University of BRISTOL

3. Clustering and Topic Modelling

Edwin Simpson

Department of Computer Science,

University of Bristol, UK.

Outline

- 25 minutes:
  - LDA distributions in detail
  - Plate notation
  - Monte Carlo sampling
  - HDP
- 20 minutes: unmarked quiz
- 15 minutes: questions.

Latent Dirichlet Allocation (LDA)

Likelihood

Prior over topic distribution

$$P(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\theta}) = \prod_{i=1}^{N} \{ P(w_i | z_i, \boldsymbol{\beta}) P(z_i | \boldsymbol{\theta}) \} P(\boldsymbol{\theta} | \alpha)$$

Probability of topic for word *i*

- w_i: token at position *i*
- z_i: cluster label for word *i*
- $\boldsymbol{\theta}$: distribution over topics in this document
- α: parameter for the prior over $\boldsymbol{\theta}$
- $\boldsymbol{\beta}$: parameters of word-topic likelihoods

David Blei, Andrew Ng and Michael I. Jordan, Latent Dirichlet Allocation, JMLR (2003)

Latent Dirichlet Allocation (LDA)

Likelihood

Prior over topic distribution

$$P(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\theta}) = \prod_{i=1}^{N} \{P(w_i | z_i, \boldsymbol{\beta}) P(z_i | \boldsymbol{\theta})\} P(\boldsymbol{\theta} | \alpha)$$

Probability of topic for word *i*

- Discrete distribution
- Comparable to the likelihoods in naïve Bayes
- Is LDA generative or discriminative?

David Blei, Andrew Ng and Michael I. Jordan, Latent Dirichlet Allocation, JMLR (2003)

Latent Dirichlet Allocation (LDA)

Prior over topic distribution

Likelihood

$$P(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\theta}) = \prod_{i=1}^{N} \{ \underbrace{P(w_i | z_i, \boldsymbol{\beta})}_{\text{Likelihood}} \underbrace{P(z_i | \boldsymbol{\theta})}_{\text{Probability of topic for word } i} \} \underbrace{P(\boldsymbol{\theta} | \alpha)}_{\text{Prior over topic distribution}}$$

- The topic of each word is drawn from the document's topic distribution
- Discrete distribution

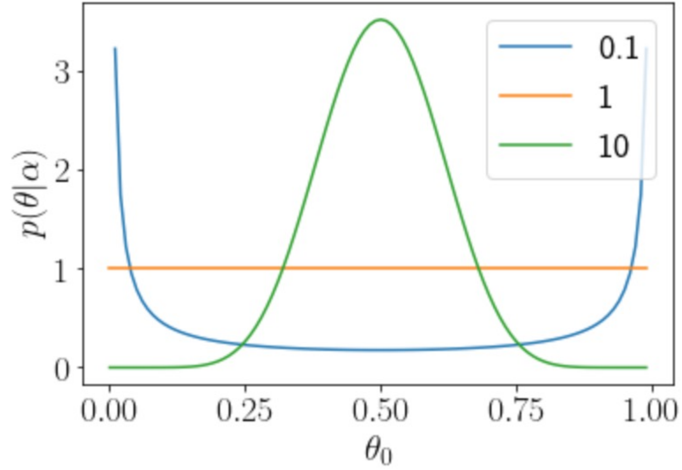
Probability of topic for word *i*

David Blei, Andrew Ng and Michael I. Jordan, Latent Dirichlet Allocation, JMLR (2003)

Latent Dirichlet Allocation (LDA)

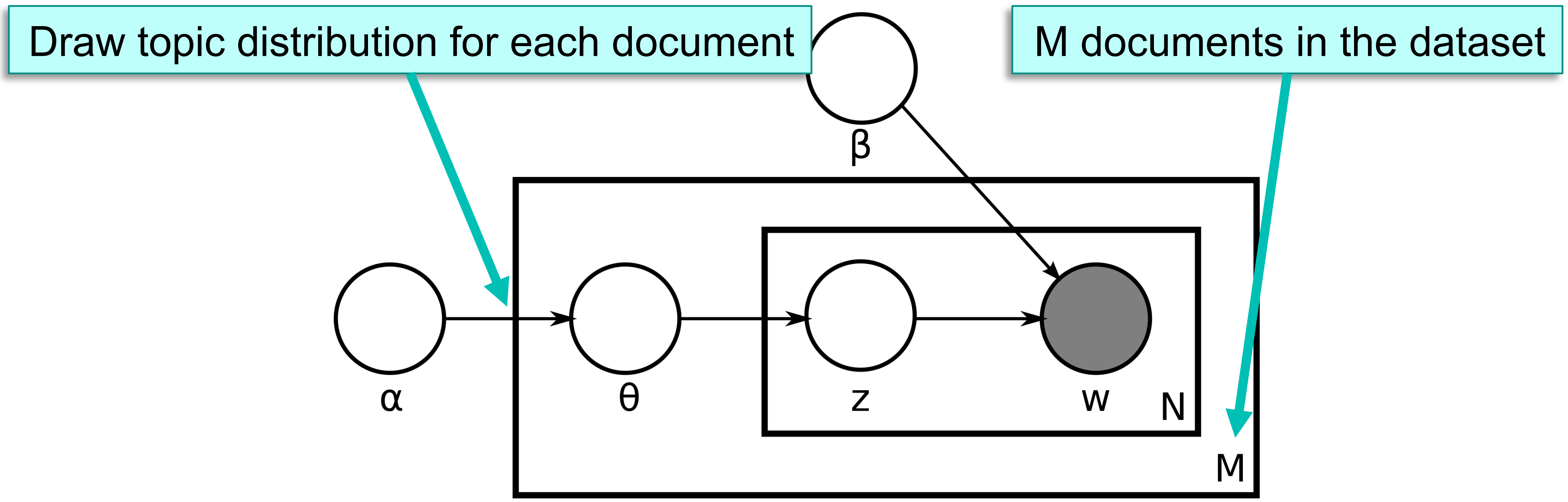
- $\boldsymbol{\theta}$ is a vector of probabilities defining the mixture of topics in the document
- Its prior is a Dirichlet distribution
- For two topics, it could look like this:

Prior over topic distribution



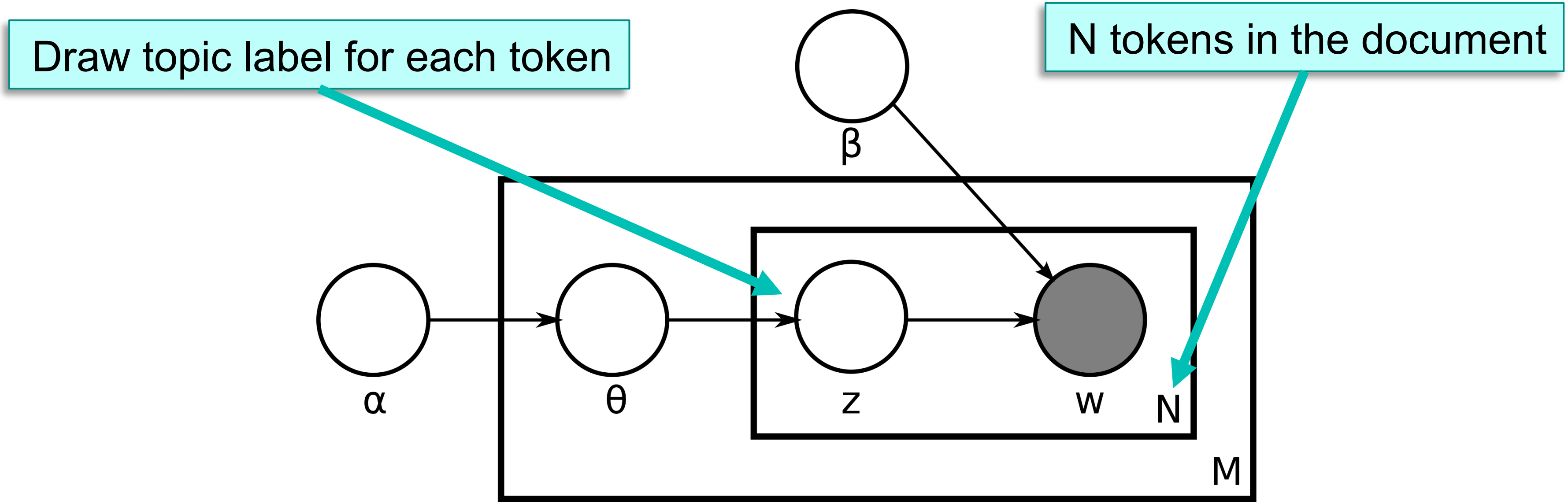
$$\boldsymbol{z}, \boldsymbol{\theta}) = \prod_{i=1}^{N} \{P(w_i|z_i, \boldsymbol{\beta}) P(z_i|\boldsymbol{\theta})\} \, P(\boldsymbol{\theta}|\alpha)$$

Latent Dirichlet Allocation (LDA)



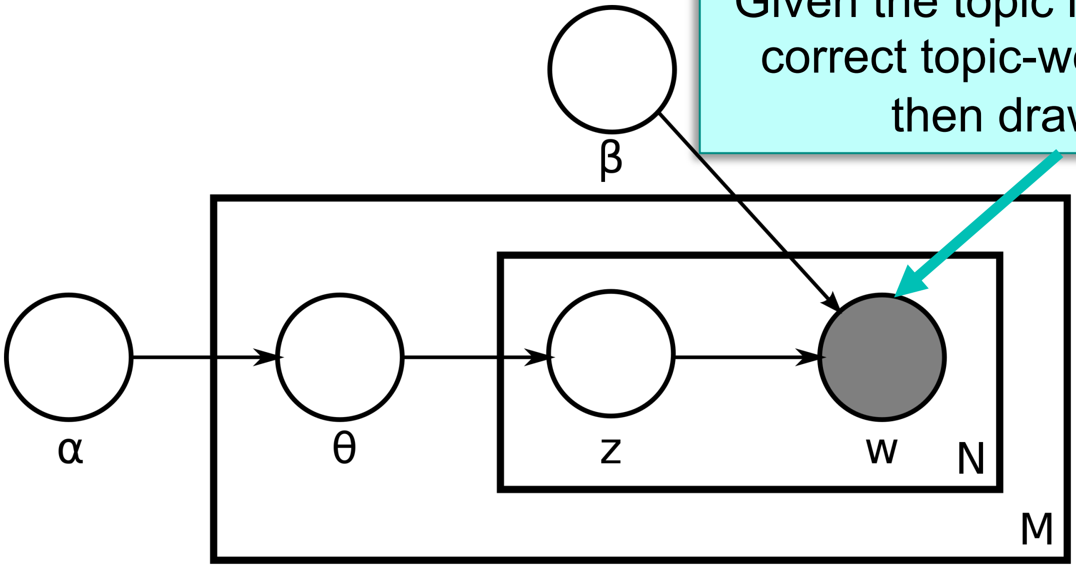
David Blei, Andrew Ng and Michael I. Jordan, Latent Dirichlet Allocation, JMLR (2003)

Latent Dirichlet Allocation (LDA)



David Blei, Andrew Ng and Michael I. Jordan, Latent Dirichlet Allocation, JMLR (2003)

Latent Dirichlet Allocation (LDA)



Given the topic label, select the correct topic-word distribution, then draw the Nth token

David Blei, Andrew Ng and Michael I. Jordan, Latent Dirichlet Allocation, JMLR (2003)

Unsupervised Learning with Variational Inference

- Randomly initialise the distribution of words in each topic, $P(x_i|z_i, \boldsymbol{\beta})$
  - $\boldsymbol{\beta}$ is a parameter we have to learn
  - Initialise it to a random value

Unsupervised Learning with Variational Inference

- Randomly initialise the distribution of words in each topic, $P(x_i|z_i, \boldsymbol{\beta})$
- Randomly initialise the distribution over each document d's topic distribution $P(\boldsymbol{\theta}^{(d)}|\boldsymbol{\alpha})$
  - Prior hyperparameter is $\boldsymbol{\alpha}$, which we set in advance
  - Controls concentration of topics: values < 1 mean few topics per document
  - $\boldsymbol{\alpha}$ will be updated for each document *d* during learning
  - So $\boldsymbol{\alpha}^{(d)}$ is initialised by adding counts to it.

Unsupervised Learning with Variational Inference

- Randomly initialise the distribution of words in each topic, $P(x_i|z_i, \boldsymbol{\beta})$
- Randomly initialise the distribution over each document d's topic distribution $P(\boldsymbol{\theta}^{(\boldsymbol{d})}|\boldsymbol{\alpha})$
- E-step: loop over documents d:
  - Compute the **expectations** of z_i given current distributions of words for all topics
    - $P\left(z_i \middle| \boldsymbol{\beta}, \theta^{(d)}, x_i\right) \propto P(x_i|z_i, \boldsymbol{\beta})P(z_i|\boldsymbol{\theta}^{(\boldsymbol{d})})$

Unsupervised Learning with Variational Inference

- Randomly initialise the distribution of words in each topic, $P(x_i|z_i, \boldsymbol{\beta})$
- Randomly initialise the distribution over each document d's topic distribution $P(\boldsymbol{\theta}^{(d)}|\boldsymbol{\alpha})$
- E-step: loop over documents d:
  - Compute the **expectations** of z_i given current distributions of words for all topics
  - Compute the counts of topics in document d given expectations of z_i for all words *I*
    - Sum up expected probabilities of z_i over all tokens in the document

Unsupervised Learning with Variational Inference

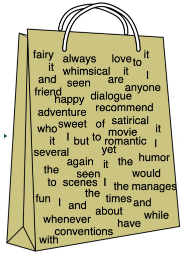
- Initialise
- E-step
- M-step:
  - Compute maximum likelihood estimates of the per-topic word distributions, $\boldsymbol{\beta}_{z_i} = P(x_i|z_i)$, using current expectations of z_i
    - ➢ For each word in the vocabulary, find all occurrences of the word, then sum up the expected probabilities of z_i
    - ➢ Thereby count how many times that word occurred in each topic

Unsupervised Learning with Variational Inference

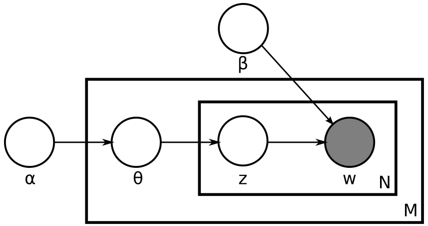
- Initialise
- E-step
- M-step:
  - Compute maximum likelihood estimates of the per-topic word distributions, $\boldsymbol{\beta}_{z_i} = P(x_i|z_i)$, using current expectations of z_i
  - Compute maximum likelihood estimates of the distribution $P(\boldsymbol{\theta}^{(\boldsymbol{d})})$, using the current counts of topics in each document
    - Sum up the expected probabilities of z_i over the words in document *d*

Machine Learning Methods

- It's useful to separate several different things:



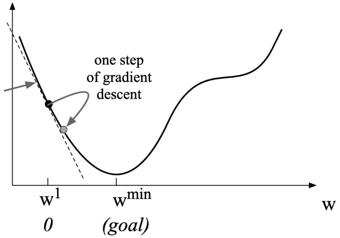
REPRESENTATION
(features)



ARCHITECTURE

$$L(\boldsymbol{\theta}; y)$$

LEARNING OBJECTIVE
e.g., maximum likelihood, max. marginal likelihood



LEARNING ALGORITHM
e.g., variational inference, stochastic gradient descent

Monte Carlo Sampling

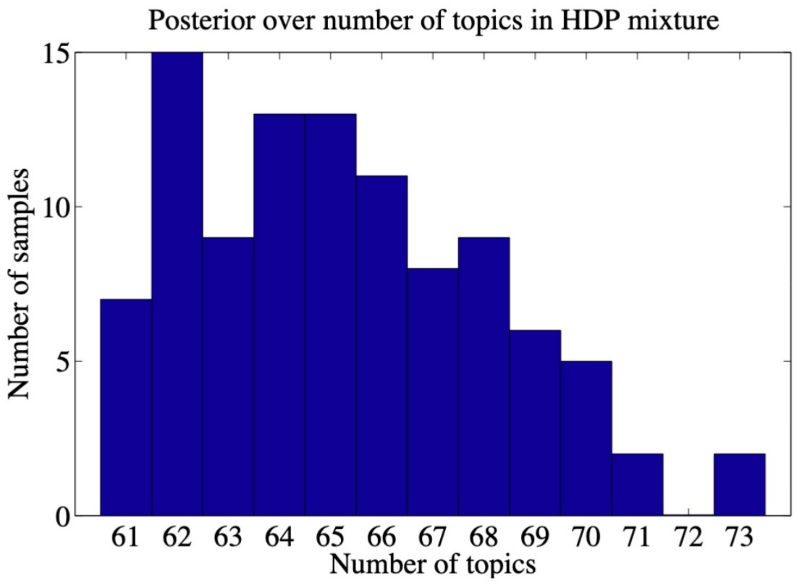
- A technique for estimating posterior probability distributions
- Say you have a coin, and you don't know the probability of heads, *p*
- How would you estimate *p*?
- Throw the coin a number of times and observe how often heads occurs → This is sampling

Monte Carlo Sampling

- Often, we cannot compute the probability distribution over a single variable in closed form
- E.g., expected topic distribution $\boldsymbol{\theta}^{(d)}$ for document *d*
- But we can *sample* from the posterior distribution
- Use a pseudo-random number generator to random values, then pass them through a function to sample parameters such as $\boldsymbol{\theta}^{(d)}$

Hierarchical Dirichlet Process (HDP)

- Can learn using variational inference of Monte Carlo sampling
- Histogram:
  - X-axis = number of active topics *k* where the sum of $z_i = k$ over all words was > 0
  - Y-axis = number of samples for each number of active topics
  - Shape gives us a distribution over the number of active topics



Teh, Y. W., et al. (2006). Hierarchical dirichlet processes. *Journal of the american statistical association*

Quiz

- 24 hours in which you can start the quiz
- Once you start, you have 2 hours
- No backtracking, random question order…
- The unmarked quizzes are examples of the kind of questions in the quiz.
- Link appears on Blackboard in the sidebar under “summative quiz”.