

Semi-supervised Categorization of Wikipedia collection by Label Expansion

Boris Chidlovskii

Xerox Research Centre Europe
6, chemin de Maupertuis, F-38240 Meylan, France

Abstract. We address the problem of categorizing a large set of linked documents with important content and structure aspects, for example, from Wikipedia collection proposed at the INEX XML Mining track. We cope with the case where there is a small number of labeled pages and a very large number of unlabeled ones. Due to the sparsity of the link based structure of Wikipedia, we apply the spectral and graph-based techniques developed in the semi-supervised machine learning. We use the content and structure views of Wikipedia collection to build a transductive categorizer for the unlabeled pages. We report evaluation results obtained with the label propagation function which ensures a good scalability on sparse graphs.

1 Introduction

The objective of the INEX 2008 XML Mining challenge is to develop machine learning methods for structured data mining and to evaluate these methods for XML document mining tasks. The challenge proposes several datasets coming from different XML collections and covering a variety of classification and clustering tasks.

In this work, we address the problem of categorizing a very large set of linked XML documents with important content and structural aspects, for example, from Wikipedia online encyclopedia. We cope with the case where there is a small number of labeled pages and a much larger number of unlabeled ones. For example, when categorizing Web pages, some pages have been labeled manually and a huge amount of unlabeled pages is easily retrieved by crawling the Web. The semi-supervised approach to learning is motivated by the high cost of labeling data and the low cost for collecting unlabeled data. Withing XML Mining challenge 2008, the Wikipedia categorization challenge has been indeed set in the semi-supervised mode, where only 10% of page labels are available at the training step.

Wikipedia (<http://www.wikipedia.org>) is a free multilingual encyclopedia project supported by the non-profit Wikipedia foundation. In April 2008, Wikipedia accounted for 10 million articles which have been written collaboratively by volunteers around the world, and almost all of its articles can be edited by anyone who can access the Wikipedia website. Launched in 2001, it is currently the largest and most popular general reference work on the Internet. Automated analysis, mining and categorization of Wikipedia pages can serve to improve its internal structure as well as to enable its integration as an external resource in different applications.

Any Wikipedia page is created, revised and maintained according to certain policies and guidelines [1]. Its edition follows certain rules for organizing the content and structuring it in the form of sections, abstract, table of content, citations, links to relevant pages, etc.. In the following, we distinguish between four different aspects (or views) of a Wikipedia page:

Content - the set of words occurred in the page.

Structure - the set of HTML/XML tags, attributes and their values in the page. These elements control the presentation of the page content to the viewer. In the extended version, we may consider some combinations of elements of the page structure, like the root-to-leaf paths or their fragments.

Links - the set of hyperlinks in the page.

Metadata - all the information present in the page Infobox, including the template, its attributes and values. Unlike the content and structure, not all pages include infoboxes [3].

We use these alternative views to generate a transductive categorizer for the Wikipedia collection. One categorizer representing the content view is based on the text of page. Another categorizer represents the structural view, it is based on the structure and Infobox characteristics of the page.

Due to the transductive setting of the XML Mining challenge, we test the graph-based semi-supervised methods which construct the similarity graph $W = \{w_{ij}\}$ and apply a function propagating labels from labeled nodes to unlabeled ones. We first build the content categorizer, with weights w_{ij} being the textual similarity between two pages. We then build the structure categorizer, where weights w_{ij} are obtained from the structure and Infobox similarity between the pages. Finally, we linearly combine the two categorizers to get the optimal performance.

2 Graph-based semi-supervised learning

In the semi-supervised setting, we dispose labeled and unlabeled elements. In the graph-based approach [4, 5] to linked documents, one node in the graph represents one page. We assume a weighted graph G having n nodes indexed from 1 to n . We associate with graph G a symmetric weight matrix W where all weights are non-negative ($w_{ij} > 0$), and weight w_{ij} represents the similarity between nodes i and j in G . If $w_{ij} = 0$, there is no edge between nodes i and j .

We assume that the first l training nodes have labels, y_1, y_2, \dots, y_l , where y_i are from the category label set C , and the remaining $u = n - l$ nodes are unlabeled. The goal is to predict the labels y_{l+1}, \dots, y_n by exploiting the structure of graph G . According to the *smoothness* assumption, a label of an unlabeled node is likely to be similar to the labels of its neighboring nodes. A more strongly connected neighbor node will more significantly affect the node.

Assume the category set C includes c different labels. We define the binary label vector Y_i for node i , where $Y_i = \{y_{ij} | y_{ij} = 1 \text{ if } j = y_i, 0 \text{ otherwise}\}$. We equally introduce the category prediction vector \hat{Y}_i for node i . All such vectors for n nodes

define a $n \times c$ -dimensional score matrix $\hat{Y} = (\hat{Y}_1, \dots, \hat{Y}_n)$. At the learning step, we determine \hat{Y} using all the available information. At the prediction step, the category labels are predicted by thresholding the score vectors $\hat{Y}_{l+1}, \dots, \hat{Y}_n$.

The graph-based methods assume the following:

1. the score \hat{Y}_i should be close to the given label vectors Y_i in training nodes, and
2. the score \hat{Y}_i should not be too different from the scores of neighbor nodes.

There exist a number of graph-based methods [5]; we test some of them and report on one called the *label expansion* [4]. According to this approach, at each step, node i in graph G receives a contribution from its neighbors j weighted by the normalized weight w_{ij} , and an additional small contribution given by its initial value. This process can be expressed iteratively using the graph Laplacian matrix $L = D - W$, where $D = \text{diag}(d_i)$, $d_i = \sum_j w_{ij}$. The normalized Laplacian $\mathcal{L} = D^{-1/2}LD^{-1/2} = I - D^{-1/2}WD^{-1/2}$ can be used instead of L to get a similar result. The process is detailed in Algorithm 1 below.

Algorithm 1 Label expansion

Require: Symmetric matrix W , $w_{ij} \geq 0$ (and $w_{ii} := 0$)

Require: Labels y_i for $x_i, i = 1, \dots, l$

Ensure: Labels for x_{l+1}, \dots, x_n

- 1: Compute the diagonal degree matrix D by $d_{ii} := \sum_j w_{ij}$
 - 2: Compute the normalized graph Laplacian $\mathcal{L} := I - D^{-1/2}WD^{-1/2}$
 - 3: Initialize $\hat{Y}^{(0)} := (Y_1, \dots, Y_l, \mathbf{0}, \mathbf{0}, \dots, \mathbf{0})$, where $Y_i = \{y_{ik} | y_{ik} = 1 \text{ if } k = y_i, 0 \text{ otherwise}\}$
 - 4: Choose a parameter $\alpha \in [0, 1)$
 - 5: **while** not converged to $\hat{Y}^{(\infty)}$ **yet do**
 - 6: Iterate $\hat{Y}^{(t+1)} := \alpha\mathcal{L}\hat{Y}^{(t)} + (1 - \alpha)\hat{Y}^{(0)}$
 - 7: **end while**
 - 8: Label x_i by $\text{argmax}_j \hat{Y}_i^{(\infty)}$
-

It has been proved that Algorithm 1 always converges [4]. Indeed, the iteration equation can be represented as follows

$$\hat{Y}^{(t+1)} = (\alpha\mathcal{L})^{t+1}\hat{Y}^{(0)} + (1 - \alpha) \sum_{i=0}^t (\alpha\mathcal{L})^i \hat{Y}^{(0)}. \quad (1)$$

Matrix \mathcal{L} is a normalized Laplacian, its eigenvalues are known to be in $[-1, 1]$ range. Since $\alpha < 1$, eigenvalues of $\alpha\mathcal{L}$ are in $(-1, 1)$ range. Therefore, when $t \rightarrow \infty$, $(\alpha\mathcal{L})^t \rightarrow 0$.

Using the matrix decomposition, we have $\sum_{i=0}^{\infty} (\alpha\mathcal{L})^i \rightarrow (I - \alpha\mathcal{L})^{-1}$, so that we obtain the following convergence:

$$\hat{Y}^{(t)} \rightarrow \hat{Y}^{(\infty)} = (1 - \alpha)(I - \alpha\mathcal{L})^{-1}\hat{Y}^{(0)}. \quad (2)$$

The convergence rate of the algorithm depends on specific properties of matrix W , in particular, the eigenvalues of its Laplacian \mathcal{L} . In the worst case, the convergence takes $O(kn^2)$ time, where k is the number of neighbors of a point in the graph.

On the other hand, the score matrix \hat{Y} can be obtained by solving a large sparse linear system $(I - \alpha\mathcal{L})\hat{Y} = (1 - \alpha)Y^{(0)}$. This numerical problem has been intensively studied [2], and efficient algorithms, whose computational time is nearly linear in the number of non-zero entries in the matrix L . Therefore, the computation gets faster as the Laplacian matrix gets sparser.

2.1 Category mass regularization

Algorithm 1 generates a c -dimensional vector \hat{Y}_i for each unlabeled node i , where c is the number of categories and each element \hat{y}_{ij} between 0 and 1 gives a score for category j . To obtain the category for i , Algorithm 1 takes the category with the highest value, $\operatorname{argmax}_j \hat{y}_{ij}$. Such a rule works well when categories are well balanced. However, in real-world data categories are often unbalanced and the categorization resulting from Algorithm 1 may not reflect the prior category distribution.

To solve this problem, we perform the category mass normalization, similarly to [6]. It rescales categories in such a way that their respective weights over unlabeled examples match the prior category distribution estimated from labeled examples.

Category mass normalization is performed in the following way. First, let p_j denote the prior probability of category j estimated from the labeled examples: $p_j = \frac{1}{l} \sum_{i=1}^l y_{ij}$. Second, the mass of category j as given by the average of estimated weights of j over unlabeled examples, $m_j = \frac{1}{u} \sum_{i=l+1}^n \hat{y}_{ij}$. Then the category mass normalization consists in scaling each category j by the factor $v_j = \frac{p_j}{m_j}$. In other words, instead of the decision function $\operatorname{argmax}_j \hat{y}_{ij}$, we categorize node i in the category given by $\operatorname{argmax}_j v_j \hat{y}_{ij}$. The goal is to make the scaled masses match the prior category distribution, i.e. after normalization we have that for all j

$$p_j = \frac{v_j m_j}{\sum_{i=1}^c v_i m_i}.$$

Generally, such a scaling gives a better categorization performance when there are enough labeled data to accurately estimate the category distribution, and when the unlabeled data come from the same distribution. Moreover, if there exists such m that each category mass is $m_j = mp_j$, i.e., the masses already reflect the prior category distribution, then the mass normalization step has no effect, since $v_j = \frac{1}{m}$ for all j .

2.2 Graph construction

The label expansion algorithm starts with a graph G and associated weighted matrix W . To build the graph G for the Wikipedia collection, we first reuse its link structure by transforming directed links into undirected ones. We analyze the number of incoming and outgoing links for all pages in the Wikipedia collection. Figure 1 shows the In-Out frequencies for the corpus; note the log scale set for all dimensions.

In the undirected graph, we remove self-links as required by Algorithm 1. We then remove links between nodes with high weights w_{ij} having different labels in order to fit the smoothness condition. It turns out that the link graph is not totally connected. Figure 2 plots the link graph with the help of the Large Graph Layout package¹. As the figure shows, the graph includes one connected component and about 160 small components covering less than 1% of collection. The right plot in Figure 2 additionally projects the category information on the link graph, where each category is shown by a particular color.

We are also interested in building graphs G which are different the original link structure. The standard approach [4] is to build the k -NN (Nearest Neighbors) graph by taking the top k weights w_{ij} for each node. Unfortunately, the exhaustive k -NN procedure is infeasible even for the Wikipedia fragment used in the challenge. Thus we build a graph G' by modifying G with randomly sampling of node pairs from Wikipedia and selecting the top $k=100$ ones per node. Note using the content or structure similarity will produce different versions of G' . In the evaluation section, we report results of tests run on both G and G' graphs.

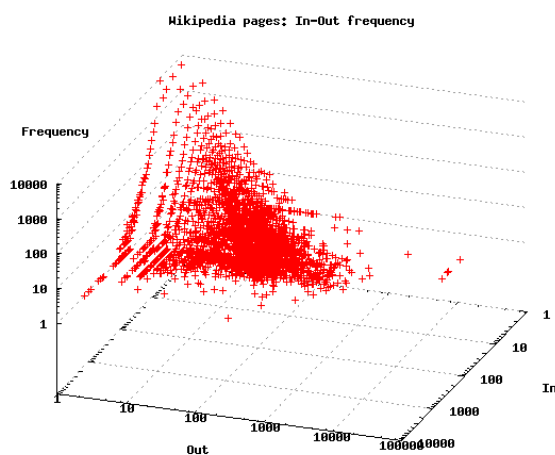


Fig. 1. Wikipedia nodes: In-Out frequencies.

Content matrix. To generate a content weighted matrix W , we extract descriptor x_i for node i in the graph by using "bag-of-words" model and the *tf-idf* values, (term frequency-inverted document frequency) as $x_{ij} = tf_{ij} \cdot idf_i$, where

¹ <http://bioinformatics.icmb.utexas.edu/lgl/>.

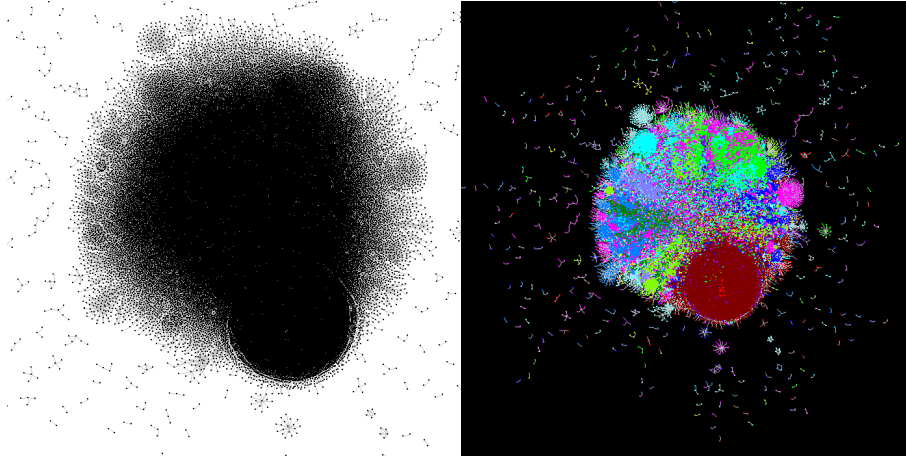


Fig. 2. Wikipedia corpus: the link graph plotted with LGL package.

- tf_{ij} is the term frequency given by $\frac{n_{i,j}}{\sum_k n_{k,j}}$, where n_{ij} is the number of occurrences of the term in document d_j , and the denominator is the number of occurrences of all terms in document d_j .
- idf_i is the inverted document frequency $\log \frac{n}{|\{d_j : t_i \in d_j\}|}$, where n is the total number of documents and $|\{d_j : t_i \in d_j\}|$ is the number of documents where the term t_i occurs.

The tf-idf weighting scheme is often used in the vector space model together with cosine similarity to determine the similarity between two documents.

Layout matrix. In the structure graph, node descriptors x_i are generated following the "bag-of-tags" approach which is similar to bag-of-words used in the content graph. Instead of words, it uses elements of the page structure. In the HTML formatted pages, the presentation is guided by instructions encoded as HTML tags, attributes and their values. The HTML structure forms a nested structure. The "bag-of-tags" model might have different instantiations, below we report some of them, where the terms form one of the following sets:

1. Set of tag names, like (table) or (font),
2. Set of descendant tag pairs, like (table, span) or (tr, td),
3. Set of root-to-leaf paths in HTML page, like (html, body, table, tr, td),
4. Set of (tag, attribute) pairs, like (table, font),
5. Set of (tag, attribute, attribute_value) triples, like (table, font, times).

For each of the above sets, we extract descriptors x_i for node i according to the conventional tf-idf weights. We build the weighted matrix W using the *structure similarity* between pages evaluated with "bag-of-tags" model and one of the listed tag sets.

Similarity measures. Once we have obtained description vectors x_i for all nodes in graph G , we can get the weighted matrix W by measuring a similarity between two nodes i and j in G . Two possible measures are the following:

1. The Gaussian (RBF) kernel of width σ , $w_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$, where the width σ is evaluated from the variance of the descriptors x_i .
2. The standard cosine function, $w_{ij} = \frac{x_i \cdot x_j}{\|x_i\| \|x_j\|}$.

3 Evaluation

The collection used in the INEX XML Mining challenge is composed of $n=114,366$ pages from the Wikipedia XML Corpus; 10% of these pages have been annotated ($l=11,437$) with $c=15$ categories, 90% of pages ($u=102,929$) are unannotated. Some global characteristics of the corpus is given in Table 1. The word set is composed of all lexemized keywords; neither non-English words nor stop words were excluded.

Set	Size	Set	Size	Set	Size
Text words	727,667	Tag+attribute pairs	5,772	Infobox templates	602
Infobox tags	1,208	Root-to-leaf paths	110,099	Hyperlinks	636,187
Tags	1,257	Tag+attribute+value triples	943,422		

Table 1. Wikipedia collection: some global characteristics.

In all experiments, we measure the accuracy of a transductive categorizer using 10-fold cross validation on the training set (in the presence of unlabeled data). As the baseline method, we used the semi-supervised learning with the multi-class SVM, with x_i node descriptors being feature values. We also combine content, structure and infobox views, by concatenating the corresponding descriptors. However, direct concatenation of these alternative views brings no benefit (see 'Content+Tag+Attr+IB' line in Table 2).

For the label expansion method, we tested the link-based graph G and the sampling-enriched link graph G' , with matrices W_c and W_s being generated with content or structure similarity measures, respectively. Using tag+attribute descriptors enriched with infoboxes generates a transductive categorizer whose performance is comparable to the content categorizer. Finally, the best performance is achieved by combining two graphs G' with weights w_{ij} obtained the content and structure similarity. The resulting weighted matrix is obtained as $W = \alpha W_s + (1 - \alpha) W_c$ with the optimal $\alpha = 0.34$ obtained by the cross validation. The right column in Table 2 reports the evaluation results for different (graph, similarity) combinations and aligns them with the SVM results.

Three submissions to the INEX challenge have been done with three values of α : 0.34, 0.37 and 0.38. They yielded the accuracy values 73.71%, 73.79% and 73.47%, respectively. Despite the high density, these results are a clear underperformance with respect to the cross validation tests and results by the relatively simpler SVM classifiers. Nevertheless, the graph-based methods clearly represent a powerful mechanism for classifying the linked data like Wikipedia; thus we intend to conduct further studies to realize their potential.

SVM Method	Accuracy(%)	LP Method	Accuracy (%)	Comment
Content	73.312	G -Content	72.104	Cosine idem
		G' -Content	75.03	
Tag+Attr	72.744	G' -Tag+Attr	72.191	Gaussian, $\delta=1.5$ idem
		G' -Paths	64.824	
Tag+Attr+InfoBox	72.921	G -Tag+Attr+IB	70.287	idem
Content+Tag+Attr+IB	73.127	G' -Tag+Attr+IB	74.753	idem $\alpha=0.34$
		G' -Content + G' -TAIB	77.572	

Table 2. Performance evaluation for different methods.

4 Conclusion

We applied the graph-based semi-supervised methods to the categorization challenge defined on Wikipedia collection. The methods benefit from the recent advances in spectral graph analysis and offer a good scalability in the case of sparse graphs. From the series of experiments on the Wikipedia collection, we may conclude that the optimal graph construction remains the main issue. In particular, the good choice of the graph generator and node similarity distance is a key to get an accurate categorizer. The use of the Wikipedia link graph offers the baseline performance, while the sampling technique brings a clear improvement. Nevertheless, its impact remains limited as the graph smoothness requirement is satisfied only partially. To better satisfy the requirement, we would need a smarter sampling technique and an extension of the method toward the graph regularization and an advanced text analysis.

5 Acknowledgment

This work is partially supported by the ATASH Project co-funded by the French Association on Research and Technology (ANRT).

References

1. D. Riehle. How and why Wikipedia works: an interview with Angela Beesley, Elisabeth Bauer, and Kizu Naoko. In *Proc. WikiSym '06*, pages 3–8, New York, NY, USA, 2006.
2. Y. Saad. *Iterative Methods for Sparse Linear Systems, 2nd Edition*. SIAM, 2008.
3. F. Wu and D. S. Weld. Autonomously semantifying Wikipedia. In *CIKM '07: Proc. 16th ACM Conf. Information and Knowledge Management*, pages 41–50, 2007.
4. D. Zhou, O. Bousquet, T. Navin Lal, J. Weston, and B. S. Olkoph. Learning with local and global consistency. In *Advances in NIPS 16*, pages 321–328. MIT Press, 2004.
5. X. Zhu. Semi-supervised learning literature survey. In *University of Wisconsin-Madison, CD Department, Technical Report 1530*, 2005.
6. X. Zhu, Z. Ghahramani, and J. Lafferty. Semisupervised learning using Gaussian fields and harmonic functions. In *Proc. 12th Intern. Conf. Machine Learning*, pages 912–919, 2003.