CS 181 Spring 2021 Section 9
Topic Models, Factor Analysis, and PCA

1 Topic Models

Topic modeling is a form of mixture modeling commonly used in analyzing text corpora. As this is a form of unsupervised learning, the goal is to extract a set of parameters from our data that we can use to make inferences about it. In particular, in topic modeling, we are interested in learning the topics that exist in a corpus - we will define this concept more formally.

Like mixture models that we study in this course, we will describe topic modeling as a generative model: one in which we assume our corpus is generated by some process. Then, we will seek to use an estimation technique, like EM, to determine the parameters of this model, and we can then interpret these parameters to understand the corpus.

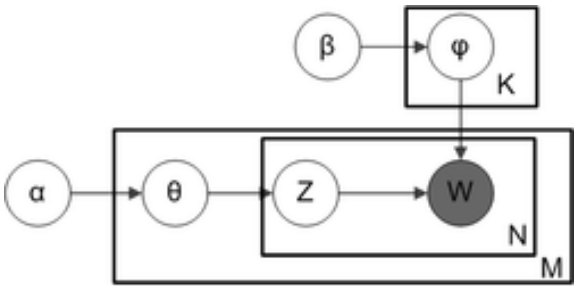
We begin by describing the generative process by which a document i is generated. For topic modeling, similar to K-means, we have to begin by picking the number of topics, K, to look for. We define a topic $\boldsymbol{\phi}_k$ to be a distribution over the words, so $\boldsymbol{\phi}_k \in [0,1]^{|\mathcal{W}|}$,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

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 $n = 1, \ldots, M$, sample a mixture over topics: $\boldsymbol{\theta}_n \sim Dir(\alpha)$.
3. For each topic $k = 1, \ldots, K$, sample a mixture over words in that topic: $\boldsymbol{\phi}_k \sim Dir(\beta)$.
4. For each word $w_{n,j}$, first sample the topic $z_{n,j} \sim Cat(\boldsymbol{\theta}_n)$, then sample the word $w_{n,j} \sim Cat(\boldsymbol{\phi}_{z_{n,j}})$.

At a high level, a topic model is a mixture over mixtures: within a single document, D_n, $\boldsymbol{\theta}_n$ 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 summarized in the following plate diagram:



Exercise 1. *In class, we discussed the EM algorithm applied to topic models. 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 nth document, the posterior distributions $q_{n,j}$ (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 $q_{n,j}$'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 $\boldsymbol{\theta}_n, \boldsymbol{\phi}_k$ to maximize this lower bound. Specifically, in the M step, we find parameters that maximize the expected complete-data log likelihood under q_n: $\mathbb{E}_{q_n}[\log p(\mathbf{W}, \mathbf{Z})]$, where $\mathbf{W}$ and $\mathbf{Z}$ denote words and topics, respectively (note that the expectation is taken over the unknown topics, $\mathbf{Z}$). □

2 Factor Analysis

In factor analysis, we have the following generative process:

1. $w \sim \mathcal{N}(0, I_K)$
2. For each n:
   - $z_n \sim \mathcal{N}(0, I_K)$
   - $x_n \sim \mathcal{N}(w^T z_n, \sigma^2)$ for some variance $\sigma^2 > 0$

In particular, the vector w and the variables z_n are *not* observable.

Estimating w (without access to the z's) and/or the distribution of the z's (without access to w) seems like a very difficult problem! However, suppose that we could see the latent variables z_n. Then finding w reduces to a linear regression problem (which we know how to solve). Alternatively, noting that $p(z_n|x_n, w) \propto p(x_n|z_n, w) \cdot p(z_n)$, if we could see the vector w, then we can write down the PDFs $p(x_n|z_n, w)$ and $p(z_n)$ and thus can find $p(z_n|x_n, w)$ (i.e., since we have its unnormalized PDF and everything is Gaussian). We can then solve for z_n by returning the mode of this PDF.

Exercise 2. *In the real-world setting where we don't have access to the latent variables or w, describe how we can use the above ideas to alternate between optimizing w give the z's and optimizing the z's given w. In particular, propose a Max-Max algorithm to do so.*

Solution. First randomly initialize the matrix w. Then we simply alternate between doing the two optimization procedures described above. Specifically, our Max-Max algorithm does the following:

1. Randomly initialize $w \in \mathbb{R}^{D \times K}$ (i.e., by drawing it from $\mathcal{N}(0, I_K)$)
2. Optimize the z_n's give w and the x_n's. That is, for each n, find the PDF of the distribution $p(z_n|x_n, w)$ as described above and choose z_n to be the mode of this distribution.
3. Optimize w given the x_n's and z_n's. That is, choose w which minimizes $\sum_{n=1}^{N}(x_n - w^T z_n)^2$ by using the OLS estimator.

Note that this algorithm is very similar, in spirit, to Lloyd's algorithm, which is also Max-Max.

□

3 Principal Component Analysis

3.1 Motivation

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 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 3. *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.

□

One method for dimensionality reduction through **linear projections** of the original data is PCA. When reducing the dimensionality of our data from m to d where $d < m$, PCA can be interpreted as minimizing the reconstruction loss of projecting data onto d basis vectors, or as maximizing the variance in data that can be explained by d basis vectors.

3.2 Finding the lower dimensional representation

To perform PCA, or project each data point $\mathbf{x}$: $(m \times 1)$ to $\mathbf{z}$: $(d \times 1)$,

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 $\mathbf{X}$: $(n \times m)$.
2. Calculate the **empirical covariance** matrix:

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

3. Decide how many dimensions d out of the original m that we want to keep in the final representation. For visualizations, often this will be $d = 2$ or $d = 3$.

4. Find the d largest eigenvalues of $\mathbf{S}$. The $m \times 1$ eigenvectors $(\mathbf{u}_1, \ldots, \mathbf{u}_d)$ corresponding to these eigenvalues will be our lower-dimensional basis.

5. Thus, we reduce the dimensionality of a data point $\mathbf{x}$ by projecting it onto this basis - we combine the eigenvectors into the $m \times d$ matrix $\mathbf{U}$, and compute

$$\langle \mathbf{u}_1^\top \mathbf{x}, \mathbf{u}_2^\top \mathbf{x}, \ldots, \mathbf{u}_d^\top \mathbf{x}, 0 \rangle = \mathbf{U}^\top \mathbf{x} = \mathbf{z}$$

   $\mathbf{z}$ is called the reconstruction coefficients where $\mathbf{U}\mathbf{z}$ is the reconstruction of $\mathbf{x}$.

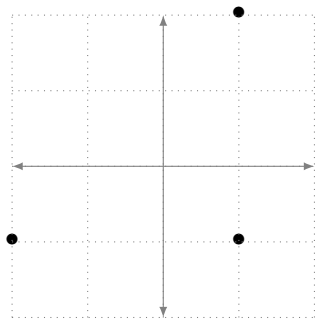
Exercise 4. *You are given the following data set:*

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

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?*

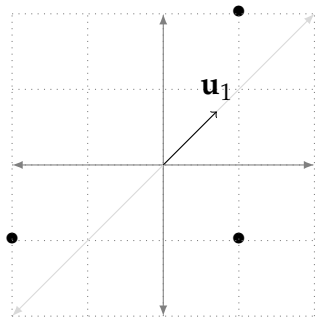
Solution. 1.



2.

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

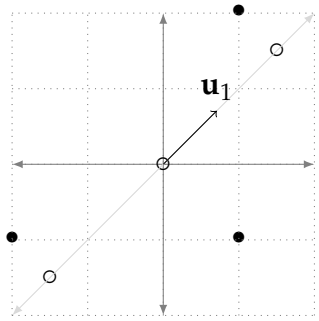
3. We select the eigenvectors with the largest eigenvalues for our basis, so our basis will contain a scalar multiple of $[1\ 1]^\top$. Normalizing $[1\ 1]^\top$ gives us that $\mathbf{u}_1 = [\frac{\sqrt{2}}{2}\ \frac{\sqrt{2}}{2}]^\top$.



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.

□

Exercise 5. *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 $U \in \mathbb{R}^{m \times d}$ (i.e., where $U^T U = I$) to minimize the reconstruction loss,*

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

we found that U_d, the matrix whose first d columns are (in order) the top d eigenvectors of the empirical covariance matrix $S = \frac{1}{n} X^T X$, will achieve the minimum (i.e., $z = U^T x$ is the projection of x into d dimensions, and 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:

Theorem 1 (Courant-Fischer)**.** *Let A be a symmetric $n \times n$ matrix with eigenvalues $\lambda_1 \leq \cdots \leq \lambda_n$ and corresponding eigenvectors $v_1, \ldots, v_n$. Then*

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

$$\lambda_2 = \min_{||x||=1, x \perp v_1} x^T A x$$

$$\vdots$$

$$\lambda_i = \min_{||x||=1, x \perp v_1, x \perp v_2, \ldots, x \perp v_{i-1}} x^T A x$$

$$\vdots$$

$$\lambda_n = \min_{||x||=1, x \perp v_1, x \perp v_2, \ldots, x \perp v_{n-1}} x^T A x.$$

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

$$x_i = \sum_{k=1}^{m} z_{i,k} \tilde{U}^{(k)}.$$

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

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

$$= \frac{1}{n}\sum_{i=1}^{n} || \sum_{k=d+1}^{m} z_{i,k} U^{(k)} ||_2^2 = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=d+1}^{m} z_{i,k}^2 = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=d+1}^{m} \left(\tilde{U}^{(k)T} x_i\right)\left(x_i^T \tilde{U}^{(k)}\right)$$

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

where Σ denotes the empirical covariance matrix.

Let $u_1, \ldots, u_m$ denote the eigenvectors of Σ (ordered by increasing corresponding eigenvalue). When $d+1 = m$, there's only one element in the sum, and since we know that $\tilde{U}$ is orthogonal, we must have that $||\tilde{U}^{(k)}||_2 = 1$ and thus Courant-Fischer says that $\tilde{U}^{(k)} = v_1$ will minimize. When $d+1 = m-1$, there will be two elements in the sum:

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

Consider solving this optimization problem by first picking $\tilde{U}^{(m)}$ and then $\tilde{U}^{(m-1)}$. Now, we either choose $\tilde{U}^{(m)}$ to be v_1 or we do not. If we choose $\tilde{U}^{(m)} = u_1$, the Courant-Fischer implies that picking $\tilde{U}^{(m-1)} = u_2$ will minimize. If we do not choose $\tilde{U}^{(m)}$ to be u_1, then Courant-Fischer implies that picking $\tilde{U}^{(m-1)} = u_1$ will minimize and that, if we did not pick $\tilde{U}^{(m)}$ to be u_2, we could've achieved an even lower value more by picking it to be so. Thus, in either case, we will take $\tilde{U}^{(m)}$ and $\tilde{U}^{(m-1)}$ to be u_1 and u_2.

Proceeding iteratively like this shows that we will chose $\tilde{U}^{(d+1)}, \ldots, \tilde{U}^{(m)}$ to be $v_1, \ldots, v_{m-d}$. Now, note that these are the columns we *exclude* from the matrix $\tilde{U}$ when we construct U (which will be the first d column of $\tilde{U}$). Since Courant-Fischer implies that $v_1, \ldots, v_n$ are orthogonal, taking U's columns to be $v_{m-d+1}, \ldots, v_m$ (i.e., the top d eigenvectors) indeed will minimize the reconstruction loss, as desired. □