A framework for streamlined statistical prediction using topic models

Vanessa Glenny[1,2] **Jonathan Tuke**[1,2] **Nigel Bean**[1,2] **Lewis Mitchell**[1,2,3]
[1]ARC Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS)
[2]School of Mathematical Sciences, The University of Adelaide, SA 5005, Australia
[3]Data to Decisions CRC Stream Lead
lewis.mitchell@adelaide.edu.au

Abstract

In the Humanities and Social Sciences, there is increasing interest in approaches to information extraction, prediction, intelligent linkage, and dimension reduction applicable to large text corpora. With approaches in these fields being grounded in traditional statistical techniques, the need arises for frameworks whereby advanced NLP techniques such as topic modelling may be incorporated within classical methodologies. This paper provides a classical, supervised, statistical learning framework for prediction from text, using topic models as a data reduction method and the topics themselves as predictors, alongside typical statistical tools for predictive modelling. We apply this framework in a Social Sciences context (applied animal behaviour) as well as a Humanities context (narrative analysis) as examples of this framework. The results show that topic regression models perform comparably to their much less efficient equivalents that use individual words as predictors.

1 Introduction

For the past 20 years, topic models have been used as a means of dimension reduction on text data, in order to ascertain underlying themes, or ‘topics’, from documents. These probabilistic models have frequently been applied to machine learning problems, such as web spam filtering (Li et al., 2013), database sorting (Krestel et al., 2009) and trend detection (Lau et al., 2012).

This paper develops a methodology for incorporating topic models into traditional statistical regression frameworks, such as those used in the Social Sciences and Humanities, to make predictions. Statistical regression is a supervised method, however it should be noted the majority of topic models are themselves unsupervised.

When using text data for prediction, we are often confronted with the problem of condensing the data into a manageable form, which still retains the necessary information contained in the text. Methods such as using individual words as predictors, or n-grams, while conceptually quite simple, have a tendency to be extremely computationally expensive (with tens of thousands of predictors in a model). Except on extremely large corpora, this inevitably leads to overfitting. As such, methods that allow text to be summarised by a handful of (semantically meaningful) predictors, like topic models, gives a means to use large amounts of text data more effectively within a supervised predictive context.

This paper outlines a statistical framework for predictive topic modelling in a regression context. First, we discuss the implementation of a relatively simple (and widely used) topic model, latent Dirichlet allocation (LDA) (Blei et al., 2003), as a preprocessing step in a regression model. We then compare this model to an equivalent topic model that incorporates supervised learning, supervised LDA (sLDA) (Blei and McAuliffe, 2008).

Using topic models in a predictive framework necessitates estimating topic proportions for new documents, however retraining the LDA model to find these is computationally expensive. Hence we derive an efficient likelihood-based method for estimating topic proportions for previously unseen documents, without the need to retrain.

Given these two models hold the ‘bag of words’ assumption (*i.e.*, they assume independence between words in a document), we also investigate the effect of introducing language structure to the model through the hidden Markov topic model (HMTM) (Andrews and Vigliocco, 2010). The implementation of these three topic models as a dimension reduction step for a regression model provides a framework for the implementation of

Proc. of the 3rd Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature, pp. 61–70
Minneapolis, MN, USA, June 7, 2019. ©2019 Association for Computational Linguistics

further topic models, dependent on the needs of the corpus and response in question.

1.1 Definitions

The following definitions are used when considering topic models.

Vocabulary (V): a set of v unique elements (generally words) from which our text is composed.

Topic (ϕ): a probability distribution over the vocabulary. That is, for word i in the vocabulary, a probability $p_i \in [0, 1]$ is assigned of that word appearing, given the topic, with $\sum_{i=1}^{v} p_i = 1$. In general, there are a fixed number k of topics, $\boldsymbol{\phi} = \{\phi_1, ..., \phi_k\}$.

Document ($\mathbf{w}$): a collection of n_j units (or words) from the vocabulary. Depending on the topic model, the order of these words within the document may or may not matter.

Corpus ($\mathbf{D}$): a collection of m documents over which the topic model is applied. That is, $\mathbf{D} = \{\mathbf{w}_1, ..., \mathbf{w}_m\}$, each with length n_j, $j = 1, 2, ..., m$.

Topic proportion (θ_j): a distribution of topics over the document j. A corpus will then have an $m \times k$ matrix $\boldsymbol{\theta}$, where each row $j = 1, 2, ..., m$ corresponds to the distribution of topics over document j.

2 LDA regression model

Latent Dirichlet allocation (LDA) (Blei et al., 2003), due to its simplicity and effectiveness, continues to be the basis for many topic models today. When considering topic regression, we take LDA as our 'baseline' model; *i.e.*, we measure all subsequent models against the performance of the LDA regression model.

LDA is an unsupervised process that assumes both topics and topic proportions are drawn from Dirichlet distributions. One reason for its simplicity is that it makes the 'bag of words' assumption. LDA assumes the process outlined in Algorithm 1 when generating documents.

```
for $l = 1, 2, ..., k$ do
    generate the $k$ topics $\phi_l \sim \text{Dir}(\beta)$;
end
for $j = 1, 2, ..., m$ do
    let $n_j \sim \text{Poisson}(\xi)$, the length of document $j$;
    choose the topic proportions $\theta_j \sim \text{Dir}(\alpha)$;
    for $i = 1, 2, ..., n_j$ do
        choose the topic assignment $z_{ji} \sim \text{Multi}(\theta_j)$;
        choose a word $w_{ji} \sim \text{Multi}(\phi_{z_{ji}})$;
    end
    create the document $w_j = \{w_{ji}\}_{i=1,2,...,n_j}$;
end
```

Algorithm 1: LDA generative process.

Here, α (length k) and β (length v) are hyperparameters of the distributions of the θ_j and ϕ_l respectively.

When topic modelling, we are generally interested in inferring topic proportions $\boldsymbol{\theta} = \{\theta_1, ..., \theta_m\}$ and topics $\boldsymbol{\phi}$ themselves, given the corpus $\mathbf{D}$. That is, we wish to find

$$P(\boldsymbol{\theta}, \boldsymbol{\phi} | \mathbf{D}, \alpha, \beta) = \frac{P(\boldsymbol{\theta}, \boldsymbol{\phi}, \mathbf{D} | \alpha, \beta)}{P(\mathbf{D} | \alpha, \beta)}.$$

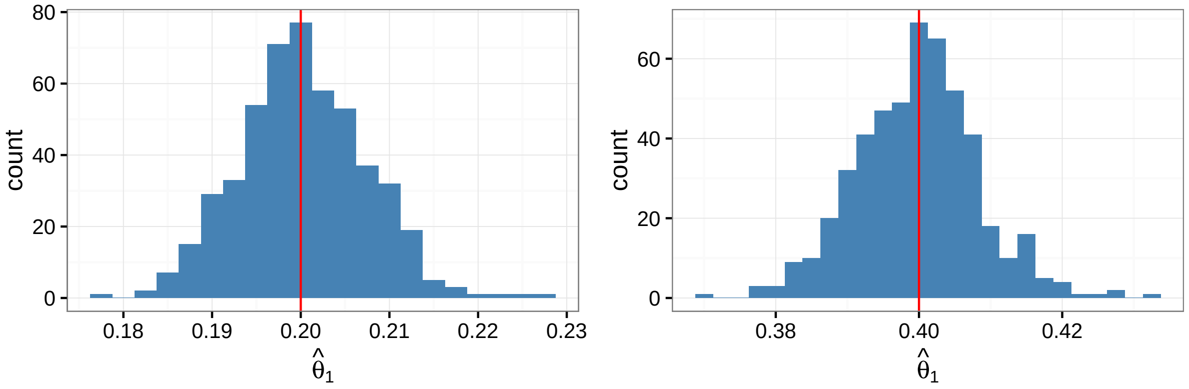
The denominator, $P(\mathbf{D}|\alpha, \beta)$, the probability of the corpus, is understandably generally intractable to compute. For the purposes of this paper, we use collapsed Gibbs sampling as outlined in Griffiths and Steyvers (2004), as an approximate method for finding the LDA model given the corpus.

2.1 Regression model and number of topics

Given an LDA model on a corpus with some corresponding response variable, we use the topic proportions generated as predictors in a regression model. More specifically, we use the topic proportions $\boldsymbol{\theta}$ as the predictors, as the amount of a document belonging to each topic may be indicative of its response.

When applying LDA as a preprocessing step to a regression model, we must also bear in mind the number of topics k we choose for the LDA model. While this number is assumed to be fixed in advance, there are various measures for determining the number that best 'fits' the corpus, such as perplexity (Blei et al., 2003) and the log likelihood measure outlined in Griffiths and Steyvers (2004).

However, given we are inferring this topic model with a specific purpose in mind, it would be prudent to include this information into the decision making process. For that reason, we choose the 'best' number of topics k to be the number



(a) $\theta_1 = 0.2$

(b) $\theta_1 = 0.4$

Figure 1: Histograms of the maximum likelihood estimates of θ_1 for corpora of two topics, given relative true values of 0.2 and 0.4.

that reduces the cross validation prediction error (CVPE) (Geisser, 1975) of the corresponding LDA regression model, found from K-fold cross validation of the model on the corpus. The CVPE is here defined to be

$$\text{CVPE}_K = \sum_{i=1}^{K} \frac{m_i}{m} \text{MSE}_i,$$

where K is the number of folds, m_i is the number of documents in the ith fold, and m the total number of documents in the corpus. The mean square error for the ith fold, denoted by MSE_i, is defined as

$$\text{MSE}_i = \sum_{j \in C_i} \frac{1}{m_i} \left(y_j - \hat{y}_j\right)^2,$$

where $\hat{y}_j$ is the model estimate of response y_j for all documents in the set C_i, the ith fold. It follows that the better a model performs, the smaller the MSE and thus the CVPE.

While we choose the best number of topics based on the information in the regression model, it should be noted that LDA is still unsupervised, and that the topics have not been generated with the response in mind.

2.2 Introducing new documents

When it comes to prediction, we generally have a corpus for which we find our regression model, and use this model to predict the response of new documents that are not in the original corpus. Because our regression model requires us to know θ_j, the topic proportion, for any new document j, we have two options. Either the topic model can be retrained with the new document added to the corpus, and the regression model retrained with the new topics on the old documents, or the topic proportions can be found based on the existing topic model.

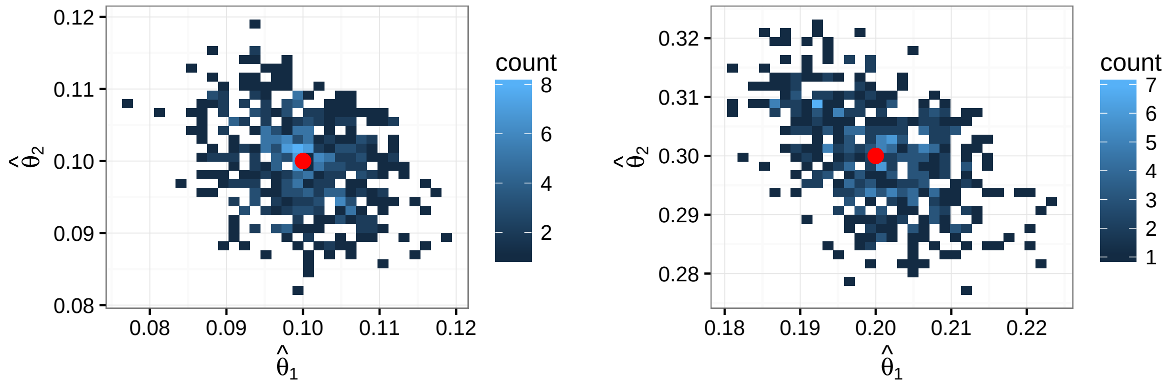
For both efficiency's sake (*i.e.*, to avoid retraining the model for every prediction), and for the sake of true prediction, the second option is preferable. Particularly in cross validation, it is necessary to have a completely distinct traning and test set of data. In retraining a topic model with new documents, we do not have a clear distinction between the two sets.

Blei et al. (2003) outline a procedure for estimating the topic proportions of a held-out document, however this procedure follows a posterior approach that requires variationally inferring the posterior parameters, which are then used to approximate the expected number of words belonging to each topic, as an estimate for θ_j.

We propose here a likelihood-based approach to estimation of topic proportions of new documents, by treating the problem as a case of maximum likelihood estimation. That is, we want to find $\hat{\theta}_j$, the estimate of θ_j that maximises the likelihood of document j occurring, given our existing topic model. Therefore, we aim to maximise

$$\begin{aligned} L(\theta_j) &= f(\mathbf{w}_j|\theta_j) \\ &= f(w_{j1}, ..., w_{jn_j}|\theta_j), \end{aligned}$$

where $w_{j1}, ..., w_{jn_j}$ are the words in document j. As LDA is a 'bag of words' model, we are able to



(a) $\{\theta_1, \theta_2\} = \{0.1, 0.1\}$

(b) $\{\theta_1, \theta_2\} = \{0.2, 0.3\}$

Figure 2: Histograms of the maximum likelihood estimates of $\{\theta_1, \theta_2\}$ for corpora of three topics, given relative true values of $\{0.1, 0.1\}$ and $\{0.2, 0.3\}$.

express this as

$$L(\theta_j) = \prod_{i=1}^{n_j} f(w_{ji}|\theta_j).$$

The law of total probability gives

$$L(\theta_j) = \prod_{i=1}^{n_j} \sum_{l=1}^{k} f(w_{ji}|z_{ji} = l, \theta_j) f(z_{ji} = l|\theta_j),$$

where z_{ji} is the topic assignment for the ith word in document j. However, as the choice of word w_{ji} is independent of the topic proportions θ_j given its topic assignment z_{ji}, we can write

$$L(\theta_j) = \prod_{i=1}^{n_j} \sum_{l=1}^{k} f(w_{ji}|z_{ji} = l) f(z_{ji} = l|\theta_j).$$

The likelihood is now expressed as the products of the topic proportions and the topics themselves.

$$L(\theta_j) \quad = \quad \prod_{i=1}^{n_j} \sum_{l=1}^{k} \phi_{l,w_{ji}} \theta_{jl}.$$

If we express the document as a set of word counts $\mathbf{N} = \{N_1, ..., N_v\}$, where N_i is the number of times the ith word of the vocabulary appears in document j, then we can write the log likelihood of θ_j as

$$l(\theta_j) = \mathbf{N} \cdot \log(\theta_j \phi).$$

In order to deal with words that appear in a new document, and not the original corpus, we assign a probability of 0 to any such word of appearing in any of the k topics; this is equivalent to removing those words from the document.

To demonstrate the effectiveness of this method for estimation, we generate documents for which we know the topics and topic proportions. Suppose there exists a corpus comprising of two topics, with a vocabulary of 500 words. Given an assumed LDA model, we generate 500 documents with lengths between 5,000 and 10,000 words.

Given our newly generated documents, and known topics ϕ, we are able to test the validity of the MLE process outlined above by finding the estimates $\hat{\theta}_j$ for each document j and comparing them to known topic proportions θ_j. Figure 1 shows the results of the MLE method for finding topic proportion estimates for documents with certain true values of θ_j. From these figures, there is a tight clustering around the true value θ_j, and thus it is reasonable to assume that the MLE process for estimating the topic proportions of a new document given previously existing topics is sound. This process also holds for greater numbers of topics, as evidenced in Figure 2, which estimates topic proportions for a three-topic document.

3 sLDA regression model

LDA is an unsupervised process, which does not take into account the response variable we are predicting when inferring topics. Several supervised methods have been developed to incorporate this knowledge, generally for the purpose of finding 'better' topics for the corpus in question. Notably, supervised LDA (sLDA) (Blei and McAuliffe, 2008) builds on the LDA model by assuming that some response y_j is generated alongside each document $j = 1, 2, ..., m$ in the corpus, based on the topics prevalent in the document. When inferring the sLDA model, we are therefore inclined to find topics that best suit the response and therefore the prediction problem at hand.

Unlike LDA, we treat the topics ϕ as unknown constants rather than random variables. That is, we are interested in maximising

$$P\left(\boldsymbol{\theta}, \mathbf{z} | \mathbf{D}, \mathbf{y}, \boldsymbol{\phi}, \alpha, \eta, \sigma^2\right),$$

where η and σ^2 are parameters of the normally distributed response variable, $y_j \sim N(\eta^T \bar{z}_j, \sigma^2)$, where $\bar{z}_j = (1/n_j)\sum_{i=1}^{n_j} z_{ji}$.

As with LDA, this probability is computationally intractable, and thus we require an approximation method for model inference. For the purposes of this paper, we use a variational expectation-maximisation (EM) algorithm, as outlined in Blei and McAuliffe (2008).

When it comes to choosing the model with the most appropriate number of topics for the regression problem at hand, we use the same method as outlined for the LDA regression model in Section 2.1.

The method behind sLDA is specifically developed for prediction. As such, we are able to compute the expected response y_j from the document $\mathbf{w}_j$ and the model $\{\alpha, \boldsymbol{\phi}, \eta, \sigma^2\}$. For a generalised linear model (as we use in this paper), this is approximated by

$$E\left[Y_j | \mathbf{w}_j, \alpha, \boldsymbol{\phi}, \eta, \sigma^2\right] \approx E_q\left[\mu\left(\eta^T \bar{\mathbf{z}}_j\right)\right],$$

where $\mu\left(\eta^T \bar{\mathbf{z}}_j\right) = E\left[Y_j | \zeta = \eta^T \bar{\mathbf{z}}_j\right]$ and ζ is the natural parameter of the distribution from which the response is taken. Again, further detail on this method is found in Blei and McAuliffe (2008).

4 HMTM regression model

Topic modelling is designed as a method of dimension reduction, and as such we often deal with large corpora that cannot otherwise be analysed computationally. Given the complexity of human language, we therefore have to choose what information about our corpus is used to develop the topic model. The previous two models, LDA and sLDA, have relied on the 'bag of words' assumption in order to maintain computational efficiency. While for some corpora, the loss of all information relating to language and document structure may not have a particularly large effect on the predictive capability of the topic model, this may not hold for all prediction problems.

One simple way of introducing structure into the model is through a hidden Markov model (HMM) structure (Baum and Eagon, 1967; Baum et al., 1970); in fact, there already exist multiple topic models which do so. We look here at the hidden Markov topic model (HMTM) (Andrews and Vigliocco, 2010), which assumes that the topic assignment of a word in a document is dependent on the topic assignment of the word before it. That is, the topic assignments function as the latent states of the HMM, with words in the document being the observations. The HMTM assumes the generative process outlined in Algorithm 2 for documents in a corpus.

```
for $l = 1, 2, ..., k$ do
    generate topics $\phi_l \sim$ Dir($\beta$);
end
for $j = 1, 2, ...m$ do
    generate starting probabilities
      $\boldsymbol{\pi}_j \sim$ Dir($\alpha$);
    for $l = 1, 2, ..., k$ do
        generate the $l$th row of the transition
          matrix, $\boldsymbol{\Theta}_j$, $\Theta_{jl} \sim$ Dir($\gamma_l$);
    end
    choose the topic assignment for the first
      word $z_{j1} \sim$ Multi($\boldsymbol{\pi}_j$);
    select a word from the vocabulary
      $w_{j1} \sim$ Multi($\phi_{z_{j1}}$);
    for $i = 2, 3, ..., n_j$ do
        choose the topic assignment $z_{ji}$ based
          on transition matrix $\boldsymbol{\Theta}_j$;
        select a word from the vocabulary
          $w_{ji} \sim$ Multi($\phi_{z_{ji}}$);
    end
    create the document $\mathbf{w}_j = \{w_{ji}\}_{i=1,...,n_j}$;
end
```

Algorithm 2: HMTM generative process.

Here, α, β and $\boldsymbol{\gamma} = \{\gamma_1, ..., \gamma_k\}$ are Dirichlet priors of the starting probabilities, topics and transition probabilities respectively.

When it comes to prediction, we are able to use the transition matrices for each document $\boldsymbol{\Theta}_j$ as predictors, but to keep consistency with the previous models we take the equilibrium distributions of the matrices as the topic proportions θ_j. That is, we find θ_j such that

$$\theta_j \boldsymbol{\Theta_j} = \theta_j, \quad \text{and} \quad \theta_j \mathbf{e} = 1.$$

This also fits with the concept of topic models as a form of dimension reduction, allowing $k - 1$

variables, as opposed to $k(k-1)$ when using the transition matrix $\mathbf{\Theta}_j$. As models are often fit using hundreds of topics (Blei, 2012; Griffiths and Steyvers, 2004), this makes models faster to compute. We choose the number of topics k here with the same method outlined in Section 2.1.

4.1 Introducing new documents

Like with the LDA regression model, we require a method for estimating the topic proportion θ_j of any new documents from which we are predicting a response, that does not involve retraining the entire model. To do so, we rely on techniques used for HMMs; specifically, we use a modified Baum-Welch algorithm.

The Baum-Welch algorithm is used as an approximate method to find an HMM $\Omega = \{\mathbf{\Theta}, \boldsymbol{\phi}, \boldsymbol{\pi}\}$, given some observed sequence (in this case, a document). However, the key difference here is that our emission probabilities (or topics) $\boldsymbol{\phi}$ are common across all documents in our corpus, and thus when introducing any new documents for prediction we assume that we already know them. Given the Baum-Welch algorithm calculates forward and backward probabilities based on an assumed model, and updates estimates iteratively, we may simply take our assumed $\boldsymbol{\phi}$ found from the initial HMTM as the truth and refrain from updating the emission probabilities.

We are generally dealing with very small probabilities in topic modelling - $\boldsymbol{\phi}$ generally has tens of thousands of columns (the length of the vocabulary) over which probabilities must sum to one. While in theory this does not change how we would approach parameter estimation, computationally these probabilities are frequently recognised as zero. To make the process more numerically stable, we implement the adapted Baum-Welch algorithm demonstrated and justified in Shen (2008).

While we are ultimately interested in finding topic proportions θ_j for prediction, the Baum-Welch algorithm finds the transition matrix $\mathbf{\Theta}_j$ for some document. We are able to deal with this in the same way as finding the original HMTM regression model, by taking θ_j to be the equilibrium probabilities of $\mathbf{\Theta}_j$.

5 Testing the topic regression models

To demonstrate the use of topic models in a regression framework, we apply them to a problem involving online advertisements. Specifically, we have a corpus containing 4,151 advertisements taken from the trading website, Gumtree[1], pertaining to the sale of cats in Australia, and hand-labelled by an expert. Of these advertisements, 2,187 correspond to relinquished cats and 1,964 to non-relinquished. We train a model to predict 'relinquished status' from the text of an advertisement, using a topic regression model. A cat is considered to be relinquished if it is being given up by its owner after a period of time, as opposed to cats that are sold, either by breeders or former owners.

In order to improve efficiency and model quality, we first clean our text data. Details on the cleaning steps can be found in Appendix A.1.

5.1 Word count model

Before investigating regression models that use topic proportions as predictors, it is worth developing a 'gold standard' model, *i.e.*, a model whose predictive capability we aim to match with our topic regression models. Because the problem here involves a relatively small corpus (advertisements with a median word count of 35), we are able to compare our topic regression models to a model that uses individual words as its predictors.

In a much larger corpus, this kind of prediction would be cumbersome to compute - hence our reliance on topic models and other dimension reduction techniques.

Because we are predicting a categorical, binary variable, we use logistic regression. Rather than using all words in the corpus (as this would drastically overfit the model), we use a step-up algorithm based on the Akaike information criterion (AIC) (Akaike, 1974) to choose the most significant words for the model, without overfitting.

Instead of applying the step-up process to the entire vocabulary (of exactly 13,000 words), we apply it to the 214 most common words (*i.e.*, words that appear in at least 2.5% of the documents in the corpus). The chosen model uses 97 predictors, with coefficients appearing consistent with what you would expect from the problem: for example, the word *kitten* is indicative of non-relinquished advertisements, while *cat* is the opposite, which is expected as younger cats are less likely to be relinquished.

To assess the predictive capability of this and other models, we require some method by which

[1] `www.gumtree.com.au`

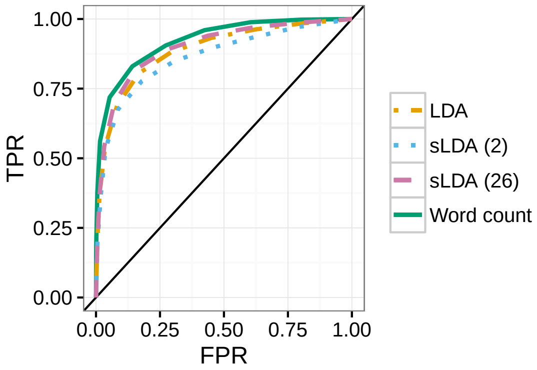


Figure 3: Threshold-averaged ROC curves of the word count model, LDA regression model, and sLDA regression models with two and 26 topics respectively.

we can compare the models. For that purpose, we use receiver operating characteristic (ROC) curves as a visual representation of predictive effectiveness. ROC curves compare the true positive rate (TPR) and false positive rate (FPR) of a model's predictions at different threshold levels. The area under the curve (AUC) (between 0 and 1) is a numerical measure, where the higher the AUC is, the better the model performs.

We cross-validate our model by first randomly splitting the corpus into a training set (95% of the corpus) and test set (5% of the corpus). We then fit the model to the training set, and use it to predict the response of the documents in the test set. We repeat this process 100 times. The threshold-averaged ROC curve (Fawcett, 2006) is found from these predictions, and shown in Figure 3. Table 1 shows the AUC for each model considered.

Model	AUC	95% CI
Word count	0.9264	$(0.9234, 0.9294)$
LDA	0.8913	$(0.8871, 0.8955)$
sLDA (2 topics)	0.8588	$(0.8534, 0.8642)$
sLDA (26 topics)	0.9030	$(0.8988, 0.9073)$

Table 1: TArea under the curve (AUC) for the models used on the Gumtree dataset, with their 95% confidence intervals.

5.2 Topic regression models

Using the method outlined in Section 2.1, we choose the LDA regression model with 26 topics as the 'best' for this problem. Inspection of the top words included in these 26 topics shows individual topics associated with different breeds (e.g., 'persian', 'manx') as well as urgency of selling (e.g., 'urgent', 'asap'), suggesting that the model is not overfit to the data. We generate a threshold-averaged ROC curve using the same cross validation method as earlier, yielding an area under the curve (AUC) of 0.8913. The curve can be seen in Figure 3. While not as high as the AUC for the word count model, the LDA regression model is significantly more efficient, taking only 3% of the time to calculate.

We can compare this result to that of an sLDA regression model. The model chosen for this problem has two topics, giving a threshold-averaged ROC curve under cross validation with an AUC of 0.8588. It is surprising that the LDA regression model should outperform sLDA, as sLDA incorporates the response variable when finding the most appropriate topics. However, this can be attributed to the number of topics in the model: the sLDA regression model with 26 topics outperforms the LDA model, with an AUC of 0.9030.

The word count model still outperforms the sLDA model, however once again the topic regression model is significantly more efficient, taking only 0.6% of the time to calculate. Further details on the models and their calculation can be found in Appendix A.2.

6 Incorporating language structure

When evaluating the usefulness of incorporating document structure into a topic model for regression, we require a corpus and problem that we would expect would be heavily influenced by this structure. To understand the predictive capability of the HMTM regression model over that of the more simplistic LDA, we therefore consider predicting the storylines of the 2003 film *Love Actually*[2], known for its interwoven yet still quite distinct storylines. We therefore ask if we are able to predict to which storyline a scene belongs, based on the dialogue in that scene.

The film consists of 79 scenes, each pertaining to one of 10 storylines. The scenes were hand-classified by storyline, and their dialogue forms the documents of our corpus. We once again clean our data; more detail can be found in Appendix A.1.

[2] www.imdb.com/title/tt0314331/

6.1 Word count model

As with the Gumtree dataset, we first construct a word count model against which we can measure the performance of our topic regression models. Once again, this can be done because we are working with a small corpus; otherwise, we would generally consider this approach to be computationally too heavy.

As we have a categorical, non-binary response variable (storyline) with 10 levels, we use a multinomial logistic regression model. We again use a step-up process with AIC as the measure to determine which words in our vocabulary to use as predictors in our model. As our vocabulary consists of only 1,607 unique words, we consider all of them in our step-up process. After applying this process, the model with three predictors, *minister*, *night* and *around*, is chosen.

We are no longer able to easily apply ROC curves as a measure of performance to this problem, as we are dealing with a non-binary response. We instead use a Brier score (Brier and Allen, 1951), a measure for comparing the predictive performance of models with categorical responses. The Brier score is

$$\text{BS} = \frac{1}{m} \sum_{j=1}^{m} \sum_{i=1}^{s} \left(\hat{y}_{ji} - o_{ji}\right)^2 ,$$

where $\hat{y}_{ji}$ is the probability of document j belonging to storyline i, and $o_{ji} = 1$ if document j belongs to storyline i, and 0 otherwise, for document $j = 1, 2, ..., m$ and storyline $i = 1, 2, ..., s$. Each term in the sum goes to zero the closer the model gets to perfect prediction, and as such our aim is to minimise the Brier score in choosing a model.

For each document in the corpus, we find the probabilities of each outcome by using the remaining 78 documents (or training dataset) as the corpus in a multinomial logistic regression model with the same three predictors as found above. Due to the fact that the training dataset here is smaller than the Gumtree dataset, we perform leave-one-out cross validation on each document in the corpus (rather than using a 95/5 split). We then predict the outcome based on the words found in the left-out document (or test dataset), and repeat for all 79 scenes. However, due to the short length of some scenes, and the fact that unique words must be thrown out, we restrict the testing to 57 of the 79 scenes: the remaining scenes do not generate a numerically stable approximation for θ_j for the HMTM regression model.

The Brier score calculated using this method for the step-up word count model is 0.8255.

6.2 Topic regression models

For the LDA regression model for this problem, we determine the 'best' number of topics k to be 16. As with the word count model, we use the Brier score to evaluate the performance of this model compared to others in the chapter. We again use the leave-one-out cross validation approach to predict the probabilities of a scene belonging to each storyline.

The Brier score found for the LDA regression model is 1.6351. While this is higher and therefore worse than the Brier score for the word count model above, this is not unexpected and we are more interested in seeing how the LDA model fares against other topic models.

We compare these results to the HMTM regression model, as outlined in Section 4. We choose the model with 12 topics, according to the CVPE. The Brier score calculated from 57 scenes for the HMTM regression model is 1.5749. While still not up to the standard of the word count model at 0.8255, this appears to be a slight improvement on the LDA model, meaning that dropping the 'bag of words' assumption may in fact improve the predictive performance of the model. However, it should be kept in mind that the LDA model is better at handling short documents. It would be worth applying these models to corpora with longer documents in future, to see how they compare. Further details on the computation of these models can be found in Appendix A.2.

One of the motivating ideas behind having topic dependencies between consecutive words, as in the HMTM model, is that some documents will have a predisposition to stay in the same topic for a long sequence, such as a sentence or a paragraph. This argument particularly applies to narrative-driven corpora such as the *Love Actually* corpus. To that end, we may adapt the HMTM described above so that the model favours long sequences of the same topic, by adjusting the Dirichlet priors of the transition probabilities, $\gamma = \{\gamma_1, ..., \gamma_k\}$, to favour on-diagonal elements. By specifying these priors to be

$$\gamma_{ls} = \begin{cases} 0.99 + 0.01/k & \text{if} \quad l = s \\ 0.01/k & \text{elsewhere,} \end{cases}$$

for $l = 1, 2, ..., k$, we choose the persistent HMTM regression model with three topics. This results in a Brier score of 0.9124, which is a massive improvement on the original HMTM regression model and makes it very competitive with the word count model. Table 2 summarises these results.

Model	**Accuracy**	**Brier score**
Word count	26.58	0.8255
LDA	12.66	1.6351
HMTM	14.04	1.5749
Persistent HMTM	15.58	0.9124

Table 2: Table of the percentage of hard classifications of storylines for each left-out scene in the corpus that are correct, alongside the Brier score, for each model.

7 Discussion and further research

This paper outlines and implements a streamlined, statistical framework for prediction using topic models as a data processing step in a regression model. In doing so, we investigate how various topic model features affect how well the topic regression model makes predictions.

While this methodology has been applied to three specific topic models, the use of any particular topic model depends heavily on the kind of corpus and problem at hand. For that reason, it may be worth applying this methodology to incorporate different topic models in future, depending on the needs of the problem at hand.

In particular, we investigate here the influence of both supervised methods, and the incorporation of document structure. A logical next step would be to propose a model that incorporates these two qualities, in order to see if this improves predictive capability on corpora with necessary language structure.

References

Hirotugu Akaike. 1974. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723.

Mark Andrews and Gabriella Vigliocco. 2010. The hidden Markov topic model: A probabilistic model of semantic representation. *Topics in Cognitive Science*, 2(1):101–113.

Leonard E Baum and J A Eagon. 1967. An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. *Bulletin of the American Mathematical Society*, 73(3):360–363.

Leonard E Baum, Ted Petrie, George Soules, and Norman Weiss. 1970. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171.

David M Blei. 2012. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84.

David M Blei and Jon D McAuliffe. 2008. Supervised topic models. In *Advances in Neural Information Processing Systems*, pages 121–128.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Glenn W Brier and Roger A Allen. 1951. Verification of weather forecasts. In *Compendium of Meteorology*, pages 841–848. Springer.

Tom Fawcett. 2006. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874.

Seymour Geisser. 1975. The predictive sample reuse method with applications. *Journal of the American Statistical Association*, 70(350):320–328.

Thomas L Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5228–5235.

Ralf Krestel, Peter Fankhauser, and Wolfgang Nejdl. 2009. Latent Dirichlet allocation for tag recommendation. In *Proceedings of the Third ACM Conference on Recommender Systems*, pages 61–68. ACM.

Jey Han Lau, Nigel Collier, and Timothy Baldwin. 2012. On-line trend analysis with topic models:\# twitter trends detection topic model online. *Proceedings of COLING 2012*, pages 1519–1534.

Jiwei Li, Claire Cardie, and Sujian Li. 2013. TopicSpam: A topic-model based approach for spam detection. In *51st Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference*, volume 2, pages 217–221.

Martin Porter. 2001. Snowball: A language for stemming algorithms. Available at *http://snowballstem.org*. Last checked 13 December 2017.

Dawei Shen. 2008. Some mathematics for HMM. *Massachusetts Institute of Technology*.

A Appendix

A.1 Text cleaning

The following steps were taken to clean the Gumtree corpus:

- removal of punctuation and numbers,

- conversion to lower case,
- removal of stop words (*i.e.*, common words such as *the* and *for* that contribute little lexically), and
- removal of grammatical information from words (*i.e.*, stemming).

When stemming words in this paper, we use the stemming algorithm developed by Porter for the Snowball stemmer project (Porter, 2001). Similarly, when removing stop words, we use the (English language) list compiled, again, in the Snowball stemmer project.

In cleaning the *Love Actually* corpus, we perform the first three steps outlined here. However, unlike with the Gumtree dataset, we do not stem words, as grammatical information is more pertinent when incorporating language structure.

A.2 Topic model inference

For each topic model, we choose the best number of topics from models generated with between two and 40 topics.

For the LDA models found in this paper, we use the *LDA* function from the R package **topicmodels**, with the following parameters:

- `burnin` = `1000`,
- `iterations` = `1000`, and
- `keep` = `50`.

The sLDA model in this paper was found using the `slda.em` function from the R package **lda**, with the following parameters:

- `alpha` = `1.0`,
- `eta` = `0.1`,
- `variance` = `0.25`,
- `num.e.iterations` = `10`, and
- `num.m.iterations` = `4`.

We use the Python code from Andrews and Vigliocco (2010) for the generation of our HMTM.