# L101: Machine Learning for Language Processing

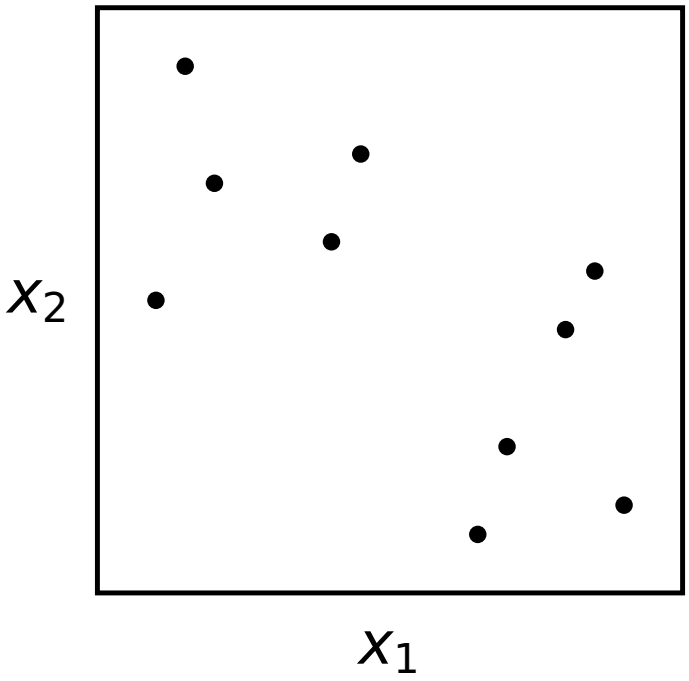
Lecture 5

Guy Emerson

# Today's Lecture

- Unsupervised Learning
  - Word Sense Induction
  - Topic Discovery
- K-Means Clustering
- Latent Dirichlet Allocation
- Approximate Inference

# Unsupervised Learning






In unsupervised learning, input points are unlabelled (no outputs given).

For these points, we might say that the points can be grouped into two clusters (five points in top left, and five in bottom right).

# Word Senses

There was even closing drama when Shelford missed a penalty, and a chance to save the game, with the last kick of the match.

Micro-routes in the Duddon are no match, after all, for a route on any of the limestone crags in Yorkshire or Derbyshire.



Examples from the British National Corpus (BNC) `http://www.natcorp.ox.ac.uk/`

A word sense is an abstraction over particular usages of the word.

We might want to say that the above examples have different senses. How should we define the set of senses?

# Word Senses

- a thin piece of wood, ignites with friction
- a formal contest
- a burning piece of wood
- an exact duplicate
- the score needed to win
- a good matrimonial prospect
- a person of equal standing
- a pair of people who live together
- something that harmonizes



In the supervised task of Word Sense Disambiguation, we have a inventory of senses from a lexical resource. The above senses are from WordNet https://wordnet.princeton.edu

However, we may not want the same set of senses for every task. Further reading:

Kilgarriff (2007) “Word Senses” http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.659.8359&rep=rep1&type=pdf#page=49

Erk et al. (2013) “Measuring Word Meaning in Context” https://www.mitpressjournals.org/doi/full/10.1162/COLI_a_00142

# Word Senses

| | | |
|---|---|---|
| ... the last kick of the | match. | It was entertaining ... |
| ... the Duddon are no | match, | after all, for a route ... |
| ... first or second round | matches | of any consequence ... |
| ... Tried soaking the | matches | in paint, he wrote, ... |
| ... is very much a | match | for Berowne; this is ... |
| ... to win and the | match | is therefore ... |
| ... to lose you the | match | even though no ... |
| ... of an elimination | match | is fought. If this ... |
| ... needed to watch the | match, | needed a diversion ... |
| ... drop in a burning | match. | The plastic of the ... |



In the unsupervised task of word sense discovery, the sense inventory is not given to us. Potentially relevant contextual features are highlighted (but it would be difficult to discover senses without a much larger set of examples).

# Topics

This dissertation describes the measurement of angular diameters of compact radio sources by the technique of interplanetary scintillation. The design, construction and testing of a four acre radio aerial functioning at a frequency of 81.5 MHz is described, and its operation during a survey of the sky.

The stunning array of features and functions exhibited by proteins in nature should convince most scientists of the power of evolutionary design processes. Natural selection acting on populations over long periods of time has generated a vast number of proteins ideally suited to their biological functions.



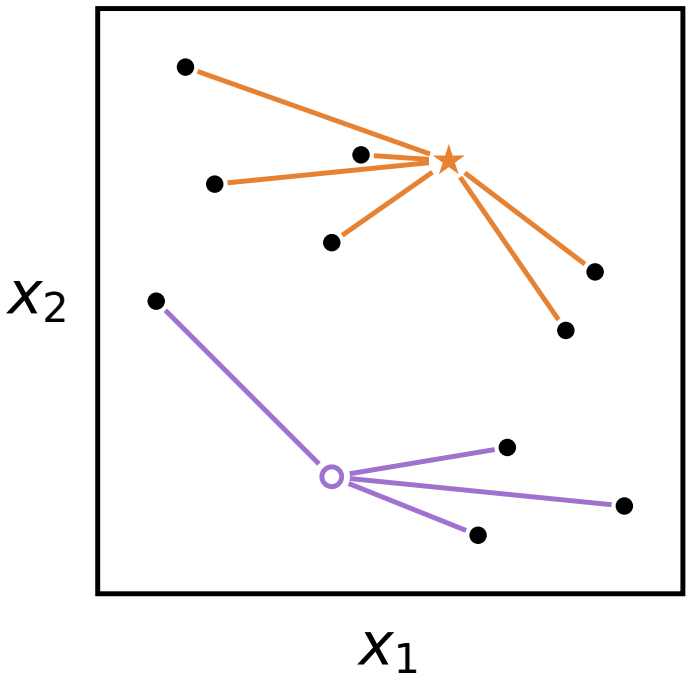
As described in the first lecture, topic classification involves labelling documents with topics. It is a supervised task, where we have access to manually labelled documents.

Topic discovery involves identifying groups of documents that share the same topic. Some potentially useful words and phrases are highlighted, for astronomy and evolution.

The first text is the beginning of Jocelyn Bell Burnell's PhD dissertation, *The measurement of radio source diameters using a diffraction method*, in which she discovered radio pulsars.

The second text is the beginning of a paper by Nobel laureate Frances Arnold, *Design by directed evolution*, in which she describes a novel method of protein engineering.

# K-Means Clustering



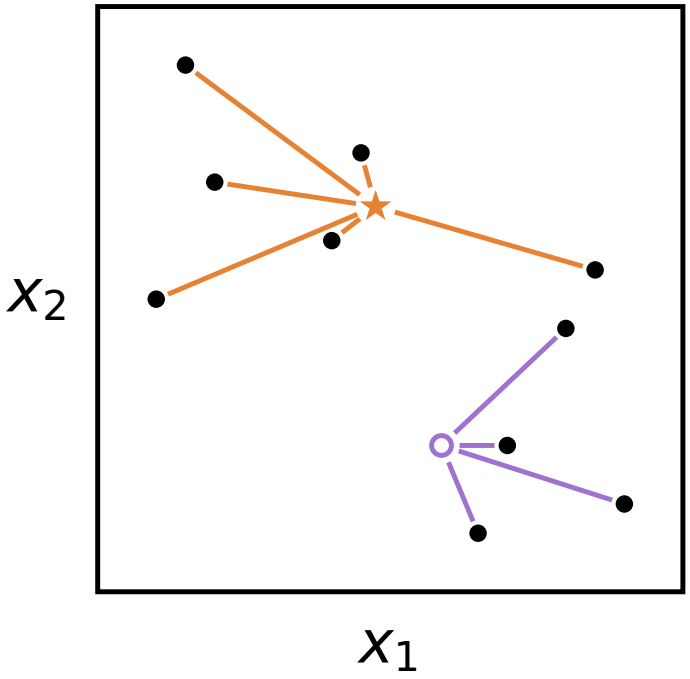

1. For each point, find closest cluster
2. For each cluster, find mean point



We specify some number of clusters $k$ (this is a hyperparameter). Each cluster is represented by a mean point. – hence the name.

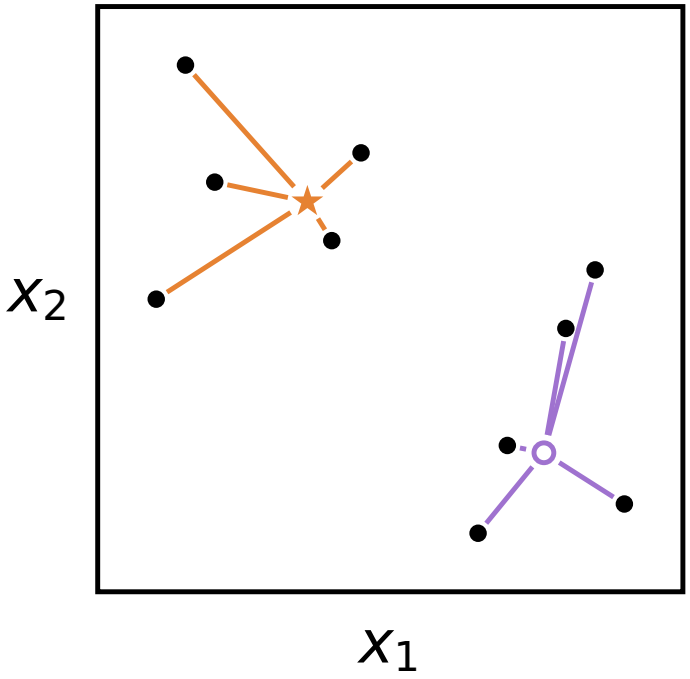
The simplest way to train the model is by initialising the clusters randomly, and then iteratively updating them (using the two steps written above).

# K-Means Clustering



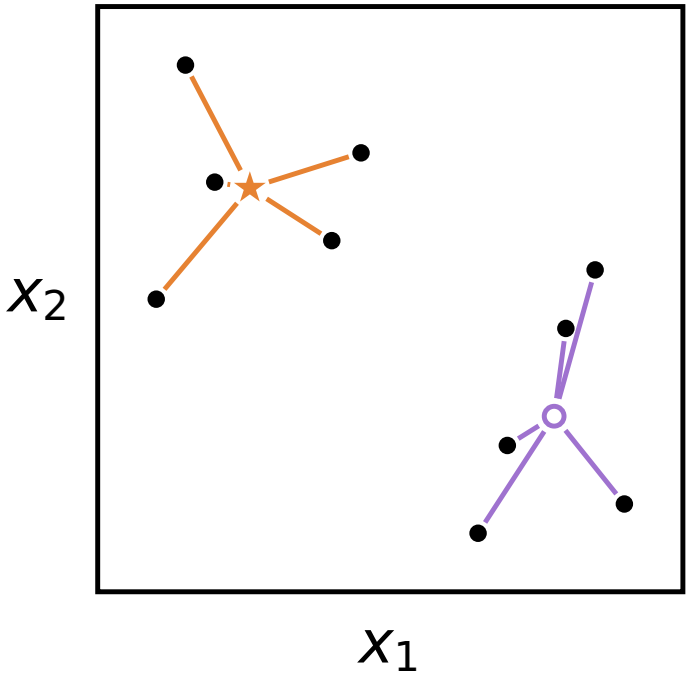

1. For each point, find closest cluster
2. For each cluster, find mean point

# K-Means Clustering



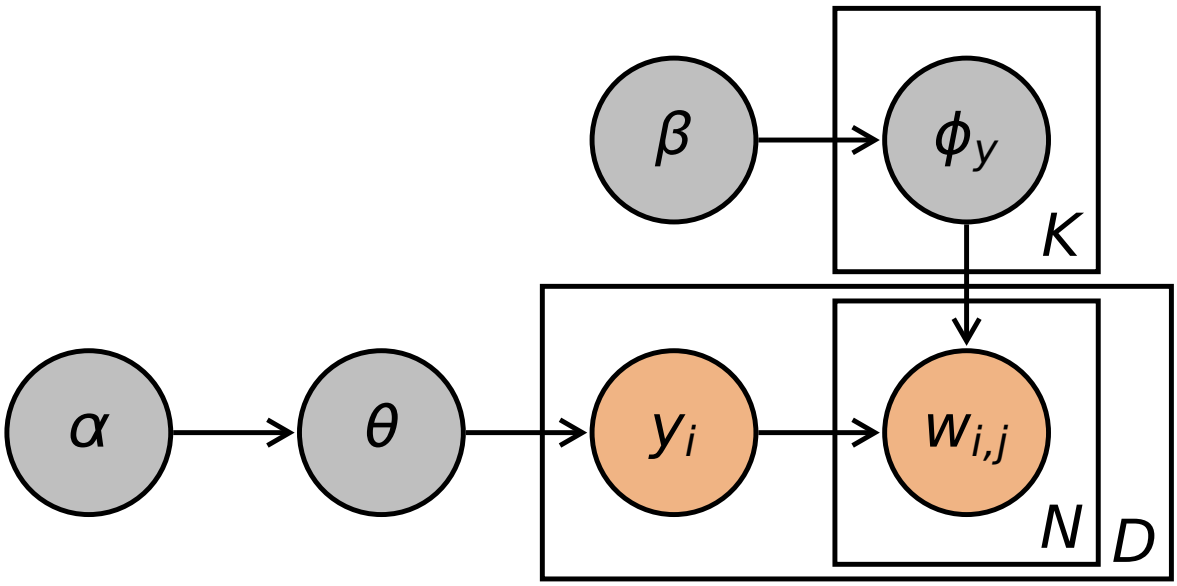

1. For each point, find closest cluster
2. For each cluster, find mean point

# K-Means Clustering




1. For each point, find closest cluster
2. For each cluster, find mean point

# Recap: Multinomial Naive Bayes




## Bayesian view of smoothing hyperparameters: Dirichlet prior



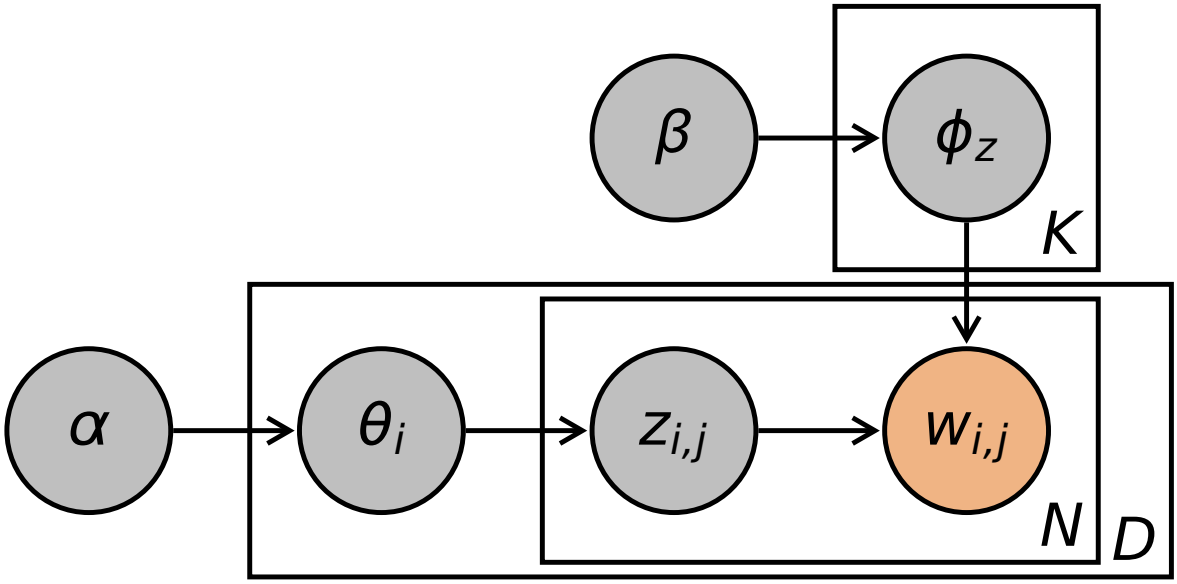
Multinomial Naive Bayes is given here as a probabilistic graphical model. Each rectangle (called a "plate") denotes repetition of nodes, with the number of repeitions given in the bottom right.

In the bottom right, we have the class labels $y$ and the words $w$. The nodes are repeated once for each document $i$ (out of $D$ documents) and the word nodes are repeated once for each token $j$ (out of $N$ tokens).

We have parameters $\theta(y) = P(y)$ and $\phi_y(w) = P(w|y)$, where there are $K$ different possible values for $y$.

Estimating the parameters without smoothing is the maximum likelihood estimate (MLE) for the parameters. Estimating the parameters with smoothing is the maximum a posteriori (MAP) estimate for the parameters, under a Dirichlet prior. The smoothing hyperparameter corresponds to the Dirichlet concentration parameter. A Dirichlet distribution may seem quite abstract (a distribution over distributions) but it was basically invented so that it corresponds to smoothing.

# Latent Dirichlet Allocation

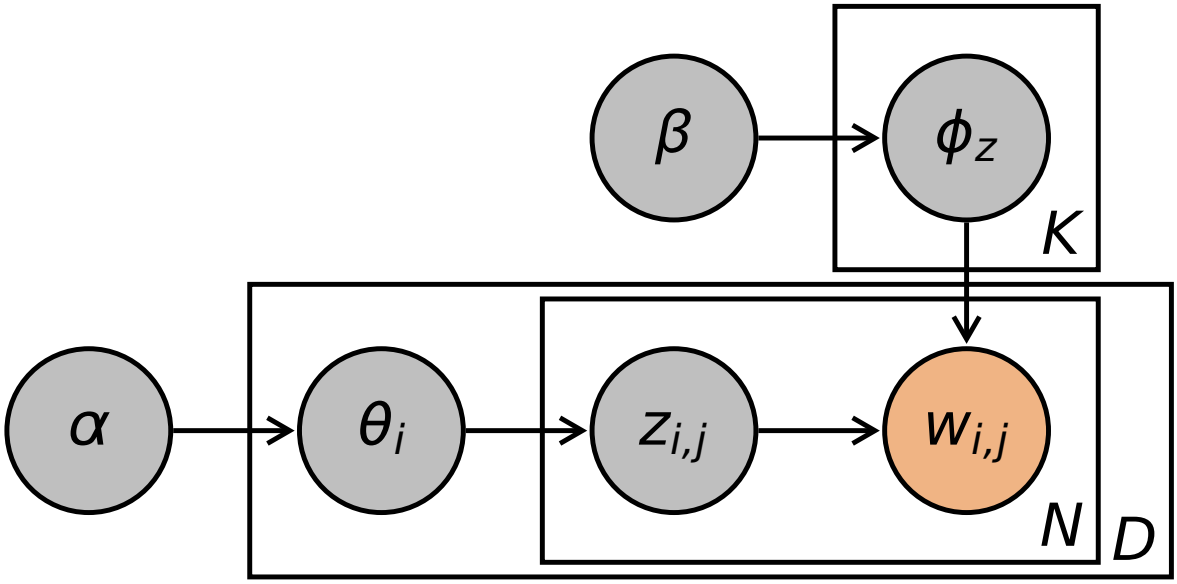





Latent Dirichlet Allocation has a couple of differences compared to Naive Bayes. Firstly, it is unsupervised, so we don't observe topics (labels). Secondly, each document has a distribution over topics. Thirdly, each token in a document may have a different topic.

The topics $z_{i,j}$ are *latent variables* – they are unobserved variables, and they are local variables (at the token level).

# Latent Dirichlet Allocation




$$\prod_z P(\phi_z \mid \beta) \prod_i P(\theta_i \mid \alpha) \prod_j P\left(z_{i,j} \mid \theta_i\right) P\left(w_{i,j} \mid z_{i,j}\right)$$



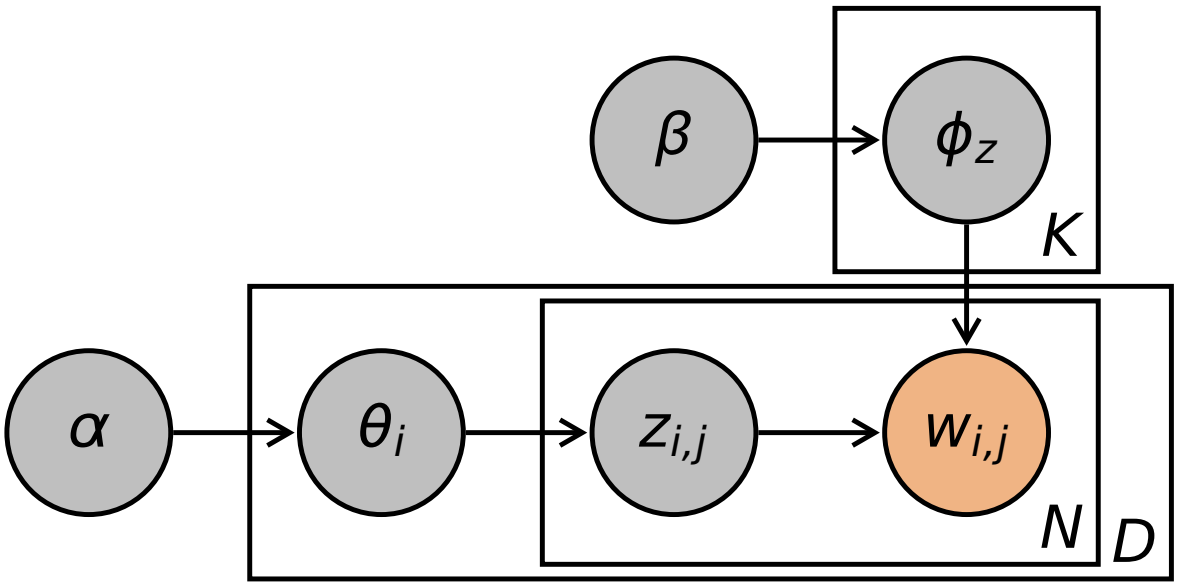
The graphical model has the following "generative story":

First, we generate distributions $\phi$ over words, once for each topic $z$.

Second, we generate distributions $\theta$ over topics, once for each document $i$.

Third, we generate words $w$, once for each token $j$, by sampling a topic $z$ from $\theta_i$, and then sampling a word from that topic's distribution $\phi_z$.

# Latent Dirichlet Allocation




$$P(\phi_z, \theta_i \mid w_{i,j}, \alpha, \beta) = \sum_{z_{i,j}} P(\phi_z, \theta_i, z_{i,j} \mid w_{i,j}, \alpha, \beta)$$



We would like to infer the distributions $\theta$ and $\phi$, given the data ($w_{i,j}$) and given the hyperparameters ($\alpha$, $\beta$, $K$).

However, the model defines a joint distribution over these variables and also the topics $z_{i,j}$. Exact inference is intractable, because we would have to sum over all possible topic assignments – but there are $K^{ND}$ possible assignments.

# Approximate Inference

- Want to know global variables (e.g. $\phi$)
- Don’t want to know local variables (e.g. $z$)
- Exact inference intractable

# Markov Chain Monte Carlo

- $E_x\left[f(x)\right] = \sum_x P(x) f(x)$
- Construct Markov chain converging to $P(x)$
- Sample from Markov chain
- $E_x\left[f(x)\right] \approx \frac{1}{N} \sum_{\text{samples}} f(x)$



A Markov Chain Monte Carlo (MCMC) method allows us to sample from a distribution $P(x)$ when exactly calculating the distribution is intractable.

This means that we can approximate a calculation involving that distribution, by considering a small set of samples, rather than considering all possible values.

# Gibbs Sampling

- $P(x)$ intractable
- $P(x_1 \mid x_2, x_3, \ldots)$ tractable
- Markov chain:
  - Initialise $x$
  - Iteratively update $x_i \sim P(x_i \mid x_{-i})$
- Distribution converges to $P(x)$

# Gibbs Sampling for LDA

- $\sum_{z_{i,j}} P\left(\phi_z, \theta_i, z_{i,j} \middle| w_{i,j}, \alpha, \beta\right)$ intractable
- $P\left(z_{i,j} \middle| z_{-i,j}, w_{i,j}, \alpha, \beta\right)$ tractable
  - Dirichlet prior ⇒ can marginalise out $\phi$, $\theta$

$$
\begin{aligned}
& P\left(z_{i,j} \middle| z_{-i,j}, w_{i,j}, \alpha, \beta\right) \\
\propto\ & P\left(z_{i,j} \middle| \theta_i\right)\ P\left(w_{i,j} \middle| z_{i,j}, \phi_{z_{i,j}}\right) \\
=\ & \frac{C_{i,z} + \alpha}{C_i + K\alpha}\ \frac{C_{z,w} + \beta}{C_z + V\beta}
\end{aligned}
$$



Because we can calculate the above conditional probabilities, we can use Gibbs Sampling.

More precisely, we should write:

$$
\begin{aligned}
& P\left(z_{i,j} \middle| z_{-i,j}, w_{i,j}, \alpha, \beta\right) \\
\propto\ & \sum_{\theta_i} \sum_{\phi_{z_{i,j}}} P(\theta_i \mid \alpha) P\left(\phi_{z_{i,j}} \middle| \beta\right) P\left(z_{i,j} \middle| \theta_i\right) P\left(w_{i,j} \middle| z_{i,j}, \phi_{z_{i,j}}\right)
\end{aligned}
$$

The fact that we can easily calculate these sums (integrals) is because of the choice of the Dirichlet distribution as a prior.

# Gibbs Sampling for LDA

$$K = 2,\ V = 4,\ \alpha = \beta = 1$$

| | | | | | |
|---|---|---|---|---|---|
| a | a | b | a | b | b |
| 1 | 2 | 2 | 1 | 1 | 2 |
| c | d | d | d | c | |
| 2 | 2 | 1 | 1 | 1 | |
| b | a | c | b | d | d |
| 1 | 2 | 1 | 1 | 1 | 2 |
| a | c | | | | |
| 1 | 2 | | | | |



A toy set of four documents is shown on the left, with vocabulary {a,b,c,d}. The topics are randomly initialised.

# Gibbs Sampling for LDA

$$K = 2,\ V = 4,\ \alpha = \beta = 1$$

| a | a | b | a | b | b |
|---|---|---|---|---|---|
| ? | 2 | 2 | 1 | 1 | 2 |

| c | d | d | d | c |
|---|---|---|---|---|
| 2 | 2 | 1 | 1 | 1 |

| b | a | c | b | d | d |
|---|---|---|---|---|---|
| 1 | 2 | 1 | 1 | 1 | 2 |

| a | c |
|---|---|
| 1 | 2 |

$$P(z_{1,1}=1) \propto P(1|\theta_1)P(a|1)$$

$$P(z_{1,1}=2) \propto P(2|\theta_1)P(a|2)$$



We iteratively go through each token, and calculate the conditional distribution for that topic, given all other topic assignments.

# Gibbs Sampling for LDA

$$K = 2,\ V = 4,\ \alpha = \beta = 1$$

| a | a | b | a | b | b |
|---|---|---|---|---|---|
| ? | 2 | 2 | 1 | 1 | 2 |

| c | d | d | d | c |
|---|---|---|---|---|
| 2 | 2 | 1 | 1 | 1 |

| b | a | c | b | d | d |
|---|---|---|---|---|---|
| 1 | 2 | 1 | 1 | 1 | 2 |

| a | c |
|---|---|
| 1 | 2 |

$$P(z_{1,1}=1) \propto \frac{2+1}{5+2} P(a|1)$$

$$P(z_{1,1}=2) \propto \frac{3+1}{5+2} P(a|2)$$

# Gibbs Sampling for LDA

$$K = 2,\ V = 4,\ \alpha = \beta = 1$$

| a | a | b | a | b | b |
|---|---|---|---|---|---|
| ? | 2 | 2 | 1 | 1 | 2 |

| c | d | d | d | c |
|---|---|---|---|---|
| 2 | 2 | 1 | 1 | 1 |

| b | a | c | b | d | d |
|---|---|---|---|---|---|
| 1 | 2 | 1 | 1 | 1 | 2 |

| a | c |
|---|---|
| 1 | 2 |

$$P(z_{1,1}=1) \propto \frac{2+1}{5+2} \frac{2+1}{10+4}$$

$$P(z_{1,1}=2) \propto \frac{3+1}{5+2} P(a|2)$$

# Gibbs Sampling for LDA

$$K = 2,\ V = 4,\ \alpha = \beta = 1$$

| a | a | b | a | b | b |
|---|---|---|---|---|---|
| ? | 2 | 2 | 1 | 1 | 2 |

| c | d | d | d | c |
|---|---|---|---|---|
| 2 | 2 | 1 | 1 | 1 |

| b | a | c | b | d | d |
|---|---|---|---|---|---|
| 1 | 2 | 1 | 1 | 1 | 2 |

| a | c |
|---|---|
| 1 | 2 |

$$P(z_{1,1}=1) \propto \frac{2+1}{5+2} \quad \frac{2+1}{10+4}$$

$$P(z_{1,1}=2) \propto \frac{3+1}{5+2} \quad \frac{2+1}{8+4}$$

# Gibbs Sampling for LDA

$$K = 2,\ V = 4,\ \alpha = \beta = 1$$

| | | | | | |
|---|---|---|---|---|---|
| a | a | b | a | b | b |
| 2 | 2 | 2 | 1 | 1 | 2 |
| c | d | d | d | c | |
| 2 | 2 | 1 | 1 | 1 | |
| b | a | c | b | d | d |
| 1 | 2 | 1 | 1 | 1 | 2 |
| a | c | | | | |
| 1 | 2 | | | | |

$$P(z_{1,1}=1) = 0.391$$

$$P(z_{1,1}=2) = 0.609$$



Once we have calculated the probabilities, we randomly sample a topic. Here we have sampled the topic 2 (which would happen with probability 0.609)

# Gibbs Sampling for LDA

$$K = 2,\ V = 4,\ \alpha = \beta = 1$$

| | | | | | |
|---|---|---|---|---|---|
| a | a | b | a | b | b |
| 2 | ? | 2 | 1 | 1 | 2 |

| | | | | |
|---|---|---|---|---|
| c | d | d | d | c |
| 2 | 2 | 1 | 1 | 1 |

| | | | | | |
|---|---|---|---|---|---|
| b | a | c | b | d | d |
| 1 | 2 | 1 | 1 | 1 | 2 |

| | |
|---|---|
| a | c |
| 1 | 2 |



We repeat this for each token. We will eventually go over the whole dataset many times.

At some point we can stop and take a sample from the Markov chain: the set of all topic assignments.

# Gibbs Sampling for LDA

- Given a sample:

$$\hat{\theta}_i(z) = \frac{C_{i,z} + \alpha}{C_i + K\alpha} \qquad \hat{\phi}_z(w) = \frac{C_{z,w} + \beta}{C_z + V\beta}$$

- Can't directly compare topics from different samples

- Can compare e.g. $D_{KL}(\text{doc 1}||\text{doc 2})$, as distributions over words



The order of the topics doesn't affect the model's predictions. For example, we can't equate topic 7 from one sample with topic 7 from another sample – the number is arbitrary.

However, we can compare quantities that don't rely on a specific topic – for example, we can compare two documents, by looking at their distributions over words. LDA will have smoothed out the distributions (using the inferred topics). We can calculate a quantity such as Kullback-Leibler Divergence, which measures how similiar or different two distributions are. We can take many samples from the Markov chain, caculate the KL-divergence for each sample, and then take an average. Taking an average over many samples makes the estimate more accurate.

# Summary

- Tasks:
  - Word Sense Induction
  - Topic Discovery
- Models:
  - K-Means
  - Latent Dirichlet Allocation
- Training:
  - Gibbs Sampling