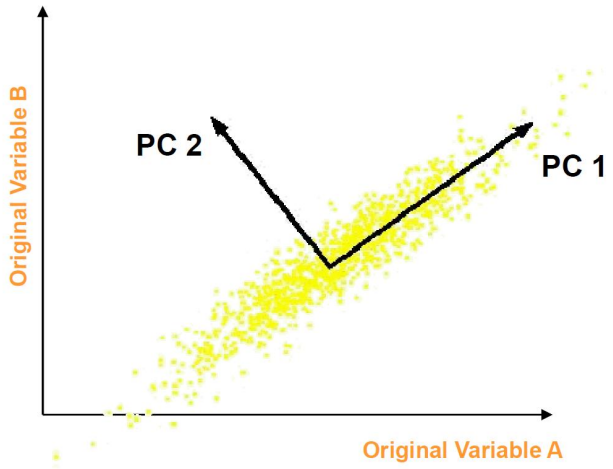
10-701 Recitation 10

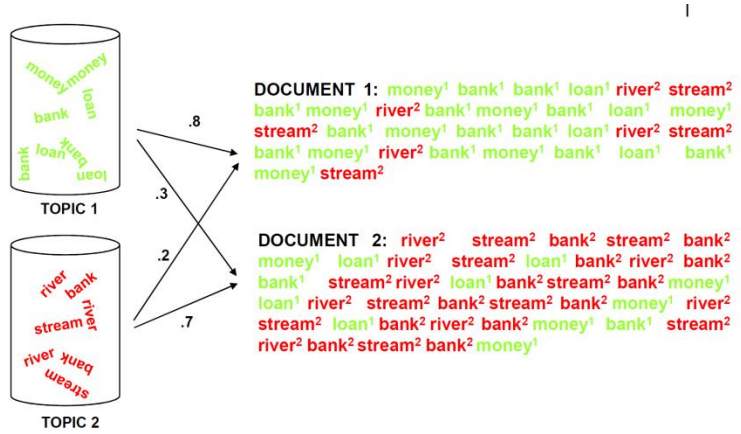
PCA and Topic Models

Latent Space Methods

- PCA, Topic Models are latent space methods
  - Unsupervised
  - Reduce data to fewer dimensions
  - Easier to visualize and understand



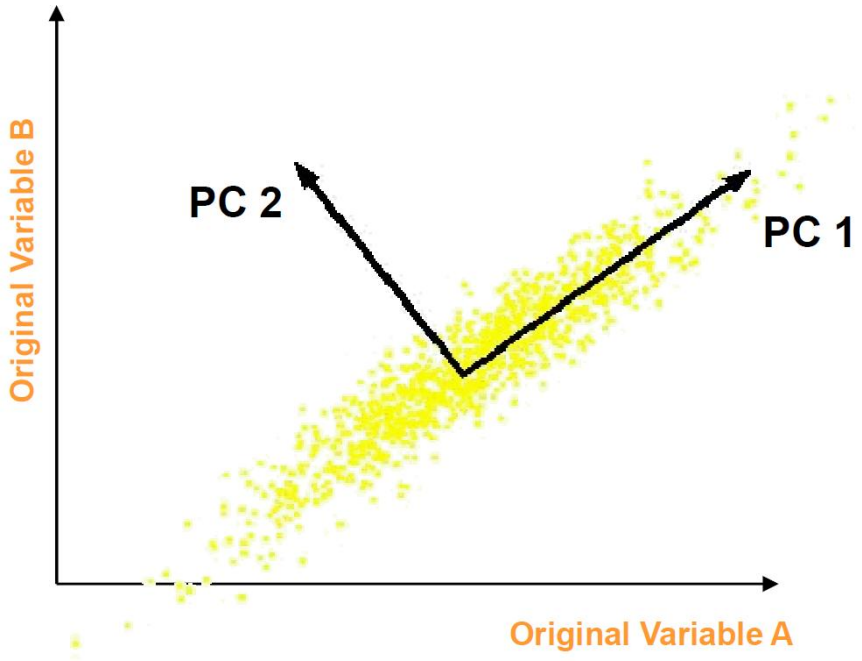
PCA



Topic Model

Principal Component Analysis

- Key idea: find vectors that capture most of the variability in the data X
  - These vectors are the principal components:



Principal Component Analysis

- Let's say the data X has n rows and p columns
  - Every row x_i is a p-dimensional data point
  - We assume that X has zero mean

- What quantity represents the variability of X?
  - Answer: The p-by-p covariance matrix X^TX, which is equal to

$$X^\top X = \sum_{i=1}^{n} x_i^\top x_i \quad \text{(sum of outer products)}$$

Principal Component Analysis

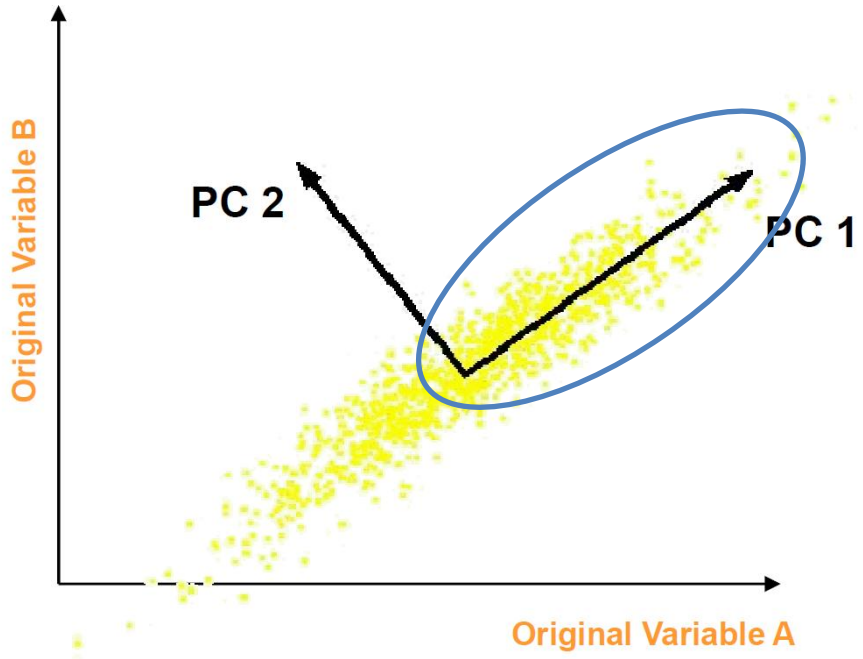
- What does it mean for a vector to “capture” variability in the data?

- Let $C = X^TX$ be the covariance matrix. We want a (unit length) vector u that maximizes u^TCu
  - Why do we want this?
  - Intuition: let $v = Cu$, so we are maximizing u^Tv
  - u^Tv is high when
    - The magnitude of v is large
    - The angle between u and v is small
  - In other words, we want to find u such that
    - C makes u longer
    - C doesn’t change the angle of u

- u^TCu is maximized when u is the principal eigenvector of C
  - Hence $Cu = \lambda u$ where λ is the principal (largest) eigenvalue of C
  - Graphically, the principal eigenvector gives the direction of highest variability

Principal Component Analysis

- Graphically, the principal eigenvector gives the direction of highest variability:



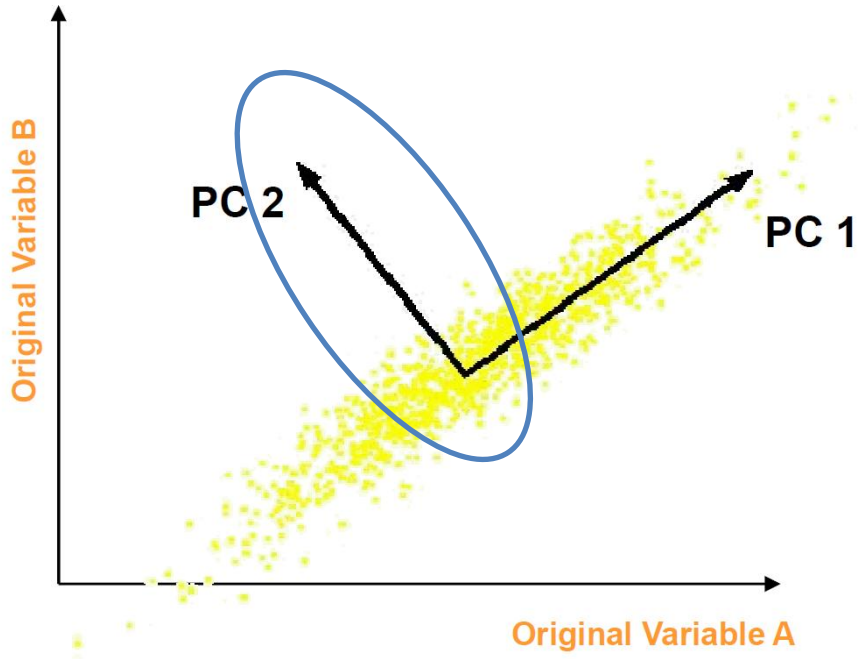
Principal Component Analysis

- We found the first principal component u_1 (the principal eigenvector).
  - How do we find the other principal components?

- Again, find a unit-length u that maximizes $u^T Cu$, such that u is perpendicular to u_1
  - The solution is the second eigenvector u_2
  - Next, maximize $u^T Cu$ s.t. u perpendicular to u_1 and u_2, which gives the third eigenvector u_3
  - And so on…

Principal Component Analysis

- We maximize $u^T C u$ s.t. u perpendicular to u_1:



Finding the eigenvectors

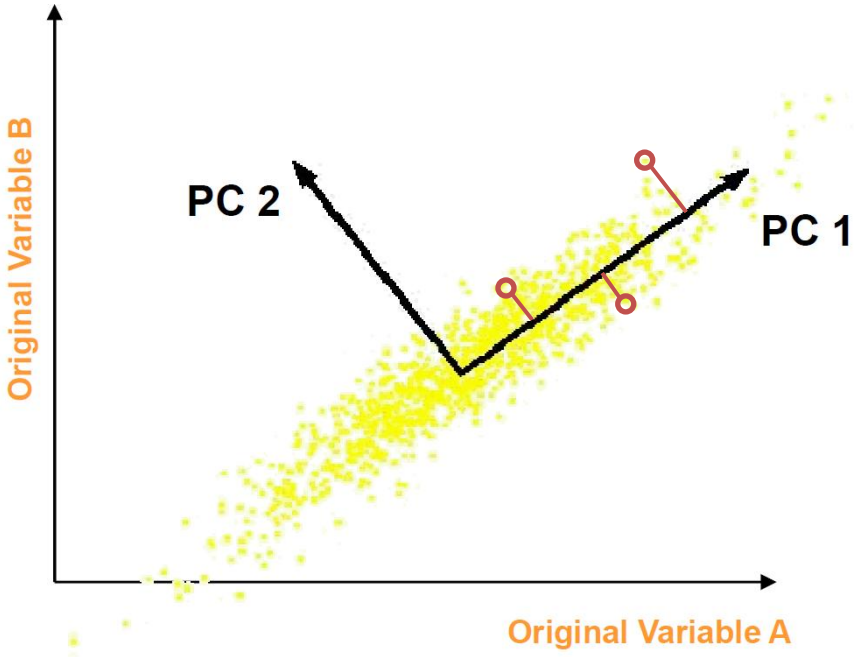
- MATLAB code to find the top k eigenvectors:
  - [V,D] = eigs(X'*X,k);

- V is p-by-k, D is k-by-k
- V contains top k eigenvectors as columns
- D contains top k eigenvalues on its diagonal

Principal Component Analysis

- So far we've talked about eigenvectors of C
  - What's the connection with a latent space?

- Let's project the data points onto the 1st PC/eigenvector
  - Notice how the points didn't move much



Principal Component Analysis

- If we pick the top k PCs and project the data X onto them, we get a lower dimensional latent space representation of the data
  - Note that k < p (original data dimensionality)
  - By picking the top k PCs, we ensure the latent space distorts the data as little as possible

Principal Component Analysis

- How do we project the data on the top k PCs?
- MATLAB code:
  - [V,D] = eigs(X'*X,k);
  - W = X*V;
- W is n-by-k, and is the projection of X onto the top k PCs
  - W_{ij} represents how much X_i depends on the j-th PC

Principal Component Analysis

- The projection W can be thought of as a “compressed” version of X
  - In some cases, we can even interpret the columns of W as “concepts”
- How do we reconstruct the data from W?
- MATLAB code:
  - [V,D] = eigs(X’*X,k);
  - W = X*V;
  - Y = W*V’;
- Y is n-by-p, and is the reconstruction of X from the top k PCs
  - If the top k PCs capture most of the data variability, then Y will be similar to X

Summary of PCA

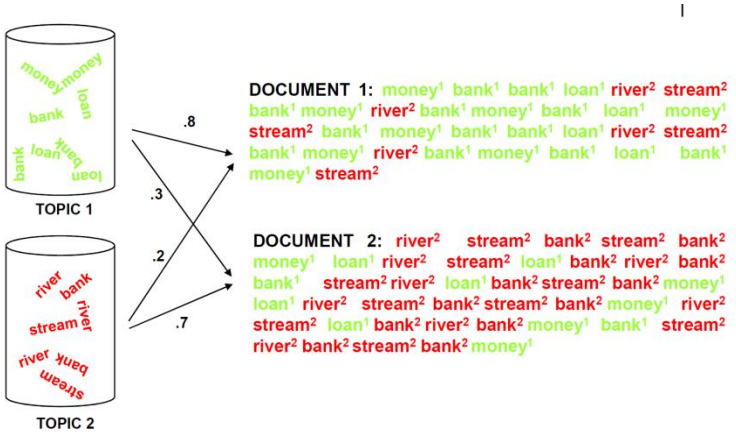
- PCA projects p-dimensional data X onto a k-dimensional latent space W, where $k < p$

- Properties of the latent space W
  - Captures most of the variability in X
  - Easier to visualize and understand than X
  - Less storage than X
  - We can reconstruct X from W with minimal error

Topic Models

- Setting: want to organize documents, represented as (high-dimensional) bags of words

- Key idea: find K topics (collections of words) so we can describe documents as combinations of topics
  - Note that topics may overlap



Topic Models

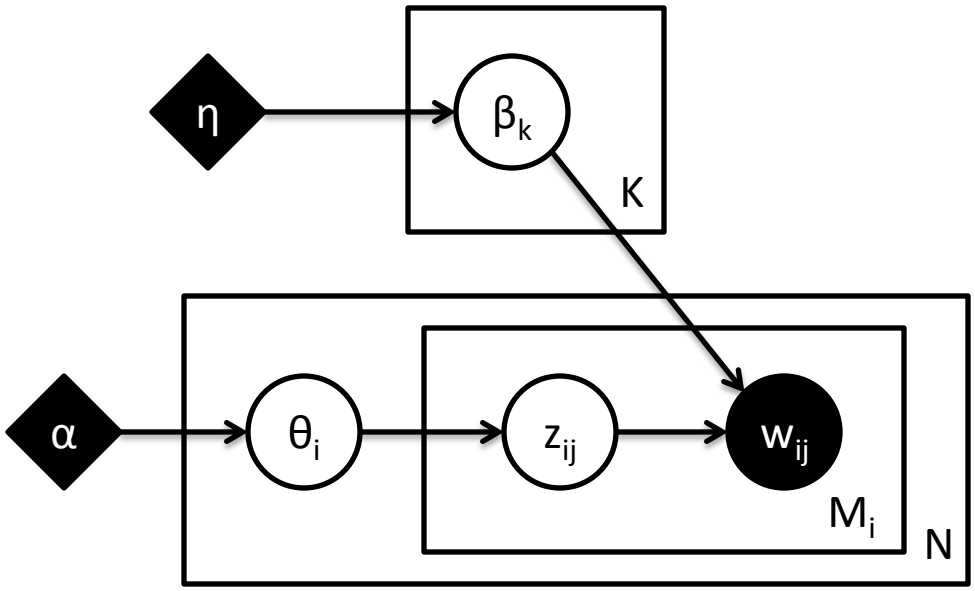
- A Topic Model is a Bayesian generative model
  - Observed data depends on hidden variables, which can depend on other hidden variables, etc.
  - Can be graphically depicted as a Bayes Net
  - You already know another generative model
    - K-Gaussians Mixture Model

Topic Models

- The Topic Model "generative process" describes the model in a compact form:
  - Draw topic vocabularies k = 1…K:
    - $\beta_k \sim \text{Dirichlet}(\eta)$
  - Draw document topic vectors i = 1…N:
    - $\theta_i \sim \text{Dirichlet}(\alpha)$
  - For each document i = 1…N:
    - Draw words j = 1…M_i:
      - $z_{ij} \sim \text{Multinomial}(\theta_i)$ (word-topic indicator)
      - $w_{ij} \sim \text{Multinomial}(\beta_{z_{ij}})$ (the word itself)

Topic Models

- "Graphical Model" illustration:

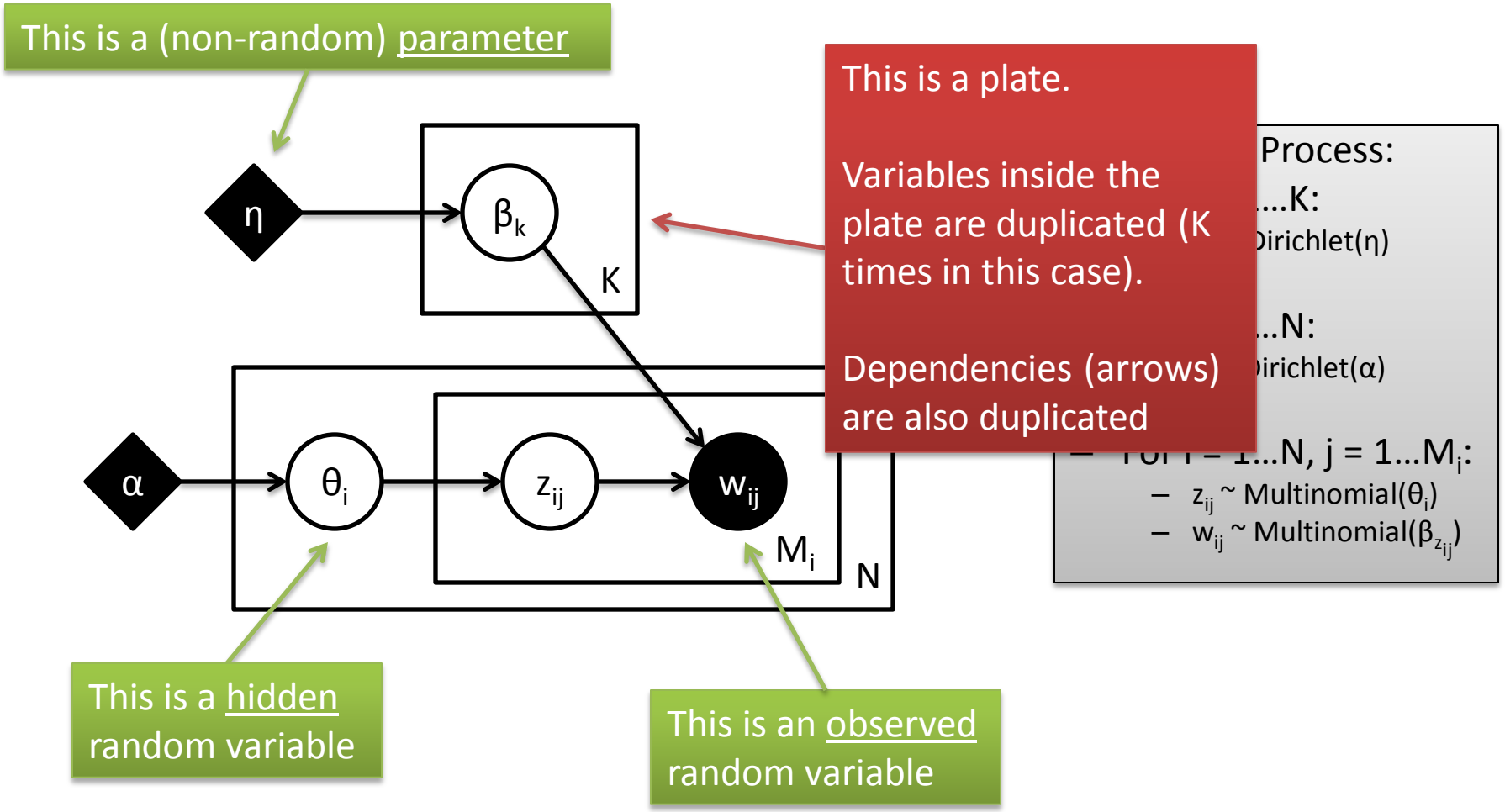


Generative Process:

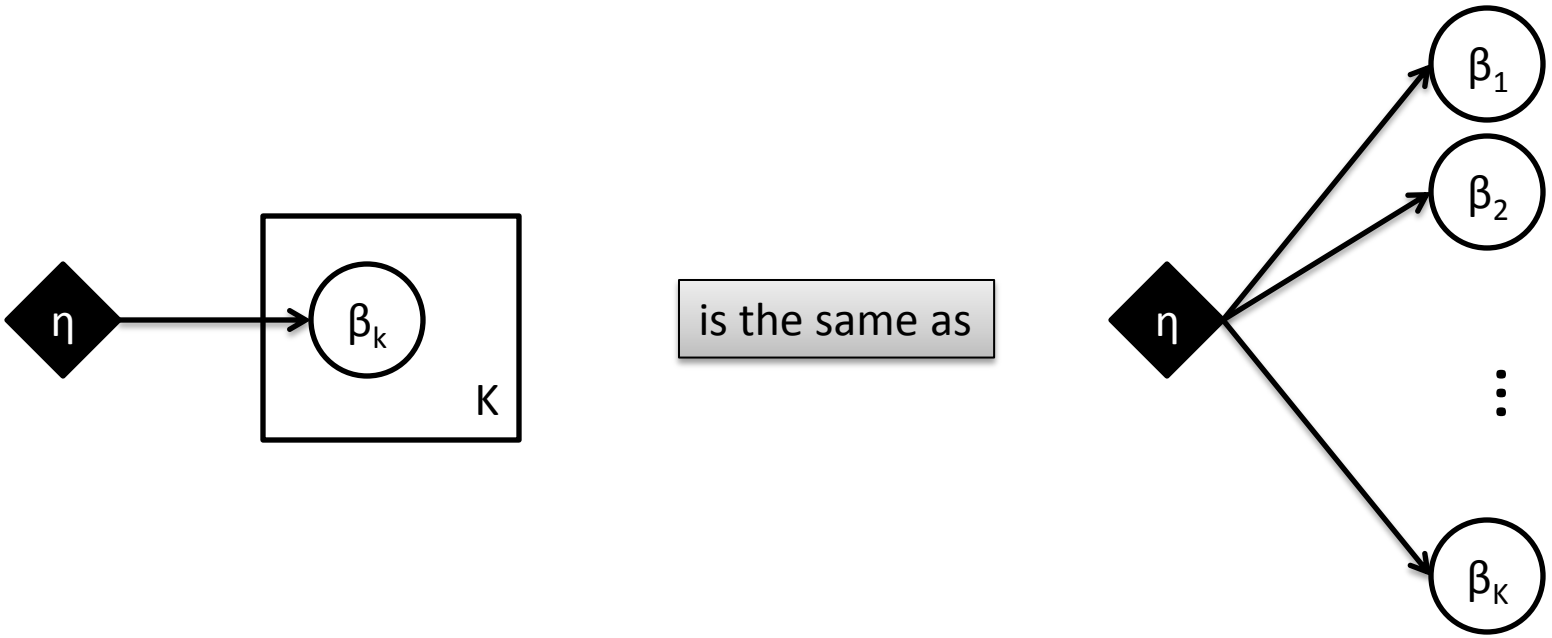
- For k = 1…K:
  - $\beta_k \sim$ Dirichlet(η)
- For i = 1…N:
  - $\theta_i \sim$ Dirichlet(α)
- For i = 1…N, j = 1…M_i:
  - $z_{ij} \sim$ Multinomial(θ_i)
  - $w_{ij} \sim$ Multinomial($\beta_{z_{ij}}$)

Reading Plate Notation

- "Graphical Model" illustration:

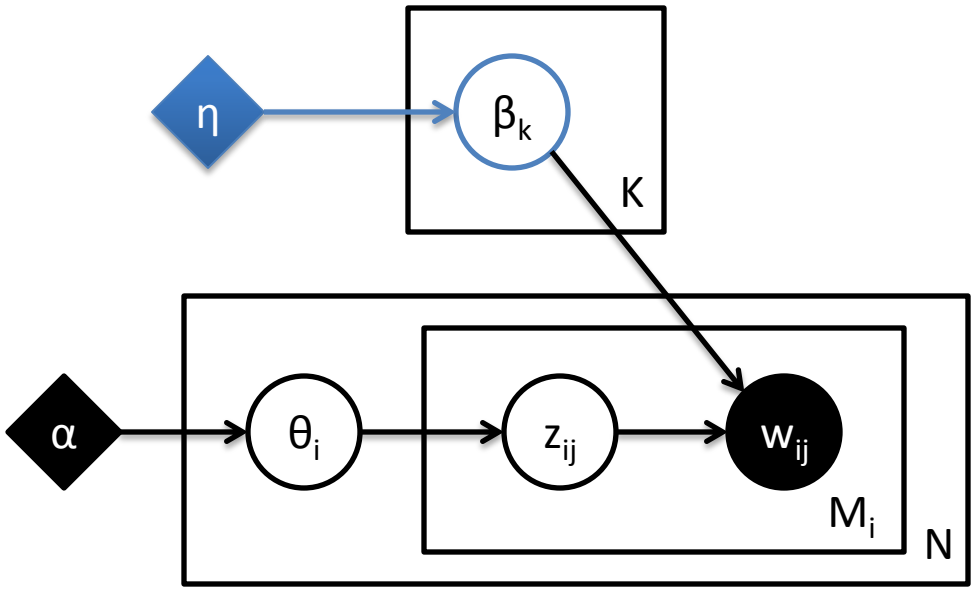


Reading Plate Notation



Correspondence with Gen. Process

- "Graphical Model" illustration:

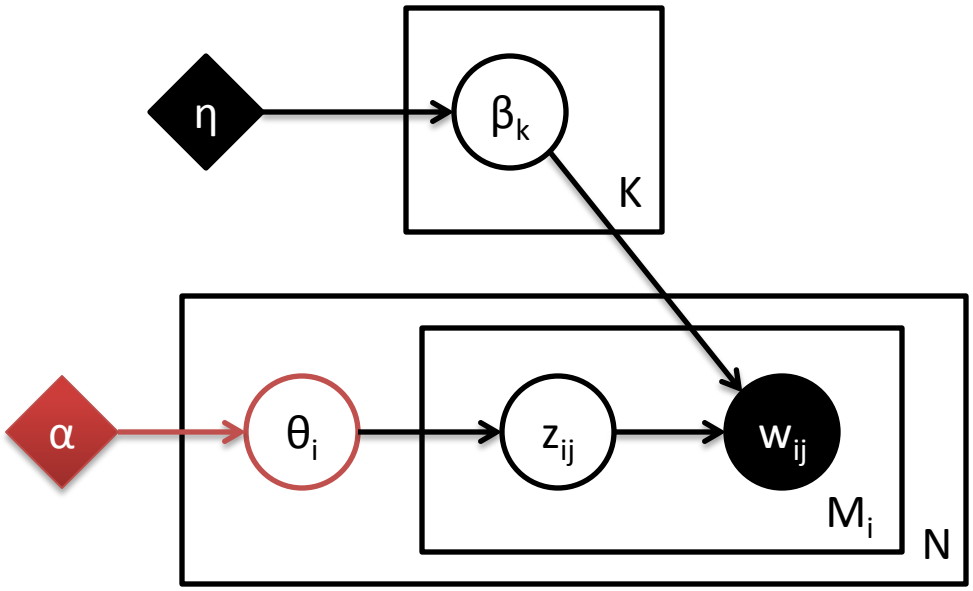


Generative Process:

- For k = 1…K:
  - $\beta_k \sim$ Dirichlet(η)
- For i = 1…N:
  - $\theta_i \sim$ Dirichlet(α)
- For i = 1…N, j = 1…M_i:
  - $z_{ij} \sim$ Multinomial(θ_i)
  - $w_{ij} \sim$ Multinomial($\beta_{z_{ij}}$)

Correspondence with Gen. Process

- "Graphical Model" illustration:

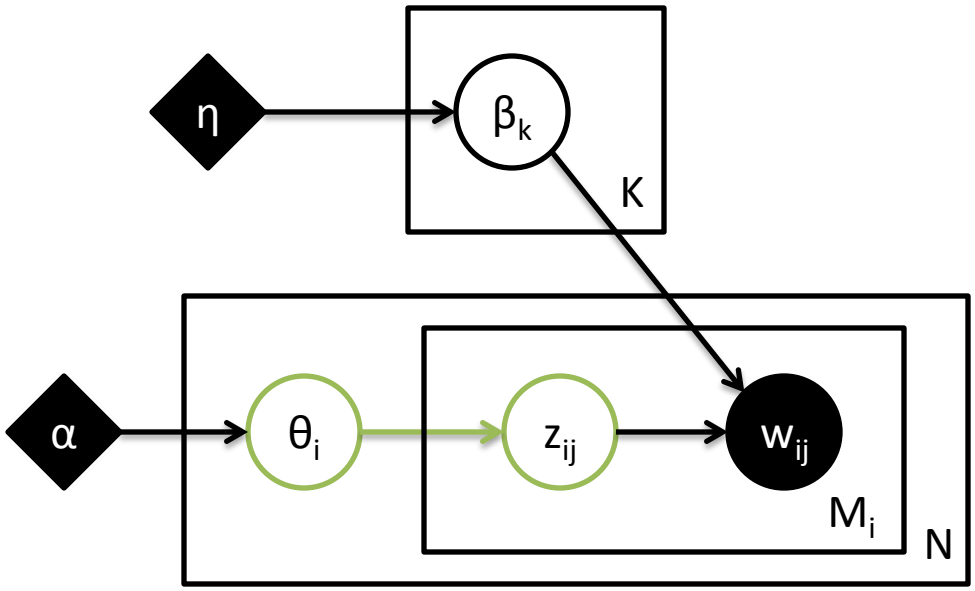


Generative Process:

- For k = 1…K:
  - $\beta_k \sim$ Dirichlet(η)
- For i = 1…N:
  - $\theta_i \sim$ Dirichlet(α)
- For i = 1…N, j = 1…M_i:
  - $z_{ij} \sim$ Multinomial(θ_i)
  - $w_{ij} \sim$ Multinomial($\beta_{z_{ij}}$)

Correspondence with Gen. Process

- "Graphical Model" illustration:

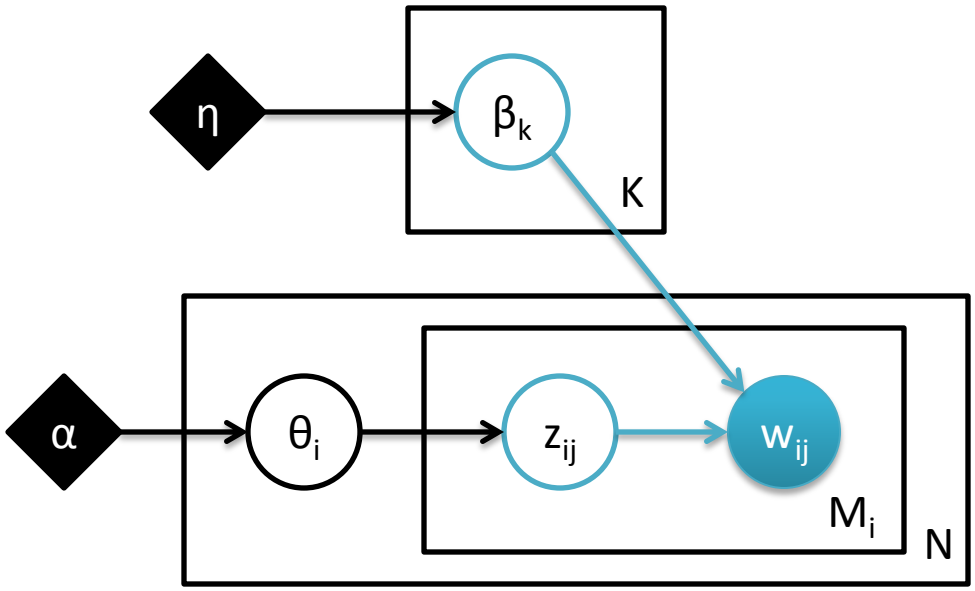


Generative Process:

- For k = 1…K:
  - $\beta_k \sim$ Dirichlet(η)
- For i = 1…N:
  - $\theta_i \sim$ Dirichlet(α)
- For i = 1…N, j = 1…M_i:
  - $z_{ij} \sim$ Multinomial(θ_i)
  - $w_{ij} \sim$ Multinomial($\beta_{z_{ij}}$)

Correspondence with Gen. Process

- "Graphical Model" illustration:



Generative Process:

- For k = 1…K:
  - $\beta_k \sim$ Dirichlet(η)
- For i = 1…N:
  - $\theta_i \sim$ Dirichlet(α)
- For i = 1…N, j = 1…M_i:
  - $z_{ij} \sim$ Multinomial(θ_i)
  - $w_{ij} \sim$ Multinomial($\beta_{z_{ij}}$)

From Generative Process to Inference

- We just saw the topic model generative process and its corresponding graphical model
  - Given just the parameters α and η, we could generate random data from the topic model

- But that's not what we want!
  - We want to represent documents in terms of topics
  - So we need to find the topic vocabularies β and the document topic vectors θ

From Generative Process to Inference

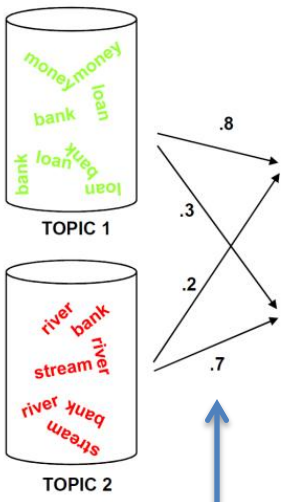
- We find β and θ via probabilistic inference
  - In other words, find the distribution of β and θ conditioned on the observed words w (and parameters α, η)
  - This is the same principle as Viterbi for HMMs and variable elimination for Bayes Nets
  - I won't go into details here, that's for HW5

From Generative Process to Inference

- Topic model inference isn't easy
  - We need to marginalize (sum out) the word-topic indicators z
    - But there are exponentially many settings to all z, so this is infeasible!
  - One popular solution is Gibbs sampling
    - In HW3, you saw Gibbs sampling for K-Gaussians
    - We'll walk you through topic model Gibbs sampling in HW5
  - Naïve EM doesn't work, so people use "variational EM"
    - You don't need to know the details, just be aware of it

Interpreting Topic Models

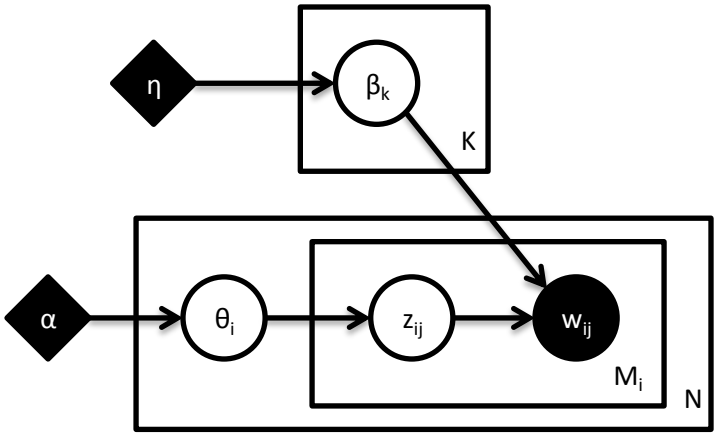
Corresponds to topic vocabularies β



.8
.3
.2
.7

DOCUMENT 1: money[1] bank[1] bank[1] loan[1] river[2] stream[2] bank[1] money[1] river[2] bank[1] money[1] bank[1] loan[1] money[1] stream[2] bank[1] money[1] bank[1] bank[1] loan[1] river[2] stream[2] bank[1] money[1] river[2] bank[1] money[1] bank[1] loan[1] bank[1] money[1] stream[2]

DOCUMENT 2: river[2] stream[2] bank[2] stream[2] bank[2] money[1] loan[1] river[2] stream[2] loan[1] bank[2] river[2] bank[2] bank[1] stream[2] river[2] loan[1] bank[2] stream[2] bank[2] money[1] loan[1] river[2] stream[2] bank[2] stream[2] bank[2] money[1] river[2] stream[2] loan[1] bank[2] river[2] bank[2] money[1] bank[1] stream[2] river[2] bank[2] stream[2] bank[2] money[1]

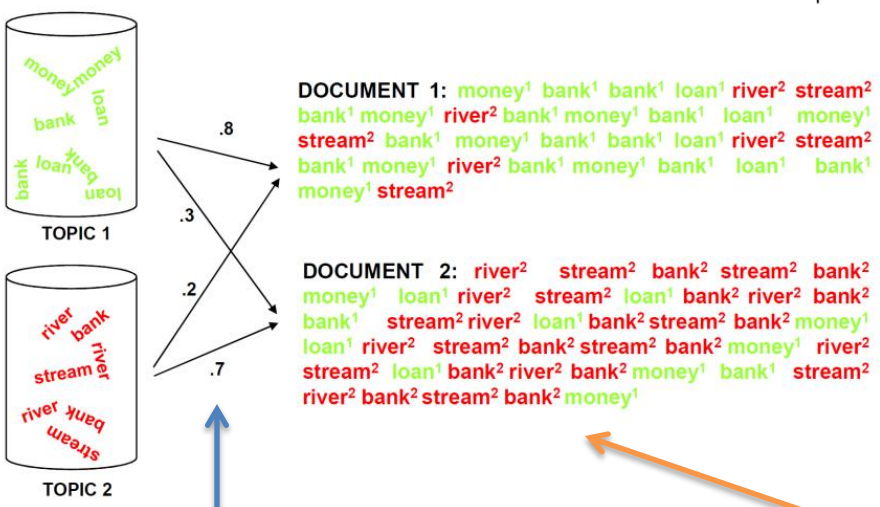


Corresponds to document topic vectors θ

These are document words w. The word colors (red/green) correspond to word-topic indicators z

Interpreting Topic Models

Corresponds to topic vocabularies β



Corresponds to document topic vectors θ

These are document words w. The word colors (red/green) correspond to word-topic indicators z

Interpretation:

Topic 1 is about finance
Topic 2 is about rivers

Document 1 is mostly about finance
Document 2 is mostly about rivers

Although "bank" appears in both topic vocabularies, in doc 1 it probably means a place to store money, whereas in doc 2 it probably means a river bank

Summary of Topic Models

- Bayesian Generative Model that represent documents in a latent space of topics
- Topics contain highly related words, corresponding to some concept
- Use probabilistic inference to find topic vocabularies β and document topic vectors θ, from document text w
  - β_k is a vector of word frequencies for topic k
  - θ_{ik} shows what proportion of document i corresponds to topic k
- Human interpretation is required to make sense of β and θ

Summary of Latent Space Methods

- PCA and Topic Models are unsupervised learning methods
- They reduce high dimensional data to a lower dimensional representation
- The lower dimensional representation is:
  - Easier to interpret
  - Compact (less storage required)