CS769 Spring 2010 Advanced Natural Language Processing

Dimensionality Reduction and Latent Topic Models

Lecturer: Xiaojin Zhu   *jerryzhu@cs.wisc.edu*

Often a document can be thought of as a mixture of a small number of topics (e.g., a news article can be about 60% "finance", 30% "politics", and 10% "war".) Latent topic models recover hidden topics from document collections. This is interesting because

- The topics can be viewed as *latent semantic concepts*. Psychologists use latent topic models to explain the concept space we operate in, which is called the latent semantic space.

- Each document can be represented at the topics level instead of the word level. This allows two documents which share no common words (one says "buy", the other says "sell") to be regarded as similar because they share the the same "finance" topic.

1 LSI and PCA

1.1 Singular Value Decomposition (SVD) and Eigen Decomposition of a Matrix

Let $M_{D \times n}$ be a rectangular (i.e. not necessarily square nor symmetric) matrix. The singular value decomposition of M is

$$M_{D \times n} = U_{D \times m} S_{m \times m} V_{n \times m}^{\top}, \tag{1}$$

where $m = \min(D, n)$; U has orthonormal columns $u_1, \ldots, u_m$, so does V; S is diagonal, where the diagonal elements $\sigma_1 \geq \cdots \geq \sigma_m \geq 0$ are called the *singular values*.

Let $M_{D \times D}$ be a *square symmetric* matrix. The eigen decomposition of M is

$$M_{D \times D} = \Phi_{D \times D} \Lambda_{D \times D} \Phi_{D \times D}^{\top}, \tag{2}$$

where the columns of Φ are orthonormal eigenvectors $\phi_1, \ldots, \phi_D$, and Λ is a diagonal matrix with the eigenvalues $\lambda_1 \geq \cdots \geq \lambda_D$ on the diagonal. Note in general, the eigenvalues can be negative, unlike the singular values.

1.2 Latent Semantic Indexing (LSI)

Let the document collection be represented by a $D \times n$ matrix $X = [x_1 | \ldots | x_n]$, where the columns are document BOW vectors, D is the vocabulary size, and n is the number of documents.

LSI is SVD applied to X:

$$X_{D \times n} = U_{D \times m} S_{m \times m} V_{n \times m}^{\top}. \tag{3}$$

Furthermore, one only keeps the largest $d \ll m = \min(D, n)$ singular values. That is, let $\hat{U}_{D \times d}$ be the first d columns of U, $\hat{S}_{d \times d}$ be the first $d \times d$ submatrix of S, $\hat{V}_{n \times d}$ be the first d columns of V. Then

$$\hat{U}_{D \times d} \hat{S}_{d \times d} \hat{V}_{n \times d}^{\top}$$

is the best rank-d approximation to X in the least square sense. The d columns of $\hat{U}_{D \times d}$ defines the new, rotated, lower dimensional coordinate system. The n columns of $\hat{S}_{d \times d} \hat{V}_{n \times d}^{\top}$ are the new coordinates of each document after dimensionality reduction.

The new coordinate system allows a natural way to perform dimensionality reduction for points not in the dataset. For a new test document x^*, its new coordinate is $\hat{U}^\top x^*$. If x^* coincide with an existing document x_i, it will have the same new coordinates $\hat{U}^\top x_i = \hat{S}\hat{V}_{i.}^\top$ as above.

An alternative would be to add x^* to the dataset, and compute SVD on the $n+1$ documents. In general, this will produce a slightly different dimension reduction solution. It is much more computational intensive.

1.3   Principal Component Analysis (PCA)

Let X be the same document collection. Let $\bar{x} = 1/n \sum_{i=1}^n x_i$ be the mean document vector. Now *center* each document by subtracting the mean $y_i = x_i - \bar{x}$. Let $Y = [y_1 | \ldots | y_n]$ be the centered matrix. Let $\Sigma = 1/n YY^\top$ be the covariance matrix. The eigen-decomposition of Σ

$$\Sigma = \Phi \Lambda \Phi^\top \tag{4}$$

is known as the Principal Component Analysis of X. Furthermore, one only keeps the top $d \ll D$ eigenvalues. That is, let $\hat{\Phi}_{D \times d}$ be the first d columns of Φ, $\hat{\Lambda}_{d \times d}$ be the first $d \times d$ submatrix of Λ. Then

$$\hat{\Phi}_{D \times d} \hat{\Lambda}_{d \times d} \hat{\Phi}_{D \times d}^\top$$

is the best rank-d approximation to Σ in the least square sense. The columns of $\hat{\Phi}$ form the new, lower dimensional coordinate system. The new coordinates for y_i is $\hat{\Phi}^\top y_i$. The reconstruction of Y is $\hat{\Phi}\hat{\Phi}^\top Y$. The reconstruction of X is $\bar{x} + \hat{\Phi}\hat{\Phi}^\top Y$.

One major usage of PCA is in data visualization. Let $d = 2$ or 3. Each y_i (or x_i) is approximated by α_i, a d-dimension vector now suitable for plotting in 2D or 3D spaces.

Another major usage of PCA is to understand the few major axes $(\phi_1, \ldots, \phi_d)$ of the dataset. For example, if X is a collection of aligned face images (each x vector is the image pixels in some scanning order), then ϕ's (each vector can be turned back into an image) tend to represent the major prototype faces. This is known as eigen-face.

A shortcoming of PCA is that it is an unsupervised learning algorithm. There is no guarantee that the ϕ's necessarily correspond to any particular goal (for classification).

1.4   LSI vs. PCA

LSI is equivalent to PCA, *when X is already centered*. In this case,

$$
\begin{aligned}
XX^\top &= n\Sigma \tag{5}\\
USV^\top(USV^\top)^\top &= n\Phi\Lambda\Phi^\top \tag{6}\\
US^2U^\top &= \Phi(n\Lambda)\Phi^\top, \tag{7}
\end{aligned}
$$

That is, the LSI (SVD) U matrix is identical to the eigenvector matrix Φ in PCA, and the singular values are the square root of eigenvalues in PCA (scaled by n): $\sigma_i = \sqrt{n\lambda_i}$.

2   Probabilistic Latent Semantic Analysis (pLSA)

Recall Naive Bayes models a document collection by K topics (classes). Each topic is a multinomial over words, and each document is generated from a single topic:

$$p(w) = \sum_{k=1}^{K} p(z = k)p(w|z = k). \tag{8}$$

The parameters $p(z = k), p(w|z = k)$ can be learned from labeled data (MLE or MAP), or with the EM algorithm if there is no class label (this corresponds to a mixture of multinomial model with K components). This is not a very flexible model, because it assumes "one document, one topic".

pLSA assumes that *each* document d (with word vector w) is generated from all topics, with document-specific topic weights. The generative process of pLSA is the following. Given a fixed document collection with n documents, we represent it as a $n \times V$ document-word matrix with entry $c(d, w)$, the count of word type w in document d. At each iteration, one picks a topic $z = 1 \ldots K \sim p(z)$, then picks a document and a word type independent of each other, but both depends on the topic: $d \sim p(d|z = k)$, $w \sim p(w|z = k)$. Generate (add one count of) word w to document d. Repeat until we generate the document-word matrix. Under this process, the probability of picking the cell (d, w) is

$$p(d, w) = \sum_{z=1}^{K} p(z)p(d|z)p(w|z). \tag{9}$$

The model parameters are $\Theta = \{p(z), p(d|z), p(w|z)\}$. We want to find the MLE to maximize the likelihood of the observed document-word matrix,

$$\max_{\Theta} \quad \sum_{d=1}^{n} \sum_{w=1}^{V} c(d, w) \log p(d, w) \tag{10}$$

$$= \quad \max_{\Theta} \quad \sum_{d=1}^{n} \sum_{w=1}^{V} c(d, w) \log \left(\sum_{z=1}^{K} p(z)p(d|z)p(w|z) \right). \tag{11}$$

Note z is a hidden variable, and note the sum inside log. We can apply the EM algorithm:

$$\sum_{d,w} c(d, w) \log p(d, w) \tag{12}$$

$$= \quad \sum_{d,w} c(d, w) \log \left(\sum_{z=1}^{K} p(z)p(d|z)p(w|z) \right) \tag{13}$$

$$= \quad \sum_{d,w} c(d, w) \log \left(\sum_{z=1}^{K} p(z|d, w, \Theta^t) \frac{p(z)p(d|z)p(w|z)}{p(z|d, w, \Theta^t)} \right) \tag{14}$$

$$\geq \quad \sum_{d,w} c(d, w) \sum_{z=1}^{K} p(z|d, w, \Theta^t) \left(\log \frac{p(z)p(d|z)p(w|z)}{p(z|d, w, \Theta^t)} \right) \tag{15}$$

Note Jensen's inequality involves $p(z|d, w, \Theta^t)$, which computes the probability of topics separately for each cell, under the current parameters Θ^t. This is exactly the E-step. They can be computed as

$$p(z|d, w, \Theta^t) \propto p(z|\Theta^t)p(d|z, \Theta^t)p(w|z, \Theta^t). \tag{16}$$

Maximizing (15) by setting the gradient to zero amounts to the M-step, which gives

$$p(z) \quad \propto \quad \sum_{d} \sum_{w} c(d, w)p(z|d, w, \Theta^t) \tag{17}$$

$$p(d|z) \quad \propto \quad \sum_{w} c(d, w)p(z|d, w, \Theta^t) \tag{18}$$

$$p(w|z) \quad \propto \quad \sum_{d} c(d, w)p(z|d, w, \Theta^t). \tag{19}$$

The E-step and M-step are repeated until convergence.

Once the model is trained, we can look at it in the following way:

- $p(w|z)$ are the topics. Each topic is defined by a word multinomial. Often people find that the topics seem to have distinct semantic meanings.

- From $p(d|z)$ and $p(z)$, we can compute $p(z|d) \propto p(d|z)p(z)$. $p(z|d)$ is the topic wights for document d.

One drawback of pLSA is that it is *transductive* in nature. That is, there is no easy way to handle a new document that is not already in the collection.

3 Latent Dirichlet Allocation (LDA)

LDA too assumes that each document is a mixture of multiple topics, and each document can have different topics weights. Unlike pLSA, LDA is a full generative model and readily generalizes to unseen documents. The LDA generative process is the following.

1. Sample K multinomial distributions (each of size V) $\phi_{1:K}$ from a Dirichlet distribution $Dir(\beta)$, these are the topics. Note β is a parameter vector of length V.

2. For each document

   (a) Sample a topic multinomial (of size K) θ from a Dirichlet distribution $Dir(\alpha)$.

   (b) For each word position

      i. Sample topic index $z \sim \theta$

      ii. Sample a word from the topic $w \sim \phi_z$

The observation is the document collection $w_{1:n}$. The parameters are α and β. The other parameters z, ϕ and θ are hidden variables that will be marginalized out.

The probability of a topic multinomial ϕ drawn from $Dir(\beta)$ is

$$p(\phi|\beta) = \frac{\Gamma(\sum_{i=1}^{V} \beta_i)}{\prod_{i=1}^{V} \Gamma(\beta_i)} \prod_{i=1}^{V} \phi_i^{\beta_i - 1}. \tag{20}$$

The probability of drawing the K topic multinomials are

$$p(\phi_{1:K}|\beta) = \prod_{j=1}^{K} p(\phi_j|\beta). \tag{21}$$

Similarly,

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{K} \alpha_i)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \prod_{i=1}^{K} \theta_i^{\alpha_i - 1}. \tag{22}$$

Let $p(z|\theta)$ and $p(w|z,\phi) = p(w|\phi_z)$ be the corresponding multinomial probabilities. The probability of generating the words $w_{1:N}$ for a single document, given θ, ϕ is

$$\prod_{n=1}^{N} \sum_{z_n=1}^{K} p(z_n|\theta)p(w_n|z_n, \phi), \tag{23}$$

where we marginalize out z for each word position. Putting things together, the probability of a single document, after marginalizing out all hidden variables, is

$$p(w|\alpha, \beta) = \int_{\phi_{1:K}} \int_{\theta} p(\phi_{1:K}|\beta)p(\theta|\alpha) \left(\prod_{n=1}^{N} \sum_{z_n=1}^{K} p(z_n|\theta)p(w_n|z_n, \phi) \right) d\theta d\phi_{1:K}. \tag{24}$$

Finally, the probability of a document collection $w^{(1)}, \ldots, w^{(M)}$ is

$$p(w^{(1)}, \ldots, w^{(M)}|\alpha, \beta) = \prod_{d=1}^{M} p(w^d|\alpha, \beta). \tag{25}$$