topicmodels: An R Package for Fitting Topic Models

Bettina Grün
WU Wirtschaftsuniversität Wien

Kurt Hornik
WU Wirtschaftsuniversität Wien

Abstract

Topic models allow the probabilistic modelling of term frequency occurrences in documents. The fitted model can for example be used to estimate the similarity between documents as well as between a set of specified keywords using an additional layer of latent variables which are referred to as topics. The R package **topicmodels** provides basic infrastructure for fitting topic models based on data structures from the text mining package **tm**. The package includes interfaces to two algorithms for fitting topic models: the Variational Expectation-Maximization algorithm provided by David M.~Blei and co-authors and an algorithm using Gibbs Sampling by Xuan-Hieu Phan and co-authors.

Keywords:~Gibbs sampling, R, text analysis, topic model, variational EM.

1. Introduction

Topic models are generative models which provide a probabilistic framework for the term frequency occurrences for documents in a given corpus. They are bag-of-word models, i.e., it is assumed that the information in which order the words occur in a document is negligible. This assumption is also referred to as the *exchangeability* assumption for the words in a document (Blei, Ng, and Jordan 2003). In order to model dependencies between words, i.e., to allow related words to occur more likely together in a document, topics are introduced as latent variables. Topic models assume that the content of each document is based on certain topics and these underlying topics induce a certain word distribution for the document. Each document therefore has its own topic distribution. The Latent Dirichlet Allocation (LDA; Blei *et~al.* 2003) model is the basic topic model where topics are assumed to be uncorrelated. The Correlated Topics Model (CTM; Blei and Lafferty 2007) is an extension of the LDA model where correlations between topics are allowed. An introduction to topic models is given in Steyvers and Griffiths (2007) and Blei and Lafferty (2009). Topic models have previously been used for ad-hoc information retrieval (Wei and Croft 2006), geographical information retrieval (Li, Wang, Xie, Wang, and Ma 2008) and the analysis of the development of ideas over time in the field of computational linguistics (Hall, Jurafsky, and Manning 2008).

C code for fitting the LDA model (`http://www.cs.princeton.edu/~blei/lda-c`) and the CTM (`http://www.cs.princeton.edu/~blei/ctm-c`) is available under the GPL from David M.~Blei and co-authors, who were introducing these models in their papers. The method used for fitting the models is the Variational Expectation-Maximization (VEM) algorithm. Other implementations for fitting topic models—especially of the LDA model—are available. The standalone program lda (Mochihashi 2004, `http://chasen.org/~daiti-m/dist/lda/`) provides standard VEM estimation. The authors of the lda package indicate that according to their experiments their package runs about 4 to 10 times faster than the code by Blei and co-authors. For Bayesian estimation using Gibbs sampling several implementations are available including the following. GibbsLDA++ (Phan, Nguyen, and Horiguchi 2008, http://gibbslda.sourceforge.net/) is available under the GPL. The Matlab Topic Modeling Toolbox 1.3.2 (Griffiths and Steyvers 2004, http://psiexp.ss.uci.edu/research/programs_data/toolbox.htm) is free for scientific use. A license must be obtained from the authors to use it for commercial purposes. MALLET (McCallum 2002, http://mallet.cs.umass.edu) is released under the CPL and is a Java-based package which is more general in allowing for statistical natural language processing, document classification, clustering, topic modeling using LDA, information extraction, and other machine learning applications to text.

The R package **topicmodels** provides an interface to the code for fitting an LDA model and a CTM with the VEM algorithm as implemented by Blei and co-authors and to the code for fitting an LDA topic model with Gibbs sampling written by Phan and co-authors. In package **topicmodels** the respective code is directly called through an interface at the C level avoiding file input and output, i.e., the functionality for data input and output in the original code was substituted to allow direct use of R objects as input and to return S4 objects as output to R. The same main function allows fitting the LDA model with different estimation methods returning objects only slightly different in structure. In addition the strategies for model selection and inference are applicable in both cases. This allows for easy use and comparison of both current state-of-the-art estimation techniques for topic models.

CRAN (http://CRAN.R-project.org) also features package **lda** (Chang 2009) which provides collapsed Gibbs sampling methods for LDA and related topic models, with the Gibbs sampler implemented in C. Similar to package **topicmodels**, package **lda** can be used to fit the LDA model using Gibbs sampling. In addition the mixed membership stochastic blockmodel (Airoldi, Blei, Fienberg, and Xing 2008) and supervised topic models (Blei and McAuliffe 2008) can be fitted using the same C code function for the Gibbs sampling step. Furthermore, the relational topic model (RTM; Chang and Blei 2009) and the Networks Uncovered By Bayesian Inference (NUBBI) model (Chang, Boyd-Graber, and Blei 2009) can be fitted with separate C code functions. All models in package **lda** are fitted using Gibbs sampling for determining the posterior probability of the latent variables. EM wrappers are provided which build on this functionality for the E-step. Note that this implementation therefore differs in general from the proposed estimation technique in the original papers introducing these model variants, where the VEM algorithm is usually applied.

This paper is structured as follows: Section˜2 introduces the specification of topic models, outlines the estimation with the VEM as well as Gibbs sampling and gives an overview of preprocessing steps and methods for model selection and inference. The main fitter functions in the package and the helper functions for analyzing a fitted model are presented in Section˜3. An illustrative example for using the package is given in Section˜4 where topic models are fitted to the corpus of abstracts in the *Journal of Statistical Software*. The corpus is rather small with only hundreds of documents and a rather limited vocabulary. In addition it consists of documents from very similar content areas. These two factors might be the reason that topic models do not perform particularly well on this data set.

2. Topic model specification and estimation

2.1. Model specification

For both models—LDA and CTM—the number of topics k has to be fixed a-priori. The LDA model and the CTM assume the following generative process for a document $w = (w_1, \dots, w_N)$ of a corpus D containing N words from a vocabulary consisting of V different words, i.e., $w_i \in \{1, \dots, V\}$ for all $i = 1, \dots, N$.

Step 1: The proportions θ of the topic distribution for the document w are determined.

LDA: Draw $\theta \sim \text{Dirichlet}(\alpha)$.

CTM: Draw $\eta \sim N(\mu, \Sigma)$ with $\eta \in \mathbb{R}^{(k-1)}$ and $\Sigma \in \mathbb{R}^{(k-1)\times(k-1)}$, set $\tilde{\eta}^\top = (\eta^\top, 0)$ and determine θ by

$$\theta_K = \frac{\exp\{\tilde{\eta}_K\}}{\sum_{i=1}^k \exp\{\tilde{\eta}_i\}}$$

for $K = 1, \dots, k$.

Step 2: For each of the N words w_i

(a) Choose a topic $z_i \sim \text{Multinomial}(\theta)$.

(b) Choose a word w_i from a multinomial probability distribution conditioned on the topic z_i: $p(w_i|z_i, \beta)$.

β is the word distribution of topics, i.e., gives the probability of a word occurring in a given topic.

The log likelihood for one document $w \in D$ is therefore given for LDA by

$$\begin{aligned}\ell(\alpha, \beta) &= \log\left(p(w|\alpha, \beta)\right) \\ &= \log \int \sum_z \prod_{i=1}^N p(w_i|z_i, \beta) p(z_i|\theta) p(\theta|\alpha) d\theta\end{aligned}$$

and for CTM by

$$\begin{aligned}\ell(\mu, \Sigma, \beta) &= \log\left(p(w|\mu, \Sigma, \beta)\right) \\ &= \log \int \sum_z \prod_{i=1}^N p(w_i|z_i, \beta) p(z_i|\theta) p(\theta|\mu, \Sigma) d\theta.\end{aligned}$$

The sum over $z = (z_i)_{i=1,\dots,N}$ includes all combinations of assigning the N words in the document to the k topics.

2.2. Estimation

For maximum likelihood (ML) estimation of the LDA model the log likelihood of the data, i.e., the sum over the log likelihoods of all documents, is maximized with respect to the model parameters α and β. For the CTM model the log likelihood of the data is maximized with respect to the model parameters μ, Σ and β. The quantities $p(w|\alpha, \beta)$ for the LDA model and $p(w|\mu, \Sigma, \beta)$ for the CTM cannot be computed tractably. Hence, a VEM procedure is used for estimation. The EM algorithm (Dempster, Laird, and Rubin 1977) is an iterative method for determining a ML estimate in a missing data framework where the complete likelihood of the observed and missing data is easier to maximize than the likelihood of the observed data only. It iterates between an Expectation (E)-step where the expected complete likelihood given the data and current parameter estimates is determined and a Maximization (M)-step where the expected complete likelihood is maximized to find new parameter estimates. For topic models the missing data in the EM algorithm are the latent variables θ and z for LDA and η and z for CTM.

For topic models a VEM algorithm is used instead of an ordinary EM algorithm because the expected complete likelihood in the E-step is also computationally intractable. Instead the posterior distribution $p(\theta, z|\alpha, \beta)$ is replaced by a variational distribution $q(\theta, z|\gamma, \phi)$. This implies that instead of

$$\mathsf{E}_p[\log p(\theta, z|w, \alpha, \beta)]$$

the following is determined

$$\mathsf{E}_q[\log p(\theta, z|w, \alpha, \beta)].$$

The parameters for the variational distributions are document specific and hence are allowed to vary over documents which is not the case for α and β. For the LDA model the variational parameters γ and ϕ for a given document w are determined by

$$(\gamma^*, \phi^*) = \arg \min_{(\gamma, \phi)} \mathrm{D}_{\mathrm{KL}}(q(\theta, z|\gamma, \phi)||p(\theta, z|w, \alpha, \beta)).$$

D_{KL} denotes the Kullback-Leibler (KL) divergence. The variational distribution is set equal to

$$q(\theta, z|\gamma, \phi) = q_1(\theta|\gamma) \prod_{i=1}^{N} q_2(z_i|\phi_i),$$

where $q_1()$ is a Dirichlet distribution with parameters γ and $q_2()$ is a multinomial distribution with parameters ϕ_i.

Analogously for the CTM the variational parameters are determined by

$$(\lambda^*, \nu^*, \phi^*) = \arg \min_{(\lambda, \nu, \phi)} \mathrm{D}_{\mathrm{KL}}(q(\eta, z|\lambda, \nu^2, \phi)||p(\eta, z|w, \mu, \Sigma, \beta)).$$

Since the variational parameters are fitted separately for each document the variational covariance matrix can be assumed to be diagonal, i.e., ν^2 consists only of the diagonal elements. The variational distribution is set to

$$q(\eta, z|\lambda, \nu^2, \phi) = \prod_{K=1}^{k} q_1(\eta_K|\lambda_K, \nu_K^2) \prod_{i=1}^{N} q_2(z_i|\phi_i),$$

where $q_1()$ is a univariate Gaussian distribution with mean λ_K and variance ν_K^2, and $q_2()$ again denotes a multinomial distribution with parameters ϕ_i.

For the LDA model it can be shown with the following equality that the variational parameters result in a lower bound for the log likelihood

$$\log p(w|\alpha,\beta) = L(\gamma,\phi;\alpha,\beta) + \mathrm{D}_{\mathrm{KL}}(q(\theta,z|\gamma,\phi)||p(\theta,z|w,\alpha,\beta))$$

where

$$L(\gamma,\phi;\alpha,\beta) = \mathsf{E}_q[\log p(\theta,z,w|\alpha,\beta)] - \mathsf{E}_q[\log q(\theta,z)]$$

(see Blei *et˜al.* 2003, p.˜1019). Maximizing the lower bound $L(\gamma,\phi;\alpha,\beta)$ with respect to γ and ϕ is equivalent to minimizing the KL divergence between the variational posterior probability and the true posterior probability. This holds analogously for the CTM.

For estimation the following steps are repeated until convergence of the lower bound of the log likelihood.

E-step: For each document find the optimal values of the variational parameters $\{\gamma,\phi\}$ for the LDA model and $\{\lambda,\nu,\phi\}$ for the CTM.

M-step: Maximize the resulting lower bound on the log likelihood with respect to the model parameters α and β for the LDA model and μ, Σ and β for the CTM.

For inference the latent variables θ and z are often of interest to determine which topics a document consists of and which topic a certain word in a document was drawn from. Under the assumption that the variational posterior probability is a good approximation of the true posterior probability it can be used to determine estimates for the latent variables. In the following inference is always based on the variational posterior probabilities if the VEM is used for estimation.

For Gibbs sampling in the LDA model draws from the posterior distribution $p(z|w)$ are obtained by sampling from

$$p(z_i = K|w,z_{-i}) \propto \frac{n^{(j)}_{-i,K} + \delta}{n^{(.)}_{-i,K} + V\delta}\frac{n^{(d_i)}_{-i,K} + \alpha}{n^{(d_i)}_{-i,.} + k\alpha}$$

(see˜ Griffiths and Steyvers 2004; Phan *et˜al.* 2008). z_{-i} is the vector of current topic memberships of all words without the ith word w_i, i.e., it is the vector of topic memberships z where the entry for the ith word is omitted. The index j indicates that w_i is equal to the jth word in the vocabulary. $n^{(j)}_{-i,K}$ gives how often the jth word of the vocabulary is currently assigned to topic K without the ith word. The dot . implies that summation over this index is performed. d_i indicates the document in the corpus to which word w_i belongs. δ denotes the parameter of the prior distribution for the word distribution of the topics, i.e., in the Bayesian model formulation β is drawn from a Dirichlet distribution with parameter δ. Note that in this model formulation α also is a parameter of a prior distribution. The predictive distributions of the parameters θ and β given w and z are given by

$$\hat{\theta}^{(j)}_K = \frac{n^{(j)}_K + \delta}{n^{(.)}_K + V\delta}, \qquad \hat{\beta}^{(d)}_K = \frac{n^{(d)}_K + \alpha}{n^{(.)}_K + k\alpha},$$

for $j = 1,\ldots,V$ and $d = 1,\ldots,D$.

2.3. Pre-processing

The input data for topic models is a document-term matrix. The rows in this matrix correspond to the documents and the columns to the terms. The entry m_{ij} indicates how often the jth word occurred in the ith document. The number of rows is equal to the size of the corpus and the number of columns to the size of the vocabulary. The data preprocessing step involves selecting a suitable vocabulary, i.e., the columns of the document-term matrix. In general the vocabulary will not be given a-priori, but determined using the available data. The mapping from the document to the term frequency vector involves tokenizing the document and then processing the tokens for example by converting them to lower case, removing punctuation characters, removing numbers, stemming, removing stopwords and omitting words with a length below a certain minimum. In addition the final document-term matrix can be reduced by selecting only the terms which occur in a minimum number of documents (see Griffiths and Steyvers 2004, who use a value of 5) or those terms with the highest term-frequency inverse document frequency (tf-idf) scores (Blei and Lafferty 2009).

2.4. Model selection

For fitting the LDA model or CTM to a given document-term matrix the number of topics needs to be fixed a-priori. In addition for estimation using Gibbs sampling values for the parameters of the prior distributions need to be specified. Griffiths and Steyvers (2004) suggest a value of $50/k$ for α and 0.1 for δ. Because the number of topics is in general not known, models with several different numbers of topics are fitted and the optimal number determined in a data-driven way. Model selection with respect to the number of topics is possible by splitting the data into training and test data sets. The likelihood for the test data is then approximated using the lower bound for VEM estimation. For Gibbs sampling the log likelihood is given by

$$\log(p(w|z)) = k \log\left(\frac{\Gamma(V\delta)}{\Gamma(\delta)^V}\right) + \sum_{K=1}^{k} \left\{ \left[\sum_{j=1}^{V} \log(\Gamma(n_K^{(j)} + \delta))\right] - \log(\Gamma(n_K^{(.)} + V\delta)) \right\}.$$

In addition the marginal likelihoods of the models with different numbers of topics can be compared for model selection if Gibbs sampling is used for model estimation. Griffiths and Steyvers (2004) determine the marginal likelihood using the harmonic mean estimator (Newton and Raftery 1994). The harmonic mean estimator is attractive from a computational point of view because it only requires the evaluation of the log likelihood for the different posterior draws of the parameters. The drawback however is that the estimator might have infinite variance.

3. Application: Main functions `LDA()` and `CTM()`

The main functions in package **topicmodels** for fitting the LDA and CTM models are `LDA()` and `CTM()`, respectively.

```
R> LDA(x, k, method = "VEM", control = NULL, model = NULL, ...)
R> CTM(x, k, method = "VEM", control = NULL, model = NULL, ...)
```

These two functions have the same arguments. `x` is a `"DocumentTermMatrix"` as defined in package **tm** (Feinerer, Hornik, and Meyer 2008; Feinerer 2010). A `"DocumentTermMatrix"` is a sparse matrix in a simple triplet matrix representation as provided by package **slam** (Hornik, Meyer, and Buchta 2010) with an additional weighting component. If the weighting is `"term frequency"` each entry indicates how often a term occurs in the document. To use `LDA()` or `CTM()` the entries of the matrix need to be integer numbers. `k` is an integer (larger than 1) specifying the number of topics. `method` determines the estimation method used and currently can be either `"VEM"` or `"Gibbs"` for `LDA()` and only `"VEM"` for `CTM()`. Users can provide their own fit functions to use a different estimation technique or fit a slightly different model variant and specify them to be called within `LDA()` and `CTM()` via the `method` argument.

Argument `control` can be either specified as a named list or as a suitable S4 object where the class depends on the chosen method. In general a user will provide named lists and coercion to an S4 object will internally be performed. The following arguments are possible for the control for fitting the LDA model with the VEM algorithm. They are set to their default values.

```
R> control_LDA_VEM <-
+    list(estimate.alpha = TRUE, alpha = 50/k, estimate.beta = TRUE,
+         verbose = 0, prefix = tempfile(), save = 0,
+         seed = as.integer(Sys.time()),
+         var = list(iter.max = 500, tol = 10^-6),
+         em = list(iter.max = 1000, tol = 10^-4),
+         initialize = "random")
```

The arguments are described in detail below.

`estimate.alpha`, `alpha`, `estimate.beta`: By default α is estimated (`estimate.alpha = TRUE`) and the starting value for α is $50/k$ as suggested by Griffiths and Steyvers (2004). If α is not estimated, it is held fixed at the initial value. If the term distributions for the topics are already given by a previously fitted model, only the topic distributions for documents can be estimated using `estimate.beta = FALSE`. This is useful for example if a fitted model is evaluated on hold-out data or for new data.

`verbose`, `prefix`, `save`: By default no information is printed during the algorithm (`verbose = 0`). If `verbose` is a positive integer every `verbose` iteration information is printed. `save` equal to 0 indicates that no intermediate results are saved in files with prefix `prefix`. If equal to a positive integer, every `save` iterations intermediate results are saved.

`seed`: For reproducibility a random seed can be set which is used in the external code.

`var`, `em`: These arguments control how convergence is assessed for the variational inference step and for the EM algorithm steps by setting a maximum number of iterations (`iter.max`) and a tolerance for the relative change in the likelihood (`tol`). If during the EM algorithm the likelihood is not increased in one step, the maximum number of iterations in the variational inference step is doubled.

If the maximum number of iterations is set to -1 in the variational inference step, there is no bound on the number of iterations and the algorithm continues until the tolerance criterion is met. If the maximum number of iterations is -1 for the EM algorithm, no M-step is made and only the variational inference is optimized. This is useful if the variational parameters should be determined for new documents. The default values for the convergence checks are chosen similar to those suggested in the code by Blei and co-authors.

`initialize`: This parameter determines how the topics are initialized and can be either equal to `"random"`, `"seeded"` or `"model"`. Random initialization means that each topic is initialized randomly, seeded initialization signifies that each topic is initialized to a distribution smoothed from a randomly chosen document. If `initialize = "model"` a fitted model needs to be provided which is used for initialization, otherwise random initialization is used.

The possible arguments controlling how the LDA model is fitted using Gibbs sampling are given below together with their default values.

```
R> control_LDA_Gibbs <-
+    list(alpha = 50/k, estimate.beta = TRUE,
+         verbose = 0, prefix = tempfile(), save = 0,
+         seed = as.integer(Sys.time()),
+         delta = 0.1,
+         iter = 2000, burnin = 0, thin = 2000,
+         best = TRUE)
```

`alpha`, `estimate.beta`, `verbose`, `prefix`, `save` and `seed` are the same as for estimation with the VEM algorithm. The additional parameters are described below in detail.

`delta`: This parameter specifies the parameter of the prior distribution of the term distribution over topics. The default 0.1 is suggested in Griffiths and Steyvers (2004).

`iter`, `burnin`, `thin`: These parameters control how many Gibbs sampling draws are made. The first `burnin` iterations are discarded and then every `thin` iteration is returned for `iter` iterations.

`best`: All draws are returned if `best = FALSE`, otherwise only the draw with the highest posterior likelihood is returned.

For the CTM model using the VEM algorithm the following arguments can be used to control the estimation.

```
R> control_CTM_VEM <-
+    list(estimate.beta = TRUE,
+         verbose = 0, prefix = tempfile(), save = 0,
+         seed = as.integer(Sys.time()),
+         var = list(iter.max = 500, tol = 10^-6),
+         em = list(iter.max = 1000, tol = 10^-4),
+         initialize = "random",
+         cg = list(iter.max = 500,  tol = 10^-5))
```

estimate.beta, verbose, prefix, save, seed, var, em and initialize are the same as for VEM estimation of the LDA model. If the log likelihood is decreased in an E-step, the maximum number of iterations in the variational inference step is increased by 10 or—if no maximum number is set—the tolerance for convergence is divided by 10 and the same E-step is continued. The only additional argument is cg.

cg: This controls how many iterations at most are used (iter.max) and how convergence is assessed (tol) in the conjugate gradient step in fitting the variational mean and variance per document.

LDA() and CTM() return S4 objects of a class which inherits from "TopicModel" (or a list of objects inheriting from class "TopicModel" in the case of Gibbs sampling and best = FALSE). Because of certain differences in the fitted objects there are sub-classes with respect to the model fitted (LDA or CTM) and the estimation method used (VEM or Gibbs sampling). The class "TopicModel" contains the call, the dimension of the document-term matrix, the control object, the number of topics, the terms and document names, the estimates for the term distributions for the topics and the topic distributions for the documents, the assignment of words to the most likely topic and the log likelihood which is $\log p(w|\alpha, \beta)$ for LDA with VEM estimation, $\log p(w|z)$ for LDA using Gibbs sampling and $\log p(w|\mu, \Sigma, \beta)$ for CTM with VEM estimation. For VEM estimation the log likelihood is returned separately for each document. The extending class "LDA" has an additional slot for α, "CTM" additional slots for μ and Σ. "LDA_Gibbs" which extends class "LDA" has a slot for δ and "CTM_VEM" which extends "CTM" has an additional slot for ν^2.

Helper functions to analyse the fitted models are contained. logLik() obtains the log likelihood of the fitted model. posterior() allows to obtain the topic distributions for documents and the term distributions for topics. There is a newdata argument which needs to be given a document-term matrix and where the topic distributions for these new documents are determined without fitting the term distributions of topics. Finally, functions terms() and topics() allow to obtain from a fitted topic model either the k most likely terms for topics or topics for documents respectively, or all terms for topics or topics for documents where the probability is above the specified threshold.

4. Illustrative example: Abstracts of JSS papers

The application of the package **topicmodels** is demonstrated on the collection of abstracts of the *Journal of Statistical Software* (JSS) (up to 2010-08-05). The JSS data is available as a list matrix in the package **corpus.JSS.papers** which can be installed and loaded by

```
R> install.packages("corpus.JSS.papers",
+    repos = "http://datacube.wu.ac.at/", type = "source")
R> data("JSS_papers", package = "corpus.JSS.papers")
```

Alternatively package **OAIHarvester** (Hornik 2010a) can be used to harvest the meta information of the papers published in JSS from its web page.

```
R> library("OAIHarvester")
R> x <- oaih_list_records("http://www.jstatsoft.org/oai")
```

```
R> JSS_papers <- oaih_transform(x[, "metadata"])
R> JSS_papers <- JSS_papers[order(as.Date(unlist(JSS_papers[, "date"]))), ]
R> JSS_papers <- JSS_papers[grep("Abstract:", JSS_papers[, "description"]), ]
R> JSS_papers[, "description"] <- sub(".*\nAbstract:\n", "",
+    unlist(JSS_papers[, "description"]))
```

For reproducibility of results we use only abstracts published up to 2010-08-05 and omit those containing non-ASCII characters in the abstracts.

```
R> JSS_papers <- JSS_papers[JSS_papers[,"date"] < "2010-08-05",]
R> JSS_papers <- JSS_papers[sapply(JSS_papers[, "description"],
+                                  Encoding) == "unknown",]
```

The final data set contains 348 documents. Before analysis we transform it to a `"Corpus"` using package **tm**. HTML markup in the abstracts for greek letters, subscripting, etc., is removed using package **XML** (Temple Lang 2010).

```
R> set.seed(1102)
R> library("topicmodels")
R> library("XML")
R> remove_HTML_markup <-
+ function(s) {
+     doc <- htmlTreeParse(s, asText = TRUE, trim = FALSE)
+     xmlValue(xmlRoot(doc))
+ }
R> corpus <- Corpus(VectorSource(sapply(JSS_papers[, "description"],
+                                       remove_HTML_markup)))
```

The corpus is exported to a document-term matrix using function `DocumentTermMatrix()` from package **tm**. The terms are stemmed and the stopwords, punctuation, numbers and words of length less than 3 are removed using the `control` argument. (We use a C locale for reproducibility.)

```
R> Sys.setlocale("LC_COLLATE", "C")

[1] "C"

R> JSS_dtm <- DocumentTermMatrix(corpus,
+     control = list(stemming = TRUE, stopwords = TRUE, minWordLength = 3,
+       removeNumbers = TRUE, removePunctuation = TRUE))
R> dim(JSS_dtm)

[1]  348 3282
```

The mean term frequency-inverse document frequency (tf-idf) over documents containing this term is used to select the vocabulary. This measure allows to omit terms which have low frequency as well as those occurring in many documents. We only include terms which have a tf-idf value of at least 0.1 which is a bit less than the median and ensures that the very frequent words are omitted.

```
R> summary(col_sums(JSS_dtm))

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  1.000   1.000   2.000   6.948   5.000 550.000 

R> term_tfidf <-
+    tapply(JSS_dtm$v/row_sums(JSS_dtm)[JSS_dtm$i], JSS_dtm$j, mean) *
+      log2(nDocs(JSS_dtm)/col_sums(JSS_dtm > 0))
R> summary(term_tfidf)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
0.02266 0.08615 0.11410 0.14470 0.16230 1.25100 

R> JSS_dtm <- JSS_dtm[,term_tfidf >= 0.1]
R> JSS_dtm <- JSS_dtm[row_sums(JSS_dtm) > 0,]
R> summary(col_sums(JSS_dtm))

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  1.000   1.000   2.000   3.383   3.250  64.000 
```

After this pre-processing we have the following document-term matrix with a reduced vocabulary which we can use to fit topic models.

```
R> dim(JSS_dtm)

[1]  348 2004
```

In the following we fit an LDA model with 30 topics using (1) VEM with α estimated, (2) VEM with α fixed and (3) Gibbs sampling with a burn-in of 1000 iterations and recording every 100th iterations for 1000 iterations. The initial α is set to the default value. By default only the best model with respect to the log likelihood $\log(p(w|z))$ observed during Gibbs sampling is returned. In addition a CTM is fitted using VEM estimation.

```
R> k <- 30
R> SEED <- 2010
R> jss_TM <-
+    list(VEM = LDA(JSS_dtm, k = k, control = list(seed = SEED)),
+         VEM_fixed = LDA(JSS_dtm, k = k,
+           control = list(estimate.alpha = FALSE, seed = SEED)),
+         Gibbs = LDA(JSS_dtm, k = k, method = "Gibbs",
+           control = list(seed = SEED, burnin = 1000,
+             thin = 100, iter = 1000)),
+         CTM = CTM(JSS_dtm, k = k,
+           control = list(seed = SEED,
+             var = list(tol = 10^-4), em = list(tol = 10^-3))))
```

The four fitted models are compared by investigating the similarity between the topics. The distance measure used between the term distributions for each topic is the Hellinger distance, which measures the dissimilarity between two probability distributions and is given by

$$d(x, y) = \sqrt{\frac{1}{2}\sum_{i=1}^{V}(\sqrt{x_i} - \sqrt{y_i})^2}.$$

$x = (x_1, \ldots, x_V)$ and $y = (y_1, \ldots, y_V)$ are vectors of probability distributions which have non-negative entries and sum to one. Package **topicmodels** provides the function `distHellinger()` for computing this distance.

The term distribution for each topic as well as the predictive distribution of topics for a document can be obtained with `posterior()`. A list with components `"terms"` for the term distribution over topics and `"topics"` for the topic distributions over documents is returned. To compare the similarity between the different solutions the topics of the different fitted models are matched using the solver for the linear sum assignment problem provided in package **clue** (Hornik 2005, 2010b). The average distance between the best-matched topics are determined for the different topic model solutions.

```
R> library("clue")
R> methods <- c("VEM", "VEM_fixed", "Gibbs", "CTM")
R> d <- matrix(0, nrow = 4, ncol = 4,
+              dimnames = rep(list(methods), 2))
R> for (i in 1:3) {
+    for (j in (i+1):4) {
+      dist_models <-
+        distHellinger(posterior(jss_TM[[methods[i]]])$terms,
+                      posterior(jss_TM[[methods[j]]])$terms)
+      matching <- solve_LSAP(dist_models)
+      d[i,j] <- d[j,i] <- mean(diag(dist_models[,matching]))
+    }
+  }
R> d
```

```
                VEM VEM_fixed     Gibbs       CTM
VEM       0.0000000 0.3808551 0.8066918 0.7778904
VEM_fixed 0.3808551 0.0000000 0.8114819 0.7989644
Gibbs     0.8066918 0.8114819 0.0000000 0.8123821
CTM       0.7778904 0.7989644 0.8123821 0.0000000
```

The two solutions from the VEM algorithm are closer to each other than to the Gibbs sampling or the CTM solution by only having half the average distance. However the discrepancy between the solutions is still rather large. In general Gibbs sampling and CTM have about the same dissimilarity to any of the other solutions. We compare the α values as a possible reason for the difference between the two solutions from the VEM estimation.

```
R> sapply(jss_TM[1:2], slot, "alpha")
```

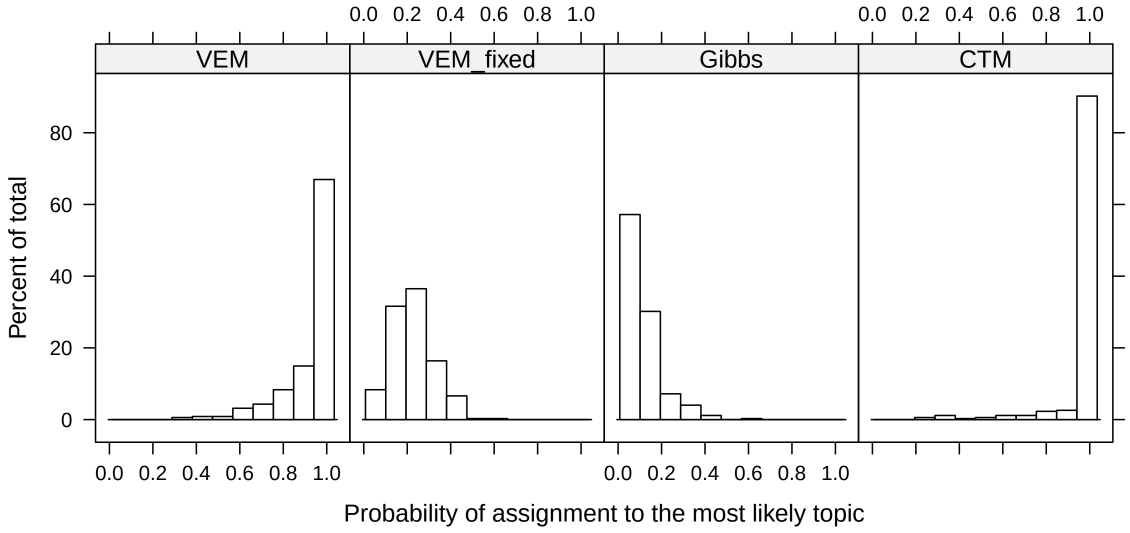

Figure 1: Histogram of the probabilities of assignment to the most likely topic for all documents for the different estimation methods.

```
       VEM  VEM_fixed
0.01064509 1.66666667
```

We see that if α is estimated it is set to a value much smaller than the default. This indicates that in this case the Dirichlet distribution has more mass at the corners and hence, documents consist only of few topics. The influence of α on the estimated topic distribution for documents is illustrated in Figure˜1 where the probabilities of the assignment to the most likely topic for all documents are given. The lower α the higher is the percentage of documents which are assigned to one single topic with a high probability. Furthermore, it indicates that the association of documents with only one topic is strongest for the CTM solution.

The estimated topics for a document and estimated terms for a topic can be obtained using the convenience functions `topics()` and `terms()`. The most likely topic for each document is obtained by

```
R> Topic <- topics(jss_TM[["VEM"]], 1)
```

The five most frequent words for each topic are obtained by

```
R> Terms <- terms(jss_TM[["VEM"]], 5)
R> Terms[,1:5]
```

```
     Topic 1   Topic 2     Topic 3  Topic 4     Topic 5
[1,] "densiti" "interv"    "random" "multivari" "matlab"
[2,] "command" "cell"      "gene"   "mixtur"    "correl"
[3,] "gui"     "intern"    "integ"  "scale"     "gee"
[4,] "fast"    "xlispstat" "recurr" "correl"    "qls"
[5,] "variat"  "pilot"     "gamlss" "nonlinear" "growth"
```

The number of topics was set to 30 rather arbitrarily. We used 10-fold cross-validation and varied the number of topics from 2 to 200 to determine the number of topics in a data-driven way. The results indicated that the number of topics has only a small impact on the model fit on the hold-out data. There is only slight indication that the solution with two topics performs best and that the performance deteriorates again if the number of topics is more than 100. For applications a model with only two topics is of little interest because it enables only to group the documents very coarsely. This lack of preference of a model with a reasonable number of topics might be due to the facts that (1) the corpus is rather small containing less than 500 documents and (2) the corpus consists only of text documents on statistical software.

5. Summary

The package **topicmodels** provides functionality for fitting topic models in R. It builds on and complements functionality for text mining already provided by package **tm**. Functionality for constructing a corpus, transforming a corpus into a document-term matrix and selecting the vocabulary is available in **tm**. The basic text mining infrastructure provided by package **tm** is hence extended to allow also fitting of topic models which are seen nowadays as state-of-the-art techniques for analyzing document-term matrices. The advantages of package **topicmodels** are that (1) it gives access within R to the code written by David M.˜Blei and co-authors, who introduced the LDA model as well as the CTM in their papers, and (2) allows different estimation methods by providing VEM estimation as well Gibbs sampling. Extensibility to other estimation techniques or slightly different model variants is easily possible via the `method` argument.

Packages **Snowball** (Hornik 2009) and **tm** provide stemmers and stopword lists not only for English, but also for other languages including for example German. To the authors' knowledge topic models have so far only been used for corpora in English. The availability of all these tools in R hopefully does not only lead to an increased use of these models, but also facilitates to try them out for corpora in other languages as well as in different settings. In addition different modelling strategies for model selection, such as for example cross-validation, can be easily implemented with a few lines of R code and the results can be analyzed and visualized using already available tools in R.

Package **topicmodels** will only work for reasonable large corpora with numbers of topics in the hundreds. Gibbs sampling needs less memory than using the VEM algorithm and might therefore be able to fit models when the VEM algorithm fails due to high memory demands. In order to be able to fit topic models to very large data sets distributed algorithms to fit the LDA model were proposed for Gibbs sampling in Newman, Asuncion, Smyth, and Welling (2009). The proposed Approximate Distributed LDA (AD-LDA) algorithm requires the Gibbs sampling methods available in **topicmodels** to be performed on each of the processors. In addition functionality is needed to repeatedly distribute the data and parameters to the single processors and synchronize the results from the different processors until a termination criterion is met. Algorithms to parallelize the VEM algorithm for fitting LDA models are outlined in Nallapati, Cohen, and Lafferty (2007). In this case the processors are used in the E-step such that each calculates only the sufficient statistics for a subset of the data. In the future we intend to look into the potential of leveraging the existing infrastructure for large data sets along the lines proposed in Newman *et˜al.* (2009).

Acknowledgments

This research was supported by the the Austrian Science Fund (FWF) under Hertha-Firnberg grant T351-N18.

References

Airoldi EM, Blei DM, Fienberg SE, Xing EP (2008). "Mixed Membership Stochastic Blockmodels." *Journal of Machine Learning Research*, **9**, 1981–2014.

Blei D, McAuliffe J (2008). "Supervised Topic Models." In J~Platt, D~Koller, Y~Singer, S~Roweis (eds.), *Advances in Neural Information Processing Systems 20*, pp. 121–128. MIT Press, Cambridge, MA.

Blei DM, Lafferty JD (2007). "A Correlated Topic Model of Science." *The Annals of Applied Statistics*, **1**(1), 17–35.

Blei DM, Lafferty JD (2009). "Topic Models." In A~Srivastava, M~Sahami (eds.), *Text Mining: Classification, Clustering, and Applications*. Chapman & Hall/CRC Press.

Blei DM, Ng AY, Jordan MI (2003). "Latent Dirichlet Allocation." *Journal of Machine Learning Research*, **3**, 993–1022.

Chang J (2009). ***lda**: Collapsed Gibbs Sampling Methods for Topic Models*. R~package version 1.2.1, URL http://CRAN.R-project.org/package=lda.

Chang J, Blei DM (2009). "Relational Topic Models for Document Networks." In D~van Dyk, M~Welling (eds.), *AISTATS '09: Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, volume~5 of *JMLR: Workshop and Conference Proceedings*, pp. 81–88. Clearwater Beach, Florida.

Chang J, Boyd-Graber JL, Blei DM (2009). "Connections Between the Lines: Augmenting Social Networks with Text." In *KDD '09: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 169–178.

Dempster AP, Laird NM, Rubin DB (1977). "Maximum Likelihood from Incomplete Data Via the EM-Algorithm." *Journal of the Royal Statistical Society B*, **39**, 1–38.

Feinerer I (2010). ***tm**: Text Mining Package*. R~package version 0.5-4., URL http://tm.r-forge.r-project.org/.

Feinerer I, Hornik K, Meyer D (2008). "Text Mining Infrastructure in R." *Journal of Statistical Software*, **25**(5). URL http://www.jstatsoft.org/v25/i05/.

Griffiths TL, Steyvers M (2004). "Finding Scientific Topics." *Proceedings of the National Academy of Sciences of the United States of America*, **101**, 5228–5235. doi:10.1073/pnas.0307752101.

Hall D, Jurafsky D, Manning CD (2008). "Studying the History of Ideas Using Topic Models." In *2008 Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, Proceedings of the Conference, 25-27 October 2008, Honolulu, Hawaii, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 363–371. ACL.

Hornik K (2005). "A CLUE for CLUster Ensembles." *Journal of Statistical Software*, **14**(12). URL `http://www.jstatsoft.org/v14/i12/`.

Hornik K (2009). *Snowball: Snowball Stemmers.* R˜package version 0.0-7, URL `http://CRAN.R-project.org/package=Snowball`.

Hornik K (2010a). *OAIHarvester: Harvest Metadata Using OAI-PMH v2.0.* R˜package version 0.1-2, URL `http://CRAN.R-project.org/package=OAIHarvester`.

Hornik K (2010b). ***clue**: Cluster Ensembles.* R˜package version 0.3-36, URL `http://CRAN.R-project.org/package=clue`.

Hornik K, Meyer D, Buchta C (2010). ***slam**: Sparse Lightweight Arrays and Matrices.* R˜package version 0.1-15, URL `http://CRAN.R-project.org/package=slam`.

Li Z, Wang C, Xie X, Wang X, Ma WY (2008). "Exploring LDA-Based Document Model for Geographic Information Retrieval." In C˜Peters, V˜Jijkoun, T˜Mandl, H˜Müller, D˜Oard, AP˜nas, V˜Petras, D˜Santos (eds.), *Advances in Multilingual and Multimodal Information Retrieval*, volume 5152 of *Lecture Notes in Computer Science*, pp. 842–849. Springer Berlin / Heidelberg. URL `http://dx.doi.org/10.1007/978-3-540-85760-0_108`.

McCallum AK (2002). *MALLET: A Machine Learning for Language Toolkit.* URL `http://mallet.cs.umass.edu`.

Mochihashi D (2004). "A Note on a Variational Bayes Derivation of Full Bayesian Latent Dirichlet Allocation." URL `http://chasen.org/~daiti-m/paper/lda-fullvb.pdf`.

Nallapati R, Cohen W, Lafferty J (2007). "Parallelized Variational EM for Latent Dirichlet Allocation: An Experimental Evaluation of Speed and Scalability." In *ICDMW '07: Proceedings of the Seventh IEEE International Conference on Data Mining Workshops*, pp. 349–354. IEEE Computer Society, Washington, DC, USA. ISBN 0-7695-3033-8. `doi:http://dx.doi.org/10.1109/ICDMW.2007.70`.

Newman D, Asuncion A, Smyth P, Welling M (2009). "Distributed Algorithms for Topic Models." *Journal of Machine Learning Research*, **10**, 1801–1828.

Newton MA, Raftery AE (1994). "Approximate Bayesian Inference with the Weighted Likelhood Bootstrap." *Journal of the Royal Statistical Society B*, **56**(1), 3–48.

Phan XH, Nguyen LM, Horiguchi S (2008). "Learning to Classify Short and Sparse Text & Web with Hidden Topics from Large-scale Data Collections." In *Proceedings of the 17th International World Wide Web Conference (WWW 2008)*, pp. 91–100. Beijing, China.

Steyvers M, Griffiths T (2007). "Probabilistic Topic Models." In TK˜Landauer, DS˜McNamara, S˜Dennis, W˜Kintsch (eds.), *Handbook of Latent Semantic Analysis*. Lawrence Erlbaum Associates.

Temple Lang D (2010). ***XML**: Tools for Parsing and Generating XML Within R and S-PLUS.* R~package version 3.1-1, URL http://CRAN.R-project.org/package=XML.

Wei X, Croft WB (2006). "LDA-Based Document Models for Ad-Hoc Retrieval." In *SIGIR '06: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 178–185. ACM, New York, NY, USA. ISBN 1-59593-369-7. doi:http://doi.acm.org/10.1145/1148170.1148204.

Affiliation:

Bettina Grün, Kurt Hornik
Institute for Statistics and Mathematics
WU Wirtschaftsuniversität Wien
Augasse 2–6
1090 Wien, Austria
E-mail: Bettina.Gruen@wu.ac.at, Kurt.Hornik@R-project.org
URL: http://statmath.wu.ac.at/~gruen/,
http://statmath.wu.ac.at/~hornik/