

EXPLORING SOCIAL TAGGING GRAPH FOR WEB OBJECT CLASSIFICATION

BY

ZHIJUN YIN

THESIS

Submitted in partial fulfillment of the requirements  
for the degree of Master of Science in Computer Science  
in the Graduate College of the  
University of Illinois at Urbana-Champaign, 2009

Urbana, Illinois

Adviser:

Professor Jiawei Han

# Abstract

We study web object classification problem with the novel exploration of social tags. Automatically classifying web objects into manageable semantic categories has long been a fundamental preprocess for indexing, browsing, searching, and mining these objects. The explosive growth of heterogeneous web objects, especially non-textual objects such as products, pictures, and videos, has made the problem of web classification increasingly challenging. Such objects often suffer from a lack of easy-extractable features with semantic information, interconnections between each other, as well as training examples with category labels.

In our study, we explore the social tagging data to bridge this gap. We cast web object classification problem as an optimization problem on a graph of objects and tags. We then propose an efficient algorithm which not only utilizes social tags as enriched semantic features for the objects, but also infers the categories of unlabeled objects from both homogeneous and heterogeneous labeled objects, through the implicit connection of social tags. Experiment results show that the exploration of social tags effectively boosts web object classification. Our algorithm significantly outperforms the state-of-the-art of general classification methods.

*To my family for all their love.*

# Acknowledgments

First, I would like to thank Professor Jiawei Han for his guidance and insightful comments for my research work. Second, I would like to thank all the faculties and colleagues in DAIS (Data and Information System) research group at the University of Illinois, Urbana-Champaign. Especially, I thank Rui Li and Qiaozhu Mei for their help throughout the writing of the thesis.

# Table of Contents

<b>List of Tables</b> . . . . .	<b>vi</b>
<b>List of Figures</b> . . . . .	<b>vii</b>
<b>Chapter 1 Introduction</b> . . . . .	<b>1</b>
<b>Chapter 2 Related Work</b> . . . . .	<b>5</b>
<b>Chapter 3 Problem Formulation</b> . . . . .	<b>7</b>
3.1 Social Tagging Graph . . . . .	7
3.2 Intuitions . . . . .	8
3.3 The Optimization Framework . . . . .	9
<b>Chapter 4 The Classification Algorithm</b> . . . . .	<b>12</b>
4.1 The Iterative Algorithm . . . . .	12
4.2 Complexity Analysis . . . . .	15
4.3 Discussion . . . . .	15
4.3.1 Parameter Setting . . . . .	15
4.3.2 Connection to Absorption Random Walk . . . . .	16
<b>Chapter 5 Experiment</b> . . . . .	<b>18</b>
5.1 Experiment Setup . . . . .	18
5.2 Experimental Result . . . . .	20
5.2.1 Overall Performance Comparison . . . . .	20
5.2.2 Effectiveness of Tag Feature . . . . .	21
5.2.3 Exploring the Interconnections of Objects . . . . .	21
5.2.4 Handling Lack of Labeling Issue . . . . .	22
5.2.5 Prior Knowledge . . . . .	23
5.2.6 Convergence . . . . .	25
<b>Chapter 6 Conclusions</b> . . . . .	<b>26</b>
<b>References</b> . . . . .	<b>27</b>

# List of Tables

5.1	Category of web pages in ODP and products in Amazon . . . . .	19
5.2	Overall comparison of different classification methods . . . . .	20
5.3	Comparison of title feature and tag feature . . . . .	21
5.4	Comparison of classification results using both homogeneous and heterogeneous objects	22
5.5	Comparison of classification results using prior knowledge . . . . .	24

# List of Figures

1.1	Social tagging in heterogeneous domains . . . . .	2
3.1	Social tagging graph of heterogeneous web objects . . . . .	8
4.1	Iterative algorithm. . . . .	14
5.1	Sensitivity of parameter $\alpha$ . . . . .	23
5.2	Accuracy change in 10 iterations . . . . .	25

# Chapter 1

## Introduction

Beyond textual web pages, various genres of *web objects* become increasingly popular, which makes the Web more vivid than ever before. Indeed, millions of products are being sold on Amazon ([www.amazon.com](http://www.amazon.com)); millions of videos are being uploaded to YouTube ([www.youtube.com](http://www.youtube.com)) every month; millions of research papers are referenced on CiteULike ([www.citeulike.com](http://www.citeulike.com)); and billions of photos are uploaded to and collected by Flickr ([www.flickr.com](http://www.flickr.com)) and Facebook ([www.facebook.com](http://www.facebook.com)). There is an urgent need to efficiently index and organize these web objects, to facilitate convenient browsing and search of the objects, and to effectively reveal interesting patterns from the objects. For all these tasks, classifying the web objects into manipulable semantic categories is an essential preprocessing procedure. Imagine a user who visits Amazon and looks for a DVD of “Harry Potter”, then the fifth book of the “Harry Potter” series, and then a “Harry Potter” costume for a Halloween party, he is able to find the first item in the department (category) of “Movies & TV”, the second item in the department of “Books”, and the third item in the department of “Apparel & Accessories”.

The general problem of text classification is well studied in literature. Classification of web objects, however, is a much more challenging task due to the specific characteristics of the data. The reasons are listed as follows.

1. *Lack of features.* Unlike text documents, web objects cannot be easily represented in a meaningful feature space. The limited text description (e.g., the name of a product on Amazon or the title of a picture on Flickr) is usually too sparse to provide enough semantic features. Content features of images or videos on the other hand, usually cannot be extracted in an accurate and efficient way.
2. *Lack of interconnections.* Web objects often exist in isolate settings, where interconnections



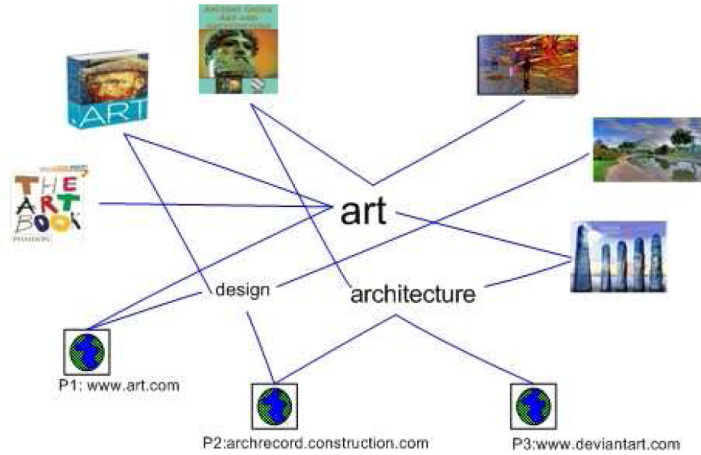


Figure 1.1: Social tagging in heterogeneous domains

between each other are both limited and untrustful. It is quite obvious to a human user that a video of Michael Jordan playing basketball should be put in the same category of a video of Kobe Bryant. This is however not so obvious to an automatic classifier which simply focuses on titles. Instead, it may mistakenly match the basketball video with the Berkeley professor Michael Jordan teaching a class.

3. *Lack of labels.* Good classification performance relies on a reasonable amount of training examples. However, web object classification usually suffers from a lack of training examples. Creating a large training set for certain types of web objects is laborious, sometimes even infeasible. Objects of different types have different feature spaces, and there are barely any interrelations across different object types. Even if there are labeled examples for some type of objects such as web pages, such information can hardly be utilized in classifying other types of objects such as products.

It is therefore desirable to find a treatment of all these deficiencies, which could bridge the gap between the limited text description of web objects and the rich semantic feature space, between the isolated settings of web objects, and between the unlabeled objects in one domain and the labeled examples in another. In our study, we show that a new type of data called *social tags* serves as such an effective bridge, which alleviates all the three problems discussed above.

Along with the burgeoning of the Web 2.0 technology, social tagging has become a common online activity of web users, and has been provided as a critical functionality in various web sites.

As shown in Figure 1.1, heterogeneous objects on the Web are tagged by users, with keywords freely chosen from their own vocabularies. In Flickr, in Amazon, and in Delicious(delicious.com), users have provided rich tags for pictures, products, and arbitrary web pages. These tags are effectively utilized for information sharing and retrieval. Moreover, news articles are tagged in Digg (digg.com); blogs are tagged in Technorati (technorati.com); and questions are tagged in Live search QnA (qna.live.com).

All these tags reflect the semantics of the web objects from users' points of view, using a ubiquitous vocabulary for heterogeneous domains of objects. This property makes social tags an ideal type of data, which overcomes the above difficulties of web object classification. First, a web object is associated with tagged keywords selected by many users, which provide enriched semantic features for web object classification. Back to Figure 1.1, the tags "art", "design", and "architecture" are good features to characterize the semantics of the book "ancient Greek art and architecture". Second, through the intermediate connection of tags, a new link structure of web objects (and tags) is established, which makes it feasible to explore the latent relationships between web objects. For example, in Figure 1.1, although web page  $P_1$  and web page  $P_3$  do not have any tags in common, there is an implicit path from  $P_1$  to  $P_3$  via two tags and  $P_2$ . This provides a way to infer the category of  $P_3$  based on the category information of  $P_1$ . Furthermore, since people are likely to tag different types of objects using the similar vocabularies, heterogeneous types of web objects are now connected through common tags. This enables us to leverage the labeled examples from one domain to infer the category labels of objects in another domain. For example, in Figure 1.1, the book "ancient Greek art and architecture" and "the art book" are linked by the common tag "art". The tag "art" also connects to the web page "www.art.com", and a picture in Flickr. If the web page has a category label *Fine Arts*, the books and the pictures are likely to be labeled with the same category. We can clearly see that social tags act as a bridge, through which the category information can be transferred between different domains.

In our study, we innovatively explore social tags for web object classification. We cast the problem of web object classification as an optimization problem on a heterogeneous graph of web objects and social tags. In such a graph, different types of social objects are linked with their associated tags. We then propose an iterative algorithm which solves the optimization problem

efficiently. In order to evaluate our proposed model, we perform comprehensive experiments on a novel data set which consists of 5536 products from Amazon, and 6123 web pages from ODP. All of them are associated with tags from either Amazon or Delicious. The experiments show that our classification model based on social tagging can solve the above problems in web object classification effectively, which significantly outperforms the state-of-the-art methods without using tags as bridges.

We summarize our contributions as follows:

1. Social tags are explored as a novel evidence to classify objects on the web. A new link structure between objects and tags is explored for classification. Tags also act as bridges to connect the heterogenous domains of objects.
2. Web object classification task is innovatively formalized as an optimization problem on the social tagging graph. An efficient iterative algorithm is proposed to solve the optimization problem.
3. Extensive experiments are conducted to demonstrate the effectiveness of social tags in web object classification.

## Chapter 2

# Related Work

To the best of our knowledge, our study is the first work to explore social tag data for web object classification. Some research works have been done in related areas.

Web object classification has been investigated for a long time, especially web page classification [6, 12] and multimedia classification [9, 11]. However, these studies mainly focus on homogeneous objects, and classification algorithms are designed for specific types of data. For web page classification, other than the classical textual feature based classification model [8], hyperlink [3], html meta data [6], and query log [14] are explored to improve classification result. For multimedia objects, both text feature and contextual information extracted from images or videos are combined to boost the performance. Besides, Xue et al. propose an iterative reinforce categorization algorithm (IRC) [17], which propagates the category information between web pages and their associated web queries based on a set of heuristic rules. Different from the above work, we explore a new resource—social tag for all kinds of web object classification tasks. We propose a general theoretic framework for explicitly modeling tagging behaviors and web object classification problem.

Social tag, a new type of user generated data, can benefit web search [1, 7], information retrieval [19, 13], semantic web [15], web page clustering [2] and user interest mining [10]. Bao et al. [1] propose an iterative algorithm to propagate the popularity within a socially linked graph for ranking web pages. Wu et al. [15] use a probabilistic model to extract semantic concept by exploring the links between tags and pages. However, there is little work on exploring social tagging for general web object classification. Tagging is a kind of “social classification”, and is different from traditional information organization mechanism that is referred as “expert classification”. In our study, we try to provide a uniform model to bridge “social classification” and “classical classification”.

Classification is also an active research topic in machine learning. In order to handle the few labeling problem, semi-supervised learning [20, 18] and transfer learning [4, 5] are proposed. Given

only a small portion of labeled data and a large amount of unlabeled data, unlabeled data can be combined with the labeled data together to improve the accuracy of the classifier. Zhu et al. [20] formulated the learning problem as a Gaussian random field on a weighted graph of labeled and unlabeled objects. Zhou et al. [18] proposed a semi-supervised learning framework to achieve local and global consistency. Both methods model the classification problem on a graph with objects as vertices. We also use a graph to explicitly model the social tagging behavior and extends the graph with heterogeneous objects. Transfer learning is proposed to use the knowledge from other domains, Xue et al. [16] use topics as bridges to transfer information from the source domain to the target domain by extending the traditional topic model. In our study, with social tag acting as a bridge, our model can transfer the category information among any tagged heterogeneous non-textual objects effectively.

## Chapter 3

# Problem Formulation

In this section, we formulate the problem of web object classification as an optimization problem based on a social tagging graph.

### 3.1 Social Tagging Graph

Let us start with the definition of the social tagging graph, an illustration of which is shown in Figure 3.1. There are various types of objects on the Web. To simplify the discussion, we assume there are two types of web objects: type  $T$  and type  $S$ . Objects of type  $T$  are the target objects which we want to assign category labels; whereas objects of type  $S$  are labeled objects from another domain. We assume that some objects of type  $T$  have initial category labels (e.g.,  $T_1, T_2$ ) and some do not (e.g.,  $T_3, T_4, T_5$  and  $T_6$ ), for which we want to infer the labels. Objects of type  $S$  could either have initial labels or not. We simply assume that all of them are labeled. In Figure 3.1,  $S_1, S_2$  and  $T_1$  belong to one category, and  $S_3, S_4$  and  $T_2$  belong to another category. All the objects are associated with some social tags, denoted as  $Tag$ . The goal of web object classification is to assign class label to each unlabeled object of type  $T$  using the existing labeled objects of type  $T$  or type  $S$ , or both. Formally, we introduce the following definitions.

- $C$ : a *category set*,  $\{c_1, c_2, \dots, c_k\}$ .
- $G = \langle V, E \rangle$ : a *social tagging graph*. Every *object*,  $u$ , and every *tag*,  $v$ , is a vertex in the graph  $G$  ( $u, v \in V$ ). If an object  $u$  is associated with a tag  $v$ , there will be an edge between  $u$  and  $v$ , denoted as  $(u, v) \in E$ .  $V$  consists of four types of vertices:
  - $V_S$ : a set of objects of type  $S$ .
  - $V_T^l$ : a set of labeled objects of type  $T$ .

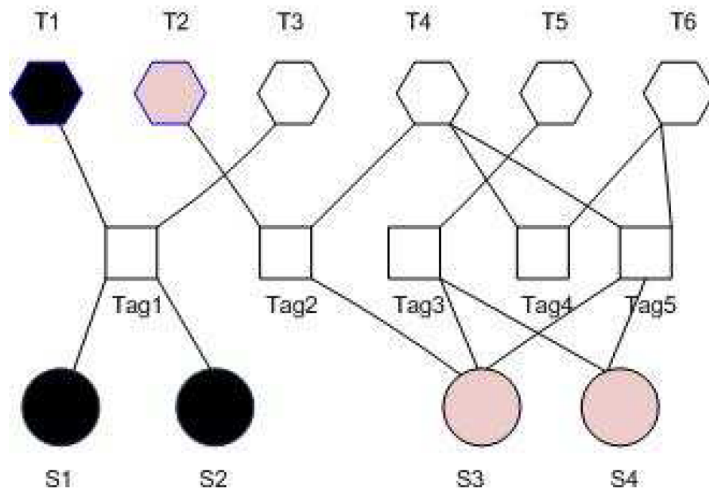


Figure 3.1: Social tagging graph of heterogeneous web objects

- $V_T^u$ : a set of unlabeled objects of type  $T$ .
- $V_{tag}$ : a set of tags.

The problem of web object classification can then be loosely defined as the problem of assigning weights of categories to each vertex in the graph. Formally, the problem is casted to assigning the weights of  $c_1, \dots, c_k$  for every  $u \in V_T^u$ , or  $u \in V$ . In the following sections, we will define it as an optimization problem.

## 3.2 Intuitions

Tags are used as bridges to connect the unlabeled objects, the labeled ones of the same type and the labeled heterogeneous ones. This is based on the following intuitive assumption:

Web users are likely to select similar tags for the objects of the same semantic category, independent of the domain/type of the objects.

This indicates that the category assignments of two objects connected by the same tag are likely to be similar, both similar to the semantic category of the tag itself.

When a reasonable classifier assigns new labels to all the vertices, the assignments should achieve the consistency on the entire social tagging graph. This consistency can be captured by the following four properties:

1. Category assignment of a vertex in  $V_S$  should not deviate much from its original label.
2. Category assignment of the vertex in  $V_T^l$  should remain the same with its original label if it is fully trustable. Even if they are not, they should not deviate too much.
3. Category of the vertex in  $V_T^u$  should take the prior knowledge into consideration if there is any.
4. Category assignment of any vertex in graph  $G$  should be as consistent as possible to the categories of its neighbors.

The category assignments of  $V_T^u$  is used as the classification results of the unlabeled objects of type  $T$ .

### 3.3 The Optimization Framework

With the intuitions discussed in Section 3.2, we now define the classification problem within an optimization framework on the social tagging graph. Formally, let us introduce the following definitions.

- $f_u$ : a  $k$ -dimension vector that represents the class distribution of vertex  $u \in V$ , where  $k$  is the number of categories.  $f_u[i]$  represents the possibility that  $u$  belongs to category  $i$ , s.t.  $\sum_{i=1}^k f_u[i] = 1$ . We denote  $\{f_u\}_{u \in V}$  as  $f$ .
- $f_u^*$ : the optimal solution of  $f_u$
- $\hat{f}_u$ : for  $u \in V_S \cup V_T^l$ ,  $\hat{f}_u$  is the class distribution estimated from the original category labels of vertex  $u$ . For  $u \in V_T^u$ ,  $\hat{f}_u$  is the class distribution estimated from some prior knowledge of the unlabeled object  $u$  (e.g., the label assignments by a domain classifier).
- $w_{uv}$ : a weight of the importance of edge  $(u, v)$ . Given an object  $u$  and its associated tag  $v$ ,  $w_{uv}$  is the frequency that  $v$  is used to tag  $u$ .

We propose the following general optimization framework for web object classification. The objective function is defined as



$$\begin{aligned}
O(f) = & \alpha \sum_{u \in V_S} \|f_u - \hat{f}_u\|^2 \\
& + \beta \sum_{u \in V_T^l} \|f_u - \hat{f}_u\|^2 \\
& + \gamma \sum_{u \in V_T^u} \|f_u - \hat{f}_u\|^2 \\
& + \sum_{(u,v) \in E} w_{uv} \|f_u - f_v\|^2
\end{aligned}$$

The objective function  $O(f)$  has four components, corresponding to the four properties that we want to achieve (see Section 3.2).

1.  $\sum_{u \in V_S} \|f_u - \hat{f}_u\|^2$  means that the category of a vertex in  $V_S$  should not deviate much from its original label(s).
2.  $\sum_{u \in V_T^l} \|f_u - \hat{f}_u\|^2$  means that the category of a vertex in  $V_T^l$  should keep close to its initial label(s).
3.  $\sum_{u \in V_T^u} \|f_u - \hat{f}_u\|^2$  means that the category of a vertex in  $V_T^u$  should keep close to the prior knowledge if any.
4.  $\sum_{(u,v) \in E} w_{uv} \|f_u - f_v\|^2$  makes sure that the class distribution of the vertices are smooth over the whole graph, i.e., the class distribution of a vertex is consistent with its neighbors.

$\alpha, \beta, \gamma$  are three parameters, which control the weights of each constraint. Different settings of these parameters can reflect different beliefs we have for the problem, and refer to different scenarios. We will discuss the parameter setting analytically in Section 4.3.1 and empirically in Section 5.2.

Our target is to find  $f = f^*$  to minimize the cost function.

$$f^* = \arg \min O(f) \tag{3.1}$$

Based on the class distribution, we can assign an object  $o$  to the class  $c$  such that

$$c = \arg \max \frac{P(o|c)}{P(o)} = \arg \max \frac{P(c|o)}{P(c)}$$

Therefore, we can infer the class label of an unlabeled object  $u \in V_T^u$  based on  $f_u^*(u \in V_T^u)$  as follows:

$$c = \arg \max_{1 \leq i \leq k} \frac{f_u^*[i]}{\sum_{u' \in V_T^l \cup V_T^u} f_{u'}^*[i]} \quad (3.2)$$

# Chapter 4

## The Classification Algorithm

Finding the close solution of equation (3.1) requires the computation of the inverse of a matrix with the size of all web objects and tags. In reality, this is usually not feasible due to the complexity of computation. In this section, we alternatively propose an efficient iterative algorithm to solve the optimization problem. We then discuss the setting of parameters in our model w.r.t the different scenarios of web object classification.

### 4.1 The Iterative Algorithm

We propose the following iterative algorithm for the optimization problem. In order to find  $f$  that minimizes the cost function, at each iteration we differentiate  $O(f)$  with regard to  $s \in V_S$ ,  $l \in V_T^l$ ,  $u \in V_T^u$  and  $v \in V_{tag}$ . We then find an update of the variables by setting the differentiated result to zero. Naturally, we have

$$\frac{\partial O}{\partial s} = 2\alpha(f_s - \hat{f}_s) + 2 \sum_{v \in V_{tag}} w_{sv}(f_s - f_v) = 0$$

$$f_s = \frac{\alpha}{\alpha + \sum_{v \in V_{tag}} w_{sv}} \hat{f}_s + \frac{\sum_{v \in V_{tag}} w_{sv} f_v}{\alpha + \sum_{v \in V_{tag}} w_{sv}} \quad (4.1)$$

$$\frac{\partial O}{\partial l} = 2\beta(f_l - \hat{f}_l) + 2 \sum_{v \in V_{tag}} w_{lv}(f_l - f_v) = 0$$

$$f_l = \frac{\beta}{\beta + \sum_{v \in V_{tag}} w_{lv}} \hat{f}_l + \frac{\sum_{v \in V_{tag}} w_{lv} f_v}{\beta + \sum_{v \in V_{tag}} w_{lv}} \quad (4.2)$$

$$\frac{\partial O}{\partial u} = 2\gamma(f_u - \hat{f}_u) + 2 \sum_{v \in V_{tag}} w_{uv}(f_u - f_v) = 0$$

$$f_u = \frac{\gamma}{\gamma + \sum_{v \in V_{tag}} w_{uv}} \hat{f}_u + \frac{\sum_{v \in V_{tag}} w_{uv} f_v}{\gamma + \sum_{v \in V_{tag}} w_{uv}} \quad (4.3)$$

$$\begin{aligned} \frac{\partial O}{\partial v} = & -2 \sum_{s \in V_S} w_{sv}(f_s - f_v) - 2 \sum_{l \in V_T^l} w_{lv}(f_l - f_v) \\ & - 2 \sum_{u \in V_T^u} w_{uv}(f_u - f_v) = 0 \end{aligned}$$

$$f_v = \frac{\sum_{s \in V_S} w_{sv} f_s + \sum_{l \in V_T^l} w_{lv} f_l + \sum_{u \in V_T^u} w_{uv} f_u}{\sum_{s \in V_S} w_{sv} + \sum_{l \in V_T^l} w_{lv} + \sum_{u \in V_T^u} w_{uv}} \quad (4.4)$$

It is easy to show that after each iteration, we still have  $\forall u \in V, \sum_i f_u[i] = 1$ .

Based on the above equations, we design an iterative algorithm to get  $f^*$ , which minimizes the cost function. At each iteration, the class distribution of a vertex is updated from its connected vertices. Therefore, the category information is propagated around the whole graph, not only from the homogeneous labeled objects to unlabeled ones, but also from and to those heterogeneous labeled objects. After the iterations converge, the class distributions of the unlabeled objects are used to generate class labels.

The iterative algorithm is shown in Figure 4.1. At steps 1-5,  $\hat{f}_x$  is initialized for all  $x$  belongs to  $V_S \cup V_T^l \cup V_T^u$  from the labeling and the prior knowledge. The iterations start from step 6. At each iteration, the class distribution is updated from the neighbor vertices. At steps 7-8, the class distributions of objects of type  $S$  are updated from the class distributions of the associated tags, which are based on equation (4.1). At steps 9-10, the class distributions of the labeled objects of type  $T$  are updated from the class distributions of the associated tags, which are based on equation (4.2). At steps 11-12, the class distributions of the unlabeled objects of type  $T$  are updated from the class distributions of the associated tags, which are based on equation (4.3). At steps 13-14,

<b>Algorithm 1. Iterative Algorithm.</b>
<b>Input:</b> category size $k$ , class labels $C(x)$ for $x \in V_S \cup V_T^l \cup V_T^u$ .
<b>Output:</b> class labels $\tilde{C}(x)$ for $x \in V_T^u$ .
<pre> // Initialization 1  foreach <math>x \in V_S \cup V_T^l \cup V_T^u</math> 2    <math>\hat{f}_x[C(x)] \leftarrow 1</math> 3  foreach <math>x \in V</math> 4    foreach <math>i \leftarrow 1</math> to <math>k</math> 5      <math>f_x[i] \leftarrow 1/k</math>  // Iteration 6  Repeat until converged 7  foreach <math>x \in V_S</math> 8    <math>f'_x \leftarrow \frac{\alpha}{\alpha + \sum_{v \in V_{tag}} w_{xv}} \hat{f}_x + \frac{\sum_{v \in V_{tag}} w_{xv} f_v}{\alpha + \sum_{v \in V_{tag}} w_{xv}}</math> 9  foreach <math>x \in V_T^l</math> 10   <math>f'_x \leftarrow \frac{\beta}{\beta + \sum_{v \in V_{tag}} w_{xv}} \hat{f}_x + \frac{\sum_{v \in V_{tag}} w_{xv} f_v}{\beta + \sum_{v \in V_{tag}} w_{xv}}</math> 11 foreach <math>x \in V_T^u</math> 12   <math>f'_x \leftarrow \frac{\gamma}{\gamma + \sum_{v \in V_{tag}} w_{xv}} \hat{f}_x + \frac{\sum_{v \in V_{tag}} w_{xv} f_v}{\gamma + \sum_{v \in V_{tag}} w_{xv}}</math> 13 foreach <math>x \in V_{tag}</math> 14   <math>f'_x \leftarrow \frac{\sum_{s \in V_S} w_{sx} f_s + \sum_{l \in V_T^l} w_{lx} f_l + \sum_{u \in V_T^u} w_{ux} f_u}{\sum_{s \in V_S} w_{sx} + \sum_{l \in V_T^l} w_{lx} + \sum_{u \in V_T^u} w_{ux}}</math> 15 foreach <math>x \in V</math> 16   <math>f_x \leftarrow f'_x</math>  // Get Class Label 17 foreach <math>x \in V_T^u</math> 18   <math>\tilde{C}(x) = \arg \max_{1 \leq i \leq k} \frac{f_x[i]}{\sum_{u \in V_T^l \cup V_T^u} f_u[i]}</math> </pre>

Figure 4.1: Iterative algorithm.

the class distributions of the tags are updated from the class distributions of the connected objects, which are based on equation (4.4). Here tag acts as a bridge of belief propagation. At steps 17-18, the class labels of unlabeled objects of type  $T$  are obtained from the class distributions based on equation (3.2).

## 4.2 Complexity Analysis

Here we analyze the computational complexity for the iterative algorithm. The initialization steps (i.e., steps 1-5) take  $O(k|V|)$  time where  $k$  is the number of the categories. At each iteration (i.e., steps 6-16), all the vertices in the graph are enumerated. When each vertex is considered, the class distribution of the vertex is updated from all the neighbor vertices. Therefore, it costs  $O(k \sum_{v \in V} degree(v))$  at each iteration, which is equal to  $O(2k|E|)$  where  $|E|$  is the number of the edges in the graph. It takes  $O(k|V_T^u|)$  time to get the class labels (i.e., steps 17-18). In total, the algorithm takes  $O(k(|V| + iter|E|))$  time where  $iter$  is the iteration number.

## 4.3 Discussion

The optimization framework we proposed is quite general. In this subsection, we show the connection between our framework and various scenarios of web object classification in reality. We then show that although the proposed optimization framework consists of complex components, it is closely related to an absorption random walk.

### 4.3.1 Parameter Setting

There are three parameters in the objective function:  $\alpha$ ,  $\beta$ , and  $\gamma$ , which represent the weights of different consistency constraints on the graph. Different settings of these parameters map the optimization problem to different scenarios of the classification problem in reality:

- $\alpha = 0, \beta \neq 0, \gamma = 0$ . In this case, labeled examples are only available in the target domain (type  $T$ ). No objects of type  $S$ , nor any prior knowledge about the unlabeled objects are available. The parameter  $\beta$  specifies how much we trust the initial label of  $l$ . Specifically, if  $\beta = +\infty$ , we fully trust the initial label, so  $f_l$  must be equal to  $\hat{f}_l$ . The probability

distributions of  $V_T^u$  are inferred through both the labeled vertices ( $V_T^l$ ) and also the unlabeled vertices ( $V_{tag}$  and  $V_T^u$ ). This reduces the problem to a semi-supervised learning task similar to [20].

- $\alpha \neq 0, \beta = 0, \gamma = 0$ . In this case, no labeled object is available in the target domain  $T$ , but there are labeled objects available from a different domain (type  $S$ ). According to Equation (4.1),  $\frac{\alpha}{\alpha + \sum_{v \in V_{tag}} w_{sv}}$  specifies the relative amount of the information obtained from the initial label of  $s$  where  $s \in V_S$ . The larger  $\alpha$  is, the more we trust the initial label of  $s$ . The category information is then propagated from objects of type  $S$  to objects of type  $T$ , through the tags  $V_{tag}$  as a transfer channel. This can be considered as a transfer learning task.
- $\gamma \neq 0$ . In this case, there is prior knowledge available for  $V_T^u$ . The prior class probability distribution can be obtained from domain experts or from the categorization results of other classification methods. In this way, our classification framework can be flexibly combined with any other classification technique, and can be easily adapted to user interactions. Here  $\gamma$  means how much we should rely on the prior knowledge. The larger the  $\gamma$  is, the more we believe in the prior knowledge.
- $\alpha \neq 0, \beta \neq 0, \gamma \neq 0$ . In this case, we have labeled examples from both the target domain (type  $T$ ) and another domain (type  $S$ ). In addition, we also have prior knowledge about the unlabeled objects of type  $T$  (i.e.,  $V_T^u$ ). This can be considered as an integrative scenario of transfer learning, semi-supervised learning, as well as prior integration.

### 4.3.2 Connection to Absorption Random Walk

The optimization framework has quite a few components. However, despite its complex appearance, it is interesting to show that the iterative algorithm we proposed is closely related to an absorption random walk process. To demonstrate this, we construct a graph  $G'$  as follows. For every vertex  $x$  in  $V$  of  $G$ , we make a corresponding vertex  $x'$  in  $G'$  and denote the set of these vertices as  $V'$ . Then for every vertex  $y$  in  $V_S, V_T^l$  and  $V_T^u$ , we make an additional copy of vertex  $\hat{y}'$  in  $G'$  and denote the set of these new vertices as  $\hat{V}'$ . We then define the transition probabilities in  $G'$  as follows.

For  $x' \in V'$  and  $\hat{y}' \in \hat{V}'$ ,

$$p(x', \hat{y}') = \begin{cases} \frac{\alpha}{\alpha + \sum_{v \in V_{tag}} w_{xv}} & \text{if } x \in V_S \text{ and } y \in V_S; \\ \frac{\beta}{\beta + \sum_{v \in V_{tag}} w_{xv}} & \text{if } x \in V_T^l \text{ and } y \in V_T^l; \\ \frac{\gamma}{\gamma + \sum_{v \in V_{tag}} w_{xv}} & \text{if } x \in V_T^u \text{ and } y \in V_T^u; \\ 0 & \text{otherwise.} \end{cases}$$

For  $\hat{y}' \in \hat{V}'$  and  $x' \in V'$ ,  $p(\hat{y}', x') = 0$

For  $x' \in V'$  and  $y' \in V'$ ,

$$p(x', y') = \begin{cases} \frac{w_{xy}}{\alpha + \sum_{v \in V_{tag}} w_{xv}} & \text{if } x \in V_S \text{ and } y \in V_{tag}; \\ \frac{w_{xy}}{\beta + \sum_{v \in V_{tag}} w_{xv}} & \text{if } x \in V_T^l \text{ and } y \in V_{tag}; \\ \frac{w_{xy}}{\gamma + \sum_{v \in V_{tag}} w_{xv}} & \text{if } x \in V_T^u \text{ and } y \in V_{tag}; \\ \frac{w_{yx}}{\sum_{s \in V_S} w_{sx} + \sum_{l \in V_T^l} w_{lx} + \sum_{u \in V_T^u} w_{ux}} & \text{if } x \in V_{tag} \text{ and } y \in V_S \cup V_T^l \cup V_T^u; \\ 0 & \text{otherwise.} \end{cases}$$

We then label all the vertices in  $\hat{V}'$  with the original labels of these objects in  $G$ . Let us imagine that there is a particle randomly walking through  $G'$  starting from vertex  $x$  that belongs to  $V'$ , w.r.t. the defined transition probabilities. These labeled vertices in  $G$  are considered as the absorbing boundary for this random walk. It is easy to show that  $f_x^*[i]$  in  $G$  is essentially the probability that the particle starts from  $x'$  and hits (be absorbed by) the set of vertices with label  $i$  in  $G'$ .

It is easy to show that with such a random walk, the absorption probability of each vertex converges. This naturally proves the convergence of our proposed iterative algorithm.



# Chapter 5

## Experiment

In this section, we conduct extensive experiments based on classification of Amazon products to demonstrate the effectiveness of our algorithm on real world data.

### 5.1 Experiment Setup

**Data Collection:** In order to conduct web product classification task, we collect 6123 products information in 8 different categories from Amazon site by using Amazon API<sup>1</sup>. Their associated tags, category information, and product title are also collected. Further, we collect web pages as external resource for helping classification of products, since there are a large number of labeled web pages and many classification methods with good performance. We collect web pages from the corresponding 8 categories under ODP Shopping category<sup>2</sup>. And tags of these web pages are collected from Delicious<sup>3</sup> and web pages whose tags are less than 5 are removed. This results in 5536 web pages after preprocessing. Table 5.1 shows our experimental data distribution.

**Evaluation Method:** We use the standard  $F1$  measure for evaluating the effectiveness of classification results.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (5.1)$$

To evaluate the average performance across multiple categories, the micro-averaging  $F1$  and macro-averaging  $F1$  are introduced. The micro-averaged scores tend to be dominated by the performance on common categories, and the macro-averaged scores are influenced by the performance in rare categories.

---

<sup>1</sup><http://developer.amazonwebservices.com/connect/entry.jspa?externalID=880>

<sup>2</sup><http://www.dmoz.org/Shopping/>

<sup>3</sup><http://delicious.com/>

ODP:Shopping		Amazon	
Name	Count	Name	Count
Publications/Books	558	Books	937
Consumer_Electronics	494	Electronics	945
Health	1009	HealthPersonCare	747
Home_and_Garden	1976	HomeGarden	841
Jewelry	452	Jewelry	386
Music	527	Music	944
Office	77	OfficeProducts	695
Pet	443	PetSupplies	628

Table 5.1: Category of web pages in ODP and products in Amazon

**Baseline Method:** In order to demonstrate the effectiveness of our model, we compare our method with two state-of-the-art classification methods:

Support Vector Machine (**SVM**) is a well-known supervised classification, which has been applied in many applications. Here we use the libsvm toolkit<sup>4</sup> and linear kernel function.

Harmonic Gaussian field method (**HG**) is a widely-used semi-supervised learning technique [20]. In this approach, the semi-supervised learning is based on a Gaussian random field model from a weighted graph, where labeled and unlabeled data are represented as vertices and the similarity between instances as the edge weight.

In order to evaluate the effectiveness of tags, we also compare tags with other features. In our experiments, we use the product title as another feature space, so products can also be represented as bags of words in the titles. We compare our method with four different baseline methods, which are listed as follows:

- SVM+TITLE: SVM using product titles as feature.
- SVM+TAG: SVM using tags as feature.
- HG+TITLE: Harmonic Gaussian field method using titles. We use cosine similarity of the titles of two products as the edge weight in the graph.
- HG+TAG: Harmonic Gaussian field method using tags. We use cosine similarity of the tags of two products as the edge weight in the graph.

---

<sup>4</sup><http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

Label Ratio	1%		5%	
	MicroF1	MacroF1	MicroF1	MacroF1
SVM+TITLE	0.4233	0.3812	0.5967	0.6091
SVM+TAG	0.4045	0.4059	0.6397	0.6435
HG+TITLE	0.6251	0.6038	0.6778	0.6689
HG+TAG	0.7174	0.7127	0.7856	0.7859
TM <sup>5</sup>	<b>0.7870</b>	<b>0.7872</b>	<b>0.8027</b>	<b>0.8030</b>

Table 5.2: Overall comparison of different classification methods

In the rest part, we use TM (Tag-based classification Model) to refer to our method proposed in the Section 4.

## 5.2 Experimental Result

In the experiments, we demonstrate that our classification model based on social tagging performs better than other classification methods significantly. Specially, our model can solve those problems that we encounter in web object classification. We show the following. First, social tags provides an ideal feature space for web object classification compared with other features. Second, our model makes good use of the link structure of objects and tags, which effectively capture the interconnections of objects through the social tags. Third, our method is effective when there is few or no label available for web objects. Besides, prior knowledge of unlabeled objects can be incorporated, so our model can be combined with other classification methods seamlessly. Furthermore, we show that our algorithm is efficient and can converge after several rounds of iterations in different scenarios.

### 5.2.1 Overall Performance Comparison

In order to compare different classification methods, we label 1% and 5% of the products separately and consider as the training set. The results of each method are shown in Table 5.2. From the table, for each classification model, using tags as the feature space achieves much better result than using titles. When only 1% of products are labeled, SVM performs poorly both on title and on tag. The reason is that there lacks of labels, which is one of the problems that we want to solve. Among all the results, our method performs the best.

---

<sup>5</sup> $\alpha = 1000, \beta = \infty, \gamma = 0.1$

$p\%$	SVM+TITLE		SVM+TAG		HG+TITLE		HG+TAG		TM <sup>6</sup>	
	MicroF1	MacroF1	MicroF1	MacroF1	MicroF1	MacroF1	MicroF1	MacroF1	MicroF1	MacroF1
5%	0.5967	0.6091	0.6397	0.6435	0.6778	0.6689	0.7856	0.7859	<b>0.7918</b>	<b>0.7919</b>
10%	0.6700	0.6789	0.7168	0.7334	0.6937	0.6802	0.7915	0.7864	<b>0.8005</b>	<b>0.7996</b>
15%	0.7181	0.7218	0.7417	0.7366	0.7139	0.7049	0.7921	0.7908	<b>0.8187</b>	<b>0.8199</b>
20%	0.7343	0.7399	0.7674	0.7722	0.7152	0.7059	0.8025	0.8004	<b>0.8217</b>	<b>0.8231</b>
25%	0.7545	0.7597	0.7763	0.7780	0.7131	0.7038	0.8109	0.8079	<b>0.8259</b>	<b>0.8273</b>

Table 5.3: Comparison of title feature and tag feature

### 5.2.2 Effectiveness of Tag Feature

This experiment is conducted to verify our assumption that tags are good features to represent web objects. We compare the results of the same classification method with different feature spaces. The experiment results are shown in Table 5.3.  $p$  represents the percentage of data used as training data. For example, 5% means 5 percent of labeled data is used as training data, and 95% is used as testing data. In Table 5.3, we find that, for both supervised learning method (SVM) and semi-supervised learning method (HG), using tags as the feature space performs better than using titles as the features space for all the  $p$ 's that we choose. It is no surprise that the improvement of tag over title in semi-supervised learning setting is even larger than in supervised learning setting, since HG+TAG makes use of labeled objects as well as unlabeled objects in classification. Social tags are good features to capture the relationship between objects. Therefore, from the experiment result, we can safely draw the conclusion that tags are meaningful features for web object classification task.

### 5.2.3 Exploring the Interconnections of Objects

Without tags, the interconnections among the objects are difficult to model. We want to show that our model can effectively explore the interconnections of objects and improve the classification results consequently. In order to evaluate the performance without the inference of objects of type  $S$  (i.e., web page), we set  $V_S$  to be empty and  $\alpha = 0$ .  $\gamma$  is set to be 0, since we do not consider the prior knowledge here. We fully trust the initial labeling and set  $\beta$  to be  $\infty$ . From the results in Table 5.3, we find that our model performs significantly better than HG+TITLE for all the  $p$ 's that we choose. Compared with title, tags connect the objects in the same category tightly. Therefore, with tag as a bridge, the interconnections of products are explored and the classification result is improved. Compared with HG+TAG, our model performs better as well. In HG+TAG,

	HG+TITLE		HG+TAG		$\alpha = 1000$	
$p\%$	MicroF1	MacroF1	MicroF1	MacroF1	MicroF1	MacroF1
0	NA	NA	NA	NA	0.7594	0.7606
1%	0.6251	0.6038	0.7174	0.7127	<b>0.7708</b>	<b>0.7719</b>
2%	0.6499	0.6334	0.7510	0.7434	<b>0.7771</b>	<b>0.7766</b>
3%	0.6368	0.6368	0.7695	0.7666	<b>0.7774</b>	<b>0.7769</b>
4%	0.6503	0.6360	0.7566	0.7513	<b>0.7885</b>	<b>0.7891</b>
5%	0.6778	0.6689	0.7856	0.7859	<b>0.7872</b>	<b>0.7866</b>

Table 5.4: Comparison of classification results using both homogeneous and heterogeneous objects tags are only used as feature space to calculate the similarity between two objects, and then belief is propagated in the weighted graph of objects. In our model, tags appear explicitly as the vertices in the graph, which act as bridges to connect the labeled objects and unlabeled objects. In this way, our classification method models the social tagging behavior more explicitly and explores the interconnections of objects better than HG+TAG.

#### 5.2.4 Handling Lack of Labeling Issue

For most web objects, such as products and images, there is few labeling data for training a reasonable good classifier. Furthermore, the labeling process for classification is a time-consuming task, and it is too difficult to follow the expansion pace of the Web. To consider such a case, we conduct the following experiments to verify that tags enable us to leverage the labeled examples from one domain to the other.

In the experiment, we set  $p$ , the percentage of data used as training data, from 0 to 5, which means there is no or few labeled data available.  $\beta$  is set to  $\infty$ , which means we fully trust the existing labels.  $\alpha$  is set to 1000, which reflects the degree of trust of information from another domain(web page). The results of different methods are shown in Table 5.4.

From Table 5.4, we find if there is no labeling ( $p = 0$ ), other baseline methods cannot be applied. It is because there is no connection between different types of objects. However, our model can achieve an impressive classification result (i.e., MicroF1 = 0.7594 and MacroF1 = 0.7606) by transferring the knowledge from web page to product via social tags. When there is a small portion of labeled products, our model using knowledge of heterogeneous objects (i.e., web page) performs better than HG+TITLE and HG+TAG methods for all the  $p$ 's that we choose here. It illustrates the idea that category information from heterogeneous objects can also help classification through

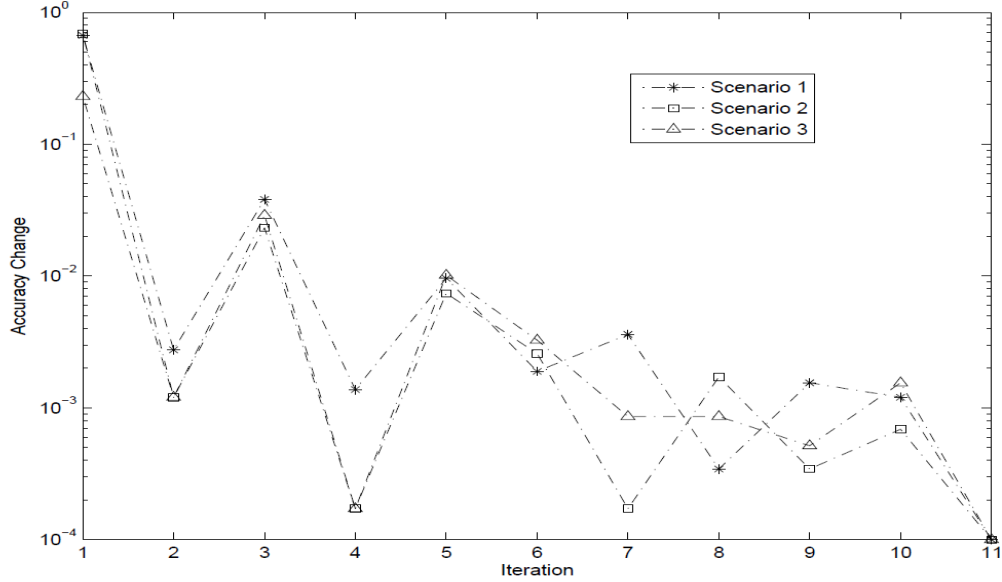


Figure 5.1: Sensitivity of parameter  $\alpha$

tags when there is not sufficient training data. When  $p$  increases to 5%, we find that the result of our model relying on external data is similar to the result of HG+TAG and the case when  $p$  is equal to 4%. The reason may be that the knowledge from heterogeneous objects are not reliable compared with the labeled objects of the same type. The results suggest that if there are no or few labels (let us say less than 5% in this specific case), the labeling of another type is a valuable source, and tag acts as a bridge to transform knowledge efficiently. When there are enough labels, we should rely on the labeling of the same type and use our model to explore the link structure among homogeneous labeled objects and unlabeled objects. If the two domains are similar, we could trust even more on the knowledge from the other domain.

In Figure 5.1, we find that the classification performance is not too much sensitive to the setting of  $\alpha$ , where  $\gamma = 0$  and no labels of type  $T$  is provided. We can set  $\alpha = 1000$  empirically.

### 5.2.5 Prior Knowledge

In our model we can take prior knowledge into consideration. If we use other classifiers to classify the unlabeled objects, we can take the classification result as the prior for our model.  $\gamma$  represent how much we trust the prior knowledge. The larger  $\gamma$  is, the more we trust the prior knowledge. We

---

<sup>6</sup> $\alpha = 0, \beta = \infty, \gamma = 0$

$p\%$	Measure	SVM+TAG	$(\gamma=0.001)+$ (SVM+TAG)	$(\gamma=0.01)+$ (SVM+TAG)	$(\gamma=0.1)+$ (SVM+TAG)	$(\gamma=1)+$ (SVM+TAG)	$\gamma=0$
5%	MicroF1	0.6397	0.7938	<b>0.7964</b>	0.7796	0.6878	0.7918
	MacroF1	0.6435	0.7914	<b>0.7932</b>	0.7673	0.6846	0.7919
10%	MicroF1	0.7168	0.8000	0.8013	<b>0.8096</b>	0.7704	0.8005
	MacroF1	0.7334	0.7987	0.8005	<b>0.8109</b>	0.7803	0.7996
15%	MicroF1	0.7417	0.8214	0.8199	<b>0.8251</b>	0.7913	0.8187
	MacroF1	0.7366	0.8198	0.8184	<b>0.8201</b>	0.7843	0.8199
20%	MicroF1	0.7674	0.8229	0.8223	<b>0.8272</b>	0.8033	0.8217
	MacroF1	0.7722	0.8238	0.8231	<b>0.8277</b>	0.8051	0.8231
25%	MicroF1	0.7763	0.8281	0.8292	<b>0.8355</b>	0.8165	0.8259
	MacroF1	0.7780	0.8295	0.8306	<b>0.8364</b>	0.8163	0.8273
$p\%$	Measure	SVM+TAG	$(\gamma=0.001)+$ (HG+TAG)	$(\gamma=0.01)+$ (HG+TAG)	$(\gamma=0.1)+$ (HG+TAG)	$(\gamma=1)+$ (HG+TAG)	-
5%	MicroF1	0.7856	0.7968	0.8012	<b>0.8038</b>	0.7950	-
	MacroF1	0.7859	0.7973	0.8028	<b>0.8043</b>	0.7951	-
10%	MicroF1	0.7915	0.8038	0.8056	<b>0.8174</b>	0.8036	-
	MacroF1	0.7864	0.8026	0.8040	<b>0.8151</b>	0.7982	-
15%	MicroF1	0.7921	0.8214	0.8222	<b>0.8233</b>	0.8082	-
	MacroF1	0.7908	0.8228	0.8233	<b>0.8238</b>	0.8065	-
20%	MicroF1	0.8025	0.8251	0.8249	<b>0.8296</b>	0.8206	-
	MacroF1	0.8004	0.8263	0.8261	<b>0.8301</b>	0.8192	-
25%	MicroF1	0.8109	0.8300	0.8313	<b>0.8381</b>	0.8339	-
	MacroF1	0.8079	0.8316	0.8329	<b>0.8387</b>	0.8308	-

Table 5.5: Comparison of classification results using prior knowledge

label  $p$  percent of products, and use SVM+TAG and HG+TAG to classify the remaining products. The classification results are considered as the prior for our model. We set  $\alpha$  to be 0 and  $\beta$  to be  $\infty$ . We vary  $p$  from 5 to 25, and test  $\gamma$  among  $\{0.001, 0.01, 0.1, 1\}$  separately. The result in Table 5.5 shows that the performance of our model using prior knowledge outperforms the one without prior (i.e.,  $\gamma = 0$ ). Compared with SVM+TAG and HG+TAG, our model using their classification results as prior performs better. The result with HG+TAG as prior is better than the one with SVM+TAG as prior as expected, since HG+TAG is better than SVM+TAG.  $\gamma = 0.1$  performs the best in most of the cases. When  $p = 5\%$ ,  $(\gamma=0.01)+(\text{SVM+TAG})$  performs better than  $(\gamma=0.1)+(\text{SVM+TAG})$ . The reason is that the result of SVM+TAG is not good when  $p = 5\%$  so it cannot be considered as a good prior, which leads to a smaller  $\gamma$  setting.

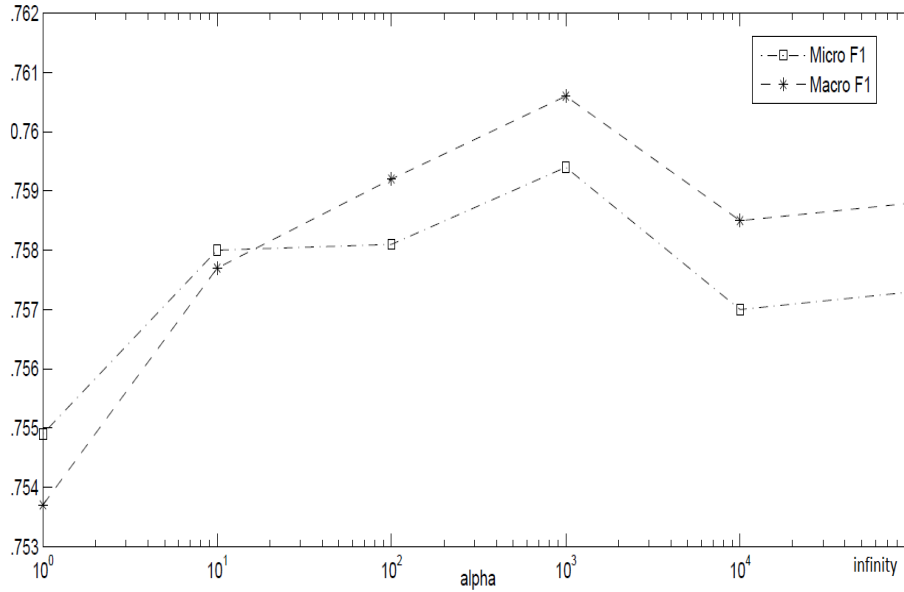


Figure 5.2: Accuracy change in 10 iterations

### 5.2.6 Convergence

Figure 5.2 shows the convergence rounds in three scenarios of classification. In scenario 1, no knowledge from web page is considered and we label 5% percent of products.  $\alpha = 0$ ,  $\beta = \infty$  and  $\gamma = 0$ . In scenario 2, we take the labels of web pages into consideration and label 5% percent of products.  $\alpha = 1000$ ,  $\beta = \infty$  and  $\gamma = 0$ . In scenario 3, we consider prior knowledge for unlabeled products, where  $\alpha = 0$ ,  $\beta = \infty$  and  $\gamma = 0.1$ . From Figure 5.2, we find that our algorithm converges quickly at about 10 rounds of iterations.



## Chapter 6

# Conclusions

Web object classification is an emerging task and becomes increasingly important as web objects, such as products, videos, and images, are bursting at a surprising speed. In our study, we explore social tagging data to deal with this problem. We find that web object classification problem can take advantage from social tags in three aspects: (1) representing web objects in a meaningful feature space, (2) interconnecting web objects to indicate implicit relationship, and (3) bridging heterogeneous objects so that category information can be propagated from one domain to another. To fully explore social tagging data for web object classification, we propose a general framework to model the problem as an optimization problem on a heterogeneous graph of web objects and social tags. The model covers different scenarios of web object classification problem. We design an efficient algorithm to solve the problem. Furthermore, we conduct extensive experiments on real world data with different scenarios, and the results demonstrate that social tag is an effective feature in web object classification and our framework models social tagging structure appropriately and outperforms the state-of-the-art of general classification methods significantly.

The proposed classification model opens up some interesting directions for future research. For example, how to consider multi-types of objects together meaningfully in an uniform setting. In our model, we only consider the setting of two types of web objects. It is interesting to generalize our model to manage multi-types of objects.

# References

- [1] Shenghua Bao, Gui-Rong Xue, Xiaoyuan Wu, Yong Yu, Ben Fei, and Zhong Su. Optimizing web search using social annotations. In *WWW*, pages 501–510, 2007.
- [2] Christopher H. Brooks and Nancy Montanez. Improved annotation of the blogosphere via autotagging and hierarchical clustering. In *WWW*, pages 625–632, 2006.
- [3] Soumen Chakrabarti, Byron Dom, and Piotr Indyk. Enhanced hypertext categorization using hyperlinks. In *SIGMOD Conference*, pages 307–318, 1998.
- [4] Wenyuan Dai, Gui-Rong Xue, Qiang Yang, and Yong Yu. Transferring naive bayes classifiers for text classification. In *AAAI*, pages 540–545, 2007.
- [5] Chuong Do and Andrew Y. Ng. Transfer learning for text classification. In *NIPS*, 2005.
- [6] Rayid Ghani, Seán Slattery, and Yiming Yang. Hypertext categorization using hyperlink patterns and meta data. In *ICML*, pages 178–185, 2001.
- [7] Paul Heymann, Georgia Koutrika, and Hector Garcia-Molina. Can social bookmarking improve web search? In *WSDM*, pages 195–206, 2008.
- [8] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *ECML*, pages 137–142, 1998.
- [9] Pedro R. Kalva, Fabrício Enembreck, and Alessandro L. Koerich. Web image classification based on the fusion of image and text classifiers. In *ICDAR*, pages 561–568, 2007.
- [10] Xin Li, Lei Guo, and Yihong Eric Zhao. Tag-based social interest discovery. In *WWW*, pages 675–684, 2008.
- [11] Wei-Hao Lin and Alexander G. Hauptmann. News video classification using svm-based multimodal classifiers and combination strategies. In *ACM Multimedia*, pages 323–326, 2002.
- [12] Xiaoguang Qi and Brian D. Davison. Web page classification: Features and algorithms. *ACM Comput. Surv.*, 41(2), 2009.
- [13] Ralf Schenkel, Tom Crecelius, Mouna Kacimi, Sebastian Michel, Thomas Neumann, Josiane Xavier Parreira, and Gerhard Weikum. Efficient top-k querying over social-tagging networks. In *SIGIR*, pages 523–530, 2008.
- [14] Dou Shen, Jian-Tao Sun, Qiang Yang, and Zheng Chen. A comparison of implicit and explicit links for web page classification. In *WWW*, pages 643–650, 2006.

- [15] Xian Wu, Lei Zhang, and Yong Yu. Exploring social annotations for the semantic web. In *WWW*, pages 417–426, 2006.
- [16] Gui-Rong Xue, Wenyuan Dai, Qiang Yang, and Yong Yu. Topic-bridged plsa for cross-domain text classification. In *SIGIR*, pages 627–634, 2008.
- [17] Gui-Rong Xue, Dou Shen, Qiang Yang, Hua-Jun Zeng, Zheng Chen, Yong Yu, Wensi Xi, and Wei-Ying Ma. IRC: An iterative reinforcement categorization algorithm for interrelated web objects. In *ICDM*, pages 273–280, 2004.
- [18] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *NIPS*, 2003.
- [19] Ding Zhou, Jiang Bian, Shuyi Zheng, Hongyuan Zha, and C. Lee Giles. Exploring social annotations for information retrieval. In *WWW*, pages 715–724, 2008.
- [20] Xiaojin Zhu, Zoubin Ghahramani, and John D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, pages 912–919, 2003.