Combining Content and Link for Classification using Matrix Factorization

Shenghuo Zhu Kai Yu Yun Chi Yihong Gong
{zsh,kyu,ychi,ygong}@sv.nec-labs.com
NEC Laboratories America, Inc.
10080 North Wolfe Road SW3-350
Cupertino, CA 95014, USA

ABSTRACT

The world wide web contains rich textual contents that are interconnected via complex hyperlinks. This huge database violates the assumption held by most of conventional statistical methods that each web page is considered as an independent and identical sample. It is thus difficult to apply traditional mining or learning methods for solving web mining problems, e.g., web page classification, by exploiting both the content and the link structure. The research in this direction has recently received considerable attention but are still in an early stage. Though a few methods exploit both the link structure or the content information, some of them combine the only authority information with the content information, and the others first decompose the link structure into hub and authority features, then apply them as additional document features. Being practically attractive for its great simplicity, this paper aims to design an algorithm that exploits both the content and linkage information, by carrying out a joint factorization on both the linkage adjacency matrix and the document-term matrix, and derives a new representation for web pages in a low-dimensional factor space, without explicitly separating them as content, hub or authority factors. Further analysis can be performed based on the compact representation of web pages. In the experiments, the proposed method is compared with state-of-the-art methods and demonstrates an excellent accuracy in hypertext classification on the WebKB and Cora benchmarks.

Categories and Subject Descriptors: H.3.3 [Information Systems]: Information Search and Retrieval

General Terms: Algorithms, Experimentation

Keywords: Link structure, Text content, Factor analysis, Matrix factorization

1. INTRODUCTION

With the advance of the World Wide Web, more and more hypertext documents become available on the Web. Some examples of such data include organizational and personal web pages (e.g, the WebKB benchmark data set, which contains university web pages), research papers (e.g., data in CiteSeer), online news articles, and customer-generated media (e.g., blogs). Comparing to data in traditional information management, in addition to content, these data on the Web also contain links: e.g., hyperlinks from a student's homepage pointing to the homepage of her advisor, paper citations, sources of a news article, comments of one blogger on posts from another blogger, and so on. Performing information management tasks on such structured data raises many new research challenges. In the following discussion, we use the task of web page classification as an illustrating example, while the techniques we develop in later sections are applicable equally well to many other tasks in information retrieval and data mining.

For the classification problem of web pages, a simple approach is to treat web pages as independent documents. The advantage of this approach is that many off-the-shelf classification tools can be directly applied to the problem. However, this approach relies only on the content of web pages and ignores the structure of links among them. Link structures provide invaluable information about properties of the documents as well as relationships among them. For example, in the WebKB dataset, the link structure provides additional insights about the relationship among documents (e.g., links often pointing from a student to her advisor or from a faculty member to his projects). Since some links among these documents imply the inter-dependence among the documents, the usual *i.i.d.* (independent and identical distributed) assumption of documents does not hold any more. From this point of view, the traditional classification methods that ignore the link structure may not be suitable.

On the other hand, a few studies, for example [25], rely solely on link structures. It is however a very rare case that content information can be ignorable. For example, in the Cora dataset, the content of a research article abstract largely determines the category of the article.

To improve the performance of web page classification, therefore, both link structure and content information should be taken into consideration. To achieve this goal, a simple approach is to convert one type of information to the other. For example, in spam blog classification, Kolari et al. [13] concatenate outlink features with the content features of the blog. In document classification, Kurland and Lee [14] convert content similarity among documents into weights of links. However, link and content information have different properties. For example, a link is an actual piece of evidence that represents an asymmetric relationship whereas the content similarity is usually defined conceptually for every pair of documents in a symmetric way. Therefore, directly converting one type of information to the other usually degrades the quality of information. On the other hand, there exist some studies, as we will discuss in detail in related work, that consider link information and content information separately and then combine them. We argue that such an approach ignores the inherent consistency between link and con-

SIGIR'07, July 23–27, 2007, Amsterdam, The Netherlands.

tent information and therefore fails to combine the two seamlessly. Some work, such as [3], incorporates link information using co-citation similarity, but this may not fully capture the *global* link structure. In Figure 1, for example, web pages v_6 and v_7 co-cite web page v_8, implying that v_6 and v_7 are similar to each other. In turns, v_4 and v_5 should be similar to each other, since v_4 and v_5 cite similar web pages v_6 and v_7, respectively. But using co-citation similarity, the similarity between v_4 and v_5 is zero without considering other information.

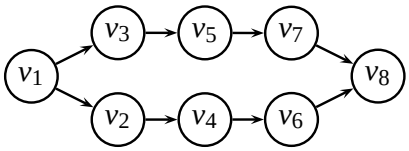


Figure 1: An example of link structure

In this paper, we propose a simple technique for analyzing inter-connected documents, such as web pages, using factor analysis[18]. In the proposed technique, both content information and link structures are seamlessly combined through a single set of *latent factors*. Our model contains two components. The first component captures the content information. This component has a form similar to that of the latent topics in the Latent Semantic Indexing (LSI) [8] in traditional information retrieval. That is, documents are decomposed into latent topics/factors, which in turn are represented as term vectors. The second component captures the information contained in the underlying link structure, such as links from homepages of students to those of faculty members. A factor can be loosely considered as a *type* of documents (e.g., those homepages belonging to students). It is worth noting that we do not explicitly define the semantic of a factor *a priori*. Instead, similar to LSI, the factors are learned from the data. Traditional factor analysis models the variables associated with entities through the factors. However, in analysis of link structures, we need to model the relationship of two ends of links, i.e., edges between vertex pairs. Therefore, the model should involve factors of both vertices of the edge. This is a key difference between traditional factor analysis and our model. In our model, we connect two components through a set of shared factors, that is, the latent factors in the second component (for contents) are tied to the factors in the first component (for links). By doing this, we search for a unified set of latent factors that best explains both content and link structures simultaneously and seamlessly.

In the formulation, we perform factor analysis based on matrix factorization: solution to the first component is based on factorizing the term-document matrix derived from content features; solution to the second component is based on factorizing the adjacency matrix derived from links. Because the two factorizations share a common base, the discovered bases (latent factors) explain both content information and link structures, and are then used in further information management tasks such as classification.

This paper is organized as follows. Section 2 reviews related work. Section 3 presents the proposed approach to analyze the web page based on the combined information of links and content. Section 4 extends the basic framework and a few variants for fine tune. Section 5 shows the experiment results. Section 6 discusses the details of this approach and Section 7 concludes.

2. RELATED WORK

In the content analysis part, our approach is closely related to Latent Semantic Indexing (LSI) [8]. LSI maps documents into a lower dimensional latent space. The latent space implicitly captures a large portion of information of documents, therefore it is called the latent semantic space. The similarity between documents could be defined by the dot products of the corresponding vectors of documents in the latent space. Analysis tasks, such as classification, could be performed on the latent space. The commonly used singular value decomposition (SVD) method ensures that the data points in the latent space can optimally reconstruct the original documents. Though our approach also uses latent space to represent web pages (documents), we consider the link structure as well as the content of web pages.

In the link analysis approach, the framework of hubs and authorities (HITS) [12] puts web page into two categories, hubs and authorities. Using recursive notion, a hub is a web page with many outgoing links to authorities, while an authority is a web page with many incoming links from hubs. Instead of using two categories, PageRank [17] uses a single category for the recursive notion, an authority is a web page with many incoming links from authorities. He et al. [9] propose a clustering algorithm for web document clustering. The algorithm incorporates link structure and the co-citation patterns. In the algorithm, all links are treated as undirected edge of the link graph. The content information is only used for weighing the links by the textual similarity of both ends of the links. Zhang et al. [23] uses the undirected graph regularization framework for document classification. Achlioptas et al[2] decompose the web into hub and authority attributes then combine them with content. Zhou et al. [25] and [24] propose a directed graph regularization framework for semi-supervised learning. The framework combines the hub and authority information of web pages. But it is difficult to combine the content information into that framework. Our approach consider the content and the directed linkage between topics of source and destination web pages in *one step*, which implies the topic combines the information of web page as authorities and as hubs in *a single set of factors*.

Cohn and Hofmann [6] construct the latent space from both content and link information, using content analysis based on probabilistic LSI (PLSI) [10] and link analysis based on PHITS [5]. The major difference between the approach of [6] (PLSI+PHITS) and our approach is in the part of link analysis. In PLSI+PHITS, the link is constructed with the linkage from the topic of the source web page to the destination web page. In the model, the outgoing links of the destination web page have no effect on the source web page. In other words, the overall link structure is not utilized in PHITS. In our approach, the link is constructed with the linkage between the factor of the source web page and the factor of the destination web page, instead of the destination web page itself. The factor of the destination web page contains information of its outgoing links. In turn, such information is passed to the factor of the source web page. As the result of matrix factorization, the factor forms a factor graph, a miniature of the original graph, preserving the major structure of the original graph.

Taskar et al. [19] propose relational Markov networks (RMNs) for entity classification, by describing a conditional distribution of entity classes given entity attributes and relationships. The model was applied to web page classification, where web pages are entities and hyperlinks are treated as relationships. RMNs apply conditional random fields to define a set of potential functions on cliques of random variables, where the link structure provides hints to form the cliques. However the model does not give an off-the-shelf solution, because the success highly depends on the arts of designing

the potential functions. On the other hand, the inference for RMNs is intractable and requires belief propagation.

The following are some work on combining documents and links, but the methods are loosely related to our approach. The experiments of [21] show that using terms from the linked document improves the classification accuracy. Chakrabarti et al.[3] use co-citation information in their classification model. Joachims et al.[11] combine text kernels and co-citation kernels for classification. Oh et al [16] use the Naive Bayesian frame to combine link information with content.

3. OUR APPROACH

In this section we will first introduce a novel matrix factorization method, which is more suitable than conventional matrix factorization methods for link analysis. Then we will introduce our approach that jointly factorizes the document-term matrix and link matrix and obtains compact and highly indicative factors for representing documents or web pages.

3.1 Link Matrix Factorization

Suppose we have a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the vertex set $\mathcal{V} = \{v_i\}_{i=1}^n$ represents the web pages and the edge set $\mathcal{E}$ represents the hyperlinks between web pages. Let $A = \{a_{sd}\}$ denotes the $n \times n$ adjacency matrix of $\mathcal{G}$, which is also called the *link matrix* in this paper. For a pair of vertices, v_s and v_d, let $a_{sd} = 1$ when there is an edge from v_s to v_d, and $a_{sd} = 0$, otherwise. Note that A is an asymmetric matrix, because hyperlinks are directed.

Most machine learning algorithms assume a feature-vector representation of instances. For web page classification, however, the link graph does not readily give such a vector representation for web pages. If one directly uses each row or column of A for the job, she will suffer a very high computational cost because the dimensionality equals to the number of web pages. On the other hand, it will produces a poor classification accuracy (see our experiments in Section 5), because A is extremely sparse[1].

The idea of link matrix factorization is to derive a high-quality feature representation Z of web pages based on analyzing the link matrix A, where Z is an $n \times l$ matrix, with each row being the l-dimensional feature vector of a web page. The new representation of web pages captures the principal factors of the link structure and makes further processing more efficient.

One may use a method similar to LSI, to apply the well-known *principal component analysis* (PCA) for deriving Z from A. The corresponding optimization problem [2] is

$$\min_{Z,U} \|A - ZU^\top\|_F^2 + \gamma\|U\|_F^2 \tag{1}$$

where γ is a small positive number, U is an $l \times n$ matrix, and $\|\cdot\|_F$ is the Frobenius norm. The optimization aims to approximate A by $ZU^\top$, a product of two low-rank matrices, with a regularization on U. In the end, the i-th row vector of Z can be thought as the hub feature vector of vertex v_i, and the row vector of U can be thought as the authority features. A link generation model proposed in [2] is similar to the PCA approach. Since A is a nonnegative matrix here, one can also consider to put nonnegative constraints on U and Z, which produces an algorithm similar to PLSA [10] and NMF [20].

[1]Due to the sparsity of A, links from two similar pages may not share any common target pages, which makes them to appear "dissimilar". However the two pages may be indirectly linked to many common pages via their neighbors.

[2]Another equivalent form is $\min_{Z,U} \|A - ZU^\top\|_F^2$, s. t. $U^\top U = I$. The solution Z is identical subject to a scaling factor.

However, despite its popularity in matrix analysis, PCA (or other similar methods like PLSA) is restrictive for link matrix factorization. The major problem is that, PCA ignores the fact that the rows and columns of A are indexed by exactly the same set of objects (i.e., web pages). The approximating matrix $\tilde{A} = ZU^\top$ shows no evidence that links are within the same set of objects. To see the drawback, let's consider a link transitivity situation $v_i \to v_s \to v_j$, where page i is linked to page s which itself is linked to page j. Since $\tilde{A} = ZU^\top$ treats A as links from web pages $\{v_i\}$ to a different set of objects, let it be denoted by $\{o_i\}$, $\tilde{A} = ZU^\top$ actually splits an "linked" object o_s from v_s and breaks down the link path into two parts $v_i \to o_s$ and $v_s \to o_j$. This is obviously a miss interpretation to the original link path.

To overcome the problem of PCA, in this paper we suggest to use a different factorization:

$$\min_{Z,U} \|A - ZUZ^\top\|_F^2 + \gamma\|U\|_F^2 \tag{2}$$

where U is an $l \times l$ full matrix. Note that U is not symmetric, thus $ZUZ^\top$ produces an asymmetric matrix, which is the case of A. Again, each row vector of Z corresponds to a feature vector of a web pages. The new approximating form $\tilde{A} = ZUZ^\top$ puts a clear meaning that the links are between *the same set of objects*, represented by features Z. The factor model actually maps each vertex, v_i, into a vector $\mathbf{z}_i = \{z_{i,k}; 1 \le k \le l\}$ in the $\mathbb{R}^l$ space. We call the $\mathbb{R}^l$ space the factor space. Then, $\{\mathbf{z}_i\}$ encodes the information of incoming and outgoing connectivity of vertices $\{v_i\}$. The factor loadings, U, explain how these observed connections happened based on $\{\mathbf{z}_i\}$. Once we have the vector $\mathbf{z}_i$, we can use many traditional classification methods (such as SVMs) or clustering tools (such as K-Means) to perform the analysis.

Illustration Based on a Synthetic Problem

To further illustrate the advantages of the proposed link matrix factorization Eq. (2), let us consider the graph in Figure 1. Given

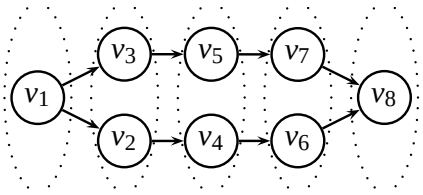


Figure 2: Summarize Figure 1 with a factor graph

these observations, we can summarize the graph by grouping as *factor graph* depicted in Figure 2. In the next we preform the two factorization methods Eq. (2) and Eq. (1) on this link matrix. A good low-rank representation should reveal the structure of the factor graph.

First we try PCA-like decomposition, solving Eq. (1) and obtaining

$$Z = \begin{bmatrix} 1. & 1. & 0 & 0 & 0 \\ 0 & 0 & -.6 & -.7 & .1 \\ 0 & 0 & .0 & .6 & -.0 \\ 0 & 0 & .8 & -.4 & .3 \\ 0 & 0 & .2 & -.2 & -.9 \\ .7 & .7 & 0 & 0 & 0 \\ .7 & .7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad U = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ .5 & -.5 & 0 & 0 & 0 \\ .5 & -.5 & 0 & 0 & 0 \\ 0 & 0 & -0.6 & -.7 & .1 \\ 0 & 0 & .0 & .6 & -.0 \\ 0 & 0 & .8 & -.4 & .3 \\ 0 & 0 & .2 & -.2 & -.9 \\ .7 & .7 & 0 & 0 & 0 \end{bmatrix}$$

We can see that the row vectors of v_6 and v_7 are the same in Z, indicating that v_6 and v_7 have the same hub attributes. The row

vectors of v_2 and v_3 are the same in U, indicating that v_2 and v_3 have the same authority attributes. It is not clear to see the similarity between v_4 and v_5, because their inlinks (and outlinks) are different.

Then, we factorize A by $ZUZ^\top$ via solving Eq. (2), and obtain the results

$$Z = \begin{bmatrix} -.8 & -.5 & .3 & -.1 & -.0 \\ -.0 & .4 & .6 & -.1 & -.4 \\ -.0 & .4 & .6 & -.1 & -.4 \\ .3 & -.2 & .3 & -.4 & .3 \\ .3 & -.2 & .3 & -.4 & .3 \\ -.4 & .5 & .0 & -.2 & .6 \\ -.4 & .5 & .0 & -.2 & .6 \\ -.1 & .1 & -.4 & -.8 & -.4 \end{bmatrix} \quad U = \begin{bmatrix} -.1 & -.2 & -.4 & .6 & .7 \\ .2 & -.5 & -.5 & -.5 & .0 \\ .1 & .1 & .4 & -.4 & .3 \\ .1 & -.2 & -.0 & .3 & -.1 \\ -.3 & .3 & -.5 & -.4 & -.2 \end{bmatrix}$$

The resultant Z is very consistent with the clustering structure of vertices: the row vectors of v_2 and v_3 are the same, those of v_4 and v_5 are the same, those of v_6 and v_7 are the same. Even interestingly, if we add constraints to ensure Z and U be nonnegative, we have

$$Z = \begin{bmatrix} 1. & 0 & 0 & 0 & 0 \\ 0 & .9 & 0 & 0 & 0 \\ 0 & .9 & 0 & 0 & 0 \\ 0 & 0 & .7 & 0 & 0 \\ 0 & 0 & .7 & 0 & 0 \\ 0 & 0 & 0 & .9 & 0 \\ 0 & 0 & 0 & .9 & 0 \\ 0 & 0 & 0 & 0 & 1. \end{bmatrix} \quad U = \begin{bmatrix} 0 & 1. & 0 & 0 & 0 \\ 0 & 0 & .7 & 0 & 0 \\ 0 & 0 & 0 & .7 & 0 \\ 0 & 0 & 0 & 0 & 1. \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

which clearly tells the assignment of vertices to clusters from Z and the links of factor graph from U. When the interpretability is not critical in some tasks, for example, classification, we found that it achieves better accuracies without the nonnegative constraints.

Given our above analysis, it is clear that the factorization $ZUZ^\top$ is more expressive than $ZU^\top$ in representing the link matrix A.

3.2 Content Matrix Factorization

Now let us consider the content information on the vertices. To combine the link information and content information, we want to use the same latent space to approximate the content as the latent space for the links. Using the bag-of-words approach, we denote the content of web pages by an $n \times m$ matrix C, each of whose rows represents a document, each column represents a keyword, where m is the number of keywords. Like the latent semantic indexing (LSI) [8], the l-dimensional latent space for words is denoted by an $m \times l$ matrix V. Therefore, we use $ZV^\top$ to approximate matrix C,

$$\min_{V,Z} \|C - ZV^\top\|_F^2 + \beta\|V\|_F^2, \tag{3}$$

where β is a small positive number, $\beta\|V\|_F^2$ serves as a regularization term to improve the robustness.

3.3 Joint Link-Content Matrix Factorization

There are many ways to employ both the content and link information for web page classification. Our idea in this paper is not to simply combine them, but rather to *fuse* them into a single, consistent, and compact feature representation. To achieve this goal, we solve the following problem,

$$\min_{U,V,Z} \Big\{ \mathcal{J}(U,V,Z) \stackrel{\text{def}}{=} \|A - ZUZ^\top\|_F^2 + \alpha\|C - ZV^\top\|_F^2 + \gamma\|U\|_F^2 + \beta\|V\|_F^2 \Big\}. \tag{4}$$

Eq. (4) is the joined matrix factorization of A and C with regularization. The new representation Z is ensured to capture both the structures of the link matrix A and the content matrix C. Once we find the optimal Z, we can apply the traditional classification or clustering methods on vectorial data Z. The relationship among these matrices can be depicted as Figure 3.

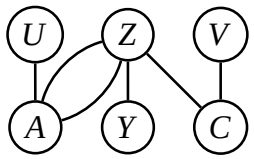


Figure 3: Relationship among the matrices. Node Y is the target of classification.

Eq. (4) can be solved using gradient methods, such as the conjugate gradient method and quasi-Newton methods. Then main computation of gradient methods is evaluating the object function $\mathcal{J}$ and its gradients against variables,

$$\begin{aligned} \frac{\partial \mathcal{J}}{\partial U} &= \left(Z^\top ZUZ^\top Z - Z^\top AZ\right) + \gamma U, \\ \frac{\partial \mathcal{J}}{\partial V} &= \alpha\left(VZ^\top Z - C^\top Z\right) + \beta V, \\ \frac{\partial \mathcal{J}}{\partial Z} &= \left(ZU^\top Z^\top ZU + ZUZ^\top ZU^\top - A^\top ZU - AZU^\top\right) \\ &\quad + \alpha\left(ZV^\top V - CV\right). \end{aligned}$$

Because of the sparsity of A, the computational complexity of multiplication of A and Z is $O(\mu_A l)$, where μ_A is the number of nonzero entries in A. Similarly, the computational complexity of $C^\top Z$ and CV is $O(\mu_C l)$, where μ_C is the number of nonzero entries in C. The computational complexity of the rest multiplications in the gradient computation is $O(nl^2)$. Therefore, the total computational complexity in one iteration is $O(\mu_A l + \mu_C l + nl^2)$. The number of links and the number of words in a web page are relatively small comparing to the number of web pages, and are almost constant as the number of web pages/documents increases, i.e. $\mu_A = O(n)$ and $\mu_C = O(n)$. Therefore, theoretically the computation time is almost linear to the number of web pages/documents, n.

4. SUPERVISED MATRIX FACTORIZATION

Consider a web page classification problem. We can solve Eq. (4) to obtain Z as Section 3, then use a traditional classifier to perform classification. However, this approach does not take data labels into account in the first step. Believing that using data labels improves the accuracy by obtaining a better Z for the classification, we consider to use the data labels to guide the matrix factorization, called *supervised matrix factorization* [22]. Because some data used in the matrix factorization have no label information, the supervised matrix factorization falls into the category of semi-supervised learning.

Let $\mathcal{C}$ be the set of classes. For simplicity, we first consider binary class problem, i.e. $\mathcal{C} = \{-1, 1\}$. Assume we know the labels $\{y_i\}$ for vertices in $\mathcal{T} \subset \mathcal{V}$. We want to find a hypothesis $h : \mathcal{V} \to \mathbb{R}$, such that we assign v_i to 1 when $h(v_i) \geq 0$, -1 otherwise. We assume a transform from the latent space to $\mathbb{R}$ is linear, i.e.

$$h(v_i) = \mathbf{w}^\top \phi(v_i) + b = \mathbf{w}^\top \mathbf{z}_i + b, \tag{5}$$

School	course	dept.	faculty	other	project	staff	student	total
Cornell	44	1	34	581	18	21	128	827
Texas	36	1	46	561	20	2	148	814
Washington	77	1	30	907	18	10	123	1166
Wisconsin	85	0	38	894	25	12	156	1210

Table 1: Dataset of WebKB

where $\mathbf{w}$ and b are parameters to estimate. Here, $\mathbf{w}$ is the norm of the decision boundary. Similar to Support Vector Machines (SVMs) [7], we can use the hinge loss to measure the loss,

$$\sum_{i:v_i \in \mathcal{T}} \left[1 - y_i h(v_i)\right]_+,$$

where $[x]_+$ is x if $x \geq 0$, 0 if $x < 0$. However, the hinge loss is not smooth at the hinge point, which makes it difficult to apply gradient methods on the problem. To overcome the difficulty, we use a smoothed version of hinge loss for each data point,

$$g(y_i h(v_i)), \tag{6}$$

where

$$g(x) = \begin{cases} 0 & \text{when } x \geq 2, \\ 1 - x & \text{when } x \leq 0, \\ \frac{1}{4}(x-2)^2 & \text{when } 0 < x < 2. \end{cases}$$

We reduce a multiclass problem into multiple binary ones. One simple scheme of reduction is the *one-against-rest* coding scheme. In the one-against-rest scheme, we assign a label vector for each class label. The element of a label vector is 1 if the data point belongs the corresponding class, -1, if the data point does not belong the corresponding class, 0, if the data point is not labeled. Let Y be the label matrix, each column of which is a label vector. Therefore, Y is a matrix of $n \times c$, where c is the number of classes, $|\mathcal{C}|$. Then the values of Eq. (5) form a matrix

$$H = ZW^\top + \mathbf{1}\mathbf{b}^\top, \tag{7}$$

where $\mathbf{1}$ is a vector of size n, whose elements are all one, W is a $c \times l$ parameter matrix, and $\mathbf{b}$ is a parameter vector of size c. The total loss is proportional to the sum of Eq. (6) over all labeled data points and the classes,

$$\mathcal{L}_Y(W, \mathbf{b}, Z) = \lambda \sum_{i:v_i \in \mathcal{T}, j \in \mathcal{C}} g(Y_{ij} H_{ij}),$$

where λ is the parameter to scale the term.

To derive a robust solution, we also use Tikhonov regularization for W,

$$\Omega_W(W) = \frac{\nu}{2} \|W\|_F^2,$$

where ν is the parameter to scale the term.

Then the supervised matrix factorization problem becomes

$$\min_{U,V,Z,W,\mathbf{b}} \mathcal{J}_{\mathrm{s}}(U, V, Z, W, \mathbf{b}) \tag{8}$$

where

$$\mathcal{J}_{\mathrm{s}}(U, V, Z, W, \mathbf{b}) = \mathcal{J}(U, V, Z) + \mathcal{L}_Y(W, \mathbf{b}, Z) + \Omega_W(W).$$

We can also use gradient methods to solve the problem of Eq. (8). The gradients are

$$\begin{aligned} \frac{\partial \mathcal{J}_{\mathrm{s}}}{\partial U} &= \frac{\partial \mathcal{J}}{\partial U}, \\ \frac{\partial \mathcal{J}_{\mathrm{s}}}{\partial V} &= \frac{\partial \mathcal{J}}{\partial V}, \\ \frac{\partial \mathcal{J}_{\mathrm{s}}}{\partial Z} &= \frac{\partial \mathcal{J}}{\partial Z} + \lambda G W, \\ \frac{\partial \mathcal{J}_{\mathrm{s}}}{\partial W} &= \lambda G^\top Z + \nu W, \\ \frac{\partial \mathcal{J}_{\mathrm{s}}}{\partial \mathbf{b}} &= \lambda G^\top \mathbf{1}, \end{aligned}$$

where G is an $n \times c$ matrix, whose ik-th element is $Y_{ik} g'(Y_{ik} H_{ik})$, and

$$g'(x) = \begin{cases} 0 & \text{when } x \geq 2, \\ -1 & \text{when } x \leq 0, \\ \frac{1}{2}(x-2) & \text{when } 0 < x < 2. \end{cases}$$

Once we obtain $\mathbf{w}$, b, and Z, we can apply h on the vertices with unknown class labels, or apply traditional classification algorithms on Z to get the classification results.

5. EXPERIMENTS

5.1 Data Description

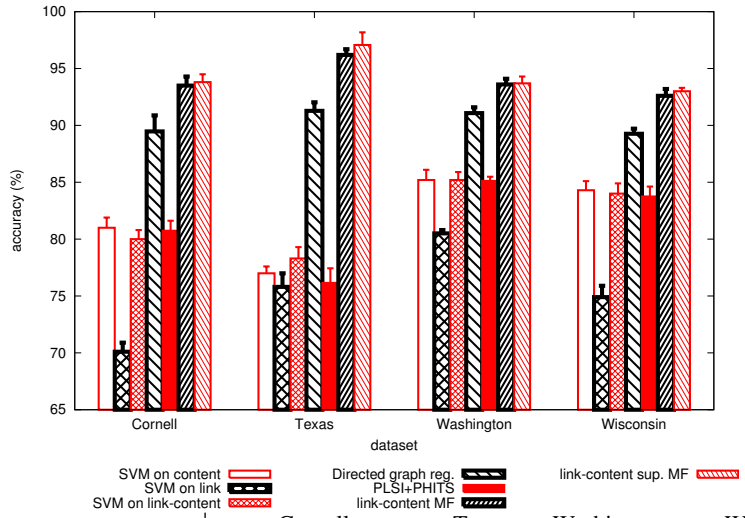
In this section, we perform classification on two datasets, to demonstrate the our approach. The two datasets are the WebKB data set[1] and the Cora data set [15]. The WebKB data set consists of about 6000 web pages from computer science departments of four schools (Cornell, Texas, Washington, and Wisconsin). The web pages are classified into seven categories. The numbers of pages in each category are shown in Table 1. The Cora data set consists of the abstracts and references of about 34,000 computer science research papers. We use part of them to categorize into one of subfields of data structure (DS), hardware and architecture (HA), machine learning (ML), and programing language (PL). We remove those articles without reference to other articles in the set. The number of papers and the number of subfields in each area are shown in Table 2.

area	# of papers	# of subfields
Data structure (DS)	751	9
Hardware and architecture (HA)	400	7
Machine learning (ML)	1617	7
Programing language (PL)	1575	9

Table 2: Dataset of Cora

5.2 Methods

The task of the experiments is to classify the data based on their content information and/or link structure. We use the following methods:



method	Cornell	Texas	Washington	Wisconsin
SVM on content	81.00 ± 0.90	77.00 ± 0.60	85.20 ± 0.90	84.30 ± 0.80
SVM on links	70.10 ± 0.80	75.80 ± 1.20	80.50 ± 0.30	74.90 ± 1.00
SVM on link-content	80.00 ± 0.80	78.30 ± 1.00	85.20 ± 0.70	84.00 ± 0.90
Directed graph regularization	89.47 ± 1.41	91.28 ± 0.75	91.08 ± 0.51	89.26 ± 0.45
PLSI+PHITS	80.74 ± 0.88	76.15 ± 1.29	85.12 ± 0.37	83.75 ± 0.87
link-content MF	93.50 ± 0.80	96.20 ± 0.50	93.60 ± 0.50	92.60 ± 0.60
link-content sup. MF	93.80 ± 0.70	97.07 ± 1.11	93.70 ± 0.60	93.00 ± 0.30

Table 3: Classification accuracy (mean $\pm$ std-err %) on WebKB data set

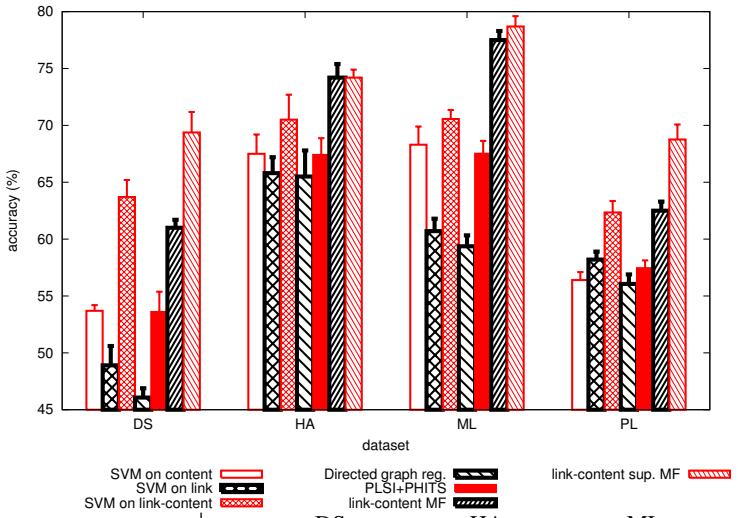
- *SVM on content* We apply support vector machines (SVM) on the content of documents. The features are the bag-of-words and all word are stemmed. This method ignores link structure in the data. Linear SVM is used. The regularization parameter of SVM is selected using the cross-validation method. The implementation of SVM used in the experiments is libSVM[4].
- *SVM on links* We treat links as the features of each document, i.e. the i-th feature is *link-to-page*$_i$. We apply SVM on link features. This method uses link information, but not the link structure.
- *SVM on link-content* We combine the features of the above two methods. We use different weights for these two set of features. The weights are also selected using cross-validation.
- *Directed graph regularization* This method is described in [25] and [24]. This method is solely based on link structure.
- *PLSI+PHITS* This method is described in [6]. This method combines text content information and link structure for analysis. The PHITS algorithm is in spirit similar to Eq.1, with an additional nonnegative constraint. It models the outgoing and in-coming structures separately.
- *Link-content MF* This is our approach of matrix factorization described in Section 3. We use 50 latent factors for Z. After we compute Z, we train a linear SVM using Z as the feature vectors, then apply SVM on testing portion of Z to obtain the final result, because of the multiclass output.
- *Link-content sup. MF* This method is our approach of the supervised matrix factorization in Section 4. We use 50 latent factors for Z. After we compute Z, we train a linear SVM on the training portion of Z, then apply SVM on testing portion of Z to obtain the final result, because of the multiclass output.

We randomly split data into five folds and repeat the experiment for five times, for each time we use one fold for test, four other folds for training. During the training process, we use the cross-validation to select all model parameters. We measure the results by the classification accuracy, i.e., the percentage of the number of correct classified documents in the entire data set. The results are shown as the average classification accuracies and it standard deviation over the five repeats.

5.3 Results

The average classification accuracies for the WebKB data set are shown in Table 3. For this task, the accuracies of SVM on links are worse than that of SVM on content. But the directed graph regularization, which is also based on link alone, achieves a much higher accuracy. This implies that the link structure plays an important role in the classification of this dataset, but individual links in a web page give little information. The combination of link and content using SVM achieves similar accuracy as that of SVM on content alone, which confirms individual links in a web page give little information. Since our approach consider the link structure as well as the content information, our two methods give results a highest accuracies among these approaches. The difference between the results of our two methods is not significant. However in the experiments below, we show the difference between them.

The classification accuracies for the Cora data set are shown in Table 4. In this experiment, the accuracies of SVM on the combination of links and content are higher than either SVM on content or SVM on links. This indicates both content and links are infor-



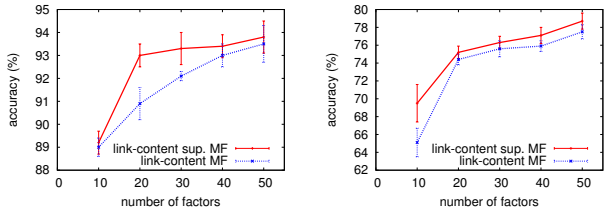
method	DS	HA	ML	PL
SVM on content	53.70 ± 0.50	67.50 ± 1.70	68.30 ± 1.60	56.40 ± 0.70
SVM on links	48.90 ± 1.70	65.80 ± 1.40	60.70 ± 1.10	58.20 ± 0.70
SVM on link-content	63.70 ± 1.50	70.50 ± 2.20	70.56 ± 0.80	62.35 ± 1.00
Directed graph regularization	46.07 ± 0.82	65.50 ± 2.30	59.37 ± 0.96	56.06 ± 0.84
PLSI+PHITS	53.60 ± 1.78	67.40 ± 1.48	67.51 ± 1.13	57.45 ± 0.68
link-content MF	61.00 ± 0.70	74.20 ± 1.20	77.50 ± 0.80	62.50 ± 0.80
link-content sup. MF	69.38 ± 1.80	74.20 ± 0.70	78.70 ± 0.90	68.76 ± 1.32

Table 4: Classification accuracy (mean $\pm$ std-err %) on Cora data set

mative for classifying the articles into subfields. The method of directed graph regularization does not perform as good as SVM on link-content, which confirms the importance of the article content in this task. Though our method of link-content matrix factorization perform slightly better than other methods, our method of link-content supervised matrix factorization outperform significantly.

5.4 The Number of Factors

As we discussed in Section 3, the computational complexity of each iteration for solving the optimization problem is quadratic to the number of factors. We perform experiments to study how the number of factors affects the accuracy of predication. We use different numbers of factors for the Cornell data of WebKB data set and the machine learning (ML) data of Cora data set. The result shown in Figure 4(a) and 4(b). The figures show that the accuracy



(a) Cornell data (b) ML data

Figure 4: Accuracy vs number of factors

increases as the number of factors increases. It is a different concept from choosing the "optimal" number of clusters in clustering application. It is how much information to represent in the latent variables. We have considered the regularization over the factors, which avoids the overfit problem for a large number of factors. To choose of the number of factors, we need to consider *the trade-off between the accuracy and the computation time*, which is quadratic to the number of factors.

The difference between the method of matrix factorization and that of supervised one decreases as the number of factors increases. This indicates that the usefulness of supervised matrix factorization at lower number of factors.

6. DISCUSSIONS

The loss functions $\mathcal{L}_A$ in Eq. (2) and $\mathcal{L}_C$ in Eq. (3) use squared loss due to computationally convenience. Actually, squared loss does not precisely describe the underlying noise model, because the weights of adjacency matrix can only take nonnegative values, in our case, zero or one only, and the components of content matrix C can only take nonnegative integers. Therefore, we can apply other types of loss, such as hinge loss or smoothed hinge loss, e.g. $\mathcal{L}_A(U, Z) = \mu h(A, ZUZ^\top)$, where $h(A, B) = \sum_{i,j} [1 - A_{ij}B_{ij}]_+$.

In our paper, we mainly discuss the application of classification. A entry of matrix Z means the relationship of a web page and a factor. The values of the entries are the weights of linear model, instead of the probabilities of web pages belonging to latent topics. Therefore, we allow the components take any possible real values. When we come to the clustering application, we can use this model to find Z, then apply K-means to partition the web pages into clusters. Actually, we can use the idea of nonnegative matrix factorization for clustering [20] to directly cluster web pages. As the example with nonnegative constraints shown in Section 3, we represent each cluster by a latent topic, i.e. the dimensionality of the latent space is set to the number of clusters we want. Then the

problem of Eq. (4) becomes

$$\min_{U,V,Z} \mathcal{J}(U,V,Z), \qquad \text{s.t.} Z \geq 0. \tag{9}$$

Solving Eq. (9), we can obtain more interpretable results, which could be used for clustering.

7. CONCLUSIONS

In this paper, we study the problem of how to combine the information of content and links for web page analysis, mainly on classification application. We propose a simple approach using factors to model the text content and link structure of web pages/documents. The directed links are generated from the linear combination of linkage of between source and destination factors. By sharing factors between text content and link structure, it is easy to combine both the content information and link structure. Our experiments show our approach is effective for classification. We also discuss an extension for clustering application.

Acknowledgment

We would like to thank Dr. Dengyong Zhou for sharing his code of his algorithm. Also, thanks to the reviewers for constructive comments.

8. REFERENCES

[1] CMU world wide knowledge base (WebKB) project. Available at `http://www.cs.cmu.edu/~WebKB/`.

[2] D. Achlioptas, A. Fiat, A. R. Karlin, and F. McSherry. Web search via hub synthesis. In *IEEE Symposium on Foundations of Computer Science*, pages 500–509, 2001.

[3] S. Chakrabarti, B. E. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. In L. M. Haas and A. Tiwary, editors, *Proceedings of SIGMOD-98, ACM International Conference on Management of Data*, pages 307–318, Seattle, US, 1998. ACM Press, New York, US.

[4] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[5] D. Cohn and H. Chang. Learning to probabilistically identify authoritative documents. *Proc. ICML 2000. pp.167-174.*, 2000.

[6] D. Cohn and T. Hofmann. The missing link - a probabilistic model of document content and hypertext connectivity. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 430–436. MIT Press, 2001.

[7] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273, 1995.

[8] 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.

[9] X. He, H. Zha, C. Ding, and H. Simon. Web document clustering using hyperlink structures. *Computational Statistics and Data Analysis*, 41(1):19–45, 2002.

[10] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the Twenty-Second Annual International SIGIR Conference*, 1999.

[11] T. Joachims, N. Cristianini, and J. Shawe-Taylor. Composite kernels for hypertext categorisation. In C. Brodley and A. Danyluk, editors, *Proceedings of ICML-01, 18th International Conference on Machine Learning*, pages 250–257, Williams College, US, 2001. Morgan Kaufmann Publishers, San Francisco, US.

[12] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 48:604–632, 1999.

[13] P. Kolari, T. Finin, and A. Joshi. SVMs for the Blogosphere: Blog Identification and Splog Detection. In *AAAI Spring Symposium on Computational Approaches to Analysing Weblogs*, March 2006.

[14] O. Kurland and L. Lee. Pagerank without hyperlinks: structural re-ranking using links induced by language models. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 306–313, New York, NY, USA, 2005. ACM Press.

[15] A. McCallum, K. Nigam, J. Rennie, and K. Seymore. Automating the contruction of internet portals with machine learning. *Information Retrieval Journal*, 3(127–163), 2000.

[16] H.-J. Oh, S. H. Myaeng, and M.-H. Lee. A practical hypertext catergorization method using links and incrementally available class information. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 264–271, New York, NY, USA, 2000. ACM Press.

[17] L. Page, S. Brin, R. Motowani, and T. Winograd. PageRank citation ranking: bring order to the web. *Stanford Digital Library working paper 1997-0072*, 1997.

[18] C. Spearman. "General Intelligence," objectively determined and measured. *The American Journal of Psychology*, 15(2):201–292, Apr 1904.

[19] B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *Proceedings of 18th International UAI Conference*, 2002.

[20] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273. ACM Press, 2003.

[21] Y. Yang, S. Slattery, and R. Ghani. A study of approaches to hypertext categorization. *Journal of Intelligent Information Systems*, 18(2-3):219–241, 2002.

[22] K. Yu, S. Yu, and V. Tresp. Multi-label informed latent semantic indexing. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 258–265, New York, NY, USA, 2005. ACM Press.

[23] T. Zhang, A. Popescul, and B. Dom. Linear prediction models with graph regularization for web-page categorization. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 821–826, New York, NY, USA, 2006. ACM Press.

[24] D. Zhou, J. Huang, and B. Schölkopf. Learning from labeled and unlabeled data on a directed graph. In *Proceedings of the 22nd International Conference on Machine Learning*, Bonn, Germany, 2005.

[25] D. Zhou, B. Schölkopf, and T. Hofmann. Semi-supervised learning on directed graphs. *Proc. Neural Info. Processing Systems*, 2004.