A Comparative Analysis of Latent Variable Models for Web Page Classification

István Bíró András Benczúr Jácint Szabó
Data Mining and Web Search Research Group
Informatics Laboratory
Computer and Automation Research Institute
Hungarian Academy of Science
Budapest, Hungary
{ibiro,benczur,jacint}@ilab.sztaki.hu

Ana Maguitman
Grupo de Investigación en Recuperación de
Información y Gestión del Conocimiento
Departamento de Cs. e Ing. de la Computación
Universidad Nacional del Sur - CONICET
Bahía Blanca, Argentina
agm@cs.uns.edu.ar

Abstract

A main challenge for Web content classification is how to model the input data. This paper discusses the application of two text modeling approaches, Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA), in the Web page classification task. We report results on a comparison of these two approaches using different vocabularies consisting of links and text. Both models are evaluated using different numbers of latent topics. Finally, we evaluate a hybrid latent variable model that combines the latent topics resulting from both LSA and LDA. This new approach turns out to be superior to the basic LSA and LDA models. In our experiments with categories and pages obtained from the ODP web directory the hybrid model achieves an averaged F-measure value of 0.852 and an averaged ROC value of 0.96.

1. Introduction

Web content classification is expected to help with the automatic generation of topic directories, community detection, advertise blocking and webspam filtering, among other applications. A main problem for classification is how to model the input data. Attempting to model text using a relatively small set of representative topics not only can help alleviate the effect of the curse of dimensionality, but it can also help avoid the false-negative match problem that arises when documents with similar topics but different term vocabulary cannot be associated.

A corpus of web pages can be characterized by the individual words and structure of each particular page (intra-document structure), through labeled hyperlinks or recurrent words relating one page to another (inter-document structure), and by the semantic relations between words, which defines the concept- or topic-space. While traditional feature selection schemes [14] have some appealing characteristics, they are deficient in revealing the inter- or intra-document statistical structure of the corpus. To address these limitations other dimensionality reduction techniques have been proposed including Latent Semantic Analysis (LSA) [6] and Probabilistic Latent Semantic Analysis (PLSA) [13]. These approaches can achieve significant reduction of the feature-space dimensionality and have found successful application not only in tasks where the huge number of features would have make it impossible to process the dataset further but also in applications where documents are to be compared in concept- or topic-space.

A successful text modeling approach is Latent Dirichlet Allocation (LDA) developed by Blei, Ng and Jordan [2]. LDA models every topic as a distribution over the terms of the vocabulary, and every document as a distribution over the topics. These distributions are sampled from Dirichlet distributions.

This paper presents an evaluation of LSA and LDA as text modeling approaches for the task of supervised text categorization. We compare the performance of both methods for Web page classification and discuss the benefits of integrating both approaches to capture various aspects of the modeled material.

2. Background

An important question that arises at the moment of implementing a Web page classifier is how to model each page. The goal is to obtain compact representations that are sufficiently descriptive and discriminative of the topics associated with the pages. Three important questions that need to be answered at the moment of designing a Web page classifier are (1) what features should be extracted from the pages, (2) how to use these features to model the content of each

page, and (3) what algorithm should be used to issue a prediction of a page category.

Web page classification is significantly different from traditional text classification because of the presence of links. While "text-only" approaches often provide a good indication of relatedness, the incorporation of link signals can considerably improve methods for grouping similar resources [24]. However, running a topic prediction algorithm by taking directly the text and links of the pages has some limitations. The main problem is that the underlying topics that can lead to a semantic representation of the pages remain hidden. Therefore, even semantically similar pages can have low similarity if they don't share a significant number of terms and links.

To address these issues, some modeling approaches that attempt to uncover the hidden structure of a document collection have been proposed. Sections 2.1 and 2.2 present an overview of LSA and LDA, two text modeling techniques where documents are associated with latent topics. Documents can then be compared by means of their topics and therefore documents can have high similarity even if they don't share any features—as long as these features are related in a sense to be described in the next two sections.

Contrary to the usual setup, links are not used to propagate topics, but instead we treat them as words and build latent topic models on them. Although links are not words of a natural language, and so one cannot take it for granted that applying latent topic models on them will work, the results of this paper justify the use of such models. LDA with a vocabulary consisting of links was first considered by Zhang, Qiu, Giles, Foley, and Yen [26]. Their model, called SSN-LDA, is exactly our LDA model on links, applied in a social network context. Aside from this paper, we are not aware of any results on latent topic models built on links.

2.1. Latent Semantic Analysis

LSA is a theory and method for extracting and representing the contextual usage meaning of words by statistical computations applied to a large corpus [6]. The method requires a corpus of documents from any domain and the vector space model [22] is used to represent this corpus. In this model a document is represented as a vector where each dimension corresponds to a separate feature from the document. A feature could be a term or any other unit that is a representative attribute of the documents in the given corpus. If a feature occurs in the document, its value in the vector is non-zero. A common way of computing these values is the tf-idf weighting scheme [21].

An important step in LSA is to transform the document-feature vector space into a document-concept and concept-document vector space. By reducing the number of concepts, the documents and their features are projected into a lower-dimension concept space. As a consequence, new and previously latent relations will arise between documents and features. In order to apply LSA we first generate a document-feature matrix **M** from the given corpus. Then, singular-value decomposition (SVD) is applied to **M**, resulting in matrices **U**, **S**, and **V**. The SVD decomposition is such that $\mathbf{M} = \mathbf{USV}^T$, **U** and **V** have orthogonal columns (i.e., $\mathbf{U}^T\mathbf{U}=\mathbf{I}$ and $\mathbf{V}^T\mathbf{V}=\mathbf{I}$) and **S** has entries only along the diagonal. The next step is to reduce the dimensions of **S** to a significant lower value k to obtain a matrix **S'**. The same reduction is performed on the row dimension and column dimension of **U** and $\mathbf{V}^T$ respectively, resulting in the lower-rank matrices **U'** and $\mathbf{V'}^T$. By multiplying matrices **U'**, **S'** and $\mathbf{V'}^T$ we obtain a new matrix **M'** that relates documents and their features through a concept-space.

The matrix **M'** can be thought of as a low-rank approximation to the document-feature matrix **M**. This reduction is sometimes necessary when the original document-feature matrix is presumed too large for the computing resources or when it is considered noisy. Most commonly, this low-rank approximation is performed to represent documents in concept space.

A problem with this approach is that the resulting dimensions might be difficult to interpret. LSA assumes that documents and features form a joint Gaussian model, while a Poisson distribution is typically observed.

To overcome some of these problems, Hofmann [13] introduced Probabilistic LSA (PLSA), which is a generative, graphical model enhancing latent semantic analysis by a sounder probabilistic model. Although PLSA had promising results, it suffers from two limitations: the number of parameters is linear in the number of documents, and it is not possible to make inference for unseen data. In this paper PLSA is not applied, only LSA.

2.2. Latent Dirichlet Allocation

In this section we present a short overview of LDA [2], for a detailed elaboration, we refer the reader to [12]. The LDA method takes a vocabulary V consisting of features, a set T of k topics and n documents of arbitrary length. For every topic z a distribution φ_z on V is sampled from $\mathrm{Dir}(\beta)$, where $\beta \in \mathbb{R}_+^V$ is a smoothing parameter. Similarly, for every document d a distribution ϑ_d on T is sampled from $\mathrm{Dir}(\alpha)$, where $\alpha \in \mathbb{R}_+^T$ is a smoothing parameter.

The words of the documents are drawn as follows: for every word-position of document d a topic z is drawn from ϑ_d, and then a term (or other useful feature) is drawn from φ_z and filled into the position. LDA can be thought of as a Bayesian network, see Figure 1.

One method for finding the LDA model via inference is using Gibbs sampling [9]. (Additional methods are varia-

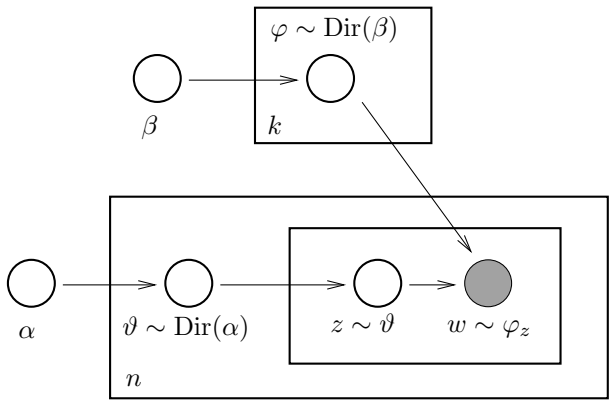

Figure 1. LDA as a Bayesian network

tional expectation maximization [2], and expectation propagation [16]). Gibbs sampling is a Monte Carlo Markov-chain algorithm for sampling from a joint distribution $p(x)$, $x \in \mathbb{R}^n$, if all conditional distributions $p(x_i|x_{-i})$ are known ($x_{-i} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$). In LDA the goal is to estimate the distribution $p(z|w)$ for $z \in T^P$, $w \in V^P$ where P denotes the set of word-positions in the documents. Thus in the Gibbs sampling one has to calculate $p(z_i|z_{-i}, w)$ for $i \in P$. This has an efficiently computable closed form (for a deduction, see [12])

$$p(z_i|z_{-i}, w) = \frac{n_{z_i}^{t_i} - 1 + \beta_{t_i}}{n_{z_i} - 1 + \sum_t \beta_t} \cdot \frac{n_d^{z_i} - 1 + \alpha_{z_i}}{n_d - 1 + \sum_z \alpha_z}. \quad (1)$$

Here d is the document of position i, t_i is the actual word in position i, $n_{z_i}^{t_i}$ is the number of positions with topic z_i and term t_i, n_{z_i} is the number of positions with topic z_i, $n_d^{z_i}$ is the number of topics z_i in document d, and n_d is the length of document d. After a sufficient number of iterations we arrive at a topic assignment sample z. Knowing z, φ and ϑ are estimated as

$$\varphi_{z,t} = \frac{n_z^t + \beta_t}{n_z + \sum_t \beta_t} \quad (2)$$

and

$$\vartheta_{d,z} = \frac{n_d^z + \alpha_z}{n_d + \sum_z \alpha_z}. \quad (3)$$

We call the above method *model inference*. After the model is built, we make *unseen inference* for every new, unseen document d. The ϑ topic-distribution of d can be estimated exactly as in (3) once we have a sample from its word-topic assignment z. Sampling z can be performed with a similar method as before, but now only for the positions i in d:

$$p(z_i|z_{-i}, w) = \frac{\tilde{n}_{z_i}^{t_i} - 1 + \beta_{t_i}}{\tilde{n}_{z_i} - 1 + \sum_t \beta_t} \cdot \frac{n_d^{z_i} - 1 + \alpha_{z_i}}{n_d - 1 + \sum_z \alpha_z}. \quad (4)$$

The notation $\tilde{n}$ refers to the union of the whole corpus and document d.

3. Comparing and Integrating LSA and LDA in Web Page Classification

Our goal is to compare LSA and LDA as text modeling approaches in the Web page classification task. For that purpose, we run a series of tests described next.

3.1 Input Data

In order to run our comparison tests we used a subset of the Open Directory Project (ODP)[1] corpus, from which we chose the 8 top-level categories: Arts, Business, Computers, Health, Science, Shopping, Society, Sports. The resulting corpus contained more than 350K documents distributed among these eight categories. We randomly split this collection of pages into training (80%) and test (20%) collections.

Text and links were extracted from the collected pages and used to generate two vocabulary sets. We refer to the vocabulary set based on text as T and to the vocabulary set based on links as L. In order to generate T we extracted all terms from the selected Web pages and kept only the 30K terms that occurred with the highest frequency. This vocabulary was filtered further by eliminating stop-words[2] and keeping only terms consisting of alphanumeric characters, including those containing the hyphen, and the apostrophe. Documents with length less than 1000 terms were discarded. Finally, the text was stemmed[3]. The final number of words in T was 21308.

The vocabulary set L combines the incoming links and outgoing links associated with our corpus, that is the vocabulary consists of web pages linking to or linked by a page in our corpus. Incoming links were obtained using the Google Web API. To avoid circularity in our classification tests, we took special care to filter out those links coming from topical directories. Finally, we extracted all the outgoing links from the pages and added them to L. Links that occur in less than 10 documents were removed from our vocabulary and we only kept (train and test) documents with at least 3 links. The size of vocabulary L was 44561. It is important to notice that distinct portions of the training and test data are kept for the link and text vocabularies. By intersecting the two test sets (i.e. the text-based and the link-based) we obtain a common test set with 1218 pages.

3.2 Experiments

Using the training collection we created models for LSA and LDA based on different vocabularies and using differ-

[1] http://dmoz.org

[2] We used the stop-word list available at http://www.lextek.com/manuals/onix/stopwords1.html.

[3] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

ent number of latent topics. By taking this approach, we were able to compare the performance of our models across different dimensions. For LSA we used the tf.idf pivoted scheme [23] and applied SVD with dimensions 5, 15 and 30. LDA was run for 1000 iterations to generate 5, 15 and 30 latent topics.

In what follows we will use a simple convention to name the tested models. For example, an LSA model that uses text vocabulary and 15 latent topics is referred to as LSA-T-15, while one that uses link vocabulary for the same number of topics is named LSA-L-15. In addition, one could explore alternative text and link combinations, as well as the integration of latent topics from LSA and LDA. For example, by combining LDA-T-15 with LDA-L-30 we obtained the LDA-T-15-L-30 model, and by combing, LDA-L-15-T-30 with LSA-L-15-T-30 we obtained LDA-L-15-T-30-LSA-L-15-T-30.

In order to generate the LSA models we used the Lanczos code of `svdpack` [1] to run SVD. For LDA we used Phan's GibbsLDA++ C++-code [19]. Once our models were generated we used the Weka machine learning toolkit [25] with 10 fold cross validation to run two different binary classifiers: C4.5 and SVM, separately for every category. The F-measures and ROC values presented in the rest of the paper are average over the 8 categories.

4. Results

In this section we report the averaged F-measures and ROC values for the tested models. Although we generated models with 5, 15 and 30 latent topics, we will omit the results for the models with 5 topics given that they performed poorly. Tables 1 and 2 compare the performance of the LSA and LDA models using different text and link vocabulary combinations (FMR stands for F-measure). We can observe that for both models the text vocabulary is superior to the link vocabulary. This is not surprising considering that the number of links associated with each pages is usually much smaller than the number of terms. However, we observe that the models that combine text and link features are appreciably superior to those based on text only.

The most effective model for LSA is LSA-L-15-T-30, which combines 15 link-based topics with 30 text-based topics. Similarly, the best LDA model is LDA-L-15-T-30. Table 3 summarizes the improvements obtained when link features are included in the vocabulary, the classifier was SVM.

We also observe that LDA was superior to LSA for all the tested combinations. Table 4 shows the averaged FMR & ROC values for the best LSA/LDA configuration, using SVM.

Finally, we looked into the combination of the two latent variable models. Interestingly, by combining the best configurations of LDA and LSA we obtain the best model with an averaged FMR value of 0.852 and an averaged ROC value of 0.96. Table 5 summarizes these results.

The improvement registered by integrating both models points to the fact that the LDA and LSA models capture different aspects of the corpus hidden structure and that the combination of these models can be highly beneficial.

Experiments	SVM	C4.5
L-15	0.105/0.514	0.198/0.575
L-30	0.136/0.532	0.433/0.732
T-15	0.531/0.824	0.487/0.722
T-30	0.562/0.839	0.446/0.687
L-15-T-15	0.666/0.881	0.558/0.753
L-15-T-30	$\mathbf{0.710/0.894}$	0.561/0.755
L-30-T-15	0.671/0.882	0.594/0.783
L-30-T-30	0.708/0.893	0.579/0.768

Table 1. Averaged FMR/ROC values with LSA

Experiments	SVM	C4.5
L-15	0.249/0.724	0.385/0.738
L-30	0.367/0.758	0.458/0.761
T-15	0.464/0.834	0.435/0.686
T-30	0.619/0.876	0.453/0.710
L-15-T-15	0.699/0.900	0.604/0.787
L-15-T-30	$\mathbf{0.765/0.938}$	0.571/0.756
L-30-T-15	0.687/0.896	0.594/0.771
L-30-T-30	0.757/0.921	0.575/0.767

Table 2. Averaged FMR/ROC values with LDA

Method	F-measure	ROC
LSA-T-30	0.562	0.839
LSA-L-15-T-30	0.710	0.894
Improvement	26.3%	6.6%
LDA-T-30	0.619	0.876
LDA-L-15-T-30	0.765	0.938
Improvement	23.6%	7.1%

Table 3. Comparison of text (T) and link (L) based classification results

5. Discussion and Conclusions

It has long been recognized that text and link features extracted from pages can help discover Web communities, which often lead to the extraction of topically coherent subgraphs useful for clustering or classifying Web pages. Many algorithms based solely on link information have been pro-

Method	F-measure	ROC
LSA-L-15-T-30	0.710	0.894
LDA-L-15-T-30	0.765	0.938
Improvement	7.7%	4.9%

Table 4. Averaged FMR & ROC values for the best LSA/LDA configuration

Method	F-measure	ROC
LSA-L-15-T-30	0.710	0.894
LDA-L-15-T-30	0.765	0.938
LSA-L-15-T-30-LDA-L-15-T-30	**0.852**	**0.96**
Improvement LDA-LSA over LSA	20%	7.4%
Improvement LDA-LSA over LDA	11.3%	2.3%

Table 5. Averaged FMR & ROC values for the best LSA, LDA, and LSA-LDA combined configuration

posed to partition hypertext environments [11, 3, 20], to identify and examine the structure of topics on the Web [8, 7, 4] and for Web content categorization [10]. Other algorithms use the hyperlink structure of the Web to find related pages [15, 5]. An approach, totally different from ours, that combines latent variable models to identify topics on the Web is Link-PLSA-LDA [18].

LSA and LDA are based on different principles. On the one hand, LSA assumes that words and documents can be represented as points in Euclidean space. On the other hand, LDA (like other statistical models) assumes that the semantic properties of words and documents are expressed in terms of probabilistic topics. Although some recent theoretical work has been carried out comparing Euclidean and probabilistic latent variable models (e.g, [17]), to the best of our knowledge this is the first attempt to provide a thorough empirical comparison of the two modeling approaches in the Web page classification task.

In our evaluations we observe that although LDA is superior to LSA for all the tested configurations, the improvements achieved by combining latent structures from both approaches are noteworthy. Despite the different underlying assumption of these two approaches and the seeming superiority of LDA, each one appears to have something unique to contribute at the moment of modeling text and links for classification.

6 Acknowledgements

We wish to thank anonymous reviewers for helpful comments and suggestions. This research work is partially supported by an international cooperation project funded by NKTH and MinCyT, by the eScience Regional Knowledge Centre, Hungary and grant OTKA NK 72845.

References

[1] M. Berry. SVDPACK: A Fortran-77 Software Library for the Sparse Singular Value Decomposition. 1992.

[2] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3(5):993–1022, 2003.

[3] R. A. Botafogo and B. Shneiderman. Identifying aggregates in hypertext structures. In *Proceedings of the third annual ACM conference on Hypertext*, pages 63–74. ACM Press, 1991.

[4] S. Chakrabarti, M. M. Joshi, K. Punera, and D. M. Pennock. The structure of broad topics on the Web. In *Proceedings of the eleventh international conference on World Wide Web*, pages 251–262. ACM Press, 2002.

[5] J. Dean and M. R. Henzinger. Finding related pages in the World Wide Web. *Computer Networks (Amsterdam, Netherlands: 1999)*, 31(11–16):1467–1479, 1999.

[6] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[7] S. Dill, S. R. Kumar, K. S. McCurley, S. Rajagopalan, D. Sivakumar, and A. Tomkins. Self-similarity in the Web. In *The VLDB journal*, pages 69–78, 2001.

[8] D. Gibson, J. M. Kleinberg, and P. Raghavan. Inferring Web communities from link topology. In *UK Conference on Hypertext*, pages 225–234, 1998.

[9] T. Griffiths. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl_ 1):5228–5235, 2004.

[10] Z. Gyongyi, H. Garcia-Molina, and J. Pedersen. Web content categorization using link information. Technical report, Stanford University, 2007.

[11] Y. Hara and Y. Kasahara. A set-to-set linking strategy for hypertext systems. In *Proceedings of the conference on Office information systems*, pages 131–135. ACM Press, 1990.

[12] G. Heinrich. Parameter estimation for text analysis. Technical report, Technical Report, 2004.

[13] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57, New York, NY, USA, 1999. ACM.

[14] H. Liu and H. Motoda. *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer Academic Publishers, Norwell, MA, USA, 1998.

[15] M. Marchiori. The quest for correct information on the Web: hyper search engines. In *Selected papers from the sixth international conference on World Wide Web*, pages 1225–1235. Elsevier Science Publishers Ltd., 1997.

[16] T. Minka and J. Lafferty. Expectation-propagation for the generative aspect model. *Uncertainty in Artificial Intelligence (UAI)*, 2002.

[17] M.Steyvers and T. Griffiths. Probabilistic topic models. In S. D. T. Landauer, D.S. McNamara and W. Kintsch, editors, *Handbook of Latent Semantic Analysis*. Mahwah, NJ: Erlbaum, 2007.

[18] R. Nallapati and W. Cohen. Link-plsa-lda: A new unsupervised model for topics and influence in blogs. In *International Conference for Weblogs and Social Media*, 2008.

[19] X.-H. Pahn. http://gibslda.sourceforge.net/.

[20] J. Pitkow and P. Pirolli. Life, death, and lawfulness on the electronic frontier. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 383–390. ACM Press, 1997.

[21] G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[22] G. Salton, C. Yang, and C. Yu. A theory of term importance in automatic text analysis. *Journal of the American Society for Information Science*, 36:33–44, 1975.

[23] A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 21–29, New York, NY, USA, 1996. ACM.

[24] R. Weiss, B. Vélez, and M. A. Sheldon. HyPursuit: a hierarchical network search engine that exploits content-link hypertext clustering. In *Proceedings of the the seventh ACM conference on Hypertext*, pages 180–193. ACM Press, 1996.

[25] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques, 2nd Edition*. Morgan Kaufmann, San Francisco, 2005.

[26] H. Zhang, B. Qiu, C. Giles, H. Foley, and J. Yen. An LDA-based Community Structure Discovery Approach for Large-Scale Social Networks. *Intelligence and Security Informatics, 2007 IEEE*, pages 200–207, 2007.