CUED - Engineering Tripos Part IIB 2019-2020 Module Coursework

Module	**4F13**	Title of report	**Latent Dirichlet Allocation**

Date submitted: **07/12/2020**	Assessment for this module is ☑ **100%** / ☐ 25% coursework of which this assignment forms **33** %

UNDERGRADUATE STUDENTS O NLY		**POST GRADUATE STUDENTS ONLY**			
Candidate number:	**5562E**	Name:		College:	

Feedback to the student

☐ **See also comments in the text**

		Very good	**Good**	Needs improvmt
C O N T E N T	**Completeness, quantity of content:** Has the report covered all aspects of the lab? Has the analysis been carried out thoroughly?			
	Correctness, quality of content Is the data correct? Is the analysis of the data correct? Are the conclusions correct?			
	Depth of understanding, quality of discussion Does the report show a good technical understanding? Have all the relevant conclusions been drawn?			
	Comments:			
P R E S E N T A T I O N	**Attention to detail, typesetting and typographical errors** Is the report free of typographical errors? Are the figures/tables/references presented professionally?			
	Comments:			

Overall assessment (circle grade)		A*	**A**	B	C	D
Guideline standard		>75%	**65-75%**	55-65%	40-55%	<40%
Penalty for lateness:		*20% of marks per week or part week that the work is late.*				

Marker: Date:

4F13 Probabilistic Machine Learning - Latent Dirichlet Allocation

Candidate: 5562E

December 9, 2020

Contents

1 Introduction

We have a document training set $\mathcal{A}$, which consists of D documents indexed by $d \in \{1 \dots D\}$. A document is simply an ordered list of words. All words are drawn from a vocabulary indexed by $m \in \mathcal{M} = \{1 \dots M\}$. We denote the n'th word in the d'th document by $w_{nd} \in \mathcal{M}$ for $n \in \{1 \dots N_d\}$. Here N_d denotes the length of document d. For simplicity, we denote the count of occurrences of word m in the training set by c_m. We hold back a test set $\mathcal{B}$ to calculate the performance of our approaches.

2 Questions

a Maximum Likelihood

We begin assuming each word is drawn independently from the same categorical distribution with parameter β such that $w_{nd} \sim Cat(\beta)$. In this case β is a $M \times 1$ vector subject to the constraints: $\sum_{m=1}^{M} \beta_m = 1$ and $\beta_i \geq 0$. The likelihood of β ($L(\beta)$) is the probability of the training dataset $\mathcal{A}$ given β:

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

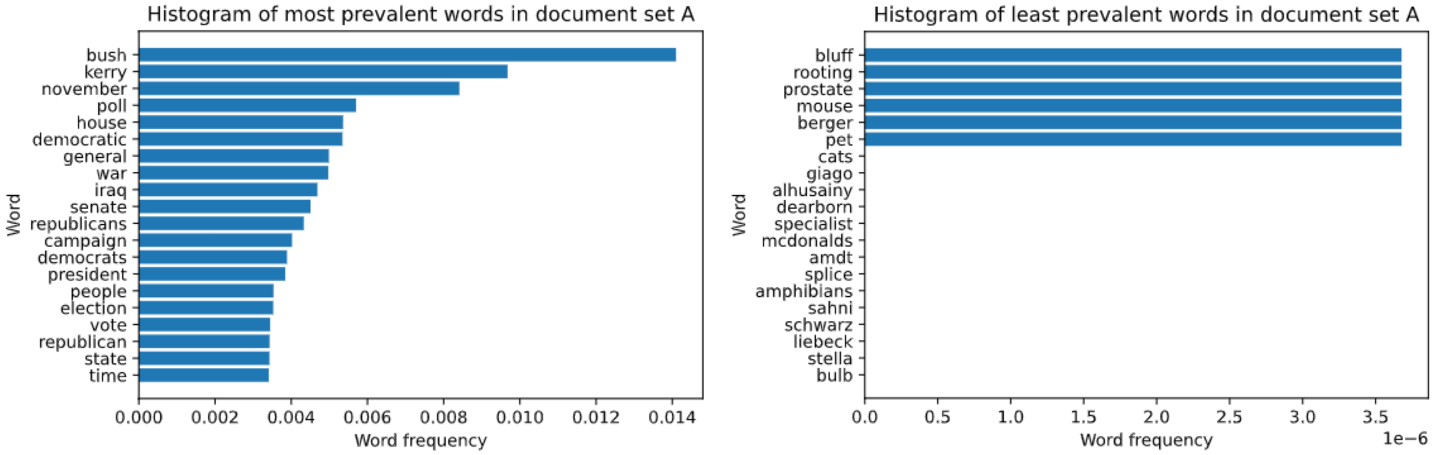
Where c_m is the count of word m in the training set. To obtain the Maximum-Likelihood estimate $\hat{\beta}^{ML} := \arg\max L(\beta)$, we maximise the log-likelihood $\mathcal{L}(\beta)$ as this is more tractable:

$$\mathcal{L}(\beta) := \log L(\beta) = \sum_{m \in \mathcal{M}} c_m \log \beta_m \tag{2}$$

We can now take derivatives and include a Lagrange multiplier term to respect the sum to 1 constraint:

$$\begin{aligned} \frac{\partial}{\partial \beta_i} \left\{ \mathcal{L}(\beta) + \lambda \left(1 - \sum_{m=1}^{M} \beta_m \right) \right\} \Bigg|_{\beta = \hat{\beta}^{ML}} &= \frac{c_i}{\hat{\beta}_i^{ML}} - \lambda = 0 \\ \therefore \hat{\beta}_i^{ML} = \frac{c_i}{\lambda} &= \frac{c_i}{\sum_{m \in \mathcal{M}} c_m} = \frac{c_i}{C} \end{aligned} \tag{3}$$

The Lagrange multiplier $\lambda = C$ to respect the sum-to-one constraint on the β components. Therefore, the ML estimate is simply the empirical frequency of each word (normalised by the sum of all counts C). For the training set $\mathcal{A}$, we have $C_{\mathcal{A}} = 271898$.



(a) Top 20 most prevalent words

(b) Top 20 least prevalent words

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

Figure 1a shows the empirical frequencies of the top 20 words in document set $\mathcal{A}$. We have that $\hat{\beta}^{ML}_{\max} := \frac{\max_i c_i}{C} = \frac{3833}{271898}$ for the word *"bush"* (the president - not the foliage). Similarly $\hat{\beta}^{ML}_{\min} := \frac{\min c_i}{C} = 0$ as there are some words in the vocabulary $\mathcal{M}$ that never appear in the training set $\mathcal{A}$ (such as *"bulb"* in 1b). Indeed, each new word w^* under ML is assumed to be drawn from equation 4:

$$P(w^* = i|\hat{\beta}^{ML}) = \frac{c_i}{C} \tag{4}$$

Therefore, if we have an arbitrary test set $\mathcal{T}$, that is T words long. The maximum probability test set would be T repetitions of *"bush"*. The lowest probability test set need only have a single word with zero probability under our ML estimate (e.g. *"bulb"*) to make the probability of the whole multiply to 0. The highest and lowest test set log-probabilities (base e) are given by:

$$\max_{|\mathcal{T}|=T} \log P(\mathcal{T}|\hat{\beta}^{ML}) = T \log \hat{\beta}^{ML}_{\max} = T \log \frac{3833}{271898} = -4.26T \tag{5}$$

$$\min_{|\mathcal{T}|=T} \log P(\mathcal{T}|\hat{\beta}^{ML}) = \log 0 = -\infty \tag{6}$$

This is clearly unsatisfactory; a feasible test set should not have zero probability. Therefore, the ML estimate is insufficient.

b Bayesian Inference

Instead, we can perform Bayesian inference to ameliorate these issues. We assume the probability vector β has a Dirichlet prior with concentration parameter α such that $\beta \sim Dir(\beta; \alpha)$. We perform Bayesian inference to obtain the posterior for β.

$$\begin{aligned} P(\beta|\mathcal{A}) &\propto P(\beta) \cdot P(\mathcal{A}|\beta) \\ &= \left(\frac{1}{B(\alpha)} \prod_{m=1}^{M} \beta_m^{\alpha_m - 1} \right) \cdot \prod_{p=1}^{M} \beta_p^{c_p} \\ &\propto \prod_{m=1}^{M} \beta_m^{(\alpha_m + c_m) - 1} \\ &\propto Dir(\beta; \alpha + c) \\ \therefore P(\beta|\mathcal{A}) &= Dir(\beta; \alpha + c) \end{aligned} \tag{7}$$

Where, c is now a vector of word counts. We say that the Dirichlet distribution is a conjugate prior to the categorical (or multinomial) distribution as the posterior is also Dirichlet (albeit with a new parameter). We now seek to compute the predictive distribution for an unseen word w^* given the posterior.

$$\begin{aligned} P(w^* = i|\mathcal{A}) &= \int_{\beta} P(w^* = i, \beta|\mathcal{A}) d\beta \\ &= \int_{\beta_i} P(w^* = i|\beta, \mathcal{A}) \int_{\beta_{\setminus i}} P(\beta|\mathcal{A}) d\beta_{\setminus i} d\beta_i \\ &= \int P(w^* = i|\beta_i) P(\beta_i|\mathcal{A}) d\beta_i \\ &= \int \beta_i P(\beta_i|\mathcal{A}) d\beta_i \\ &= \mathbb{E}_{\beta_i|\mathcal{A}}[\beta_i] \\ &= \frac{\alpha_i + c_i}{\sum_{m=1}^{M} \alpha_m + c_m} := \hat{\beta}_i^* \end{aligned} \tag{8}$$

Where the last line is a simple a property of the Dirichlet: the mean of each component is proportional to the corresponding parameter value (subject to normalisation). An interesting observation is that the predictive distribution is exactly equal to that computed using the MAP estimate ($\hat{\beta}^* = \hat{\beta}^{MAP}$). If we consider only a symmetric Dirichlet such that $\alpha = a\mathbf{1}$. The previous expression reduces to:

$$P(w^* = i|\mathcal{A}) = \frac{a + c_i}{Ma + C} \tag{9}$$

By comparing equations 4 and 9, we see that the Bayesian approach is equivalent to ML if we add a pseudo-count a to each word count c_m observed in the training set $\mathcal{A}$. We wish to see which words gain probability as a result:

$$\begin{aligned} P(w^* = i|\mathcal{A}) &> P(w^* = i|\hat{\beta}^{ML}) \\ \frac{a + c_i}{Ma + C} &> \frac{c_i}{C} \\ Ca + Cc_i &> Mac_i + Cc_i \\ c_i &< \frac{C}{M} \end{aligned} \tag{10}$$

Those with $c_i < C/M$ gain probability and those with $c_i > C/M$ lose it. All word probabilities are drawn closer to $1/M$ but probability rank orderings are unchanged (the overall distribution is simply drawn closer to the uniform). The larger the value of a, the stronger this effect and the less relative importance is assigned to the observed counts in $\mathcal{A}$.

c Testing Performance of Bayesian Predictor

We now apply this Bayesian analysis to an unseen test document. The set of all words in a document d is denoted $\{w^*_{nd}\}_{n=1}^{N_d}$. To compute $l(d)$, the log probability of document d:

$$\begin{aligned}
l(d) &:= \log P(\{w^*_{nd}\}_{n=1}^{N_d}|\mathcal{A}) \\
&= \log \prod_{n=1}^{N_d} P(w^*_{nd}|\mathcal{A}) \\
&= \log \prod_{m=1}^{M} P(w^* = m|\mathcal{A})^{c^*_m} \qquad (11) \\
&= \sum_{m=1}^{M} c^*_m \log P(w^* = m|\mathcal{A}) \\
&= (c^*)^T (\log \hat{\beta}^*)
\end{aligned}$$

Where c^*_m is the count of word m in the unseen document d. We factorised the second line by noting all words are independent.

Nevertheless, word order matters. Therefore, we treat a document as a sequence of categorical r.v.'s rather than a single multinomial over word counts; we omit a combinatoric term for all the various permutations of words in a document. The phrase *"Bush beats Kerry"* is very different to *"Kerry beats Bush"* so we treat each phrase as a separate document rather than summing their probabilities.

We set the pseudo-count parameter to $a = 0.1$ (there is no integer requirement) and apply equation 11 to test document $d = 2001$, to obtain:

$$l(d = 2001)|_{a=0.1} = -3691.2 \qquad (12)$$

As well as $l(d)$, we are interested in the per-word perplexity $p(d)$ - as defined in equation 13.

$$p(d) := \exp\left(-\frac{l(d)}{N_d}\right) \qquad (13)$$

The expected perplexity for rolls from an n-sided die is n and so the expected perplexity for samples drawn from a uniform multinomial with n total categories is also n. In our case, we have M total words in our vocabulary; if we draw N words from an arbitrary multinomial with M categories, the perplexity will always be bounded below this value: $\lim_{N\to\infty} p(\{w_n\}_{n=1}^{N}) \leq M$. Though if we draw only a few words then we can exceed this perplexity.

We compute the perplexities for a single document $d = 2001$ and for all documents in $\mathcal{B}$. We can compute overall perplexity by treating the whole set $\mathcal{B}$ as one long document. The results are given in table 1.

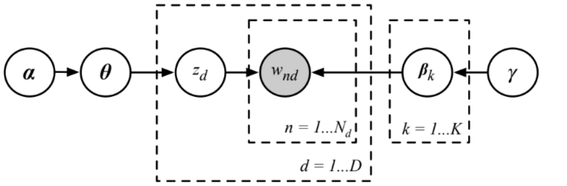
	One doc $d = 2001 \in \mathcal{B}$	All docs $\forall d \in \mathcal{B}$	Uniform multinomial
per-word perplexity	4399.0	2697.1	$M = 6906$

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

Documents with a higher prevalence of common words have higher log-probability and hence lower perplexity. Indeed document $d = 2001$ has a higher perplexity than the average in set $\mathcal{B}$ so we conclude that this document must contain a higher proportion of rare words (with respect to the training set $\mathcal{A}$) than the rest of the test set $\mathcal{B}$.

d Bayesian Mixture Model (BMM)

We extend our model by introducing the concept of document categories (topics). We define K distinct categories and for each document d we define a new latent variable $z_d \in \{1 \dots K\}$ denoting class membership. We assume the class memberships are drawn from a categorical distribution: $z_d \sim Cat(\theta)$, where the parameter θ has Dirichlet prior with concentration α[1] Each document category k has now a different vector β_k from which words are drawn categorically: $(w_{nd}|z_d = k) \sim Cat(\beta_k)$. Each β_k has prior $\beta_k \sim Dir(\gamma)$. This model is summarised in figure 2.



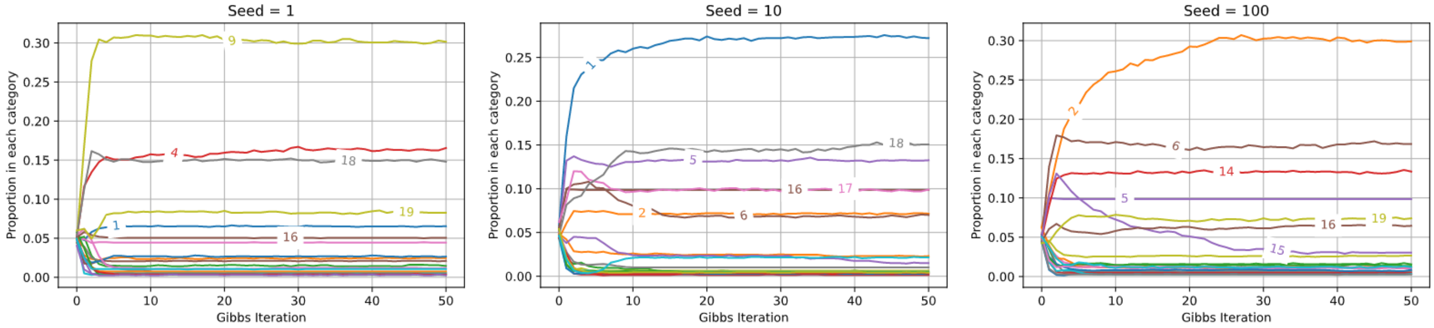
$$
\begin{aligned}
\boldsymbol{\theta} &\sim \mathrm{Dir}(\alpha) \\
\boldsymbol{\beta}_k &\sim \mathrm{Dir}(\gamma) \\
z_d|\boldsymbol{\theta} &\sim \mathrm{Cat}(\boldsymbol{\theta}) \\
w_{nd}|z_d, \boldsymbol{\beta} &\sim \mathrm{Cat}(\boldsymbol{\beta}_{z_d})
\end{aligned}
$$

Figure 2: Bayesian Mixture Model (Mixture of Multinomials)

We choose uniform Dirichlet priors parameters $\alpha_i = 10, \gamma_i = 0.1 \;\; \forall i$ and perform Gibbs sampling using the training set $\mathcal{A}$ to obtain samples for θ, z_d and β_k. We plot the topic posterior as a function of Gibbs iteration i (equation 14). This is simply the fraction of $\{z_d^{(i)}\}_{d=1}^{D}$ assigned to topic k plus the prior term:

$$
\theta_k^{(i)} \approx \frac{1}{K\alpha_k + D}\left(\alpha_k + \sum_{d=1}^{D} \mathbb{1}(z_d^{(i)} = k)\right) \tag{14}
$$

The results are given in figure 3. We see that only a handful of categories have high posterior proportions. This suggests that our value of $K = 20$ is unnecessarily high. We could have fewer topics that still explain the majority of variation well.



(a) Seed = 1

(b) Seed = 10

(c) Seed = 100

Figure 3: BMM - topic posterior against Gibbs iteration for various initialisations

Furthermore, different initialisations (seeds) yield different stationary distributions. All three Gibbs processes in figure 3 converge well enough after 30 iterations but the distribution they converge to is different in each case. This remains the case even after an arbitrary relabelling of categories. As different initialisations yield different final states, we cannot say we ever converge to the "true" posterior but rather a local optimum that is in some sense "good enough".

[1] Note that we are using new notation in this section. α is now the Dirichlet parameter over categories and γ takes the role of Dirichlet parameter over words.

e Latent Dirichlet Allocation (LDA)

We make one final extension to our model. Now each document can be an arbitrary blend of the K topics. Every nth word in each document d is given its own latent topic $z_{nd} \sim Dir(\theta_d)$ and the per document topic proportions parameter θ_d is specific to each document. This extension is summarised in figure 4.

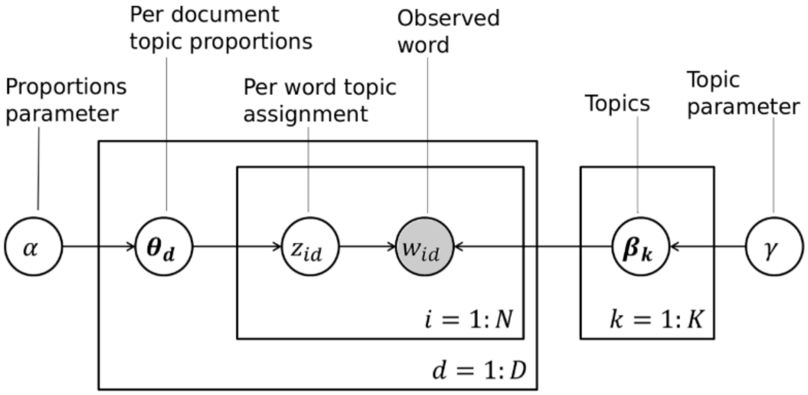
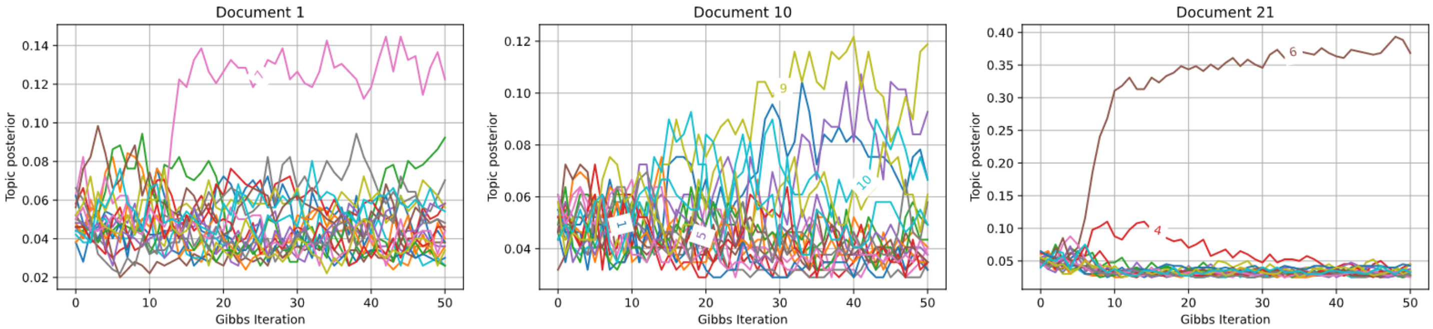


Figure 4: Latent Dirichlet Allocation model (LDA)

With this extension, the words in each document can be drawn from different topics, each drawn from $Cat(\theta_d)$. Each document's topic posterior can be empirically calculated from the Gibbs samples (in a similar way to equation 14):

$$[\theta_d^{(i)}]_k = \frac{1}{K\alpha_k + N_d}\left(\alpha_k + \sum_{n=1}^{N_d} \mathbb{1}(z_{nd}^{(i)} = k)\right) \tag{15}$$

We use equation 15 to compute the topic posteriors for a handful of documents against Gibbs iteration and plot the results on figure 5. Each document behaves differently. Document 21 appears the most stable after 50 Gibbs iterations. Topics that continue to oscillate may be similar to one another (a consequence of having K too high) so the same word has similar probability of being drawn from either topic.



(a) Document 1

(b) Document 10

(c) Document 21

Figure 5: LDA - topic posterior for specific documents against Gibbs iteration

As it is rather hard to analyse convergence by looking at a single document, we compute the posteriors by considering $\mathcal{A}$ as one single, very long document; this yields figure 6. We see that topic 6 is by far the most prevalent in the training set. This is partly a consequence of the "rich get richer" property of this Gibbs sampler. Nevertheless, the posteriors seem to stabilise sufficiently after 30 Gibbs iterations so 50 is more than sufficient.

With the topic posteriors in hand, we can now compare the computed perplexities[2] for the test set $\mathcal{B}$ for each of the methods outlined so far (table 2). As the models get more sophisticated (left to right), the perplexity decreases. In other words, the test set $\mathcal{B}$ has lower overall log-likelihood under the more complex models. A model with more degrees of freedom will always fit the training set $\mathcal{A}$ better than a simpler model. However, it is reassuring to note that we are not over-fitting as performance still improves on the unseen test data $\mathcal{B}$.

[2] For LDA, the number of Gibbs sweeps can be different for computing posteriors based on the training set and computing perplexities of the test set. In our case, both are set to 50.

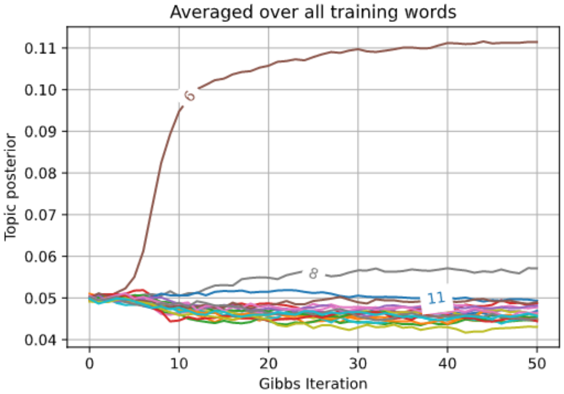


Figure 6: Topic posterior averaged over all words in training set $\mathcal{A}$

	Maximum Likelihood	Simple Bayes Predictive	BMM (seed=100, its=50)	LDA (seed=1, its=50)
Perplexity	∞	2697.1	2100.7	2072.5

Table 2: Test set $\mathcal{B}$ perplexity

It would be possible to compute the perplexity at each Gibbs iteration and plot but this was too computationally intensive to run (each perplexity computation took half an hour for $K = 20$). Instead, to test convergence, we compute the word entropy for each topic as a function of Gibbs iteration. If an unknown word $w*$ is generated from topic k, we know it is distributed as $(w^*|z^* = k) \sim Cat(\hat{\beta}^*_k)$. Where $\hat{\beta}^*_k$ is given by modifying equation 8 to use the updated notation:

$$\hat{\beta}^*_{km} = \frac{\gamma_m + c_{km}}{\sum_{i=1}^{M} \gamma_i + c_{ki}} \tag{16}$$

Where, c_{km} denotes the count of word m assigned to topic k. Therefore, the entropy of a word drawn from topic k is given by:

$$H(w^*|z^* = k) \approx \sum_{m=1}^{M} \hat{\beta}^*_{km} \log \frac{1}{\hat{\beta}^*_{km}} = -(\hat{\beta}^*_k)^T (\log \hat{\beta}^*_k) \tag{17}$$

Where the log operation is applied element-wise. If we choose log to be the natural logarithm (base e) then this entropy would be in units of nats. Base 2 would give us the familiar unit of bits. We choose to work in nats as that gives us easier comparison with the perplexity. Indeed, the natural logarithm of the perplexity for an infinitely long document drawn from a single topic k is expected to be the the entropy in nats, as shown below:

$$\begin{aligned} \log(p(d|z = k)) &= -\frac{1}{N_d} l(d) \\ &= -\frac{1}{N_d} \sum_{m=1}^{M} c^*_m \log \hat{\beta}^*_{km} \\ \therefore \lim_{N_d \to \infty} [\log(p(d|z = k))] &= \sum_{m=1}^{M} \lim_{N_d \to \infty} \left[\frac{c^*_m}{N_d} \right] \log \frac{1}{\hat{\beta}^*_{km}} \\ &= \sum_{m=1}^{M} \hat{\beta}^*_{km} \log \frac{1}{\hat{\beta}^*_{km}} \\ &= H(w^*|z^* = k) \end{aligned} \tag{18}$$

We plot the word entropy of each topic as a function of Gibbs iteration in figure 7. The general trend is for a reduction in entropy with Gibbs iteration. This is because topics become more specific as the Gibbs sampler progresses. A more specific topic, has a smaller typical vocabulary and hence lower uncertainty (lower entropy).

The entropy of each topic has stabilised after 50 iterations. This, in conjunction with the overall stability of the topic posterior (figure 6), shows 50 Gibbs sweeps to be sufficient for computing the perplexity. The log-perplexity of the test set under LDA is $\log 2072.5 = 7.63$. This is modestly higher than the entropy of any particular topic. This result is expected as the overall test set is a blend of topics and hence higher entropy than any individual specific topic.

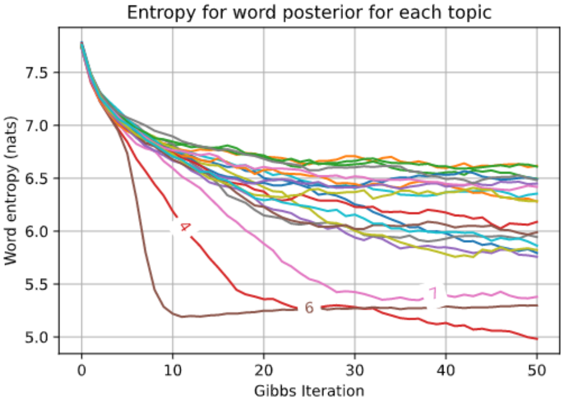
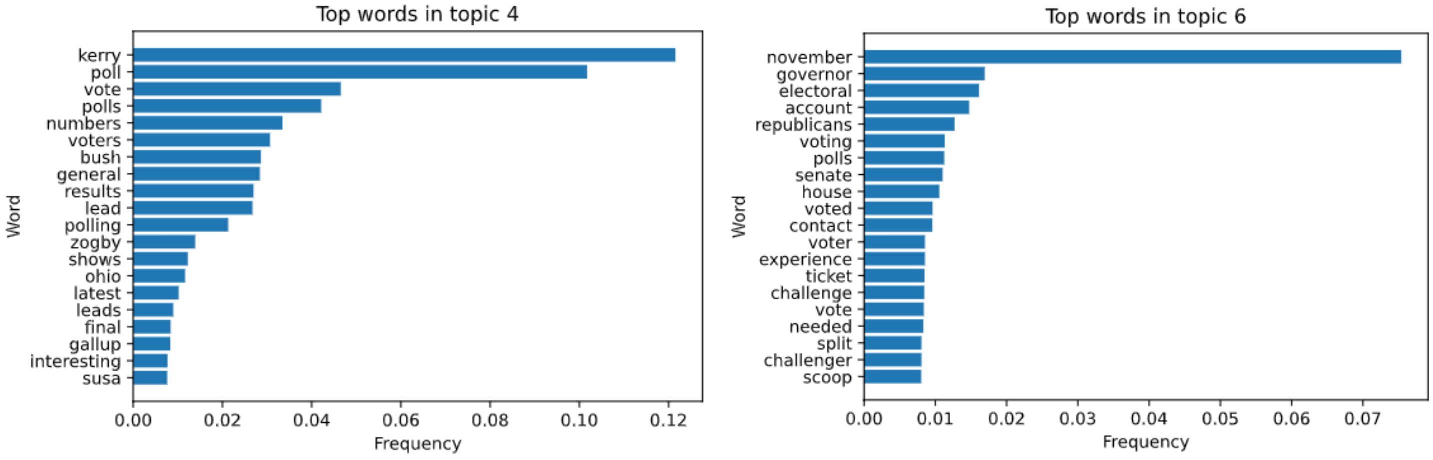


Figure 7: Entropy for each topic's categorical distribution over words

For interest we plot the top 20 words for the topic with lowest entropy ($k = 4$) and that with highest posterior ($k = 6$). This is shown on figure 8. The lowest entropy topic 4 appears to be rather technical analysis of polling data. A high fraction of the probability is contained in just a few top words - hence low entropy. On the other hand, topic 6 (the most prevalent) gives generic vocabulary for speaking about the November general election.



(a) Lowest word entropy - topic 4

(b) Highest posterior - topic 6

Figure 8: LDA - comparison of lowest entropy and highest posterior topics

Words: 979