

Module	4F13	Title of report	Latent Dirichlet Allocation			
Date submitted: 04/12/2020		Assessment for this module is <input checked="" type="checkbox"/> 100% / <input type="checkbox"/> 25% coursework of which this assignment forms <u>33</u> %				
UNDERGRADUATE STUDENTS ONLY		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 7, 2020

Contents

1	Introduction	1
2	Questions	2
a	Maximum Likelihood	2
b	Bayesian Inference	3
c	Testing Performance of Bayesian Predictor	4
d	Bayesian Mixture Model (BMM)	5
e	Latent Dirichlet Allocation (LDA)	6

1 Introduction

We have a document test set \mathcal{A} , which consists of D documents indexed by $d \in \{1 \dots D\}$. A document is an ordered list of words. The vocabulary \mathcal{M} has $M = |\mathcal{M}|$ unique words. We denote the n 'th word in the d 'th document by $w_{nd} \in \{1 \dots N_d\}$. Where N_d is 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 by assuming that each word is drawn independently from a categorical distribution with parameter β : $w_{nd} \stackrel{iid}{\sim} \text{Cat}(\beta)$. In this case β is a $M \times 1$ vector with the conditions that $\sum_{m=1}^M \beta_m = 1$ and $\beta_i \geq 0$. The likelihood of the parameter β is the probability of the dataset given β :

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

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

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

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

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

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 $C_{\mathcal{A}} = 271898$. Figure 1a shows the empirical frequencies of the top 20 words in document set \mathcal{A} . We

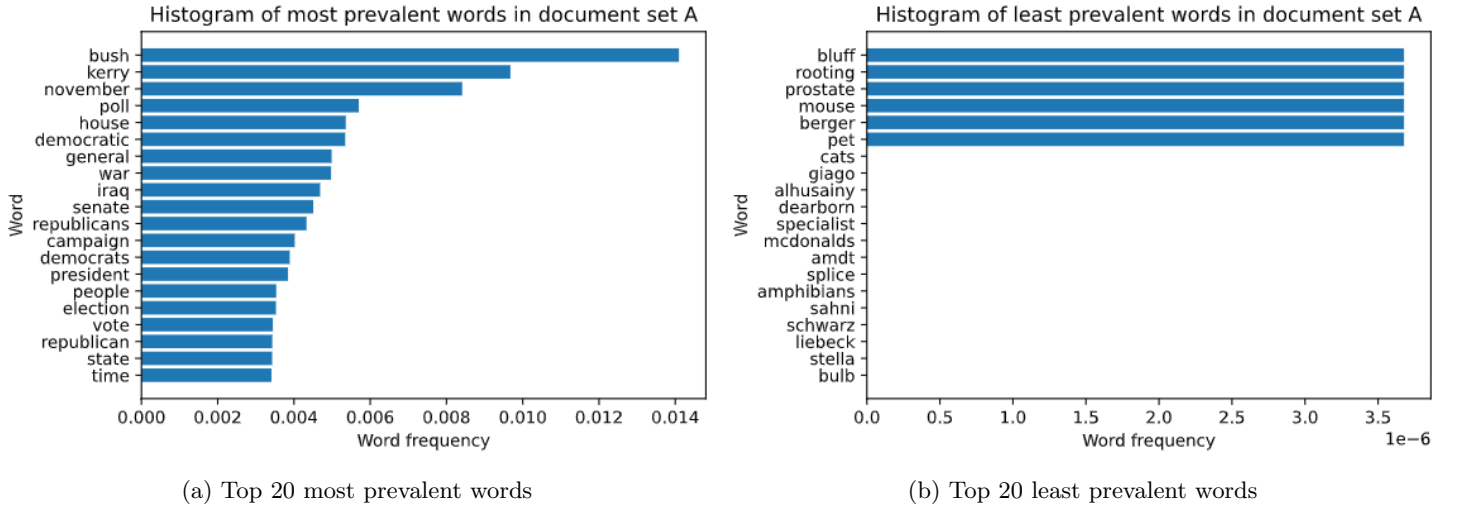


Figure 1: ML estimate of word probabilities trained on set \mathcal{A}

have that $\hat{\beta}_{\max}^{ML} := \frac{\max_i c_i}{C} = \frac{3833}{271898}$ for the word “bush” (the president - not the foliage). Likewise $\hat{\beta}_{\min}^{ML} := \frac{\min_i c_i}{C} = 0$ 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:

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

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.

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

$$\log P(\mathcal{T})|_{\min} = \log 0 = -\infty \quad (6)$$

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 $\beta \sim \text{Dir}(\beta; \alpha)$ with concentration parameter α - where α and β are the same shape. We perform Bayesian inference to obtain the posterior for β .

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

Where, c is now a vector of word counts. We say that the Dirichlet distribution is a conjugate prior to the categorical / multinomial distribution because the posterior is also Dirichlet (albeit with a new parameter). We now seek to compute the predictive distribution for a new 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 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 posterior has the effect of adding a pseudo-count a to each word prior to observing the training set \mathcal{A} .

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

As the prior is symmetric the same pseudo-count a is added to each word. Those with $c_i < C/M$ gain probability and those with $c_i > C/M$ lose it. The effect is that all word probabilities are drawn closer to C/M but importantly no probability rank orderings stay the same. The larger the value of a , the stronger this effect and the less importance is assigned to the frequency observations in \mathcal{A} .

c Testing Performance of Bayesian Predictor

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

$$\begin{aligned}
l(d) &:= \log P(\{w_{nd}^*\}_{n=1}^{N_d} | \mathcal{A}) \\
&= \log \prod_{n=1}^{N_d} P(w_{nd}^* | \mathcal{A}) \\
&= \log \prod_{m=1}^M P(w^* = m | \mathcal{A})^{c_{md}^*} \\
&= \sum_{m=1}^M c_{md}^* \log P(w^* = m | \mathcal{A}) \\
&= \sum_{m=1}^M c_{md}^* \log \frac{a + c_m}{Ma + C} \\
&= (c_d^*)^T \log \hat{\beta}^*
\end{aligned} \tag{11}$$

Where c_{md}^* is the count of word m in the unseen document d . All words are independent, hence how we can factorise the first line. 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 combining 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:

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

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

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

The expected perplexity of rolls from an n -sided die is n and so the expected perplexity for samples drawn from a uniform multinomial with M total categories would be M . This is the maximum perplexity of a multinomial with M categories as we have maximum uncertainty. We can compute the perplexities for a single document or for all documents by simply concatenating documents after one another. The results are given in table 1. Documents with a higher prevalence of common

	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} ($a = 0.1$)

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 contain rarer words than in the rest of the set.

d Bayesian Mixture Model (BMM)

We extend our model by introducing the concept of document categories. 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: $z_d \sim \text{Cat}(\theta)$ 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: $(w_{nd}|z_d = k) \sim \text{Cat}(\beta_k)$. Each β_k has prior $\beta_k \sim \text{Dir}(\gamma)$. This model is summarised in figure 2.

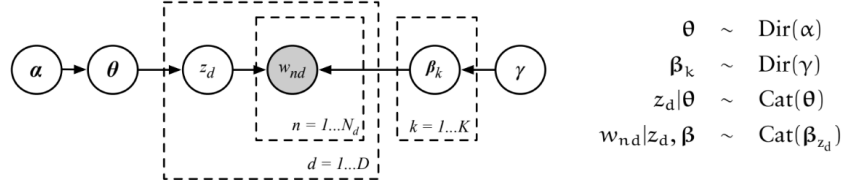


Figure 2: Bayesian Mixture Model (Mixture of Multinomials)

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 . This is simply the fraction of $\{z_d^{(i)}\}_{d=1}^D$ assigned to topic k :

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

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.

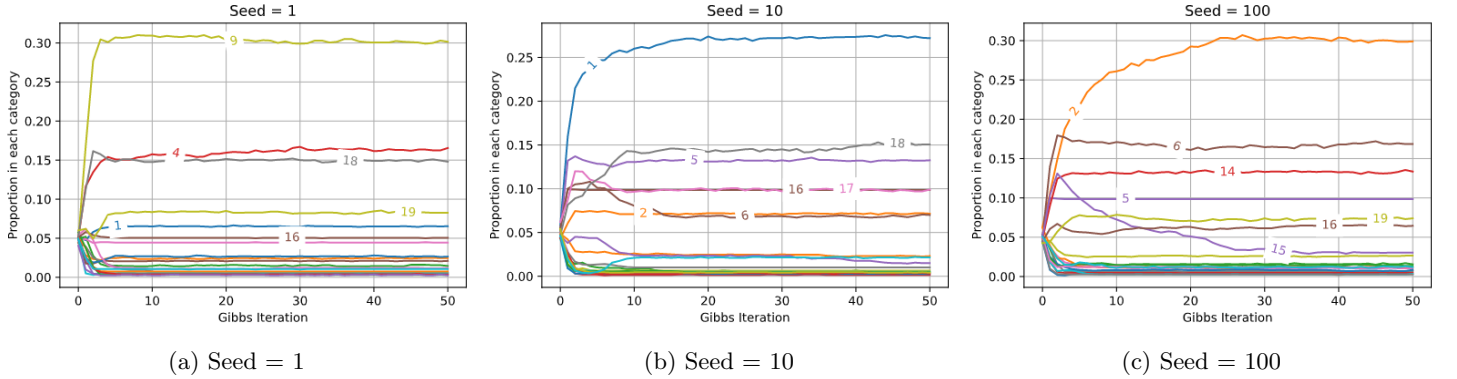


Figure 3: BMM - topic posterior against Gibbs iteration for various initialisations

Furthermore, different initialisations (seeds) yield different stationary distributions. All three Gibbs processes in figure 2 converge well enough after 30 iterations, however the exact distribution they converge to is different - even after an arbitrary relabelling of categories. The forms of the posterior are in each case different so we cannot say that we converge to the "true" posterior but rather a local optimum that is in some sense "good enough".

¹Note 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. Now every n th word in each document d has its own topic $z_{nd} \sim \text{Dir}(\theta_d)$ and the per document topic proportions parameter θ_d is specific to each document. This extension is summarised in figure 4.

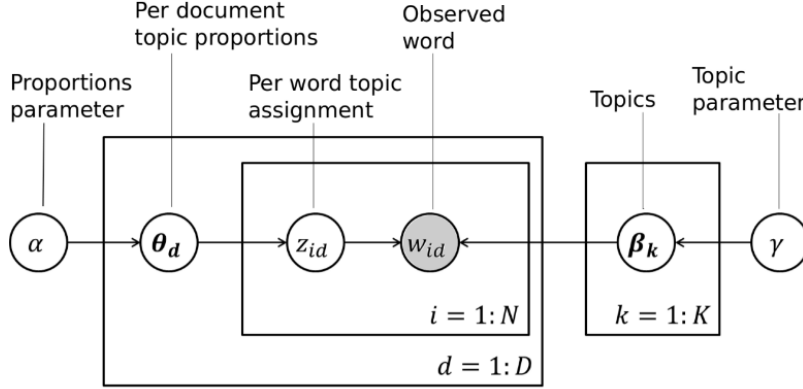


Figure 4: Latent Dirichlet Allocation model (LDA)

Now each document can have a mixture of topics drawn from θ_d . Each document’s topic posterior can be empirically calculated from the Gibbs samples (in a similar way to equation 14):

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

We use equation 15 to compute the topic posteriors for a handful of documents against Gibbs iteration and plot the results on figure 5. We see that in general each document exhibits different structure. Document 21 appears the most stable after 50 Gibbs iterations. Topics that continue to oscillate may in general be similar to one another (a consequence of having K too high - as we saw before).

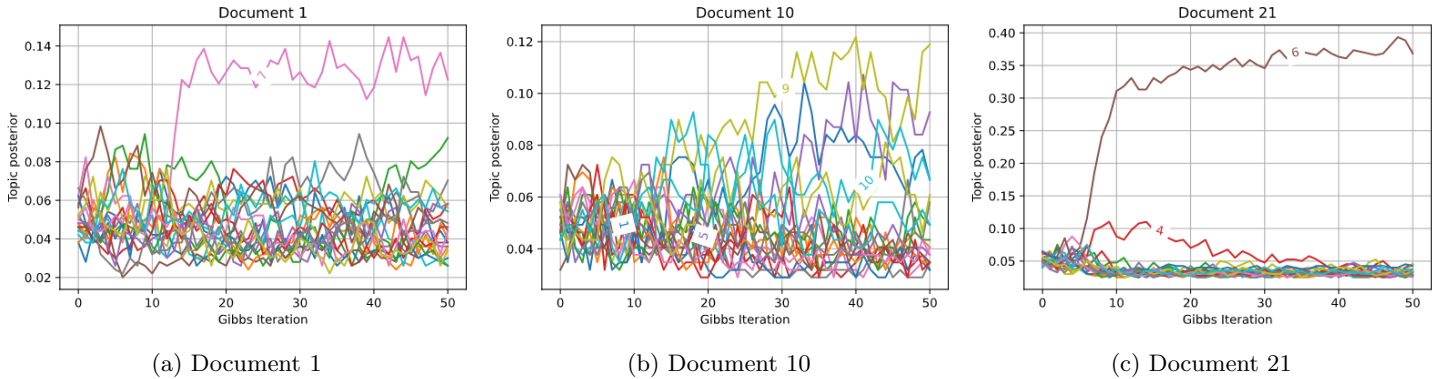


Figure 5: LDA - topic posterior for specific documents against Gibbs iteration

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 perplexities² 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 increases as the test set \mathcal{B} has higher overall log-likelihood. A model with more degrees of freedom will always fit the training set better but it is reassuring to note that we are not over-fitting as performance still improves on unseen test data.

²For 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.

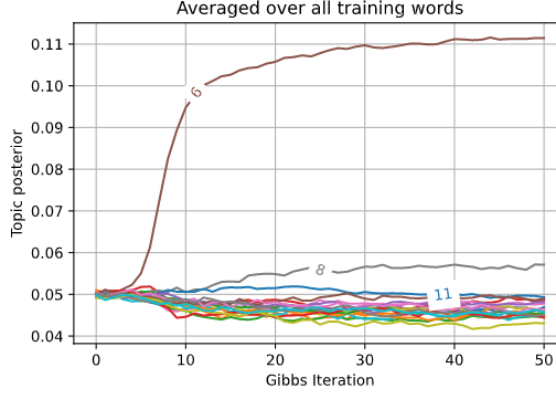


Figure 6: Topic posterior averaged over all words in training set \mathcal{A}

	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 (took several hours). Instead, to test convergence we can 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 $(w^*|z^* = k) \sim \text{Dir}(\beta_k)$. Therefore, its entropy is given by:

$$H(w^*|z^* = k) \approx \sum_{m=1}^M \beta_{km} \log \frac{1}{\beta_{km}} = -\beta_k^T \log \beta_k \quad (16)$$

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 expected value of the perplexity for a document drawn from a single topic is the exponential of the entropy as shown below:

$$\begin{aligned} \log(p(d|z = k)) &= -\frac{1}{N_d} l(d) \\ &= -\frac{1}{N_d} \sum_{m=1}^M c_m \log \beta_{km} \\ \therefore \mathbb{E}[\log(p(d|z = k))] &= \sum_{m=1}^M \mathbb{E} \left[\frac{c_m}{N_d} \right] \log \frac{1}{\beta_{km}} \\ &= \sum_{m=1}^M \beta_{km} \log \frac{1}{\beta_{km}} \\ &= H(w^*|z^* = k) \end{aligned} \quad (17)$$

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 stability of the posterior, shows 50 Gibbs sweeps to be sufficient for computing the perplexity. The log of perplexity of the test set $\log 2072.5 = 7.63$. This is modestly higher than the entropy of any particular topic, which is expected as the overall test set is a blend of topic and hence higher entropy than any individual specific topic.

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.

Words: 933

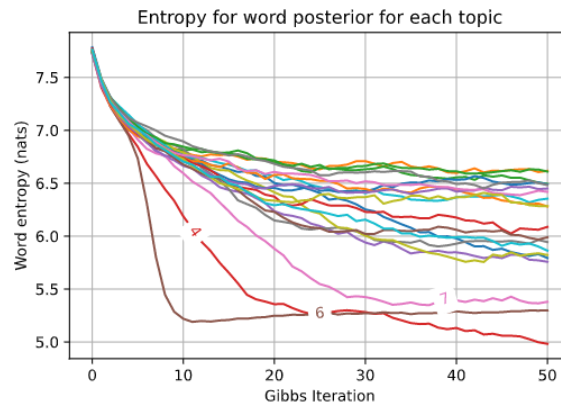
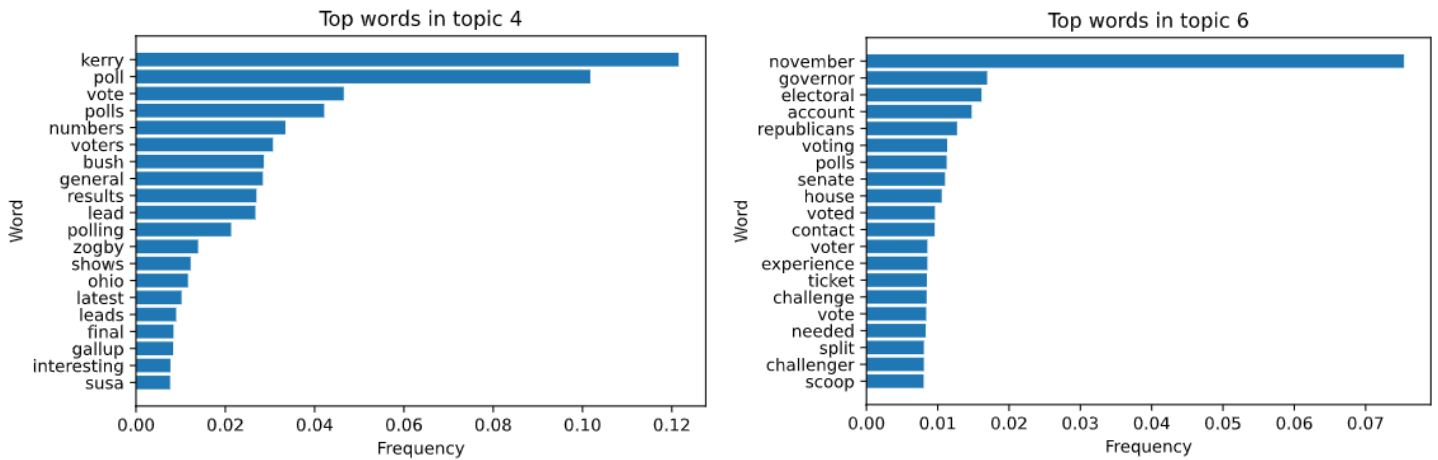


Figure 7: Entropy for each topic's categorical distribution over words



(a) Lowest word entropy - topic 4

(b) Highest posterior - topic 6

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