

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 6, 2020

Contents

1	Introduction	1
2	Questions	1
a	Maximum Likelihood	1
b	Bayesian Inference	2
c	Testing Performance	3

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 test 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 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:

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

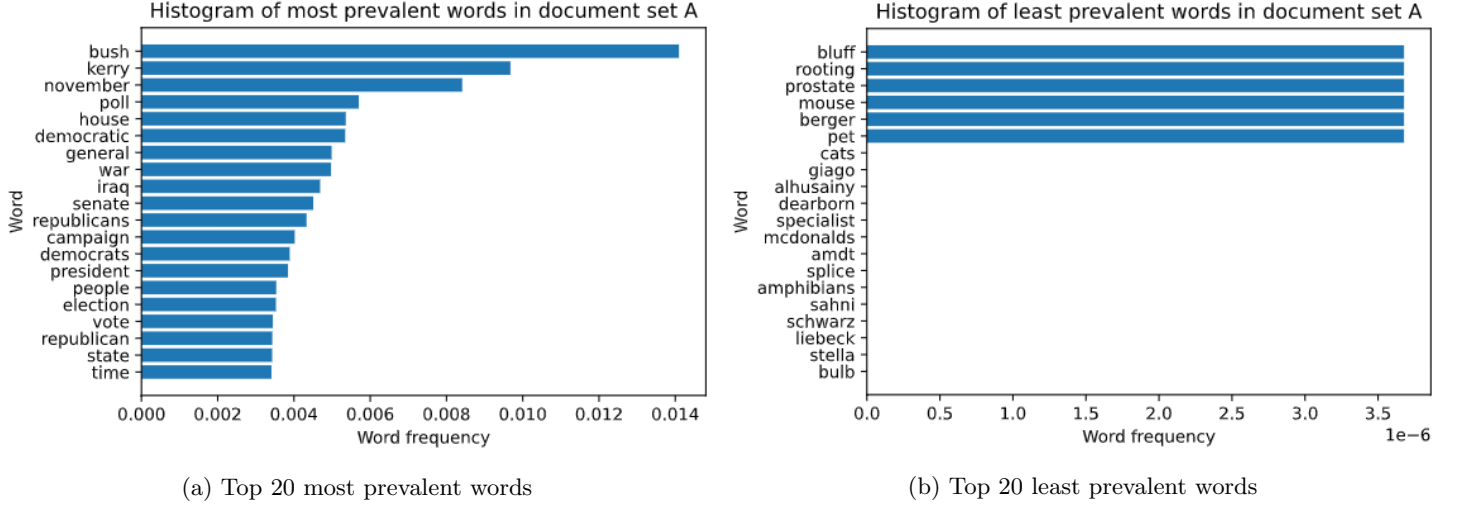


Figure 1: ML estimate of word probabilities trained on set \mathcal{A}

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.

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

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

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 .

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

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.

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

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:

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

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} .

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

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

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 $l(d)$:

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

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 $a = 10$ and apply equation 11 to test document $d = 2001$, to obtain:

$$l(d = 2001)|_{a=10} = -3682.3 \quad (12)$$

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

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

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 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.

Words: XX

	One doc $d = 2001 \in \mathcal{B}$	All docs $\forall d \in \mathcal{B}$	Uniform multinomial
per-word perplexity	4310.8	2687.1	$M = 6906$

Table 1: Perplexities of documents in test set \mathcal{B}