4F13 Coursework 3: Latent Dirichlet Allocation

Part a)

The maximum likelihood (ML) multinomial over words is a statistical method for estimating probability of observing different words w, in a collection of documents d, such that $d \in \{1, \dots, D\}$, where D documents make up our training set, $\mathcal{A}$. All words are drawn from a vocabulary $\mathcal{M}$, such that $m \in \mathcal{M} = \{1, \dots, M\}$. Thus the n'th word of the d'th document is represented by $w_{nd} \in \mathcal{M}$ for $n \in \{1, \dots, N_d\}$, where N_d denotes the length of document d. Finally, let c_m define the count of occurrences of word m in $\mathcal{A}$. For testing, we use a held-back document set, $\mathcal{B}$.

Draw each word independently from the same categorical distribution β, where $w_{nd} \sim Cat(\beta)$ and β is a $M \times 1$ vector subject to $\Sigma_{m=1}^{M} \beta_m = 1$ and $\beta_i \geq 0$. The log-likelihood $\mathcal{L}(\beta)$ which we seek to maximise, or hence the log of the probability of the training set $\mathcal{A}$ given β, is defined as follows:

$$\mathcal{L}(\beta) = \log P(\mathcal{A}|\beta) = \log \prod_{d=1}^{D} \prod_{n=1}^{N_d} P(w_{nd}|\beta) = \log \prod_{m \in M} (\beta_m)^{c_m} = \sum_{m \in M} c_m \log \beta_m$$

The maximum likelihood estimate $\hat{\beta}^{ML} \coloneqq argmax\, L(\beta)$ requires a few more steps. By taking derivatives of the above term with respect to β_i and including a Lagrange multiplier, λ, we come to:

$$\hat{\beta}^{ML} = \frac{c_i}{\lambda} = \frac{c_i}{\Sigma_{m \in M} c_m} = \frac{c_i}{C}$$

We find $\lambda = C$ is necessary to uphold the sum-to-one constraint on β components. As such, the ML estimate simplifies to the normalized frequency of each word across the training set, where $C_{\mathcal{A}} = 271898$ and $M_{\mathcal{A}} = 6906$.

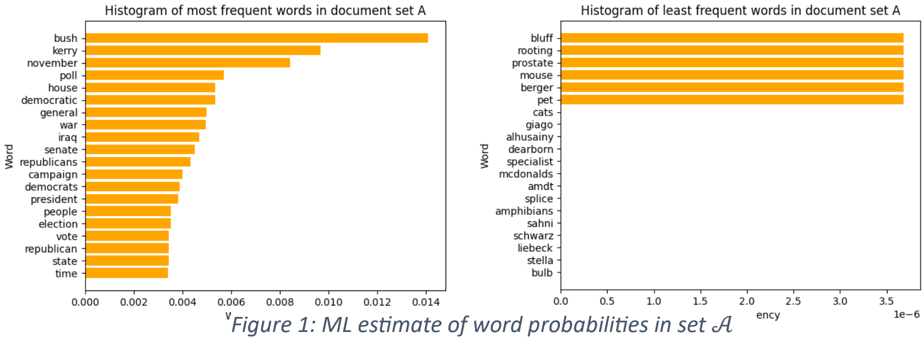


Figure 1: ML estimate of word probabilities in set $\mathcal{A}$

Figures 1a & 1b show the most and least frequently appearing words in the training set $\mathcal{A}$, respectively. 'bush' is the modal word, thus $\hat{\beta}_{MAX}^{ML} = \frac{3833}{271898}$, whilst numerous words found in $\mathcal{M}$ do not appear at all in $\mathcal{A}$, for example 'cats', thus $\hat{\beta}_{MIN}^{ML} = \frac{0}{271898} = 0$. Next, consider a subsample of $\mathcal{A}$, or in other words, a test set $\mathcal{T}$ with T words drawn from $\mathcal{A}$. The maximum (log) probability would arise from a $\mathcal{T}$ which contains all instances of 'bush' within $\mathcal{A}$ and solely entries of 'bush', or hence $T = c_{"bush"}$:

$$\max_{|\mathcal{T}|=T} \log P(\mathcal{T}|\,\hat{\beta}^{ML}) = T \log \hat{\beta}_{MAX}^{ML} = T \log \frac{3833}{271898} = -4.26T$$

The minimum (log) probability occurs when $\mathcal{T}$ contains at least one word with zero probability under the ML estimate, such as 'cats', equating the product to zero. All probabilities must be non-zero, hence $\mathcal{T}$ is unfeasible under the ML estimate.

$$\min_{|\mathcal{T}|=T} \log P(\mathcal{T}|\,\hat{\beta}^{ML}) = \log 0 = -\infty$$

Part b)

An alternative to a ML fit is Bayesian inference, which can incorporate prior beliefs about parameters and is less prone to overfitting. Suppose the probability vector β now has a symmetric Dirichlet prior with a concentration parameter α on the word probabilities, such that $\beta{\sim}Dir(\beta;\alpha)$. We can express the posterior $P(\beta|\mathcal{A})$ with c as a word count vector to track history as follows:

$$P(\beta|\mathcal{A}) \propto P(\beta)\cdot P(\mathcal{A}|\beta)$$

$$\therefore P(\beta|\mathcal{A}) = \left(\frac{1}{B(\alpha)}\prod_{m=1}^{M}\beta_m^{\alpha_m-1}\right)\cdot\prod_{p=1}^{M}\beta_p^{c_p} = Dir(\beta;\alpha+c)$$

Clearly, the Dirichlet distribution above is a conjugate prior of the multinomial distribution from before, subject to a different posterior featuring the new word count parameter, c. Now consider the predictive distribution for a new unseen word w^* given the posterior, making use of the marginals of the categorical distribution β over the training set $\mathcal{A}$:

$$P(w^*=i|\mathcal{A}) = \int_B P(w^*=i,\beta|\mathcal{A})d\beta = \int P(w^*=i|\beta_i)P(\beta_i|\mathcal{A})d\beta_i = \int \beta_i P(\beta_i|\mathcal{A})d\beta_i$$

$$= \mathbb{E}_{\beta_i|\mathcal{A}}[\beta_i]$$

$$= \frac{\alpha_i + c_i}{\sum_{m=1}^{M}\alpha_m + c_m} \coloneqq \hat{\beta}_i^*$$

Simply, the mean of each probability component β_i is proportional to the normalized parameter value, $\hat{\beta}_i^*$. Define α as $\alpha = a\mathbf{1}$ to uphold the symmetric condition, then the above simplifies to:

$$P(w^*=i|\mathcal{A}) = \frac{a+c_i}{Ma+C} \coloneqq \hat{\beta}_i^*$$

The Bayesian expression is equivalent to the ML approach should we add a count a to each word count c_m observed in the training set $\mathcal{A}$. All word probabilities are now closer to $1/\mathcal{M}$ however the ranking is unchanged, despite better resembling a uniform distribution. Rare words will gain probability, whilst more common words will lose it. The larger the value of a, the stronger this normalizing effect and the less relative importance the observed counts c_i in $\mathcal{A}$ have.

Part c)

Now consider applying the Bayesian model to a specific test document, for example $d = 2001$, and computing the log-probability of all the words in it. To capture the importance of word order, a categorical distribution function is paramount. Each phrase within each document must be considered separately, despite the words themselves being independent, thus the document is treated as a sequential collection of categorical random variables. Start with $a = 0.1$ as to minimize the aforementioned normalizing effect, we define the log probability, $l(d)$, as follows, where c_m^* is the count of word m in d:

$$l(d) = \log P(w_{nd}^* \in \{1,\dots,N_d\}|\mathcal{A}) = \log\prod_{n=1}^{N_d}P(w_{nd}^*|\mathcal{A})$$

$$= \log\prod_{m=1}^{M}P(w^*=m|\mathcal{A})^{c_m^*} = \sum_{m=1}^{M}c_m^*\log P(w^*=m|\mathcal{A}) = (c^*)^T(\log\hat{\beta}^*)$$

$$\therefore l(d = 2001)|_{a=0.1} = -3691.2$$

Furthermore, we define per-word perplexity $p(d)$, a measure of how well a language model predicts words in a test dataset, as $p(d) \coloneqq \exp\left(-\frac{l(d)}{N_d}\right)$. Lower perplexity values indicate better performance of the model on unseen data. Should we use $N_d = M_{\mathcal{A}} = 6906$, we provide an upper bound for $p(d) \forall d \in B$ of 6906. Clearly this is equivalent to drawing from a uniform multinomial, thus assuming no prior information on any word and achieving a perplexity equal to the word count. The overall perplexity is also of interest, found by treating the test set $\mathcal{B}$ as one long document.

Table 1: Perplexities of documents in test set $\mathcal{B}$ given $a = 0.1$

	Single: $d = 2001 \in \mathcal{B}$	All: $\forall d \in \mathcal{B}$	Uniform multinomial
$p(d)$	4399.0	2697.1	6906

Documents frequently featuring common words will have higher log-probabilities and hence lower perplexities. Given the higher-than-average perplexity of document $d = 2001$ compared to test set $\mathcal{B}$, $d = 2001$ must have a higher proportion of rare words than the overall training set $\mathcal{A}$ (ref. table 1).

Part d)

Oftentimes, we want to know the topic, or category z_d, of the document d, where $z_d \in \{1, \dots, K\}$ is a latent variable representing one of the K distinct categoies. The extended Bayesian Mixture Model (BMM) is summarised in figure 2.

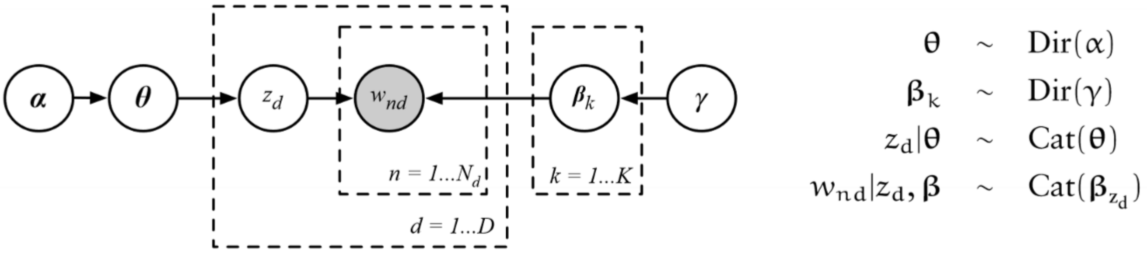
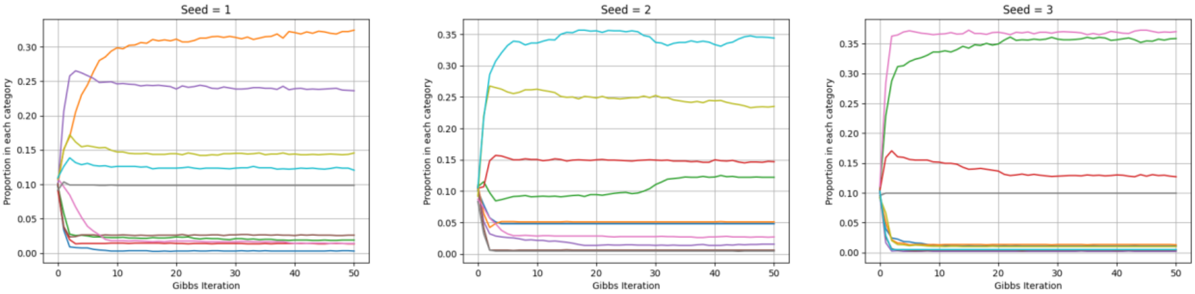

α
θ
z_d
w_{nd}
β_k
γ
$n = 1 \dots N_d$
$k = 1 \dots K$
$d = 1 \dots D$
$\theta \sim \mathrm{Dir}(\alpha)$
$\beta_k \sim \mathrm{Dir}(\gamma)$
$z_d | \theta \sim \mathrm{Cat}(\theta)$
$w_{nd} | z_d, \beta \sim \mathrm{Cat}(\beta_{z_d})$

Figure 2: Bayesian Mixture Model (BMM), Mixture of Multinomials

We perform Gibbs sampling over i Gibbs iterations using the training set $\mathcal{A}$ to obtain values for θ, z_d and β_k, subject to a chosen α_i and γ_i. We can plot the mixture proportions over i, or simply the posterior probabilities plus the prior term:

$$\theta_k^{(i)} \approx \frac{1}{K\alpha_k + D}\left(\alpha_k + \sum_{d=1}^{D} \mathbf{1}\left(z_d^{(i)} = k\right)\right)$$



Figures 3a-c: BMM – topic posteriors against Gibbs iteration for 3 different initialisations | $\alpha_i = 10, \gamma_i = 0.1, i = 50, K = 10$

Figure 3 shows topic posteriors against Gibbs iteration. Note how at $i = 0, \theta_k^{(0)} = \frac{1}{K} \forall k$, indicating that all posteriors start evenly distributed. It is clear from the topics with near-zero converged posteriors in the three examples above that $K = 10$ is excessively high; fewer topics could have described the whole document sufficiently accurately.

Different initialisations (or random states (r.s.)) lead to different stationary distributions. As such, we can't say we ever converge to the 'true' posterior but rather some local optimum, who's accuracy we must determine ourselves.

Part e)

Finally, we can improve our model by allowing each document to be a blend of the K topics, rather than one distinct class, achieved by giving each word its own latent topic. This is summarised below:

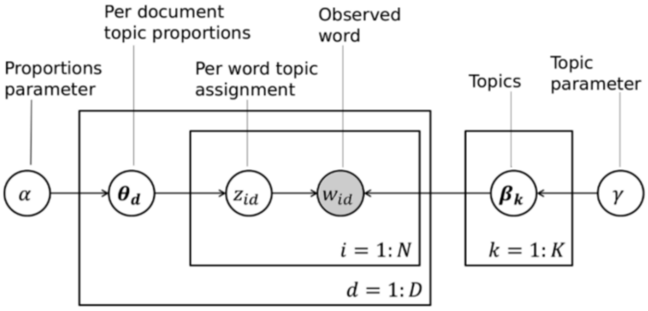
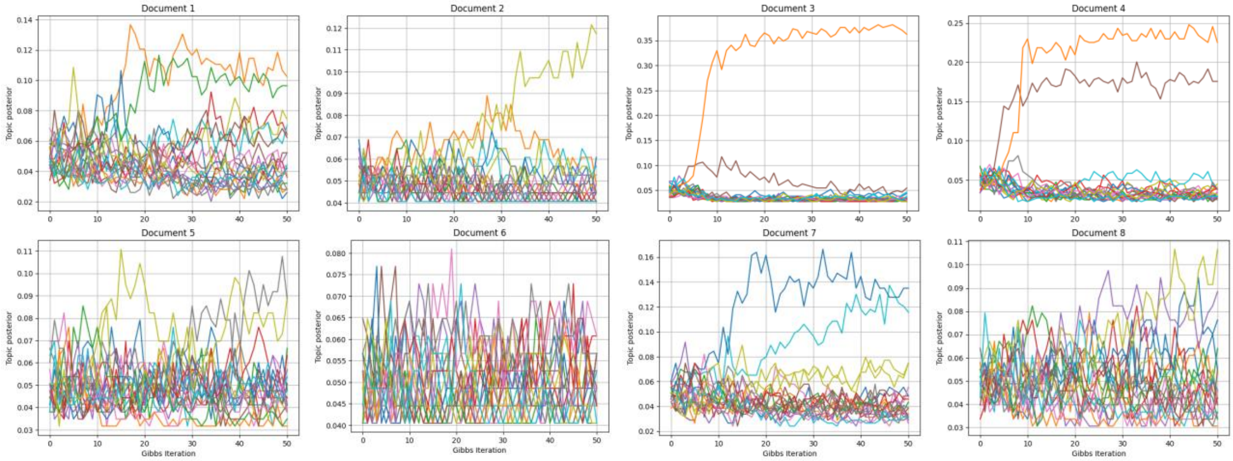


Figure 4: Latent Dirichlet Allocation

We come to a slightly different expression for each document's topic posterior seeing as each word can be drawn from different topics:

$$\left[\theta_k^{(i)}\right]_k \approx \frac{1}{K\alpha_k + N_d}\left(\alpha_k + \sum_{n=1}^{N_d} \mathbf{1}\left(z_{nd}^{(i)} = k\right)\right)$$

We first compute the posteriors for arbitrary documents against Gibbs iteration, shown below:



Figures 5i-viii) Topic posterior for specific documents against Gibbs iteration using LDA $|\alpha_i = 10, \gamma_i = 0.1, i = 50, K = 20$

Clearly, each document behaves differently and not all appear stable after 50 Gibbs iterations, some far from it. Topics that are similar will follow similar sampling paths, thus the probability of drawing the same word from either topic is indistinguishable. We expand on this by computing the posteriors over $\mathcal{A}$ to analyse convergence, noting the model stabilises after 30 iterations (ref fig.6).

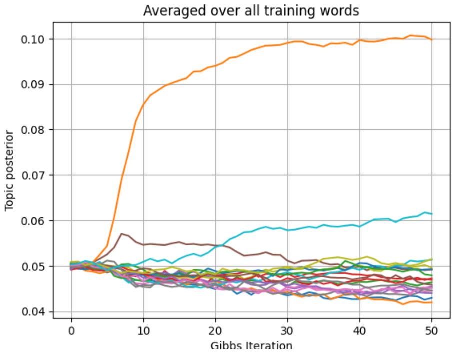


Figure 6: Topic posterior averaged over all words in training set $\mathcal{A}$

Moving on to per-word perplexities, or more specifically comparing previous, less sophisticated models to our current LDA model, we notice perplexity over $\mathcal{B}$ decreases as model complexity increases (ref table 2).

Table 2: Perplexity of test set $\mathcal{B}$

	Maximum Likelihood	Simple Bayes Predictive	BMM $r.s. = 100, i$=50	LDA $r.s. = 1, i$=50
$p(d)$	∞	2697.1	2100.7	2072.5

Next, we investigate the convergence by computing the word entropy for each topic as a function of Gibbs iteration (ref. fig.7). An unknown word w^* is drawn from topic k with distribution $(w^*|z^* = k)\sim Cat(\beta_k^*)$. Let c_{km} denote the count of word m assigned to topic k. We compute its entropy, $H(w^*|z^* = k)$, as follows:

$$\beta_{km}^* = \frac{\gamma_m + c_{km}}{\sum_{i=1}^{M} \gamma_i + c_{ki}}$$

$$H(w^*|z^* = k) \approx \sum_{m=1}^{M} \hat{\beta}_{km}^* \log \frac{1}{\hat{\beta}_{km}^*} = -\left(\hat{\beta}_k^*\right)^T \left(\log \hat{\beta}_k^*\right)$$

Where $\log(\cdot)$ is the natural logarithm applied element-wise, making the units of $H(\cdot)$ nats, which are easier to compare with perplexity versus bits (our unit should we use base 2 instead). Overall, figure 7 shows that entropy and iteration are inversely proportional. This is because topic specificity increases as the sampler progresses, thus vocabulary size and uncertainty (entropy) both reduce.

We can conclude the model has stabilised after 50 iterations. Of the 20 categories, there are three distinct ones, represented by the lowest entropies. The log-perplexity under LDA is $\log 2072.5 = 7.63$, significantly higher than the entropy of any one topic. However, this is expected as the test set $\mathcal{B}$ contains a mixture of topics thus has higher entropy than a singular topic would.

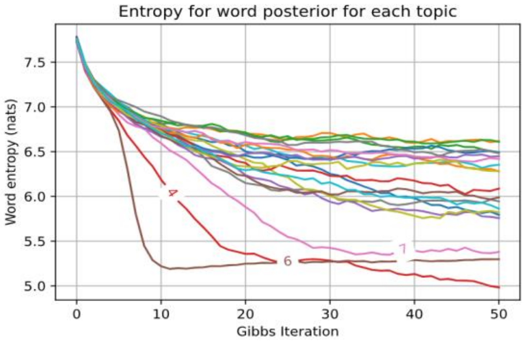


Figure 7: Entropy for each topic's categorical distribution