Latent Dirichlet Allocation in Web Spam Filtering *

István Bíró Jácint Szabó András A. Benczúr
Data Mining and Web search Research Group, Informatics Laboratory
Computer and Automation Research Institute of the Hungarian Academy of Sciences
{ibiro, jacint, benczur}@ilab.sztaki.hu

ABSTRACT

Latent Dirichlet allocation (LDA) (Blei, Ng, Jordan 2003) is a method in information retrieval to model the content and topics of a collection of documents. In this paper we apply a modification of LDA, the novel *multi-corpus LDA* technique, for supervised webspam classification. We treat the web-corpus in site-level, creating a bag-of-words document for every site, and run LDA both on the collection of sites labeled as spam, and as non-spam. In this way spam and non-spam topics are created in the training phase. In the test phase we take the union of these topics, and an unseen site is deemed spam if its total spam topic distribution is above a threshold. As far as we know, this is the first web retrieval application of LDA. Compared to other text classification methods, our LDA-based implementation works surprisingly well. We test it on the UK2007-WEBSPAM corpus and reach an absolute improvement of 55% in F-measure over the publicly available feature set of Web Spam Challenge 2008.

Categories and Subject Descriptors

H.3 [**Information Systems**]: Information Storage and Retrieval; I.2.7 [**Computing Methodologies**]: Artificial Intelligence—*Natural Language Processing*

General Terms

Text Analysis, Feature Selection, Document Classification, Document Clustering, Information Retrieval

Keywords

Web content spam, latent Dirichlet allocation, topic distribution

1. INTRODUCTION

*This work was supported by the EU FP7 project LiWA - Living Web Archives and by grants OTKA NK 72845, *ASTOR* NKFP 2/004/05

Several topic-based text analysis techniques have been developed in the field of information retrieval. Hofmann [14] introduced probabilistic latent semantic indexing (PLSI), which is a generative, graphical model enhancing latent semantic analysis by a sounder probabilistic model. Though PLSI 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.

These issues are addressed by latent Dirichlet allocation developed by Blei, Ng and Jordan [3]. LDA is a fully generative graphical model for analyzing the latent topics of documents. 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. The words of the documents are drawn from the term-distribution of a topic which was just drawn for this word from the topic-distribution of the document. Note that by *term* we mean an element of the vocabulary, and by *word* we mean a position in the document. There are several methods developed for making inference in LDA, like variational expectation maximization [3], expectation propagation [15], and Gibbs sampling [10]. LDA is an intensively studied model, and the experiments are really impressive compared to other known information retrieval techniques.

LDA has several applications, like in entity resolution [2], fraud detection in telecommunication systems [23], image processing [6, 7, 20] and ad-hoc retrieval [21], in addition to the field of text retrieval. To our best knowledge our experiments provide the first application of LDA in webspam filtering, and even in web-retrieval.

In this paper we introduce and apply a slight modification of LDA, called *multi-corpus LDA*, which works as follows. Assume we have a semi-supervised document-classification task with m classes, which are some kind of semantic themes. As usual, part of the documents are labeled with the appropriate themes. Run LDA for all m sub-corpora consisting of the identically labeled documents, then take the union of the resulting m topic-collections, and make inference w.r.t. this aggregated collection of topics for every unseen document d. The fraction of theme-i topics in the topic distribution of d may serve as a measure to what extent d belongs to theme i. For a more detailed description, see Subsection 2.1.

In our experiments we run multi-corpus LDA with 2 classes and thus 2 corpora: spam and non-spam. The inference is performed using Gibbs-sampling. The fraction of spam-topics in the topic-distribution of an unseen document serves as a so-called *spam-measure* in the decision of being spam or not.

Identifying and preventing spam is cited as one of the top challenges in web search engines in [13, 18]. As all major search engines incorporate anchor text and link analysis algorithms into their ranking schemes, web spam appears in sophisticated forms that manipulate content as well as linkage [11]. In addition to link-based, spam hunters use a variety of content based features [8, 16, 9] to detect web spam; a recent measurement of their combination appears in [4]. Content based email spam detection methods worked well for web spam already at the Web Spam Challenge 2007 [5].

1.1 Data set, evaluation and experimental setup

We test the multi-corpus LDA method in combination with the Web Spam Challenge 2008 public features [1], and with the pivoted tf.idf scheme features introduced in [19]. The improvement in F-measure over the public features is 55% with C4.5 and over the pivoted text features is 18% with Bayes-net. The calculations were performed with the machine learning toolkit Weka [22]. For a detailed explanation, see Section 3.

2. METHOD

First we describe latent Dirichlet allocation [3]. For a detailed elaboration, we refer to Heinrich [12]. We have a vocabulary V consisting of terms, 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 is drawn from φ_z and filled into the position.

LDA can be thought of as a Bayesian-network, see Figure 1.

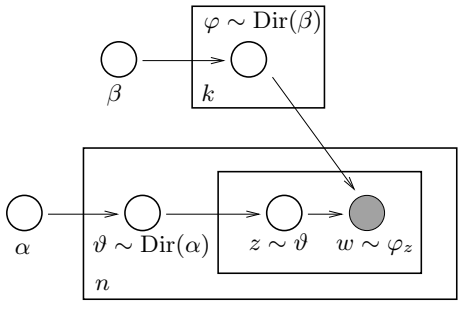

Figure 1: LDA as a Bayesian network

One method for making inference for LDA is Gibbs-sampling [10]. 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 ($x_{-i} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$). The k^{th} transition $x^{(k)} \to x^{(k+1)}$ of the Markov-chain is generated as follows. Choose an index $1 \le i \le n$ (usually $i = k \bmod n$), and let $x^{(k+1)} = x^{(k)}$ everywhere except at index i where $x_i^{(k+1)}$ is sampled from $p(x_i|x_{-i}^{(k)})$.

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 an enough number of iterations we arrive at a topic assignment sample z. Knowing z, we can estimate φ and ϑ as

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

and

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

For an unseen document d the ϑ topic-distribution 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.

We will make use of the observation that the first factor in product (4) is approximately equal to φ_{z_i,t_i} by (2).

2.1 Multi-corpus LDA

As outlined in the introduction, in the multi-corpus setting we run two distinct LDA's: one in the collection of labeled spam sites with $k^{(s)}$ topics, called *spam topics*, and one in the collection of labeled non-spam sites with $k^{(n)}$ topics, called *non-spam topics*. The vocabulary is the same for both LDA's. After both inferences have been done, we have term-distributions for all $k^{(s)} + k^{(n)}$ topics.

From now on we think of the obtained term-distributions of the unified collection of spam and non-spam topics as if they were estimated from only one presumed corpus. To make inference for an unseen document d, we shall perform Gibbs-sampling on this presumed unique distribution using (4). Observe that the $\tilde{n}$ terms in the first factor of the product are not known, as also the topic-assignments of the presumed corpus are not known. Thus we approximate this first factor

by φ_{z_i,t_i}, and $p(z_i|z_{-i},w)$ by

$$p(z_i|z_{-i},w) \approx \varphi_{z_i,t_i} \cdot \frac{n_d^{z_i} - 1 + \alpha_{z_i}}{n_d - 1 + \sum_z \alpha_z}, \qquad (5)$$

which is a computable expression. To distinguish this method from the original Gibbs-sampling inference developed in [10], we call it the *multi-corpus inference*. This is applied only to unseen documents.

After an enough number of iterations we calculate ϑ_d as in (3), and define *spam-measure* to be $\sum\{\vartheta_{d,z} : z \text{ is a spam topic}\}$. A *trivial classification* is to classify d as spam if its spam-measure is above a certain threshold.

An appealing property of multi-corpus LDA is that after the inferences have been made for all m corpora ($m = 2$ in our case), we can calculate the theme-distribution for every unseen document without using any advanced classification methods. This is in contrast to, say, using term frequency features in classification tasks, like the pivoted tf.idf scheme features considered in Subsection 2.2. Another advantage is that the spam-measures are really expressive.

2.2 Text features

We also used the pivoted tf.idf scheme features introduced in [19]. For document d and word w this feature is defined as

$$\frac{1 + \ln(\mathrm{tf}_d(w))}{(1-s) + s \cdot \frac{\mathrm{dl}(d)}{\mathrm{avgdl}}} \cdot \ln\left(\frac{N+1}{\mathrm{df}(w)}\right),$$

where $\mathrm{tf}_d(w)$ is the frequency of w in d, s is the slope, chosen to be 0.2 in our case, avgdl is the average length of a document, $\mathrm{dl}(d)$ is the length of d, N is the number of documents and finally, $\mathrm{df}(w)$ is the frequency of w in the whole corpus.

After dropping those words which appear in less then 10% of the documents, we are left with 3500 text features.

3. EXPERIMENTS

The data-set we used is the UK2007-WEBSPAM corpus. We kept only the labeled sites, which amount to 203 sites labeled as spam, and 3589 labeled as non-spam. We aggregated the words and meta keywords appearing in all pages of the sites to form one document per site in a bag-of-words model, that is only the numbers of words appearing in the document are interesting and not their order. We kept only the alphanumeric characters and the hyphen, and then removed all words containing a hyphen, where this hyphen does not connect two alphabetical words. Next we deleted all stopwords enumerated in the list of `http://www.lextek.com/manuals/onix/stopwords1.html`, and then we used a tree-tagger software from `http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/`. After this procedure a total number of 22.000 terms formed the vocabulary.

We used Phan's GibbsLDA++ C++-code [17] to run LDA and a modified version of it to run the multi-corpus inference for unseen documents in multi-corpus LDA. We applied 5-fold cross validation. Two LDA's were run on the training spam and non-spam corpora, and then multi-corpus inference were made to the test documents by Gibbs-sampling as in (5). Then we calculated the spam-measure for every test site.

The Dirichlet parameter β was chosen to be constant 0.1 throughout, while $\alpha^{(s)} = 50/k^{(s)}$, $\alpha^{(n)} = 50/k^{(n)}$, and during multi-corpus inference α was constant $50/(k^{(s)} + k^{(n)})$ (these are the default values in GibbsLDA++).

We stopped Gibbs-sampling after 2000 steps for inference on the training data, and after 1000 steps for the multi-corpus inference for an unseen document.

The number of topics were chosen to be $k^{(s)} = 2, 5, 10, 20$ and $k^{(n)} = 10, 20, 50$. Consequently, we performed altogether 12 runs, and thus obtained 12 one-dimensional spam-measures as features. Based on the trivial classification using these one-dimensional features, we selected the 5 best choices, whose F-measure curves are shown in Figure 2, and F-measures and ROC values are shown in Table 1. The best result belongs to the choice $k^{(s)} = 10$ and $k^{(n)} = 50$, where the F-measure is 0.46. This is a notable improvement of 97% over the baseline F-measure of the public features with C4.5 classifier (see Table 2). In the rest of the experiments we used only these 5 pairs of topic-numbers, called *chosen spam-measures*.

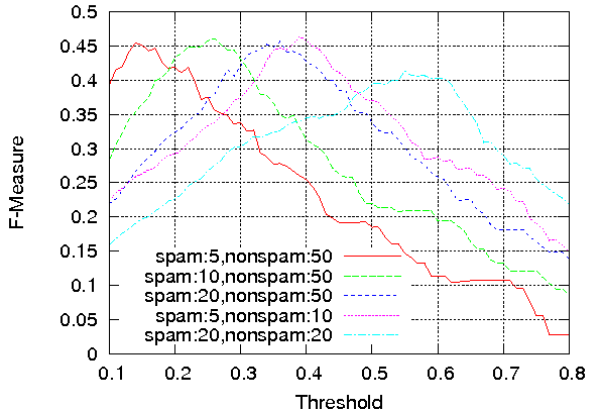

Figure 2: F-measure curves of 5 spam-measures with the trivial classification

Figure 2 indicates that the multi-corpus method is robust to the parameter of topic-numbers, as the performance does not really change by changing the topic-numbers. As one can expect, the maximum of such an F-measure curve is approximately $k^{(s)}/k^{(n)}$.

pair of topic-numbers	FMR	ROC
5-50	0.451	0.855
10-50	0.458	0.861
20-50	0.448	0.873
5-10	0.458	0.868
20-20	0.414	0.868

Table 1: F-measures and ROC values of 5 spam-measures with the trivial classification

We put the 5 chosen spam-measures, the public features and the text features as described in Subsection 2.2 into Weka [22]. The classifiers were C4.5 and Bayes-network. The F-measures and ROC values are shown in Table 2 in the form

FMR/ROC. As the combined text features performed surprisingly well, we tested them with plain SVM and with a cost-sensitive SVM classifier with cost-ratio 7, too. In all cases, the addition of the 5 spam-features resulted in remarkable improvement in F-measure, with 18% over the text features with Bayes-net, and with 55% over the public feature-set with C4.5. Note that though text features performed very well with SVM, multi-corpus LDA raised F-measure from 0.699 to 0.762 with 7-SVM.

feature-set	C4.5	Bayes-net
lda	0.393 / 0.682	0.319 / 0.819
public	0.233 / 0.669	0.233 / 0.721
public & lda	0.362 / 0.738	0.309 / 0.833
text	0.651 / 0.823	0.593 / 0.748
text & lda	**0.691 / 0.885**	**0.699 / 0.937**
text & public & lda	0.542 / 0.872	0.537 / 0.889

feature-set	SVM	7-SVM
text	0.729 / 0.826	0.699 / 0.851
text & lda	**0.792 / 0.841**	**0.762 / 0.848**
text & public & lda	0.667 / 0.772	0.695 / 0.842

Table 2: FMR/ROC values or different classifiers

We also performed the trivial classification for the 5 chosen spam-measure with the UK2006-WEBSPAM corpus. Here we have 2125 sites labeled as spam, and 8082 labeled as non-spam. The parameters and the setup was the same as above. The results can be seen at Table 3. Recall that a spam-measure is easy to calculate with multi-corpus inference, after the term-distributions of the spam and non-spam topics are available, thus the results are quite good for a trivial classification. It is also apparent that multi-corpus LDA is robust to the parameter of topic-numbers.

pair of topic-numbers	FMR	ROC
5-50	0.704	0.881
10-50	0.735	0.902
20-50	0.746	0.919
5-10	0.723	0.906
20-20	0.744	0.925

Table 3: F-measures and ROC values for UK2006-WEBSPAM

4. REFERENCES

[1] Web Spam Challenge 2008, `http://webspam.lip6.fr/`.

[2] I. Bhattacharya and L. Getoor. A latent dirichlet model for unsupervised entity resolution. *SIAM International Conference on Data Mining*, 2006.

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

[4] C. Castillo, D. Donato, A. Gionis, V. Murdock, and F. Silvestri. Know your neighbors: Web spam detection using the web topology. Technical report, DELIS – Dynamically Evolving, Large-Scale Information Systems, 2006.

[5] G. Cormack. Content-based Web Spam Detection. In *Proceedings of the 3rd international workshop on adversarial information retrieval, AIRWeb 2007*, Banff, Alberta, Canada, 2007.

[6] P. Elango and K. Jayaraman. Clustering Images Using the Latent Dirichlet Allocation Model.

[7] L. Fei-Fei and P. Perona. A Bayesian hierarchical model for learning natural scene categories. *Proc. CVPR*, 5, 2005.

[8] D. Fetterly, M. Manasse, and M. Najork. Spam, damn spam, and statistics – Using statistical analysis to locate spam web pages. In *Proceedings of the 7th International Workshop on the Web and Databases (WebDB)*, pages 1–6, Paris, France, 2004.

[9] D. Fetterly, M. Manasse, and M. Najork. Detecting phrase-level duplication on the world wide web. In *Proceedings of the 28th ACM International Conference on Research and Development in Information Retrieval (SIGIR)*, Salvador, Brazil, 2005.

[10] T. Griffiths. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl_1):5228–5235, 2004.

[11] Z. Gyöngyi and H. Garcia-Molina. Web spam taxonomy. In *Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, Chiba, Japan, 2005.

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

[13] M. R. Henzinger, R. Motwani, and C. Silverstein. Challenges in web search engines. *SIGIR Forum*, 36(2):11–22, 2002.

[14] T. Hofmann. Probabilistic latent semantic indexing. *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57, 1999.

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

[16] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly. Detecting spam web pages through content analysis. In *Proceedings of the 15th International World Wide Web Conference (WWW)*, pages 83–92, Edinburgh, Scotland, 2006.

[17] X.-H. Phan. `http://gibbslda.sourceforge.net/`.

[18] A. Singhal. Challenges in running a commercial search engine. In *IBM Search and Collaboration Seminar 2004*. IBM Haifa Labs, 2004.

[19] A. Singhal, G. Salton, M. Mitra, and C. Buckley. Document length normalization. *Information Processing and Management*, 32(5):619–633, 1996.

[20] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman. Discovering Objects and their Localization in Images. *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, 1, 2005.

[21] X. Wei and W. Croft. LDA-based document models for ad-hoc retrieval. *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185, 2006.

[22] I. H. Witten and E. Frank. *Data Mining: Practical*

Machine Learning Tools and Techniques. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, second edition, June 2005.

[23] D. Xing and M. Girolami. Employing Latent Dirichlet Allocation for fraud detection in telecommunications. *Pattern Recognition Letters*, 28(13):1727–1734, 2007.