Topics in Empirical Economics, Part III
Introduction to Unsupervised Learning

Stephen Hansen
Universitat Pompeu Fabra

Introduction

There are two main divisions in machine learning:

1. *Supervised learning* seeks to build models that predict labels associated with observations.
2. *Unsupervised learning* seeks to model hidden structure in unlabeled observations.

We begin with unsupervised learning. This can be seen as an intermediate step in empirical analysis in which we first quantify text on a low-dimensional space before using that representation in econometric work.

Important differences with respect to dictionary methods:

1. Use all dimensions of variation in the document-term matrix to estimate the latent space.
2. Let data reveal which are the most important dimensions for discriminating among observations.

Latent Variable Models

The implicit assumption of dictionary methods is that the set of words in the dictionary map back into an underlying theme of interest.

For example we might have that
$\mathfrak{D} = \{\text{school}, \text{university}, \text{college}, \text{teacher}, \text{professor}\} \rightarrow \text{education}.$

Latent variable models formalize the idea that documents are formed by hidden variables that generate correlations among observed words.

In a natural language context, these variables can be thought of as topics; other applications will have other interpretations.

Latent variable models generally share the following features:

1. Associate words with latent variables; the same word can be associated to more than one latent variable.
2. Associate documents with (one or more) latent variables.

Information Retrieval

Latent variable representations can more accurately identify document similarity.

The problem of *synonomy* is that several different words can be associated with the same topic. Cosine similarity between following documents?

school	university	college	teacher	professor
0	5	5	0	2
school	university	college	teacher	professor
10	0	0	4	0

The problem of *polysemy* is that the same word can have multiple meanings. Cosine similarity between following documents?

tank	seal	frog	animal	navy	war
10	10	3	2	0	0
tank	seal	frog	animal	navy	war
10	10	0	0	4	3

If we correctly map words into topics, comparisons become more accurate.

Outline

Single-membership models:

1. K-means algorithm
2. Multinomial mixture model and the EM algorithm

Mixed-membership models:

1. Latent semantic indexing and singular value decomposition
2. Probabilistic latent semantic indexing

K-Means

Recall we can represent document d as a vector $\vec{x}_d \in \mathbb{R}_+^V$. In the k-means model, every document has a single cluster assignment.

Let D_k be the set of all documents that are in cluster k. The *centroid* of the documents in cluster k is $\vec{u}_k = \frac{1}{|D_k|} \sum_{d \in D_k} \vec{x}_d$.

In k-means we choose cluster assignments $\{D_1, \ldots, D_K\}$ to minimize the sum of squares between each document and its cluster centroid:

$$\sum_k \sum_{d \in D_k} \|\vec{x}_d - \vec{u}_k\|^2$$

Solution groups similar documents together, and centroids represent prototype documents within each cluster.

Normalize document lengths to cluster on content, not length.

Solution Algorithm

First initialize the centroids $\vec{u}_k$ for $1, \ldots, K$.

Repeat the following steps until convergence:

1. Assign each document to its closest centroid, i.e. choose an assignment k for d that minimizes $\|\vec{x}_d - \vec{u}_k\|$.
2. Recompute the cluster centroids as $\vec{u}_k = \frac{1}{|D_k|} \sum_{d \in D_k} \vec{x}_d$ given the updated assignments in previous step.

The objective function is guaranteed to decrease at each step → convergence to local minimum.

Proof: for step 1 obvious; for step 2 choose elements of vector $\vec{y} \in \mathbb{R}_+^V$ to minimize $\sum_{d \in D_k} \|\vec{x}_d - \vec{y}\|^2 \equiv \sum_{d \in D_k} \sum_v (x_{d,v} - y_v)^2$. Solution is exactly $\vec{u}_k$.

Selection of Number of Clusters

K is a parameter in k-means that the researcher must select.

What value of K will maximize the goodness of fit of the model?

In practice we can:

1. Use our own judgment
2. Plot objective function for different values of K, look for kinks
3. Use a formal statistical criterion that trades off goodness-of-fit and model complexity

Probabilistic Modeling

As presented so far, the k-means model might produce useful groupings, but is ad hoc and has no obvious statistical foundations.

A more satisfactory approach might be to write down a statistical model for documents whose parameters we estimate—allows us to incorporate and make inferences about relevant structure in the corpus.

Akin to structural modeling in econometrics (although no behavioral model), and called *generative* modeling in the machine learning literature.

Simple Language Model

We can view a document d as an (ordered) list of words $\mathbf{w}_d = (w_{d,1}, \dots, w_{d,N_d})$.

In a unigram model, $\Pr[\mathbf{w}_d] = \Pr[w_{d,1}] \Pr[w_{d,2}] \Pr[w_{d,3}] \dots \Pr[w_{d,N_d}]$.

Let $\Pr[w_{d,n} = v] = \beta_v$ and let $\boldsymbol{\beta} \equiv (\beta_1, \dots, \beta_V) \in \Delta^{V-1}$. This defines a categorical distribution parametrized by $\boldsymbol{\beta}$.

The probability of the data given the parameters is

$$\Pr[\mathbf{w}_d \mid \boldsymbol{\beta}] = \prod_v \beta_v^{x_{d,v}}$$

Note that term counts are all we need to compute this likelihood, which is a consequence of the independence assumption. This provides a statistical foundation for the bag-of-words model. $\Pr[\mathbf{w}_d \mid \boldsymbol{\beta}] = \Pr[\mathbf{x}_d \mid \boldsymbol{\beta}]$.

Maximum Likelihood Inference

We can estimate $\boldsymbol{\beta}$ with maximum likelihood. The Lagrangian is

$$\mathfrak{L}(\boldsymbol{\beta}, \lambda) = \underbrace{\sum_v x_{d,v} \log(\beta_v)}_{\text{log-likelihood}} + \lambda \underbrace{\left(1 - \sum_v \beta_v\right)}_{\text{Constraint on } \boldsymbol{\beta}} .$$

First order condition is $\frac{x_{d,v}}{\beta_v} - \lambda = 0 \Rightarrow \beta_v = \frac{x_{d,v}}{\lambda}$.

Constraint gives $\frac{\sum_v x_{d,v}}{\lambda} = 1 \Rightarrow \lambda = \sum_v x_{d,v} = N_d$.

So MLE estimate is $\widehat{\beta}_v = \frac{x_{d,v}}{N_d}$, the observed frequency of term v in document d.

Multinomial Mixture Model

A basic probabilistic model for unsupervised learning with discrete data is the multinomial mixture model.

This builds on the simple language model above, but introduces k separate categorical distributions, each with parameter vector $\boldsymbol{\beta}_k$.

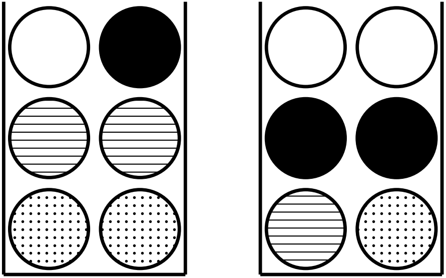
Every document belongs to a single category $z_d \in \{1, \ldots, K\}$, which is independent across documents and drawn from $\Pr[\,z_d = k\,] = \rho_k$.

The probability that documents with category k generate term v is $\beta_{k,v}$.

Topics as Urns

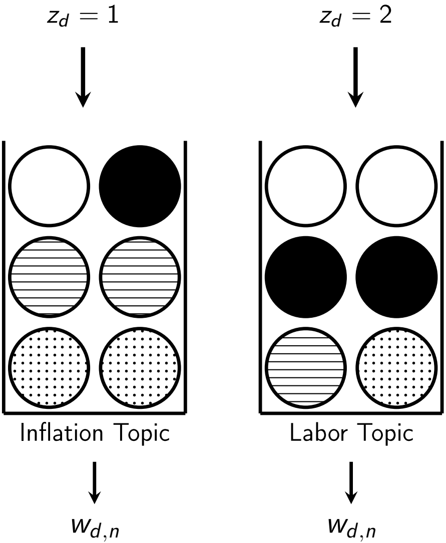
○ = wage   ● = employ

▤ = price   ⁘ = increase



“Inflation” Topic   “Labor” Topic

Mixture Model for Document



Probability of Document

Now we can again write down the probability of observing a document given the vector of mixing probabilities $\boldsymbol{\rho}$ and the matrix of term probabilities $\mathbf{B}$.

Suppose that $z_d = k$. Then the probability of $\mathbf{w}_d$ is $\prod_v (\beta_{k,v})^{x_{d,v}}$.

To compute the unconditional probability of document d, we need to marginalize over the latent assignment variable z_d

$$\Pr[\mathbf{x}_d \mid \boldsymbol{\rho}, \mathbf{B}] = \sum_{z_d} \Pr[\mathbf{x}_d \mid z_d; \boldsymbol{\rho}, \mathbf{B}] \Pr[z_d \mid \boldsymbol{\rho}, \mathbf{B}] = \sum_k \rho_k \prod_v (\beta_{k,v})^{x_{d,v}}.$$

Probability of Corpus

By independence of latent variables across documents, the likelihood of entire corpus, which we can summarize with document-term matrix **X** is

$$L\left(\mathbf{X} \mid \boldsymbol{\rho}, \mathbf{B}\right) = \prod_d \sum_k \rho_k \prod_v (\beta_{k,v})^{x_{d,v}}$$

and log-likelihood is

$$\ell\left(\mathbf{X} \mid \boldsymbol{\rho}, \mathbf{B}\right) = \sum_d \log \left(\sum_k \rho_k \prod_v (\beta_{k,v})^{x_{d,v}} \right).$$

Inference

Here the sum over the latent variable assignments lies within the logarithm, which makes MLE intractable.

Inference

Here the sum over the latent variable assignments lies within the logarithm, which makes MLE intractable.

On the other hand, suppose that we knew the category assignment of each document. Then MLE would be very easy.

So we could “fill-in” values for the latent variables by taking expectations, and then compute the MLE parameter estimates.

This provides the motivation for the expectation-maximization (EM) algorithm.

Inference

Here the sum over the latent variable assignments lies within the logarithm, which makes MLE intractable.

On the other hand, suppose that we knew the category assignment of each document. Then MLE would be very easy.

So we could "fill-in" values for the latent variables by taking expectations, and then compute the MLE parameter estimates.

This provides the motivation for the expectation-maximization (EM) algorithm.

See Arcidiacono and Jones (ECMA 2003) and Arcidiacono and Miller (ECMA 2011) for applications in economics.

Complete Data Log-Likelihood

The joint distribution of $\mathbf{x}_d$ and z_d is $\prod_k \left[\rho_k \prod_v (\beta_{k,v})^{x_{d,v}}\right]^{\mathbb{1}(z_d=k)}$ and so the joint distribution of $\mathbf{X}$ and $\mathbf{z} = (z_1, \ldots, z_D)$ is

$$L_{\text{comp}}(\mathbf{X}, \mathbf{z} \mid \boldsymbol{\rho}, \mathbf{B}) = \prod_d \prod_k \left[\rho_k \prod_v (\beta_{k,v})^{x_{d,v}}\right]^{\mathbb{1}(z_d=k)}$$

The *complete data log-likelihood* is

$$\ell_{\text{comp}}(\mathbf{X}, \mathbf{z} \mid \boldsymbol{\rho}, \mathbf{B}) = \sum_d \sum_k \mathbb{1}(z_d = k) \left[\log(\rho_k) + \sum_v x_{d,v} \log(\beta_{k,v})\right].$$

Note this function is much easier to maximize with respect to the parameters than the original log-likelihood function.

Expectation Step

In the expectation step of the EM algorithm, we compute the expected value of the complete data log-likelihood with respect to the latent variables given the current value of the parameters $\boldsymbol{\rho}^i$ and $\mathbf{B}^i$ and the data $\mathbf{X}$.

Expectation Step

In the expectation step of the EM algorithm, we compute the expected value of the complete data log-likelihood with respect to the latent variables given the current value of the parameters $\boldsymbol{\rho}^i$ and $\mathbf{B}^i$ and the data $\mathbf{X}$.

Clearly $\mathbb{E}\left[\,\mathbb{1}(z_d = k) \mid \boldsymbol{\rho}^i, \mathbf{B}^i, \mathbf{X}\,\right] = \Pr\left[\,z_d = k \mid \boldsymbol{\rho}^i, \mathbf{B}^i, \mathbf{X}\,\right] \equiv \widehat{z}_{d,k}$.

By Bayes' Rule we have that

$$\begin{aligned}\widehat{z}_{d,k} = & \Pr\left[\,z_d = k \mid \boldsymbol{\rho}^i, \mathbf{B}^i, \mathbf{x}_d\,\right] \propto \\ & \Pr\left[\,\mathbf{x}_d \mid \boldsymbol{\rho}^i, \mathbf{B}^i, z_d = k\,\right] \Pr\left[\,z_d = k \mid \boldsymbol{\rho}^i, \mathbf{B}^i\,\right] = \rho_k \prod_v \left(\beta_{k,v}\right)^{x_{d,v}}.\end{aligned}$$

So the expected complete log-likelihood becomes

$$Q(\boldsymbol{\rho}, \mathbf{B}, \boldsymbol{\rho}^i, \mathbf{B}^i) = \sum_d \sum_k \widehat{z}_{d,k} \left[\log(\rho_k) + \sum_v x_{d,v} \log\left(\beta_{k,v}\right)\right]$$

Maximization Step

In the maximization step, we maximize the expected complete log-likelihood with respect to $\boldsymbol{\rho}$ and $\mathbf{B}$ to obtain updated parameter estimates $\boldsymbol{\rho}^{i+1}$ and $\mathbf{B}^{i+1}$.

Maximization Step

In the maximization step, we maximize the expected complete log-likelihood with respect to $\boldsymbol{\rho}$ and $\mathbf{B}$ to obtain updated parameter estimates $\boldsymbol{\rho}^{i+1}$ and $\mathbf{B}^{i+1}$.

The Lagrangian for this problem is

$$Q(\boldsymbol{\rho}, \mathbf{B}, \boldsymbol{\rho}^i, \mathbf{B}^i) + \nu \left(1 - \sum_k \rho_k\right) + \sum_k \lambda_k \left(1 - \sum_v \beta_{k,v}\right).$$

Maximization Step

In the maximization step, we maximize the expected complete log-likelihood with respect to $\boldsymbol{\rho}$ and $\mathbf{B}$ to obtain updated parameter estimates $\boldsymbol{\rho}^{i+1}$ and $\mathbf{B}^{i+1}$.

The Lagrangian for this problem is

$$Q(\boldsymbol{\rho}, \mathbf{B}, \boldsymbol{\rho}^i, \mathbf{B}^i) + \nu \left(1 - \sum_k \rho_k \right) + \sum_k \lambda_k \left(1 - \sum_v \beta_{k,v} \right).$$

Standard maximization gives

$$\rho_k^{i+1} = \frac{\sum_d \hat{z}_{d,k}}{\sum_k \sum_d \hat{z}_{d,k}},$$

or the average probability that documents have topic k and

$$\beta_{k,v}^{i+1} = \frac{\sum_d \hat{z}_{d,k} x_{d,v}}{\sum_d \hat{z}_{d,k} \sum_v x_{d,v}},$$

or the expected number of times documents of type k generate term v over the expected number of words generated by type k documents.

Solution Algorithm

First initialize parameter values $\boldsymbol{\rho}^0$ and $\mathbf{B}^0$. Then, at iteration i:

1. Compute expected complete data log-likelihood $Q(\boldsymbol{\rho}, \mathbf{B}, \boldsymbol{\rho}^{i-1}, \mathbf{B}^{i-1})$.
2. Update parameter estimates to $\boldsymbol{\rho}^i, \mathbf{B}^i$ using equations on previous slide.
3. If convergence criterion met, stop; otherwise proceed to iteration $i = i + 1$.

The log-likelihood $\ell\,(\mathbf{X} \mid \boldsymbol{\rho}, \mathbf{B})$ is guaranteed to increase at each iteration. (Proof omitted, but see Murphy textbook). We converge to a local maximum.

Initially intractable MLE problem solved through relatively simple procedure.

No guarantee of global optimality; can be especially problematic for high-dimensional data. One diagnostic is to check sensitivity of parameter estimates to starting values.

Example

Let $K = 2$ and consider the corpus of 1,232 paragraphs of State-of-the-Union Addresses since 1900.

Topic	Top Terms
0	tax.job.help.must.congress.need.health.care.busi.let.school.time
1	world.countri.secur.must.terrorist.iraq.state.energi.help.unit

$(\rho_0, \rho_1) = (0.42, 0.58)$.

No *ex ante* labels on clusters, so any interpretation is *ex post*, and potentially subjective, judgment on the part of the researcher.

K-Means as EM

The k-means algorithm can be viewed as the EM algorithm under a special case of a Gaussian mixture model in which the distribution of data in cluster k is $\mathcal{N}(\boldsymbol{\mu}_k, \Sigma_k)$ where $\Sigma_k = \sigma^2 \mathbf{I}_V$ and σ^2 is small.

The probability that document d is generated by the cluster with the closest mean is then close to 1, so the assignment of documents to the closest centroid in the k-means algorithm is the E-step.

Given the spherical covariance matrix, the probability of observing documents within cluster k is proportional to the sum of squared distances between documents and the mean. So recomputing the cluster centroids is the M-step.

Good news is that k-means has statistical foundations; bad news is that the appropriateness of these for count data is doubtful.

Mixed-Membership Models

In both k-means and the multinomial mixture model, documents are associated with a single topic.

NB: the EM algorithm provides a probability distribution over topic assignments.

In practice, we might imagine that documents cover more than one topic.

Examples: State-of-the-Union Addresses discuss domestic and foreign policy; monetary policy speeches discuss inflation and growth.

Models that associated observations with more than one latent variable are called *mixed-membership* models. Also relevant outside of text mining: in models of group formation, agents can be associated with different latent communities (sports team, workplace, church, etc).

Latent Semantic Analysis

One of the first mixed-membership models in text mining was the Latent Semantic Analysis/Indexing model of Deerwester et. al. (1990).

A linear algebra rather than probabilistic approach that applies a singular value decomposition to document-term matrix.

Closely related to classical principal components analysis.

Examples in economics: Hendry and Madeley (2010); Acosta (2014).

Review

Let $\mathbf{X}$ be an $N \times N$ symmetric matrix with N linearly independent eigenvectors.

Then there exists a decomposition $\mathbf{X} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^T$, where $\mathbf{Q}$ is an orthogonal matrix whose columns are eigenvectors of $\mathbf{X}$ and $\boldsymbol{\Lambda}$ is a diagonal matrix whose entries are eigenvalues of $\mathbf{X}$.

When we apply this decomposition to the variance-covariance matrix of a dataset, we can perform principal components analysis.

The eigenvalues in $\boldsymbol{\Lambda}$ give a ranking of the columns in $\mathbf{Q}$ according to the variance they explain in the data.

Singular Value Decomposition

The document-term matrix $\mathbf{X}$ is not square, but we can decompose it using a generalization of the eigenvector decomposition called the *singular value decomposition.*

Proposition

The document-term matrix can be written $\mathbf{X} = \mathbf{A}\mathbf{\Sigma}\mathbf{B}^T$ *where* $\mathbf{A}$ *is a* $D \times D$ *orthogonal matrix,* $\mathbf{B}$ *is a* $V \times V$ *orthogonal matrix, and* $\mathbf{\Sigma}$ *is a* $D \times V$ *matrix where* $\mathbf{\Sigma}_{ii} = \sigma_i$ *with* $\sigma_i \geq \sigma_{i+1}$ *and* $\mathbf{\Sigma}_{ij} = 0$ *for all* $i \neq j$*.*

Singular Value Decomposition

The document-term matrix $\mathbf{X}$ is not square, but we can decompose it using a generalization of the eigenvector decomposition called the *singular value decomposition.*

Proposition

The document-term matrix can be written $\mathbf{X} = \mathbf{A}\boldsymbol{\Sigma}\mathbf{B}^T$ *where* $\mathbf{A}$ *is a* $D \times D$ *orthogonal matrix,* $\mathbf{B}$ *is a* $V \times V$ *orthogonal matrix, and* $\boldsymbol{\Sigma}$ *is a* $D \times V$ *matrix where* $\boldsymbol{\Sigma}_{ii} = \sigma_i$ *with* $\sigma_i \geq \sigma_{i+1}$ *and* $\boldsymbol{\Sigma}_{ij} = 0$ *for all* $i \neq j$*.*

Some terminology:

- Columns of $\mathbf{A}$ are called left singular vectors.
- Columns of $\mathbf{B}$ are called right singular vectors.
- The diagonal terms of $\boldsymbol{\Sigma}$ are called singular values.

Interpretation of Left Singular Vectors

Note that $\mathbf{X}\mathbf{X}^T = \mathbf{A}\mathbf{\Sigma}\mathbf{B}^T\mathbf{B}\mathbf{\Sigma}^T\mathbf{A}^T = \mathbf{A}\mathbf{\Sigma}\mathbf{\Sigma}^T\mathbf{A}^T$.

This is the eigenvector decomposition of the matrix $\mathbf{X}\mathbf{X}^T$, whose (i,j)th element measures the overlap between documents i and j.

Left singular vectors are eigenvectors of $\mathbf{X}\mathbf{X}^T$ and σ_i^2 are associated eigenvalues.

Interpretation of Right Singular Vectors

Note that $\mathbf{X}^T\mathbf{X} = \mathbf{B}\mathbf{\Sigma}^T\mathbf{A}^T\mathbf{A}\mathbf{\Sigma}\mathbf{B}^T = \mathbf{B}\mathbf{\Sigma}^T\mathbf{\Sigma}\mathbf{B}^T$.

This is the eigenvector decomposition of the matrix $\mathbf{X}^T\mathbf{X}$, whose (i, j)th element measures the overlap between terms i and j.

Right singular vectors are eigenvectors of $\mathbf{X}^T\mathbf{X}$ and σ_i^2 are associated eigenvalues.

Approximating the Document-Term Matrix

We can obtain a rank k approximation of the document-term matrix $\mathbf{X}_k$ by constructing $\mathbf{X}_k = \mathbf{A}\mathbf{\Sigma}_k\mathbf{B}^T$, where $\mathbf{\Sigma}_k$ is the diagonal matrix formed by replacing $\mathbf{\Sigma}_{ii} = 0$ for $i > k$.

The idea is to keep the "content" dimensions that explain common variation across terms and documents and drop "noise" dimensions that represent idiosyncratic variation.

Often k is selected to explain a fixed portion p of variance in the data. In this case k is the smallest value that satisfies $\sum_{i=1}^{k} \sigma_i^2 / \sum_i \sigma_i^2 \geq p$.

We can then perform the same operations on $\mathbf{X}_k$ as on $\mathbf{X}$, e.g. cosine similarity.

Example

Suppose the document-term matrix is given by

$$\mathbf{X} = \begin{array}{c} \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \end{array} \begin{array}{cccc} \text{car} & \text{automobile} & \text{ship} & \text{boat} \\ \end{array}$$

$$\mathbf{X} = \begin{array}{c|cccc} & \text{car} & \text{automobile} & \text{ship} & \text{boat} \\ d_1 & 10 & 0 & 1 & 0 \\ d_2 & 5 & 5 & 1 & 1 \\ d_3 & 0 & 14 & 0 & 0 \\ d_4 & 0 & 2 & 10 & 5 \\ d_5 & 1 & 0 & 20 & 21 \\ d_6 & 0 & 0 & 2 & 7 \end{array}$$

Matrix of Cosine Similarities

$$
\begin{array}{c}
 \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6
\end{array}
\begin{array}{c}
\begin{array}{cccccc} d_1 & d_2 & d_3 & d_4 & d_5 & d_6 \end{array} \\
\begin{bmatrix}
1 & \cdot & \cdot & \cdot & \cdot & \cdot \\
0.70 & 1 & \cdot & \cdot & \cdot & \cdot \\
0.00 & 0.69 & 1 & \cdot & \cdot & \cdot \\
0.08 & 0.30 & 0.17 & 1 & \cdot & \cdot \\
0.10 & 0.21 & 0.00 & 0.92 & 1 & \cdot \\
0.02 & 0.17 & 0.00 & 0.66 & 0.88 & 1
\end{bmatrix}
\end{array}
$$

SVD

The singular values are $(31.61, 15.14, 10.90, 5.03)$.

$$\mathbf{A} = \begin{bmatrix} 0.0381 & 0.1435 & -0.8931 & -0.02301 & 0.3765 & 0.1947 \\ 0.0586 & 0.3888 & -0.3392 & 0.0856 & -0.7868 & -0.3222 \\ 0.0168 & 0.9000 & 0.2848 & 0.0808 & 0.3173 & 0.0359 \\ 0.3367 & 0.1047 & 0.0631 & -0.7069 & -0.2542 & 0.5542 \\ 0.9169 & -0.0792 & 0.0215 & 0.1021 & 0.1688 & -0.3368 \\ 0.2014 & -0.0298 & 0.0404 & 0.6894 & -0.2126 & 0.6605 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0.0503 & 0.2178 & -0.9728 & 0.0595 \\ 0.0380 & 0.9739 & 0.2218 & 0.0291 \\ 0.7024 & -0.0043 & -0.0081 & -0.7116 \\ 0.7088 & -0.0634 & 0.0653 & 0.6994 \end{bmatrix}$$

Rank-2 Approximation

$$\mathbf{X}_2 = \begin{array}{c|cccc|} & \text{car} & \text{automobile} & \text{ship} & \text{boat} \\ d_1 & 0.5343 & 2.1632 & 0.8378 & 0.7169 \\ d_2 & 1.3765 & 5.8077 & 1.2765 & 0.9399 \\ d_3 & 2.9969 & 13.2992 & 0.3153 & 0.4877 \\ d_4 & 0.8817 & 1.9509 & 7.4715 & 7.4456 \\ d_5 & 1.1978 & 0.0670 & 20.3682 & 20.6246 \\ d_6 & 0.2219 & 0.1988 & 4.4748 & 4.5423 \end{array}$$

Matrix of Cosine Similarities

$$\begin{array}{c|cccccc} & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 \\ \hline d_1 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ d_2 & 0.97 & 1 & \cdot & \cdot & \cdot & \cdot \\ d_3 & 0.91 & 0.97 & 1 & \cdot & \cdot & \cdot \\ d_4 & 0.60 & 0.43 & 0.23 & 1 & \cdot & \cdot \\ d_5 & 0.45 & 0.26 & 0.05 & 0.98 & 1 & \cdot \\ d_6 & 0.47 & 0.29 & 0.07 & 0.98 & 0.99 & 1 \end{array}$$

Probabilistic Mixed-Membership Model

As with k-means, LSA provides a useful tool for data exploration, but its statistical foundations are unclear.

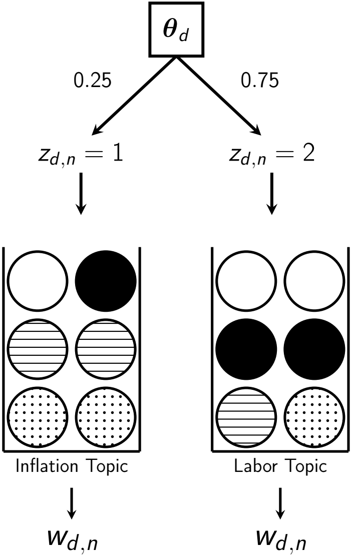
We now turn to exploring probabilistic mixed-membership models. An initial contribution is the probabilistic LSI model of Hofmann (1999).

Instead of assigning each document to a topic, we can assign each word in each document to a topic.

Let $z_{d,n}$ be the topic assignment of the nth word in document d. Suppose it is drawn from a document-specific vector of mixing weights $\boldsymbol{\theta}_d \in \mathbb{R}^K$.

As before, words in a document are conditionally independent.

Mixed-Membership Model for Document



Estimation

The likelihood function for this model is

$$\prod_{d} \Pr\left[\,\boldsymbol{\theta}_d\,\right] \prod_{n} \sum_{z_{d,n}} \Pr\left[\, w_{d,n} \,\middle|\, \boldsymbol{\beta}_{z_{d,n}} \right] \Pr\left[\, z_{d,n} \mid \boldsymbol{\theta}_d \,\right].$$

We can fit the parameters by EM, but:

1. Large number of parameters $KV + DK$, prone to over-fitting.
2. No generative model for $\boldsymbol{\theta}_d$.

We will come back to these issues in the next lecture.

Conclusion

Key ideas from this lecture:

1. The goal of unsupervised learning is to estimate latent structure in observations. Useful for dimensionality reduction.
2. Ad hoc data exploration tools are a good starting point, but probabilistic models are more flexible and statistically well-founded.
3. EM algorithm for likelihood functions with latent variables.
4. Mixture versus mixed-membership models.