbf{X}$ is mean-centered, so that the mean of the rows is the $\mathbf{0}$ vector. We choose to mean-center $\mathbf{X}$ because we don't want our bases $\mathbf{v}_i$ to capture the mean of the data, only the variation. (We can always re-add the mean $\overline{\mathbf{x}}$ of the rows at the end.)

PCA as minimizing reconstruction loss. To determine what the vectors $\mathbf{v}_i$ should be, we first impose the constraint that $\mathbf{v}$ must be *orthonormal*, i.e., that for any distinct rows k, k' in $\mathbf{v}$, we have $\mathbf{v}_k \cdot \mathbf{v}_k = 1$ and $\mathbf{v}_k \cdot \mathbf{v}_{k'} = 0$. This enforces that the $\mathbf{v}_i$ have unit length and that they are linearly independent, so that each $\mathbf{v}_i$ contains new information compared to all the other $\mathbf{v}_j$. Observe that using orthonormal basis vectors yields the property $\mathbf{x}_n^\top \mathbf{v}_k = z_{n,k}$, that is, the projection of $\mathbf{x}_n$ onto $\mathbf{v}_k$ has length exactly $z_{n,k}$.

Now, we define the $\mathbf{v}_i$ to be the vectors minimizing the *reconstruction loss*, defined as

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

which is the average distance from the linear approximation of each $\mathbf{x}_n$ to $\mathbf{x}_n$ itself, and where again $\mathbf{v}$ is orthonormal.

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 $\mathbf{x}_n$ since we'd be able to preserve all the

features. Thus, $\mathbf{x}_n = \sum_{k=1}^{D} z_{n,k}\mathbf{v}_k$. Thus, we can simplify our loss as follows:

$$\begin{aligned}
\mathcal{L}(\{\mathbf{z}_n\}, \mathbf{v}) &= \frac{1}{N}\sum_{n=1}^{N}\left\|\sum_{k=1}^{D} z_{n,k}\mathbf{v}_k - \sum_{k=1}^{K} z_{n,k}\mathbf{v}_k\right\|_2^2 \\
&= \frac{1}{N}\sum_{n=1}^{N}\left\|\sum_{k=K+1}^{D} z_{n,k}\mathbf{v}_k\right\|_2^2 \\
&= \frac{1}{N}\sum_{n=1}^{N}\sum_{k=K+1}^{D} z_{n,k}^2 \qquad \text{from orthonormality of the } \mathbf{v}_i \\
&= \frac{1}{N}\sum_{k=K+1}^{D}\sum_{n=1}^{N}(\mathbf{x}_n^\top \mathbf{v}_k)^\top(\mathbf{x}_n^\top \mathbf{v}_k) \qquad \text{from } \mathbf{x}_n^\top \mathbf{v}_k = z_{n,k} \\
&= \frac{1}{N}\sum_{k=K+1}^{D}\sum_{n=1}^{N}\mathbf{v}_k^T \mathbf{x}_n \mathbf{x}_n^\top \mathbf{v}_k \\
&= \sum_{k=K+1}^{D}\mathbf{v}_k^T\left[\frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n\mathbf{x}_n^\top\right]\mathbf{v}_k \\
&= \sum_{k=K+1}^{D}\mathbf{v}_k^T\left[\frac{1}{N}\mathbf{X}^\top\mathbf{X}\right]\mathbf{v}_k.
\end{aligned}$$

Principal components are the directions along which the data varies most. Now in this paragraph, we discuss the most important intuitions about PCA. Note that in the fourth line of our above derivations, our loss $\mathcal{L}$ is the sum over of the directions $\mathbf{v}_k$ of

$$\frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n^\top \mathbf{v}_k)^\top(\mathbf{x}_n^\top \mathbf{v}_k),$$

where $\mathbf{x}_n\mathbf{v}_k$ takes the projection of each point $\mathbf{x}_n$ onto the direction $\mathbf{v}_k$. Noting that the vectors $\mathbf{x}_n^\top \mathbf{v}_k$ have mean $\mathbf{0}$ over all n (via linearity of expectation, since the $\mathbf{x}_n$ are mean-centered), this quantity is thus the sample variance over the projections $\mathbf{X}\mathbf{v}_k$ of each data point onto the direction $\mathbf{v}_k$.

Hence, looking back to our expression for $\mathcal{L}$, minimizing the reconstruction loss is equivalent to choosing an orthonormal basis $\mathbf{v}$ and discarding the directions $\mathbf{v}_i$ along which the data $\mathbf{X}$ has least variance, while keeping the directions (principal components) $\mathbf{v}_j$ which *capture the most variance in the data*.

Principal components are the eigenvectors of the sample covariance matrix. Of note, it may be shown that the directions $\mathbf{v}_k$ are the *eigenvectors* of $\frac{1}{N}\mathbf{X}^\top\mathbf{X}$, which is the *empirical covariance matrix of* $\mathbf{X}$, and that eigenvectors with a higher eigenvalue capture more variance in the data than those with a lower eigenvalue!

We will not expect you to know the exact proof of this, but we will briefly explain why eigenvectors with larger eigenvalues capture more variance. In particular, we consider a singular value decomposition of $\mathbf{X}$ as $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$, where $\mathbf{D}$ is a diagonal matrix and $\mathbf{U}, \mathbf{V}$ are orthonormal. Now, we can see that the columns $\mathbf{v}_i$ of $\mathbf{V}$ are the eigenvectors of $\mathbf{X}^\top\mathbf{X}$ with eigenvalues d_i corresponding to the diagonal entries of $\mathbf{D}$: to see this, note that

$$\frac{1}{N}\mathbf{X}^T\mathbf{X} = \frac{1}{N}\left(\mathbf{U}\mathbf{D}\mathbf{V}^\top\right)^\top \mathbf{U}\mathbf{D}\mathbf{V}^\top = \frac{1}{N}\mathbf{V}\mathbf{D}\mathbf{U}^\top\mathbf{U}\mathbf{D}\mathbf{V}^\top = \frac{1}{N}\mathbf{V}\mathbf{D}^2\mathbf{V}^\top.$$

Now observe that any column $\mathbf{v}_i$ of $\mathbf{V}$ is an eigenvector of $\frac{1}{N}\mathbf{X}^T\mathbf{X}$: denoting b_i the i^{th} canonical basis vector (with 0s everywhere except for a 1 in the i^{th} entry) and using the orthonormality of the columns of $\mathbf{V}$, we have

$$\frac{1}{N}\mathbf{X}^T\mathbf{X}\mathbf{v}_i = \frac{1}{N}\mathbf{V}\mathbf{D}^2\mathbf{V}^\top\mathbf{v}_i = \frac{1}{N}\mathbf{V}\mathbf{D}^2 b_i = \frac{1}{N}\mathbf{V}d_i^2 b_i = \frac{d_i^2}{N}\mathbf{v}_i,$$

so $\mathbf{v}_i$ is an eigenvector of $\frac{1}{N}\mathbf{X}^T\mathbf{X}$ with eigenvalue proportional to d_i^2, where against d_i is the i^{th} diagonal entry of $\mathbf{D}$. Now if we were to consider the sample variance of the projections $\mathbf{X}\mathbf{v}_i$ of all the data points onto $\mathbf{v}_i$, this would be $\mathrm{Var}(\mathbf{X}\mathbf{v}_i)$, which is

$$\mathrm{Var}(\mathbf{X}\mathbf{v}_i) = \frac{1}{N}(\|(\mathbf{X}\mathbf{v}_i)^2\| - \|(\mathbf{X}\mathbf{v}_i)\|^2) = \frac{1}{N}(\mathbf{X}\mathbf{v}_i)^\top(\mathbf{X}\mathbf{v}_i) = \frac{1}{N}\mathbf{v}_i^\top\mathbf{X}^\top\mathbf{X}\mathbf{v}_i = \mathbf{v}_i^\top\frac{d_i^2}{N}\mathbf{v}_i = \frac{d_i^2}{N},$$

where in the second equality the term $\|\mathbf{X}\mathbf{v}_i\|^2$ disappeared since the data is mean-centered. Hence, the larger the eigenvalue d_i of $\mathbf{v}_i$, the greater the variance of the data along the direction $\mathbf{v}_i$.

1.3 Summary of PCA

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 then performed on the centered data $\mathbf{X}$, which is still $N \times D$.

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 $(\mathbf{v}_1, \ldots, \mathbf{v}_K)$ 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 $\mathbf{z}$:

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

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 $\mathbf{X}^\top \mathbf{X}$ of the data; in particular, they are the eigenvectors of this matrix.

1.4 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.

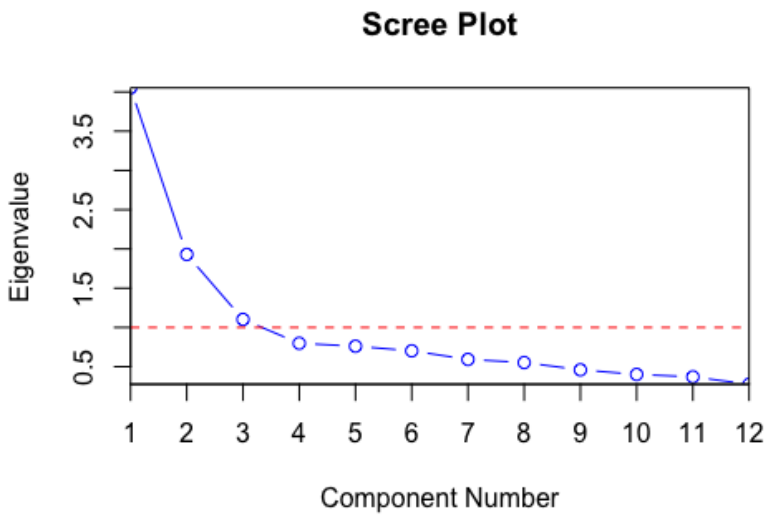


Figure 2: Scree Plot from Wikipedia.

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:

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

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 $\mathbf{v}_1$ of your 1-dimensional representation? Add the basis vector $\mathbf{v}_1$ to your plot.
4. Compute the coefficients z_1, z_2, z_3. Add the lower-dimensional representations $z_1\mathbf{v}_1, z_2\mathbf{v}_1, z_3\mathbf{v}_1$ to your plot. Based on your plot, what is the relationship between $z_i\mathbf{v}_1$ 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?

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)

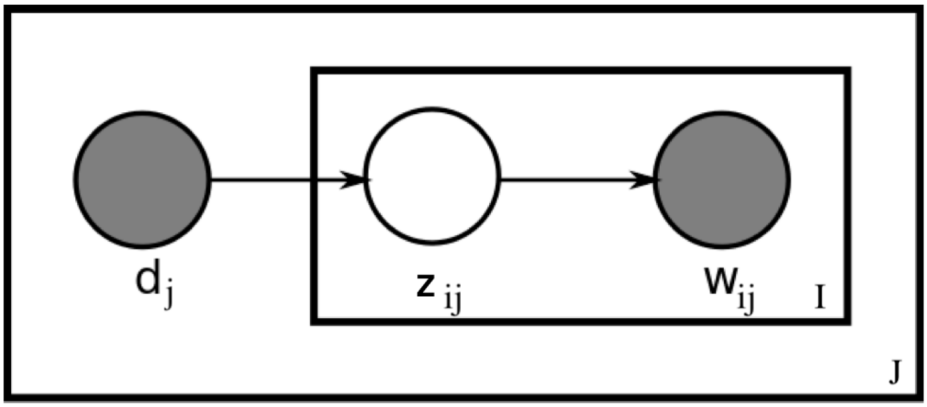
To motivate topic models (LDA), we first introduce the pLSA model and scenario. Consider a collection of documents, where each document is a mixture of various topics. pLSA is a generative model that assumes each word in a document is generated by sampling from a topic, and the topic is sampled from a per-document distribution. 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_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:



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 α and β 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 ϕ_k to be a distribution over the words, so $\phi_k \in [0,1]^{|\mathcal{W}|}$, where $\mathcal{W}$ is the set of words. For each document, we have a document-topic distribution $\theta_m \in [0,1]^K$. These are the parameters to estimate in LDA.

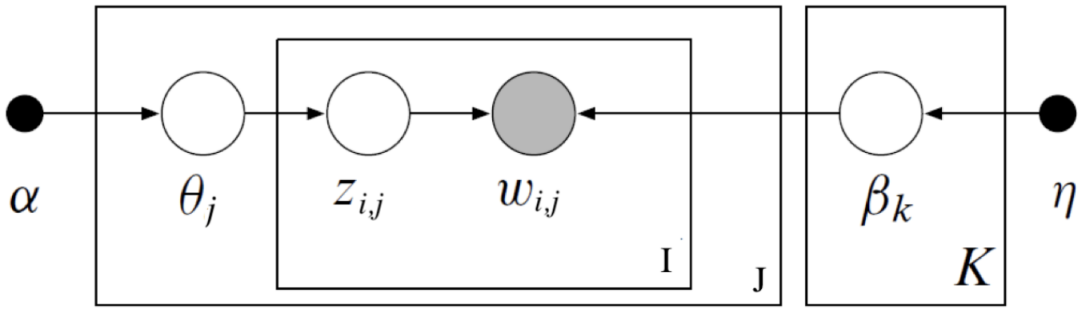
We now describe the data generation process:

1. Let $\alpha \in \mathbb{R}_+^K$ and $\beta \in \mathbb{R}_+^{|\mathcal{W}|}$.
2. For each document $m = 1, \ldots, M$, sample a mixture over topics: $\theta_n \sim Dir(\alpha)$.
3. For each topic $k = 1, \ldots, K$, sample a mixture over words in that topic: $\phi_k \sim Dir(\beta)$.

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 $z_{m,n} \sim Cat(\theta_m)$, then sample the word $w_{m,n} \sim Cat(\phi_{z_{m,n}})$.

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, θ_m specifies a distribution over topics in that document, and for each topic, k, in that document, ϕ_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 α 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 θ 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?