

# Topic-Specific Scoring of Documents with Discrete PCA<sup>\*</sup>

Wray Buntine and Kimmo Valtonen

Complex Systems Computation Group,  
Helsinki Institute for Information Technology  
P.O. Box 9800, FIN-02015 HUT, Finland.  
First.Last@HIIT.FI

**Abstract.** The random surfer model for scoring of documents, for instance using PageRank, works when good link structure exists for a collection. Here, we develop a topic-specific version using a topic structure developed automatically via discrete PCA methods. To evaluate the resultant method, scores are developed on the Wikipedia, the public domain encyclopedia on the web, because it has a good internal link structure, and results can be readily interpreted from the page titles.

More sophisticated language models are starting to be used in information retrieval [8] and real successes are being achieved in their use [4]. A document modelling approach based on discrete versions of PCA [7, 1, 2] has been applied to the language modelling task in information retrieval [2, 3]. Here we apply the same discrete PCA method to topic specific versions of page rank [6, 10]. Our intent is that this can be used as a secondary score to add topical scoring to retrieval in conjunction with a separate key-word based score such as TFIDF.

## 1 BACKGROUND

### 1.1 Topic Specific Ranking

Here, we use the term “random surfer model” in a broad sense, to encompass general Monte Carlo Markov chain methods, modelling eye-balls on pages, used to determine scores for documents. Examples are [6, 10]. A general method for topic-specific ranking goes as follows ([10], with a modified presentation here): Our surfer restarts with probability  $\alpha$  at a page  $i$  with probability  $r_i$ . From that page, they uniformly select a link to document  $i'$ , and jump to this next page. They then consider the topic of the new page, whose strength of relevance is determined by another probability  $t_{i'}$ . With probability  $t_{i'}$  they accept the new page, and with probability  $1 - t_{i'}$  they go back to the page  $i$  to try a new link. The stationary distribution of the Markov Chain for the probability of being on page  $p_i$  is then given by the update equations:

$$p_i \leftarrow \alpha r_i + (1 - \alpha) \sum_{i': i' \rightarrow i} p_{i'} \frac{t_i}{\sum_{j: i' \rightarrow j} t_j}$$

---

<sup>\*</sup> Poster submission for ECIR 2005

where we perform the calculation only for those pages  $i$  with  $r_i > 0$ , and  $i' \rightarrow i$  denotes page  $i'$  links to page  $i$ . The vectors  $\mathbf{r}$  and  $\mathbf{t}$  can be specialized to a specific topic, and so a set of such rankings  $\mathbf{p}$  developed:  $\mathbf{r}$  represents the starting documents for a topic and  $\mathbf{t}$  represents the probability that someone interested in the topic will stay at the page.

## 1.2 Discrete PCA

Principal component analysis (PCA) latent semantic indexing, and independent component analysis (ICA) are key methods in the statistical engineering toolbox. They have a long history, are used in many different ways: genotype inference using admixtures [9], probabilistic latent semantic indexing [7] latent Dirichlet allocation [1], discrete PCA [2] and GaP models [3] are just a few of the known versions. These methods are variations of one another, ignoring statistical methodology and notation, and form a discrete version of ICA [2, 3].

Each document is represented as an integer vector,  $\mathbf{w}$ , usually sparse. The vector may be as simple as bag of words, or it may be more complex, separate bags for title, abstract and content, separate bags for nouns and verbs, etc. The model also assigns a set of independent components to a document somehow representing the topical content. In the GaP model the  $k$ -th component is a Gamma( $\alpha_k, \beta_k$ ) variable. In multinomial PCA or LDA it is a Gamma( $\alpha_k, 1$ ) variable, but then the set of variables is also normalized to yield a Dirichlet [2]. Finally, component distributions complete the model: each component  $k$  has proportion vector  $\boldsymbol{\Omega}_k$  giving the proportion of each word/lexeme in the vector  $\mathbf{w}$ , where  $\sum_j \Omega_{j,k} = 1$ . We denote the total count of  $\mathbf{w}$  by  $w_0 = \sum_k w_k$ . The distribution for document  $\mathbf{w}$ , is then given using hidden components  $\mathbf{m}$  and model parameters  $\boldsymbol{\Omega}$ :

$$\begin{aligned} m_k &\sim \text{Gamma}(\alpha_k, \beta_k) && \text{for } k = 1, \dots, K \\ w_j &\sim \text{Poisson} \left( \sum_k \Omega_{j,k} m_k \right) && \text{for } j = 1, \dots, J \end{aligned}$$

Alternatively, the distribution on  $\mathbf{w}$  can be represented as:  $w_0 \sim \text{Poisson}(\sum_k m_k)$  and  $\mathbf{w}$  as a multinomial with total count  $w_0$  and proportions given by the normalized Poisson parameters  $\sum_k \Omega_{j,k} (m_k / \sum_k m_k)$ . If  $\beta_k = \beta$  is constant as in LDA then this normalized  $\mathbf{m}$  is a Dirichlet and the totals safely ignored.

With the  $\beta_k$  constant, the models can be fit using mean field, maximum likelihood, or two different kinds of Gibbs estimation. These introduce an additional set of quantities which are the word/lexeme counts  $w_j$  broken out into a term for each component,  $w_{j,k}$  where  $\sum_k w_{j,k} = w_j$  and  $w_{j,k} \sim \text{Poisson}(\Omega_{j,k} m_k)$ . According to Pritchard *et al.*'s Gibbs algorithm, we resample the set of  $w_{j,k}$  and  $\mathbf{m}$  for each document and then the component proportions  $\boldsymbol{\Omega}_k$ . By Griffiths and Steyvers, we first integrate out  $\boldsymbol{\Omega}_k$  (using Dirichlet normalizations) and  $\mathbf{m}$  (using Gamma normalizations) leaving only the  $w_{j,k}$  to sample.

### 1.3 Setting up Topic Specific Ranking

Topic specific page rank can work off the normalized component values  $m_k^* = m_k / \sum_k m_k$  for each document. For documents  $i = 1, \dots, I$ , let these be  $m_{i,k}^*$ . The restart vector  $\mathbf{r}$  for topic  $k$  is then given by  $r_i = m_{i,k}^* / \sum_i m_{i,k}^*$ . The topic relevance is more complicated. In general under the Gamma( $\alpha_k, 1$ ) version of discrete PCA, most pages have a mix of topics with perhaps 5-10 different topics or components occurring for one document. Thus a document with  $m_k^* = 0.2$  in these cases can be said to have the relevant topical content, we rarely expect much more. Thus, to derive the staying vector  $\mathbf{t}$  from discrete PCA, we put the  $m_{i,k}^*$  through a scaled tanh function so that when  $m_{i,k}^* = 0.2$ ,  $t_i$  will already be near 1.

## 2 EXPERIMENTS

We downloaded the Wikipedia in July 2004. It has approximately 290,000 documents with over 1.5Gb of text, and a rich link structure. We ran discrete PCA using Pritchard *et al.*'s algorithm with  $K = 100$  components with Gamma(1/100, 1) priors, and using empirical priors for the component proportions  $\Omega_k$ . This uses the MPCA software<sup>1</sup>, using a 600 cycle burn-in and 200 recording cycles, about 2 days on a dual 3GHz CPU under Linux. Computing the set of 100 topic specific scores for the 290,000 documents takes 15 minutes using a naive algorithm with no handling of sparsity. We compared some typical URLs (those with a high topic proportion) with those having a high rank for the topic in the table below.

Common Words	Typical URLs	High-Ranked URLs
District, County, département, formed, Island, colony, territory, South species, tree, Genus, found mythology, God, goddess, spirit food, wine, fruit, WA, meat, made, popular	Acqueville, Calvados; Litateau; Villars-sous-Yens; Mossel Bay; Proprierty Governor; Parateuthis Spotted Owlet; Giant squid; Lunaria Maris; Tekkeitsertok; Lopes mate Banana chips; Chow mein; Tomato sauce	Département: County, Switzerland Island; Pacific Ocean; New Zealand Scientific classification; Animal; Plant Greek mythology; Roman mythology; Celtic myth. Food; Cooking; Wine; Fruit; Sugar

## 3 CONCLUSION

Topic specific scoring provided by the adapted random surfer model, as shown by the Wikipedia examples, provides a far more characteristic score for documents than the proportion of component. This score could be used in a variety of ways for determining the relevance of a query for a given user.

<sup>1</sup> Previous version is published at <http://cosco.hiit.fi/search/MPCA>

**Acknowledgements** The work was supported by the ALVIS project, funded by the IST Priority of the EU's 6th framework programme.

## References

1. D. Blei, A.Y. Ng, and M. Jordan. Latent Dirichlet allocation. In *Jnl. of Machine Learning Research*, **3**, 2003.
2. W. Buntine and A. Jakulin. Applying discrete PCA in data analysis. In *UAI 2004*, Banff, Canada.
3. J. Canny. GaP: a Factor Model for Discrete Data. In *SIGIR 2004*, Sheffield, UK.
4. N. Craswell and D. Hawking. Overview of the TREC 2003 web track. In *Proc. TREC 2003*.
5. T. Griffiths and M. Steyvers. Finding scientific topics. In *PNAS Colloquium*, 2004.
6. T. Haveliwala. Topic-specific pagerank. In *11th World Wide Web*, 2002.
7. T. Hofmann. Probabilistic latent semantic indexing. In *Research and Development in Information Retrieval*, 50–57, 1999.
8. R. Nallapati. Discriminative models for information retrieval. In *ACM SIGIR Conference*, 2004.
9. J. Pritchard, M. Stephens, and P. Donnelly. Inference of population structure using multilocus genotype data. In *Genetics* **155**, 945–959, 2000.
10. M. Richardson and P. Domingos. The Intelligent Surfer: Probabilistic Combination of Link and Content Information in PageRank. In *NIPS\*14*, 2002.