Document Classification with Latent Dirichlet Allocation

Ph.D. Thesis Summary

István Bíró

Supervisor: András Lukács Ph.D.

Eötvös Loránd University
Faculty of Informatics
Department of Information Sciences

Informatics Ph.D. School
János Demetrovics D.Sc.

Foundations and Methods of Informatics Ph.D. Program
János Demetrovics D.Sc.

Budapest, 2009

1 Introduction

One of the main consequences of the "Age of Internet" is the enormous amount of electronic data, mostly in the form of text. In the last two decades we observe increasing need for very efficient content-based document management tasks including document classification (or alternatively text categorization or classification), the process of labeling documents with categories from a predefined set. Although the history of text categorization dates back to the introduction of computers, it is only from the early 90's that text categorization has become an important part of the mainstream research of text mining, thanks to the increased application-oriented interest and to the rapid development of more powerful hardware.

The efficiency, scalability and quality of document classification algorithms heavily rely on the representation of documents. Among the set-theoretical, algebraic and probabilistic approaches, the vector space model [10] representing documents as vectors in a vector space is used most widely. Dimensionality reduction of the term vector space is an important concept that, in addition to increasing efficiency by a more compact document representation, is also capable of removing noise such as synonymy, polysemy or rare term use. Examples of dimensionality reduction include Latent Semantic Analysis (LSA) [2] and Probabilistic Latent Semantic Analysis (PLSA) [5].

The central concept of this thesis is Latent Dirichlet Allocation (LDA) [1], a generative document model capable of dimensionality reduction as well as topic modeling. LDA models every topic as a distribution over the words of the vocabulary, and every document as a distribution over the topics, thereby one can use the latent topic mixture of a document as a reduced representation.

In our research, we applied various LDA models to Web document classification. First, we compared LDA to LSA based on different kinds of vocabularies and links, and we evaluated a hybrid latent variable model that combined the latent topics from both LSA and LDA. Then, we proposed an explicit topic model, called **multi-corpus LDA (MLDA)**, which, in contrast to the classical LDA, is able to assign category-dependent topic mixtures to Web documents. We showed that MLDA is more accurate by a thorough evaluation based on the ODP Web Directory. Next, we developed **linked LDA**, a fully generative model of web documents taking linkage into account. The experimental results demonstrated the superiority of our method compared to state of the art link based models. One of the main drawbacks of LDA is its poor scalability to large corpora, which is the direct consequence of the Gibbs Sampler within the model. We proposed three different approximate strategies to speed up the sampling process accompanied with only a slight deterioration in performance. Finally, we conducted experiments with our previously developed models — MLDA and linked LDA — in the Web Spam classification subtask.

2 Background

Our results are based on Latent Dirichlet Allocation (LDA), a powerful generative latent topic model. We have a vocabulary $\mathcal{V}$ consisting of V terms, a set $\mathcal{T}$ consisting of K topics and M documents of arbitrary length. For every topic $k \in \mathcal{T}$ a distribution $\vec{\varphi}_k$ on $\mathcal{V}$ is sampled from Dir($\vec{\beta}$), where Dir($\vec{\beta}$) is a V-dimensional Dirichlet distribution, $\vec{\beta} \in \mathbb{R}_+^V$ is a smoothing parameter. Similarly, for every document d a distribution $\vec{\vartheta}_d$ on $\mathcal{T}$ is sampled from Dir($\vec{\alpha}$), where Dir($\vec{\alpha}$) is a K-dimensional Dirichlet distribution, $\vec{\alpha} \in \mathbb{R}_+^K$ is a smoothing parameter.

The words of the documents are drawn as follows: for every word position n of document m a topic $z_{m,n} = k$ is drawn from $\vec{\vartheta}_m$, and then a term $w_{m,n} = t$ is drawn from $\vec{\varphi}_k$ and filled into that position. The notation is summarized in the widely used Bayesian Network representation in Figure 1.

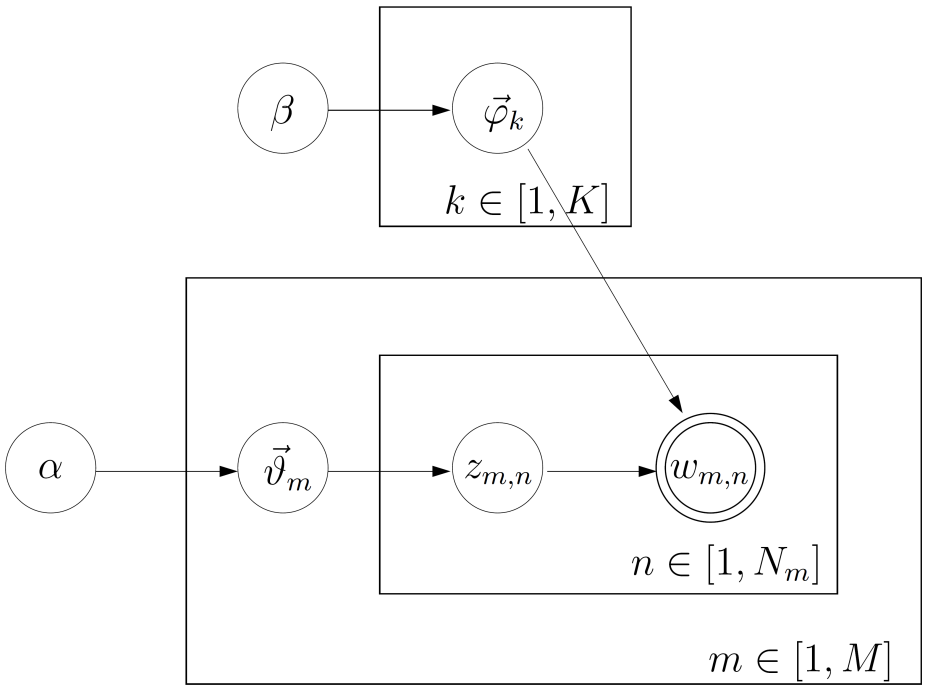

Figure 1: LDA as a Bayesian Network

However, we have to "invert" the generative process and generate the model parameters from given observations (corpus), or in other terms we have to make *model inference*. We use Gibbs sampling [4] for this purpose. Gibbs sampling is a Markov chain Monte Carlo algorithm for sampling from a joint distribution $p(\vec{x})$, $\vec{x} \in \mathbb{R}^N$, if all conditional distributions $p(x_i|\vec{x}_{\neg i})$ are known where $\vec{x}_{\neg i} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_N)$. In LDA the goal is to estimate the distribution $p(\vec{z}|\vec{w})$ for $\vec{z} \in \mathcal{T}^N$, $\vec{w} \in \mathcal{V}^N$ where

$N = \sum_{i=1}^{M} N_i$ denotes the number of word positions in the documents, hence Gibbs sampling makes use of the values $p(z_{m,n} = k|\vec{z}_{\neg(m,n)}, \vec{w})$ for $m \in [1, M]$, $n \in [1, N_m]$:

$$p(z_{m,n} = k|\vec{z}_{\neg(m,n)}, \vec{w}) \propto \frac{N_{kt}^{\neg(m,n)} + \beta_t}{\sum_{v=1}^{V}(N_{kv} + \beta_v) - 1} \cdot \frac{N_{mk}^{\neg(m,n)} + \alpha_k}{\sum_{z=1}^{K}(N_{mz} + \alpha_z) - 1} \tag{2.1}$$

N_{kt} denotes the count of term t in topic k, similarly N_{mk} denotes the count of topic k in document m. The superscript $^{\neg(m,n)}$ denotes that position n in document m is excluded from the corpus when computing the corresponding count. After a sufficient number of iterations we stop with the current topic assignment sample $\vec{z}$. From $\vec{z}$, the variables $\vec{\varphi}_k$ and $\vec{\vartheta}_m$ are estimated as:

$$\varphi_{k,t} = \frac{N_{kt} + \beta_t}{\sum_{v=1}^{V}(N_{kv} + \beta_v)}, \qquad \vartheta_{m,k} = \frac{N_{mk} + \alpha_k}{\sum_{z=1}^{K}(N_{mz} + \alpha_z)}. \tag{2.2}$$

After the model is built, we make *unseen inference* for every new, unseen document d. The topic distribution $\vec{\vartheta}_d$ of d can be estimated exactly as in (2.2) once we have a sample from its word-topic assignment $\vec{z}^{\,d}$. Sampling $\vec{z}^{\,d}$ can be performed with a similar method as above, but now only for the positions i in d:

$$p(z_i^d = k|\vec{z}_{\neg i}^{\,d}, \vec{w}) \propto \frac{\tilde{N}_{kt}^{\neg i} + \beta_t}{\sum_{v=1}^{V}(\tilde{N}_{kv} + \beta_v) - 1} \cdot \frac{N_{dk}^{\neg i} + \alpha_k}{\sum_{z=1}^{K}(N_{dz} + \alpha_z) - 1} \tag{2.3}$$

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

3 New results

Claim 1 Comparative Analysis of LSA and LDA [C4]

We discuss the application of two text modeling approaches, LSA and LDA for Web page classification. We report results on a comparison of these two approaches using different vocabularies consisting of links and text. Both models are evaluated using varying 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.

We study the classification efficiency of the LSA and LDA based latent topic models built on different vector space representations of Web documents. Generally, the vector space is spanned over documents words. We refer to this representation as *text based*. However, in a hyperlinked environment, one can represent documents with their in- and outneighbors as well, naturally defining a *link based* representation.

Claim 1.1 Experimental results

We use a thoroughly prepared subset of the Open Directory Project (ODP) [] Web directory. To a given page, we assign its main category label from the ODP category hierarchy. After generating our models, we use the Weka machine learning toolkit with 10-fold cross validation to run two different binary classifiers: C4.5 and SVM, separately for each category. The key results are as follows:

- LDA outperforms LSA by 8% in *F*-measure and 5% in *AUC*.
- Link based representation is weaker than text based, however the combination of the two performs the best. This reveals the possibility that different representations capture different pieces of information hidden in the documents. The improvement of a text and link based representation over a simple text based one is 27% in *F*-measure and 7% in *AUC*, respectively.
- LDA and LSA models capture different aspects of the hidden structure of the corpus and the combination of these models can be highly beneficial. The improvement of an LDA-LSA hybrid model over LSA is 20% in *F*-measure and 7% in *AUC*, while over LDA is 11% in *F*-measure and a 2% in *AUC*.

Claim 2 Multi-corpus LDA [C2]

We introduce and evaluate a novel probabilistic topic exploration technique, called **Multi-corpus Latent Dirichlet Allocation (MLDA)**, for supervised semantic categorization of Web pages. The appealing property of MLDA is that right after unseen inference the resulting topic distribution directly classifies the documents. This is in contrast to, say, plain LDA, where the topic distribution of the documents serves as features for a classifier.

Claim 2.1 Development of our MLDA model

MLDA is essentially a hierarchical method with two levels: categories and topics. Assume we have a supervised document categorization task with m categories. Every document is assigned to exactly one category, and this assignment is known only for the training corpus. For every category we assign separate topic collection, and the union of these collections forms the topic collection of LDA. In LDA a Dirichlet parameter vector $\vec{\alpha}$ is chosen for every document so that topics assigned to the document words are drawn from a fixed multinomial distribution drawn from $\mathrm{Dir}(\vec{\alpha})$. In MLDA, we require for every training document that this Dirichlet parameter $\vec{\alpha}$ has component zero for all topics outside the document's category. In

this way, only topics from the document's category are sampled to generate the document's words. This is equivalent to building separate LDA models for every category with category-specific topics. For an unseen document d the fraction of topics in the topic distribution of d that belong to a given category measures then how well d fits into that category. As a Dirichlet distribution allows only positive parameters, we will extend the notion of Dirichlet distribution in a natural way by allowing zeros.

Claim 2.3 Vocabulary term selection and ϑ-smoothing with Personalized PageRank

We perform a careful term selection based on the term frequency calculated for each different category and on the entropy of the normalized term frequency vector over the categories. The goal is to find terms with high coverage and discriminability. There are several results published on term selection methods for text categorization tasks [3, 7]. However, we do not directly apply these here, as our setup is different in that the features put into the classifier come from discovered latent topics, and are not derived directly from terms.

After the model inference is over, every document has an estimated topic distribution $\vec{\vartheta}$. Averaging these $\vec{\vartheta}$'s over the documents gives a distribution vector $\widehat{\vec{\vartheta}}$, estimating the overall presence of the topics in the corpus. We call $\widehat{\vec{\vartheta}}$ the **observed importance distribution** of the topics in the corpus.

There is a possibility that MLDA infers two very similar topics in two distinct categories. In such a case it can happen that multi-corpus unseen inference for an unseen document puts very skewed weights on these two topics, endangering classification performance. To avoid such a situation, we apply a modified 1-step Personalized PageRank on the topic space, taking topical similarity and the observed importance into account. The proposed method is as follows.

Fix a document m. After unseen inference, its topic distribution is $\vec{\vartheta}_m$. We smooth $\vec{\vartheta}_m$ by recomputing the components of $\vec{\vartheta}_m$ as follows. For all topics $h \in \mathcal{K}$, replace $\vartheta_{m,h}$ by

$$\vartheta_{m,h}^{new} = c \cdot \vartheta_{m,h} + \sum_{j \in \mathcal{K} \setminus h} S_j \cdot \frac{\widehat{\vartheta}_j}{JSD(\varphi_j, \varphi_h) + \epsilon} \cdot \vartheta_{m,j} \tag{3.1}$$

where c is a smoothing constant, $\widehat{\vec{\vartheta}}$ is the observed importance distribution, ε is a small value to avoid dividing by zero, φ_j, φ_h are the corresponding word distributions representing topics j and h, respectively and JSD is the Jensen-Shannon divergence function. Note that $c = 1$ corresponds to no ϑ-smoothing. S_j is chosen in such a way that the contribution of $\vartheta_{m,j}$ to

the rest of the topics satisfies

$$\sum_{h \in \mathcal{K} \setminus j} S_j \cdot \frac{\widehat{\vartheta}_j}{JSD(\varphi_j, \varphi_h) + \epsilon} \cdot \vartheta_{m,j} = (1 - c)\vartheta_{m,j}, \tag{3.2}$$

which makes sure that $\vartheta_{m,j}^{new}$ is a distribution. The consequence of ϑ-smoothing is if two topic have very similar distributions (that is, $JSD \approx 0$), then the corresponding weights will be balanced, therefore decreasing the classification error.

Claim 2.4 Experimental results

We use the same Web document collection as in Claim 1.1. We compare the classification performance of MLDA with that of several baseline methods. We also test the effectiveness of our proposed term selection and ϑ-smoothing. In the MLDA model, one can easily exploit the statistics of the hierarchical category structure of the Web collection, therefore experiments on this feature are done as well. MLDA can be used as a direct classifier, since the inferred topic distribution gives an estimation of how well the document fits into the category. The key results are as follows:

- Improvement in running time over LDA. In addition MLDA can be run in parallel.
- 15% increase in *AUC* with MLDA based direct classification over $tf \times idf$ based linear SVM.
- A slight 2% increase in *AUC* by choosing the number of subcategories of the main category as the number of latent topics.
- 10% increase in *AUC* by combining a careful vocabulary selection and the ϑ-smooting heuristic over a "heuristic-free" MLDA baseline.

Claim 3 Linked LDA [C1]

We propose **linked LDA**, a novel influence model that gives a fully generative model for hyperlinked documents, such as a collection of Web pages. We demonstrate the applicability of our model for classifying large Web document collections. In our setup, topics are propagated along links in such a way that linked documents directly influence the words in the linking document. The inferred LDA model can be applied to dimensionality reduction in classification similarly to Claims 1 and 2. In addition, the model yields link weights that can be applied in algorithms to process the graph defined by Web hyperlinks; as an example, we deploy link weights extracted by linked LDA in stacked graphical learning [6].

Claim 3.1 Development of our linked LDA model

The key idea is to sample the topic of a position in a document in two steps: first we sample either an outlink or the document itself, then sample a topic from the document topic distribution corresponding to the outlink. Finally we sample a word from the word distribution belonging to the chosen topic. The generative process is summarized in Figure 2.

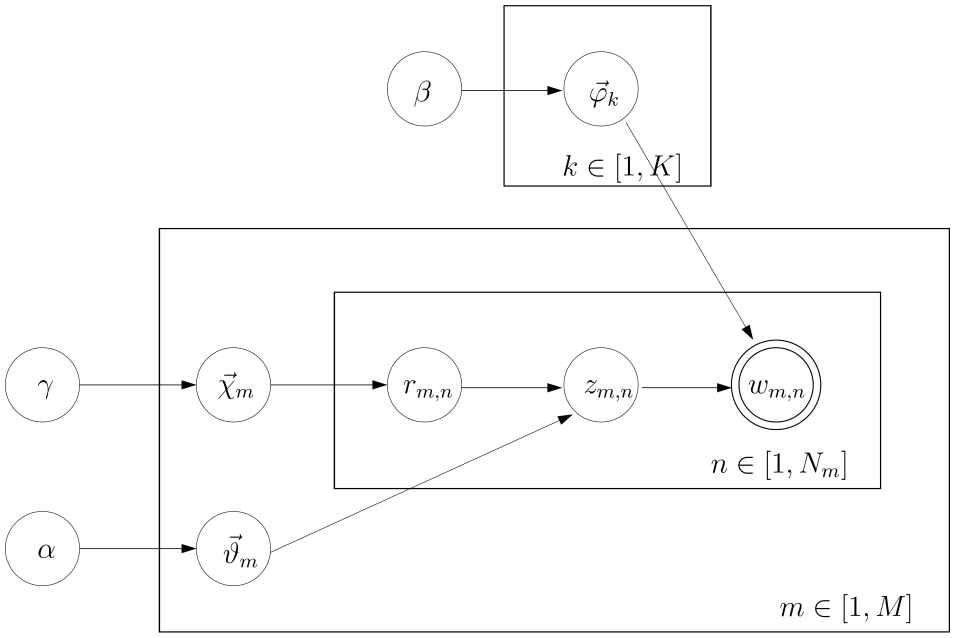

Figure 2: Linked LDA as a Bayesian Network

Our model also relies on the LDA distributions $\vec{\varphi}_k$ and $\vec{\vartheta}_m$. We introduce an additional distribution $\vec{\chi}_m$ on the set $\mathcal{S}_m$, which contains the document m and its outneighbors. $\vec{\chi}_m$ is sampled from Dir$(\vec{\gamma}_m)$, where $\vec{\gamma}_m$ is a positive smoothing vector of size $|\mathcal{S}_m|$.

We develop a Gibbs sampling inference procedure for linked LDA. The goal is to estimate the distribution $p(\vec{r}, \vec{z}|\vec{w})$ for $\vec{r} \in \mathcal{D}^N$, $\vec{z} \in \mathcal{T}^N$, $\vec{w} \in \mathcal{V}^N$ where $N = \sum_{i=1}^{M} N_i$ denotes the set of word positions in the documents. In Gibbs sampling one has to calculate $p(z_{m,n} = k, r_{m,n} = l|\vec{z}_{\neg(m,n)}, \vec{r}_{\neg(m,n)}, \vec{w})$ for $m \in [1, M]$, $n \in [1, N_m]$. We derive the following update

equation:

$$p(z_{m,n} = k, r_{m,n} = l | \vec{z}_{\neg(m,n)}, \vec{r}_{\neg(m,n)}, \vec{w}) \propto$$

$$\frac{N_{lk}^{\neg(m,n)} + \alpha_k}{\sum_{z=1}^{K}(N_{lz} + \alpha_z) - 1} \cdot \frac{N_{ml}^{\neg(m,n)} + \gamma_l}{\sum_{r=1}^{S_m}(N_{mr} + \gamma_r) - 1} \cdot \frac{N_{kt}^{\neg(m,n)} + \beta_t}{\sum_{v=1}^{V}(N_{kv} + \beta_v) - 1}, \tag{3.3}$$

where N_{lk} is the number of words in document l with topic k, similarly N_{ml} denotes the number of words in document m influenced by outneighbor $l \in S_m$, and N_{kt} is the count of word t in topic k in the corpus.

Similarly to LDA, we stop after a sufficient number of iterations with the current topic assignment sample $\vec{z}$, and estimate $\vec{\varphi}_k$ and $\vec{\vartheta}_m$ as in (2.2), and $\vec{\chi}_m$ by

$$\chi_{ml} = \frac{N_{ml} + \gamma_{ml}}{\sum_{r=1}^{|S_m|}(N_{mr} + \gamma_{mr})}. \tag{3.4}$$

Claim 3.2 Experimental results

In our experiments, we use the host-level aggregation of the WEBSPAM-UK2007 crawl of the `.uk` domain. We perform topical classification into one of the 14 top-level English language categories of the ODP. We make the following experiments in order to study the behavior of the linked LDA model:

- We experimentally show the convergence of the log-likelihood of the model.
- We experimentally show a strong correlation between the model log-likelihood and the *AUC*.
- We compare linked LDA with a standard LDA model: linked LDA outperforms LDA by about 4% in *AUC* of classification.
- We compare linked LDA with link-PLSA-LDA [8], the state of the art link based topic model. In this experiment, we use different link weight functions derived from the χ distribution of linked LDA, from link-PLSA-LDA and the cocitation graph as well. We perform the experiments with a stacked graphical learning classification scheme. The best model was linked LDA; it improves the classification over link-PLSA-LDA by 3% in *AUC*.

Claim 4 Fast Gibbs samplers

We develop LDA specific boosting of Gibbs samplers resulting in a significant speedup in our experiments. The crux in the scalability of LDA for large corpora lies in the understanding of the Gibbs sampler for inference. Originally, the unit of sampling or, in other terms, a transition step of the underlying Markov chain, is the redrawing of one sample for a single term occurrence in the Gibbs sampler to LDA [4]. The storage space and update time of all these counters prohibit sampling for very large corpora. Since however the order of sampling is neutral, we may group occurrences of the same term in one document together. Our main idea is then to re-sample each of these term positions in one step, and to assign a joint storage for them. We introduce three strategies:

Claim 4.1 Aggregated Gibbs Sampler

We calculate the conditional topic distribution F as in Equation (2.1) for the first occurrence i of a word in a given document. Next we draw a topic from F for every position with the same word without recalculating F, and update all counts corresponding to the same word. In this way, the number of calculations of the conditional topic distributions is the number of different terms in the document instead of the length of the document, moreover, the space requirement remains unchanged. This can be further improved by maintaining the aggregated topic count vector for terms with large frequency in the document, instead of storing the topic at each word.

Claim 4.2 Limit Gibbs Sampler

This sampling is based on the "bag of words" model assumption suggesting that the topic of a document remains unchanged by multiplying all term frequencies by a constant. In the limit hence we may maintain the calculated conditional topic distribution F for the set of all occurrences of a word, without drawing a topic for every single occurrence. Equation (2.1) can be adapted to this setting by a straightforward redefinition of counts N. Similarly to aggregated Gibbs sampling, limit sampling may result in compressed space usage depending on the size and term frequency distribution of the documents.

Claim 4.3 Sparse Gibbs Sampler

Sparse sampling with sparsity parameter ℓ is a lazy version of limit Gibbs sampling where we ignore some of the less frequent terms to achieve faster convergence on the more important ones. On every document m with d_m words, we sample d_m/ℓ times from a multinomial distribution on the distinct terms with replacement by selecting a term by a probability proportional to its term frequency tf_w in the document. Hence with $\ell = 1$ we expect a performance

similar to limit Gibbs sampling, while large ℓ results in a speedup of about a factor of ℓ with a trade-off of lower accuracy.

Claim 4.4 Results

- We prove that the heuristics have rationale, because they all have the required distribution $p(\vec{z}|\vec{w})$ as their unique stationary distribution.
- We compare the running time of the proposed heuristics with the standard Gibbs Sampling and with a recently developed fast Gibbs Sampler [9]. We measure an astonishing $10 - 11$ times speedup over the fast sampler with only 1% decrease in terms of *AUC*.
- We experimentally show the convergence of the model log-likelihood of the various samplers and the strong correlation between the model log-likelihood and the *AUC*.

Claim 5 Web Spam Filtering [C3, C5]

We participated in the Web Spam Challenge contest in 2008. The goal of the Web Spam Challenge series is to identify and compare machine learning methods for automatically labeling Websites as spam or normal. We made experiments with both MLDA and linked LDA and we ran our tests on the WEBSPAM-UK2007 dataset. The organizers provided baseline content and link based features, thus we were given a controlled environment to measure the performance of our models. In both cases, we applied feature combination schemes in order to improve classification performance. In the case of MLDA, we used a logistic regression based combination, whilst in case of linked LDA, we combined the results of the different classifiers by the random forest classifier.

Claim 5.1 Experimental results with MLDA [C3]

In MLDA, we build two separate LDA models, one for the spam and one for the normal Web pages, then we take the union of the resulting topic collections and make inference with respect to this aggregated collection of topics for every unseen document d. Finally, the total probability of spam topics in the topic distribution of an unseen document gives a prediction of being spam or honest. The key results are as follows:

- We reached a relative improvement of 11% in *AUC* over the baseline classifiers and we ranked fourth in the competition with an $AUC = 0.80$.

Claim 5.2 Experimental results with linked LDA [C5]

In case of linked LDA, we simply apply our model to get the topic distribution of the Web pages, and then we perform spam classification. The key results are as follows:

- We tested linked LDA on the same corpus and we compared our results to the winner of 2008. Our random forest based combinations achieve an $AUC = 0.854$, while the winner's result was $AUC = 0.848$.

References

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

[2] 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.

[3] G. Forman, I. Guyon, and A. Elisseeff. An Extensive Empirical Study of Feature Selection Metrics for Text Classification. *Journal of Machine Learning Research*, 3(7-8):1289–1305, 2003.

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

[5] T. Hofmann. Unsupervised Learning by Probabilistic Latent Semantic Analysis. *Machine Learning*, 42(1):177–196, 2001.

[6] Z. Kou. *Stacked graphical learning*. PhD thesis, School of Computer Science, Carnegie Mellon University, December 2007.

[7] J. Li and M. Sun. Scalable Term Selection for Text Categorization. *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 774–782, 2007.

[8] 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.

[9] I. Porteous, D. Newman, A. Ihler, A. Asuncion, P. Smyth, and M. Welling. Fast Collapsed Gibbs Sampling for Latent Dirichlet Allocation. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM Press New York, NY, USA, 2008.

[10] G. Salton, A. Wong, and A. C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18:229–237, 1975.

Publications

Journal papers

[J1] A. A. Benczúr, I. Bíró, M. Brendel, K. Csalogány, B. Daróczy, and D. Siklósi. Multimodal Retrieval by Text-segment Biclustering. *ADVANCES IN MULTILINGUAL AND MULTIMODAL INFORMATION RETRIEVAL. 8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007, Budapest, Hungary, September 19-21, 2007, Revised Selected Papers. Series: Lecture Notes in Computer Science , Vol. 5152*.

[J2] P. Schönhofen, I. Bíró, A. A. Benczúr, and K. Csalogány. Cross-language Retrieval with Wikipedia. *ADVANCES IN MULTILINGUAL AND MULTIMODAL INFORMATION RETRIEVAL. 8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007, Budapest, Hungary, September 19-21, 2007, Revised Selected Papers. Series: Lecture Notes in Computer Science , Vol. 5152*.

[J3] Z. Szamonek and I. Bíró. Similarity Based Smoothing in Language Modeling. *Acta Cybernetica*, 18(2):303–314, 2007.

Conference papers

[C1] I. Bíró, J. Szabó, and A. A. Benczúr. Large Scale Link Based Latent Dirichlet Allocation for Web Document Classification. Submitted.

[C2] I. Bíró and J. Szabó. Latent Dirichlet Allocation for Automatic Document Categorization. In *Proceedings of the 19th European Conference on Machine Learning and 12th Principles of Knowledge Discovery in Databases*, 2009.

[C3] I. Bíró, A. A. Benczúr, J. Szabó, and D. Siklósi. Linked Latent Dirichlet Allocation in Web Spam Filtering. In *Proceedings of the 5th international workshop on Adversarial Information Retrieval on the Web*, 2009.

[C4] I. Bíró, J. Szabó, A. A. Benczúr, and A. G. Maguitman. A Comparative Analysis of Latent Variable Models for Web Page Classification. *LA-WEB*, pages 23–28. IEEE Computer Society, 2008.

[C5] I. Bíró, J. Szabó, and A. A. Benczúr. Latent Dirichlet Allocation in Web Spam Filtering. In *Proceedings of the 4rd international workshop on Adversarial Information Retrieval on the Web*, 2008.

[C6] D. Siklósi, A. A. Benczúr, Z. Fekete, M. Kurucz, I. Bíró, A. Pereszlényi, S. Rácz, A. Szabó, and J. Szabó. Web Spam Hunting @ Budapest. In *Proceedings of the 4rd international workshop on Adversarial Information Retrieval on the Web*, 2008.

[C7] A. Benczúr, D. Siklósi, J. Szabó, I. Bíró, Z. Fekete, M. Kurucz, A. Pereszlényi, S. Rácz, and A. Szabó. Web Spam: A Survey with Vision for The Archivist. In *8th International Web archiving workshop.*, 2008.

[C8] I. Bíró, C. Szepesvári, and Z. Szamonek. Sequence Prediction Exploiting Similarity Information. *IJCAI 2007: Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 1576–1581, 2007.

[C9] P. Schönhofen, I. Bíró, A. A. Benczúr, and K. Csalogány. Performing Cross Language Retrieval with Wikipedia. In *Working Notes for the CLEF 2007 Workshop*, 2007.

[C10] A. A. Benczúr, I. Bíró, M. Brendel, K. Csalogány, B. Daróczy, and D. Siklósi. Cross-modal Retrieval by Text and Image Feature Biclustering. In *Working Notes for the CLEF 2007 Workshop*, 2007.

[C11] A. A. Benczúr, I. Bíró, K. Csalogány, and T. Sarlós. Web Spam Detection via Commercial Intent Analysis. In *AIRWeb '07: Proceedings of the 3rd international workshop on Adversarial Information Retrieval on the Web*, pages 89–92, 2007.

[C12] E. P. Windhager, L. Tansini, I. Bíró, and D. Dubhashi. Iterative Algorithms for Collaborative Filtering with Mixture Models. In *proceedings of International Workshop on Intelligent Information Access (IIIA)*, 2006.

[C13] A. A. Benczúr, I. Bíró, K. Csalogány, B. Rácz, T. Sarlós, and M. Uher. Pagerank és azon túl: Hiperhivatkozások szerepe a keresésben. *Magyar Tudomány*, (11):1325–1331, November 2006.

Conference posters

[P1] A. A. Benczúr, I. Bíró, K. Csalogány, and M. Uher. Detecting Nepotistic links by Language Model Disagreement. *WWW '06: Proceedings of the 15th International Conference on World Wide Web*, pages 939–940, 2006.

[P2] I. Bíró, Z. Szamonek, and C. Szepesvári. Simítás hasonlósági információ felhasználásával. In *Proceedings of Association for Hungarian Computational Linguistics*, 2006.