World Wide Web (2012) 15:139–170
DOI 10.1007/s11280-011-0126-4

Graffiti: graph-based classification in heterogeneous networks

Ralitsa Angelova · Gjergji Kasneci · Gerhard Weikum

Received: 30 August 2010 / Revised: 25 January 2011 /
Accepted: 8 March 2011 / Published online: 30 March 2011

Abstract We address the problem of multi-label classification in heterogeneous graphs, where nodes belong to different types and different types have different sets of classification labels. We present a novel approach that aims to classify nodes based on their neighborhoods. We model the mutual influence of nodes as a random walk in which the random surfer aims at distributing class labels to nodes while walking through the graph. When viewing class labels as "colors", the random surfer is essentially spraying different node types with different color palettes; hence the name Graffiti of our method. In contrast to previous work on topic-based random surfer models, our approach captures and exploits the mutual influence of nodes of the same type based on their connections to nodes of other types. We show important properties of our algorithm such as convergence and scalability. We also confirm the practical viability of Graffiti by an experimental study on subsets of the popular social networks *Flickr* and *LibraryThing*. We demonstrate the superiority of our approach by comparing it to three other state-of-the-art techniques for graph-based classification.

Keywords graph-based classification · social networks · heterogeneous networks

R. Angelova (✉)
Google, Zurich, Switzerland
e-mail: angelova@google.com

G. Kasneci
Microsoft Research, Cambridge, UK
e-mail: gjergjik@microsoft.com

G. Weikum
Max-Planck Institute for Informatics, Saarbrücken, Germany
e-mail: weikum@mpi-inf.mpg.de

1 Introduction

Heterogeneous graphs are developing faster than ever. In the context of rapidly growing social networks (e.g. *Flickr, Deli.cio.us, LibraryThing, LinkedIn*), heterogeneous graphs are formed by encoding users, their postings like photos, bookmarks, book descriptions, ratings, etc., and other contextual information as graph nodes, belonging to different node types but co-existing and mutually influencing each other in the graph [18–20, 36, 47, 50]. Heterogeneous graphs are formed in many other areas like medical domains (containing information about patients, treatments, diseases, contacts) or e-commerce platforms (representing the complex interactions between different types of nodes like users, products, customers, rating, reviews, etc.). This, by far not exhaustive list of forms of heterogeneous networks, poses a hard classification problem [18] where objects need to be labeled with one or more classes from a *type-specific finite set* of classes.

Consider the social network *Flickr* as an example. It can be modeled as a graph with nodes belonging to one of three different types: users, photos, or tags. Users post and annotate photos. This is captured by links between the corresponding users and photos, i.e. between different node types. We refer to them as cross-type links and denote them as X-edges. However, users are also connected among themselves by explicit friendship links. Thus, links formed between nodes of the same type also exist in the heterogeneous graph and we refer to them as same-type edges or S-edges. A classification procedure over the set of users could output their primary interests: Sport, Music, etc., while classification over the set of photos could reveal classes of photos like Black and White Photography, Urban Photography, Nature Photography, etc. Accordingly, a classification over the set of tags might output tag topics such as Happiness or Nature.

We argue that the labelings of different node types mutually influence each other since heterogeneous nodes are connected primarily by X-edges. Let us illustrate this influence with a small toy example. If you consider the social network *Flickr*, a user is more likely to belong to the class *Outdoor activity fans* if she has frequently used tags belonging to the class *Mountain* and/or she has viewed, commented and posted photos classified as *Nature Photography*. Furthermore, the likelihood that she is an *Outdoor activity fan* is even higher if many *outdoor activity fans* exhibit similar tagging and photo-viewing behavior as her. Our method prioritizes these mutual influences, as opposed to solely relying on the node's features and direct neighborhood as a source of information driving the classification process.

The social web and the semantic web complement each other in the way they approach content generation and organization. Combining the data flexibility and portability of the semantic web, and the scalability and authorship advantages of the social web is focus of many initiatives [10], most notably the Linked Open Data initiative (LOD). Their ultimate goal is to facilitate the convergence of the semantic web and the social web to the next generation Web—Web 3.0, where humans and machines can better understand and query information. Our approach, coined Graffiti, can generate crucial semantic markup related to any of the different types of nodes in the heterogeneous graph, forming a network of interlinked and semantically-rich content and knowledge.

1.1 Problem statement

A classifier is given a set of m data objects and has to map each of them to one or more of n different classes (labels) so that the resulting mapping is consistent with some prior statistical observations about the data. These observations are based on a decision model, constructed by the classifier from a set of training data labeled a priori by a human expert. In the following, we use the terms "classes", "topics", and "class labels" interchangeably; all referring to the associated mapping of an object to a class. As we will later introduce a random surfer model where the surfer sprays nodes of different types with different color palettes, we also use the term "color" as a synonym for "class".

We consider a heterogeneous directed graph $G = (V, E)$ with nodes V and edges E where V comprises nodes of t different types. We assume that all X-edges (between nodes of different types) are bidirectional; for example, each instance of the user-tag relation or tag-photo relation can be viewed as two different edges between the same node pair. This is a natural assumption as all these relations have naturally interpretable inverse relations. Note, however, that this does not necessarily hold for the S-edges; social acquaintance relations may well be asymmetric. An example of this model is illustrated in Figure 1.

The problem now is to label each node of the graph with one or more of its type-specific possible classes, such that we are largely consistent with empirical training data and can predict the class(es) of a node with unknown label with high accuracy. To this end, we propose a novel classification approach. The algorithm assigns initial class labels to all nodes, based on their content, and adjusts these assignments accounting for the existing graph connections: overwhelmingly many X-edges and too few S-edges. The problem is two-fold. Firstly, the small number of S-edges is insufficient to approach the problem by simply applying homogeneous graph classification methods. Secondly, the large number of X-edges entails complex heterogeneous graph analysis.

An example output of the proposed classification algorithm is depicted in Figure 2. A node can belong to one or more classes of their type-specific finite set of classes. In the toy example here, the first type is associated with the classes Red and Yellow; the second one with the classes Blue, Green, and Lavender; and the third one with the classes Pink and Yellow.

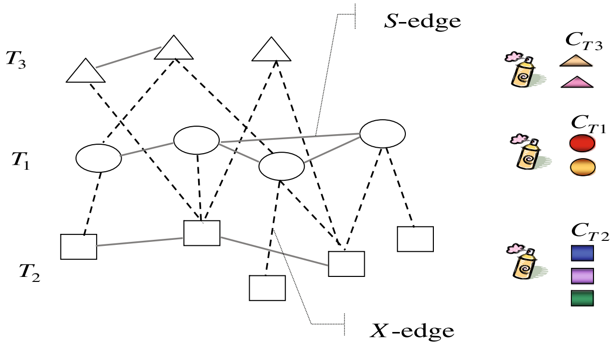
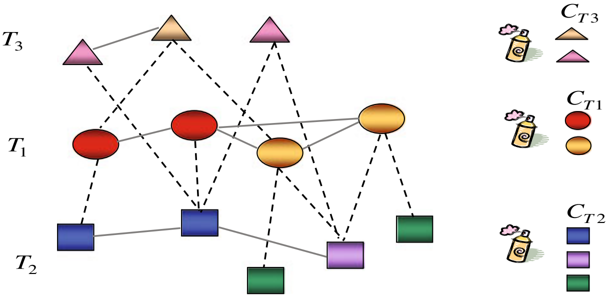


Figure 1 *Triangle*, *circle*, and *square nodes* represent users, photos, and tags, simulating the social network *Flickr*. As in real life, the number of cross-type (X-)edges is orders of magnitude higher than the number of same-type (S-)edges.

Figure 2 Graph labeled with type-specific classes.



This kind of graph-oriented classification problem has been intensively studied for *homogeneous* environments. Most notably, the large family of graphical models, such as Markov Random Fields (MRFs), is well suited to address such supervised learning tasks [20]. In *heterogeneous* graphs, however, additional complications arise: training data tends to be much sparser than in homogeneous settings (assuming the same amount of human labeling effort), and the inferencing procedures tend to be more expensive and pose severe scalability challenges. In our experimental settings, with densely connected nodes of three different types, MRF-based approaches did not perform well and were clearly inferior to the new method developed in this paper.

1.2 Contribution

This paper introduces a classification model and algorithm for *heterogeneous* graphs, based on a novel random surfer model. The approach is related to prior models for random graph walks, and also to previous work on graph-based classification. However, our model goes beyond the prior work by emphasizing the latent value of cross-type edges in heterogeneous networks. As we will see later, this allows our method to outperform competitors from the family of previously proposed graphical-model methods [20].

The main contributions of this paper are:

- The new algorithm, coined Graffiti, takes into account node contents (e.g., photo descriptions), S-edges among nodes of the same type, and X-edges across different node types.
- Our method is based on a mathematically principled random-walk model. We precisely characterize the stochastic process that drives it and prove desired properties like convergence and unique solution.
- We present an extensive experimental study of the behavior of Graffiti and related methods, using two real-world data set extracted from *Flickr* and *LibraryThing*.

2 Related work

In standard *content-based* classification when the classifier is faced with unknown data and has to predict its labels, it will apply the learned decision model to each data item in a "context-free" manner, i.e., blindly omitting any information about the relations among the data items in the set. However, links are important for the classification process [26] and the existence of relational data sets inspired significant work on *homogeneous graph-based classification* [14, 19, 20, 49]. The graph-labeling problem has been formalized as a metric labeling problem [4, 30], relaxation labeling problem [1, 11], neighborhood content influence [39], regression model [35] comprising the link structure into the document representation. Another view over the problem setting is employed in [7, 8] where graph min-cuts are used in the learning process.

The work by Kleinberg and Tardos [30] views the classification problem for nodes in an undirected graph as a *metric labeling problem* aiming to optimize a combinatorial function that consists of *assignment* costs and *separation* costs. The assignment costs are based on the individual choice of label made for each object while the separation costs are based on the *pair* of label choices for two neighboring objects. The combination of the assignment and the separation costs gives the total cost. A labeling that minimizes the total cost is called a *maximally likely labeling* of the test graph. [30] shows that the metric labeling problem is NP-hard and presents an approximation scheme for this problem, based on integer linear programming. However, this approach is not practically feasible for large graphs.

Instead of seeking a global optimum, which obtains a unique labeling for all nodes in the test graph, S. Chakrabarti et al. [11] propose to start with a greedy labeling of the graph, paying attention only to the node-labeling (assignment) cost, and then iteratively "correct" the neighborhood labeling where the presence of edges leads to a very high penalty in terms of the separation costs. In the context of link information, the relaxation labeling algorithm first uses a Naive Bayes text classifier to assign class probabilities to each node. Then it considers each page in turn and re-evaluates its class probabilities in light of the latest estimates of the class probabilities of its neighbors. A robust algorithm based on the relaxation labeling paradigm is presented in [1]. The algorithm captures and exploits the link/class patterns in the complete test graph dynamically as the relaxation-labeling procedure iterates, in contrast to previous work [11] which relies solely on the link/class patterns in the training data. As additional information, it considers a metric distance between each pair of category labels in the set of possible labels and selects a "reliable" set of neighbors for each test document avoiding deterioration of the classification results due to noise.

Oh et al. [39] present a hybrid classification approach in which the label of each node d in the graph is influenced by the popularity of this label among all immediate neighbors of d and the level of confidence in the labels of the documents in the neighborhood. An important consideration in this algorithm is the term weight adjustment applied to all documents by using the term frequencies in the neighboring documents and a parameter α that controlls the degree of neighbor influence. In a single pass trough the data collection, following a Naive Bayesian type of reasoning, the algorithm then tries to "correct" the label of each document. This correction step depends on the new feature vector and the confidence the algorithm has gained in the already assigned labels to the neighboring documents of d.

Probabilistic *graphical* models are an elegant framework that combines uncertainty (probabilities) and logical structure (independence constraints) to compactly represent complex, real-world phenomena. The framework is quite general in that many of the commonly proposed statistical models can be described as graphical models [20, 43]. The two most common types of networks are Bayesian Networks [27] and Markov Networks (known as Markov Random Fields or MRFs) [41]. Due to their flexibility, probabilistic graphical models have become quite popular for addressing learning problems on graphs.

We perform a comparative study between the proposed approach, coined Graffiti, and an MRF-based approach in Section 4. We also present detailed comparisons to models that incorporate the neighborhood features into the representation of the document while modeling the knowledge about the graph. We conclude that homogeneous classification methods show severe limitations when applied on heterogeneous networks.

Heterogeneous-graph mining has recently gained attention, but so far the work is limited to *unsupervised* methods like clustering and spectral decomposition. Wang et al. [48] constructs a co-occurrence matrix to capture pairwise relations among any two types of objects. Traditional LSA is performed to discover the strongest cross-relationships among objects. Kolda et al. [31] discusses a simultaneous analysis of hyperlink structure and anchor text or page content by means of representing the data as a tensor and applying Parallel Factors (PARAFAC) decomposition [22, 46]. Gao et al. [17] performs "high-order" mining of heterogeneous relations (as opposed to decomposing the data into pair-wise relations between different object types); it is based on the principles of information-theoretic co-clustering [13]. None of these unsupervised methods can actually label the nodes of heterogeneous networks; so they are not suitable for classification.

Related to our approach is the random surfer model [21]. Some of its most prominent implementations that are used to determine a topic distribution per node in the underlying graph are topic-sensitive PageRank [24] and topical link analysis [38]. The latest work along these lines was presented in [16] and assumes existence of labeled nodes in the network, thus covering the semi-supervised classification scenario. The intuition behind the topic-sensitive PageRank model [38] is that the random walk of an intelligent surfer is influenced by two factors: a topic distribution for each page and an importance score of a page reflecting the graph topology. The latter is referred to as an *authority* for page v, while the first is a *content* vector. The authority propagation is influenced by using the topic distribution which embodies the context in which a link is created. This way, a resource that is highly popular for one topic can not dominate the results of another topic in which it is less authoritative—an effect which easily occurs in traditional link analysis. Therefore, the score vector for each page is calculated to distinguish the contribution from different topics, using a random walk model that probabilistically combines page topic distribution and link structure. Topical HITS [6], the intelligent surfer model [42], and bookmark coloring [5] are along similar lines (although the latter also uses a coloring metaphor, it is not particularly related to our work and addresses a very different problem).

There are numerous semi-supervised learning methods for graph inference. The main difference among them is their way of realizing the assumption of label consistency in the graph. This assumption states that nearby points are likely to have the same label and points on the same structure are likely to have the same or similar class labels. Transductive inference methods for learning with local and global consistency [52, 53] use the analogy of spreading activation networks from experimental psychology [45]. In each iteration of the process, each node in the network receives information from its neighbors, and also retains its initial information. After sufficient number of iterations, the network reaches a consistent state, where each node is set to be the class of which it has received most information during the iteration process. Closely related to these inference methods is recently proposed work on ranking with local regression and global alignment [51] for media retrieval. The optimization function differs from [53] in the used Laplacian matrix, which is learned by local regression and global alignment. Such learning algorithms have been successfully applied on the task of classification of *homogeneous* graphs where near-by nodes indeed comply with the assumption of local consistency. It is a challenge to use such methods to label *heterogeneous* graphs where the local consistency assumption holds only among nodes that share the same type. It is not clear how to spread the information across types and reach a globally consistent state without ignoring the relations between different node-types. Two major questions are to be solved when confronted with labeling of heterogeneous graphs: how do we leverage the overwhelmingly large number of cross-type X-edges and too few S-edges linking nodes of the same type, and how to achieve label consistency on a local as well as global level given that the set of labels for each of the node types in the heterogeneous graph are typically disjoint. Yang et al. [51] presents a ranking algorithm that deals with heterogeneous media content (text, image, audio) that is part of a single multimedia document (MMD). The ranking algorithm regards all MMDs as *homogeneous* nodes in a database and ranks them when presented a user query. In contrast, Graffiti aims at classifying *heterogeneous* graphs avoiding any mapping of nodes into a homogeneous space first.

3 Model and algorithm

Our algorithm takes a prior label distribution over the graph nodes as input. This labeling can be obtained using any of the existing techniques for text classification or homogeneous graph classification, in case the heterogeneous graph at hand has many S-edges. Graffiti adjusts this labeling by utilizing the significantly higher number of cross-type (X-)edges to devise a propagation scheme of the mutual influence among nodes. A sketch of the model is presented in [2]. In the following we explain the process in detail.

3.1 Notation

We denote the set of nodes by $V = V_1 \cup V_2 \cup \ldots \cup V_t$, where each V_i is the set of nodes of type i (with t different types). Note that G is a directed graph, in particular, all X-edges are bidirectional (see Section 1). Each node v has a heterogeneous neighborhood of immediate neighbors N_v. For each node set V_j, $1 \leq j \leq t$ there are sets of possible class labels $\mathcal{C}_j = \{c_{ji} | 1 \leq i \leq n_j\}$ where n_j is the cardinality of the label set of type j. These type-specific label sets $\mathcal{C}_j$ need not to be completely disjoint. However, the completely disjoint case is most challenging. The union of all possible class labels in the graph $\mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2 \cup \ldots \cup \mathcal{C}_t$ is of size $n = |\mathcal{C}|$; this is usually a relatively small number (e.g., a few tens of different class labels altogether).

With each node v we associate two vectors:

- A *prior class-label probability distribution* of v, denoted by

$$\lambda(v) = (\lambda_1(v), \ldots, \lambda_n(v))^T.$$

- A vector

$$\mu(v) = (\mu_1(v), \ldots, \mu_n(v))^T$$

that captures the *class-specific (graph-topological) importance* of a node. That is, a high μ_i value indicates that v is among the most important nodes within the class c_i of nodes labeled c_i. Summing up all vector components yields the total importance of a node:

$$M(v) = \sum_i \mu_i(v).$$

$M(v)$ is a measure of *topic-biased* authority, and we will later show that it equals PageRank-like authorities for the nodes of the graph G (when ignoring the types of nodes).

To ease comprehension, we give a brief summary of notation in Table 1.

3.2 Intuition

Graffiti models the influence among nodes of the same type by looking at their common neighbors, which typically belong to other types. We base the strength of the mutual influence between two nodes $v, v' \in V_k$ on two criteria:

- The bigger the number of shared neighbors, the higher the influence of v and v' on each other.

Table 1 Summary of notation.

Notation	Meaning
V_k	nodes of type k ($k = 1..t$)
$type(v)$	the type of node v
$\mathcal{C}_k$	class labels for nodes in the set V_k
$\mathcal{C}$	class labels in $\bigcup_i \mathcal{C}_k$
n	number of all possible classes in $\mathcal{C}$
c_i	class label (or color)
N_v	direct neighbors of node v
$\lambda(v)$	vector comprising the prior label distribution of node v
$\mu(v)$	vector comprising the class-specific importance of node v
$\lambda_i(v)$	ith component of $\lambda(v)$ describing the prior that v has label c_i
$\mu_i(v)$	ith component of $\mu(v)$ describing the importance of v for class c_i
$M(v)$	total importance of v given by $\sum_i \mu_i(v)$
$Out(v)$	number of outgoing edges of v
$Out_k(v)$	number of outgoing edges of v that point to nodes of type k

- The higher the mutual influence among shared neighbors, the higher the influence of v and v' on each other.

These criteria are modeled as a *random walk* that the intelligent Graffiti surfer takes on the graph G. To describe the intuition, let us think of class labels as colors. The random surfer would start her walk from a node $v' \in V_k$ picked uniformly at random from all nodes in G. She would try to "paint" nodes v that belong to the same type V_k as v', based on the coloring of v' if $N_v \cap N_{v'} \neq \emptyset$ (i.e. influenced by their common neighbors). In order to reach the node v from v', the surfer can either take a one-hop walk to v (following an S-edge between v' and v), or perform a two-hop walk, by first following a link to a common neighbor $u \in V_l$, and then following a link from u to v (i.e. following two edges).

Hence, our random surfer model includes the following components:

- the probability, J, that the random surfer performs a random jump to any node v.
- the probability, $F^2_{same\ color}$, that the surfer reaches $v \in V_k$ from a node $v' \in V_k$ through a common neighbor and uses the same color as at v'.
- the probability, $F^2_{change\ color}$, that the surfer reaches $v \in V_k$ from a node $v' \in V_k$ through a common neighbor and uses another color than at v'.
- and the probability, F^1, of following a link to any node of the heterogeneous neighborhood.

Three damping factors q, α, β are used to control the random walk. We describe the algorithmic details in the following Section 3.3. The walk process is depicted in Algorithm 1.

3.3 Algorithm

In the random surfer model as implemented by the PageRank algorithm [40], the random surfer walks through a directed graph $G = (V, E)$. At any node v, she may continue her walk by following an outgoing edge from v with a probability inversely proportional to the out-degree of v. Alternatively, she may decide to restart her walk by jumping to any random node with a probability proportional to $1/|V|$. Finally, the probability that the random surfer is at a node v is given by

$$P(v) = \frac{(1-q)}{|V|} + q \cdot \sum_{v \to v'} \frac{P(v)}{Out(v)} \tag{1}$$

where $Out(v)$ stands for the number of the outgoing edges of v, and q is a damping factor that is usually set to 0.85 [40]. This model defines an ergodic Markov chain whose stationary state probabilities yield the PageRank values $P(v)$. The solution is unique and independent of the starting condition; it is typically computed by the Jacobi power-iteration method [32].

When modeling Graffiti, we maintain the damping factor q and denote the probability that the surfer jumps to a random node $v \in G$, and uses color c_i there, by

$$P[v, c_i, J] = (1-q) \cdot \frac{\lambda_i(v)}{|V|}. \tag{2}$$

Algorithm 1 Random walk

```
Initialize damping factors $q, \alpha, \beta$
Initialize color $c$
Initialize node $v$
$k = type(v)$;
$n = |V|$;
$V_k = \emptyset$;
for all $v \in V$ do
    if $type(v) = k$ then
        $V_k = V_k \cup v$;
    end if
end for
while true do
    coinflip = Random(0, 1);
    if coinflip $\leq (1 - q)$ then
        // Perform random jump $(J)$
        $v = u$, where $u$ picked u.a.r. from $V$;
        $k = type(v)$;
        continue;
    else
        coinflip = Random(0, 1);
        if coinflip $\leq (1 - \alpha)$ then
            // Perform $F^2_{same\ color}$
            $v = v'$, where $v'$ picked u.a.r. from $V_k$;
            continue;
        else
            coinflip = Random(0, 1);
            if coinflip $\leq (1 - \beta)$ then
                // Perform $F^2_{change\ color}$
                $v = v'$, where $v'$ picked u.a.r. from $V_k$;
                $c = c'$, where $c' \in C_k$ picked with a probability $\lambda(c')$;
                continue;
            else
                // Perform $F^1$
                $v = u$, where $u$ picked u.a.r. from $Neighborhood(v) \setminus V_k$;
                $k = type(v_{next})$;
                $c = c'$, where $c' \in C_k$ picked with a probability $\lambda(c')$;
            end if
        end if
    end if
end while
```

As explained in Section 3.2, the Graffiti surfer can perform three characteristic actions: (1) a random jump denoted by J, (2) a one-hop walk denoted by F^1, and (3) a two-hop walk denoted by F^2. Hence, apart from the damping factor q we introduce two more damping factors α and β whose roles are depicted in Figure 3 and explained below.

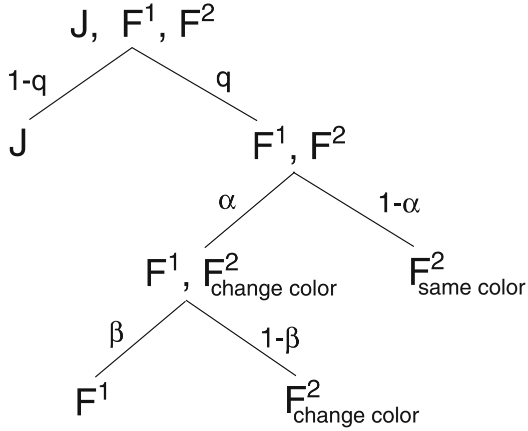


Figure 3 Role of damping factors.

While performing a two-hop walk F^2 from a node $v' \in V_k$ to a node u and back to a node $v \in V_k$ the surfer keeps her color c_i unchanged with probability

$$P[v', v, c_i, c_i, F^2_{same\ color}] = q \cdot (1-\alpha) \cdot \sum_{u:v' \to u \to v} \frac{\mu_i(v')}{|N_{v'}| \cdot Out_k(u)} \tag{3}$$

That is, the "stickiness" of color c_i is proportional to $\mu_i(v')$. In this case, $\mu_i(v')$ can be read as the probability that the Graffiti surfer is using color c_i at node v'.

Analogously, the surfer might decide to change the color in use (i.e. c_i) to any $c_j \in \mathcal{C}_k$ in order to paint v with probability

$$P[v', v, c_i, c_j, F^2_{change\ color}] = q \cdot \alpha \cdot (1-\beta) \cdot \sum_{u:v' \to u \to v} \frac{\mu_i(v') \cdot \lambda_j(v')}{|N_{v'}| \cdot Out_k(u)} \tag{4}$$

In this case, $\lambda_j(v')$ is a prior that reflects how "good" the color c_j is for the node v'. Consequently, the probability that the Graffiti surfer uses c_i at v has to be proportional to $\lambda_j(v')$.

On the other hand, the surfer can decide to perform a one-hop step F^1 to a node u of any type, and paint this node using a color c_j with probability

$$P[u, v, c_i, c_j, F^1] = q \cdot \alpha \cdot \beta \cdot \frac{\mu_i(u) \cdot \lambda_j(u)}{|N_u|} \tag{5}$$

Finally, the importance of a node $v \in V_k$ with respect to class c_i is given by:

$$\begin{aligned} \mu_i(v) = {} & P[v, c_i, J] + \sum_{v' \in V_k} P\left[v', v, c_i, c_i, F^2_{same\ color}\right] \\ & + \sum_{v' \in V_k} \sum_j P\left[v', v, c_j, c_i, F^2_{change\ color}\right] \\ & + \sum_j P\left[u, v, c_j, c_i, F^1\right] \end{aligned} \tag{6}$$

Plugging the previous equations into the last one, we arrive at:

$$\mu_i(v) = (1-q)\cdot\frac{\lambda_i(v)}{|V|} + q\cdot(1-\alpha)\cdot\sum_{\substack{v'\in V_k,\\ u:v'\to u\to v}}\frac{\mu_i(v')}{|N_{v'}|\cdot Out_k(u)}$$

$$+q\cdot\alpha\cdot(1-\beta)\cdot\sum_{\substack{v'\in V_k,\\ u:v'\to u\to v}}\sum_j\frac{\mu_j(v')\cdot\lambda_i(v')}{|N_{v'}|\cdot Out_k(u)}$$

$$+q\cdot\alpha\cdot\beta\cdot\sum_{u:u\to v}\sum_j\frac{\mu_j(u)\cdot\lambda_i(u)}{|N_u|} \qquad (7)$$

The Graffiti algorithm computes these $\mu_i(v)$ values for each node v. So the output of the algorithm is a class probability distribution for each node, restricted to the corresponding node type.

We show in the following section that the presented Equation (7) for computing the values of $\mu_i(v)$ leads to a unique solution and can be computed effectively. We will see that we can use the PageRank-style power iteration (Jacobi method) to compute these values.

A natural question arising at this point is whether the F^2 walk (plus the F^1 walk and the jump J) model is indeed sufficient to capture the mutual influence between nodes. Starting at a node of a given type, the surfer could also walk through a series of nodes of different types and return to a node of the origin's type after three or more hops. Following the arguments of [3], such walks have diminishing influence of longer distances in these kinds of stochastic processes but could be potentially informative. Note that such pattern of walking is *incorporated* in Graffiti, namely by combining a series of F^2 and F^1 walks.

3.4 Properties

The result that the $\mu_i(v)$ values can be computed by power iteration is not obvious from the definition of our random surfer model. In this section we state a number of theorems for our framework that together constitute this result.

Theorem 1 *The Graffiti model, as defined by* (7)*, is a finite-state Markov chain with* $|V|\cdot|C|$ *states.*

Proof We show this result by constructing the Markov chain. This will later prove useful for other theorems as well, most notably, for characterizing the convergence rate of the Graffiti method.

We decompose each node $u \in V$ into a set of sub-nodes $u_1, ..., u_{|C|}$, leading to a new graph $G' = (V', E')$ with node set $V' = \{u_1, ..., u_{|C|} | u \in V\}$, i.e., $|V'| = |V| \times |C|$ as shown in Figure 4. The edge set of G' is defined by $E' = \{(u_i, u'_j) \in V' \times V' |\, u, u' \in V_k; \exists (u, v), (v, u') \in E, v \in V_l\} \cup \{(u_i, u'_j) \in V' \times V' | (u, u') \in E\}$. That is, there is an edge from sub-node u_i to a sub-node u'_j, with the two sub-nodes belonging to nodes from the original V if there is a two-hop path between them via node v of any type, or if the original E already contains an edge from u to u'.

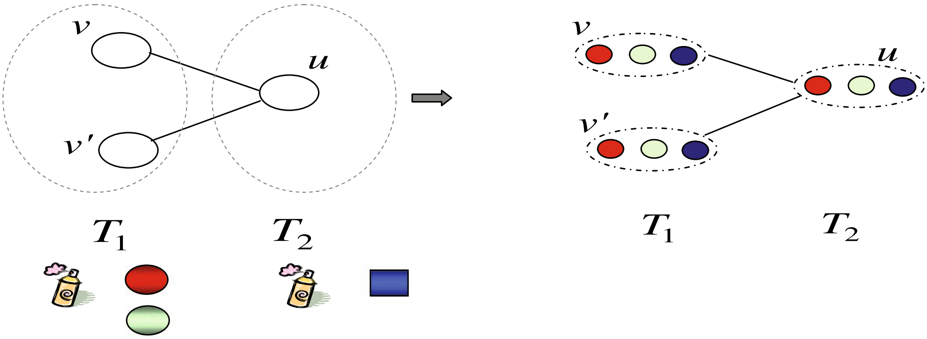


Figure 4 Node representation. On the left hand side, we show the example graph G that consists of two types of nodes: T_1 and T_2, and the set of possible classes consists of three labels: *red*, *green* and *blue*. The graph G' is shown on the right hand side and represents the decomposed graph, used by the Graffiti model, where each node u in G' is represented by a set of sub-nodes $u_1, ..., u_{|\mathcal{C}|}$.

As illustration, the small toy graph G depicted in Figure 4 leads to the sub-node-level graph G' depicted in Figures 5, 6 and 7.

Two nodes v and v' from V_1 are connected to a node u from V_2 in G. Then each sub-node of v will be connected to each other sub-node of v (the same holds for v'). Each sub-node of v will be connected to each sub-node of v' in G', representing the indirect connection of v and v' through u. Additionally, each sub-node of v and v' will be connected to each sub-node of u.

More formally, each edge in E' is assigned a weight between 0 and 1. A directed edge (v_i, v_i'), where nodes v_i, v_i' belong to the same type V_k, has weight:

$$\sum_{v,u:v\rightarrow u\rightarrow v'} \frac{(1-\alpha)+\alpha\cdot(1-\beta)\cdot\lambda_i(v)}{|N_v|\cdot|Out_{type(v)}(u)|}. \tag{8}$$

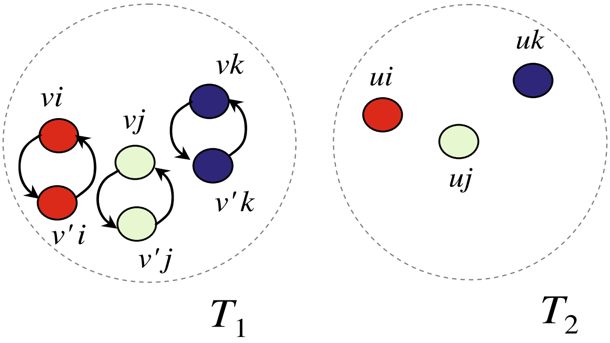
(The walk from v_i to v_i' can be the result of an $F^2_{same\_color}$ or an $F^2_{change\_color}$ step.)

Accordingly, a directed edge (v_i', v_i) has weight

$$\sum_{v',u:v'\rightarrow u\rightarrow v} \frac{(1-\alpha)+\alpha\cdot(1-\beta)\cdot\lambda_i(v')}{|N_{v'}|\cdot|Out_{type(v')}(u)|}. \tag{9}$$

This type of edge is depicted in Figure 5.

Figure 5 Link formation for the toy example of Figure 4. The edges represent the surfer's walk in which he follows links to same-type neighbors and keeps the previously picked color.



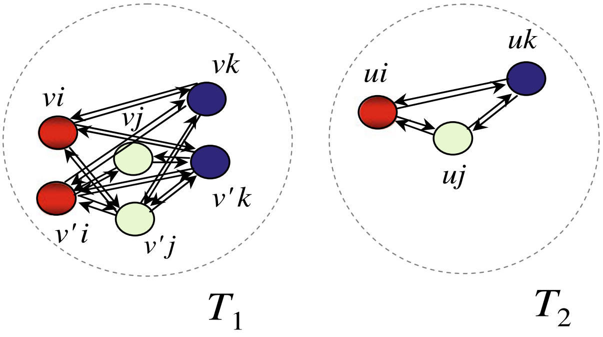


Figure 6 Link formation for the toy example of Figure 4. The edges represent the surfer's walk in which he follows links to same-type neighbors but changes color.

A directed edge $(v_i, v'_j), i \neq j$ has weight

$$\sum_{v,u:v\to u\to v'} \frac{\alpha \cdot (1-\beta) \cdot \lambda_j(v)}{|N_v| \cdot |Out_{type(v)}(u)|}. \tag{10}$$

(The walk from v_i to v'_j, $i \neq j$ can only be the result of an $F^2_{change\_color}$ step.)

The corresponding type of edges is shown in Figure 6.

Finally, a directed edge $(v'_i, v_j), i \neq j$ has weight

$$\sum_{v',u:v'\to u\to v} \frac{\alpha \cdot (1-\beta) \cdot \lambda_j(v')}{|N_{v'}| \cdot |Out_{type(v')}(u)|}.$$

This type of edges are depicted in Figure 7.

All directed edges that connect sub-nodes of the same node (e.g. sub-nodes of v) are weighted analogously.

Apart from the edges described above, the graph G' also contains edges that connect each sub-node from v and v' to each sub-node from u. An edge (v_j, u_l) has weight

$$\sum_{v:v\to u} \frac{\alpha \cdot \beta \cdot \lambda_l(v)}{|N_v|}, \tag{11}$$

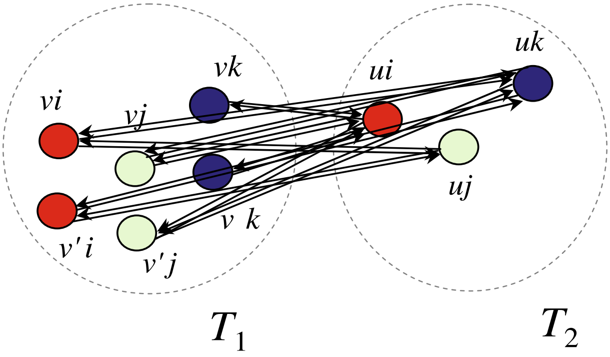


Figure 7 Link formation for the toy example of Figure 4. The edges represent the surfer's walk in which he follows links to nodes of other types.

and an edge (u_l, v_j) has weight

$$\sum_{u:u\to v} \frac{\alpha \cdot \beta \cdot \lambda_j(u)}{|N_u|}. \tag{12}$$

The corresponding $|V| \cdot |C| \times |V| \cdot |C|$-dimensional adjacency matrix P is a stochastic matrix, which defines a finite-state Markov chain with P_{ij} denoting the transition probability from state i to state j.

To see that P is stochastic, i.e., its row sums equal 1, consider a sub-node v'_i of G' that belongs to a node v' of G. Such a node has

$$\sum_{v,u:v\to u\to v'} |N_v| \cdot |Out_{type(v)}(u)|$$

linked edges of weight

$$\sum_{v,u:v\to u\to v'} \frac{(1-\alpha) + \alpha \cdot (1-\beta) \cdot \lambda_i(v)}{|N_v| \cdot |Out_{type(v)}(u)|}. \tag{13}$$

Furthermore, such a node v'_i has

$$\sum_{j\neq i} \sum_{v,u:v\to u\to v'} |N_v| \cdot |Out_{type(v)}(u)|$$

linked edges of weight

$$\sum_{v,u:v\to u\to v'} \frac{\alpha \cdot (1-\beta) \cdot \lambda_i(v)}{|N_v| \cdot |Out_{type(v)}(u)|}. \tag{14}$$

Consequently, the total weight of edges linked to v'_i, that result from two-hop paths, is $(1-\alpha) + \alpha \cdot (1-\beta) = 1 - \alpha \cdot \beta$. Besides these edges, v'_i is linked to other $\sum_i \sum_{u:u\to v'} |N_u|$ of weight

$$\sum_{u:u\to v'} \frac{\alpha \cdot \beta \cdot \lambda_i(u)}{|N_u|}.$$

The overall weight of these edges is $\alpha \cdot \beta$. Accordingly, the total weight of edges connected to v'_i in G' is given by $1 - \alpha \cdot \beta + \alpha \cdot \beta = 1$.

Finally, we define a $|V| \cdot |C| \times |V| \cdot |C|$-dimensional matrix E where E_{ij} denotes the random jump probability from a subnode u_i to a subnode v_j, hence we set E_{ij} to $\lambda_j(v)/|V|$. The matrix $A = [qP + (1-q)E]^T$ is the transition probability matrix that corresponds to (7). □

Theorem 2 *The Graffiti model (based on (7)) is an ergodic Markov chain. This means that there exist stationary state probabilities that are unique and independent of the initial state probabilities (when the Markov-chain process is started).*

Proof We show that the Markov chain is aperiodic and irreducible. According to the definition of the jump probability J in (7), each state in our Markov chain is reachable from any state, which means that the Markov chain is irreducible.

As for aperiodicity, a state v_i of a Markov chain is called non-periodic if the greatest common divisor of the lengths of all cycles that lead back to v_i is one. Since for each state in our Markov chain, there is a cycle of length 1 that leads back to the same state with probability $(1-q)\cdot\frac{\lambda_i(v)}{|V|}$, the aperiodicity condition is fulfilled.

Finally, for finite-state Markov chains, aperiodicity and irreducibility guarantee ergodicity [15]. □

These results lead to the conclusion that there exists a unique solution for the Graffiti random surfer model.

The PageRank estimation is typically carried out by applying the iterative Jacobi method [32]. Equation (1) shows that the rank of page v is a function of the ranks of the pages pointing to it. In other words, the rank of each page v is boosted by a fraction of the rank of the pages pointing to v inversely weighted by the number of outgoing links of these pages. To ease comprehension, the recursive definition in (1) can be rewritten as:

$$p_i = (1-q) + q \sum_{(j,i)\in E} Out(j)^{-1} p_j, \forall i \tag{15}$$

where p_i is the page rank of page i, (j, i) denotes an edge between pages j and i, and $Out(j)$ is the number of links pointing out of j. On the other hand, (15) can be represented in a matrix form as:

$$p = (1-q)e + qAp \tag{16}$$

where $p = (p_1, \ldots, p_i, p_j \ldots, p_n)^T$ is a page rank vector; e is a vector of n entries all of which are 1's; and $a_{ij} = Out(j)^{-1}$ if $e(i, j) \in E$ and 0 otherwise.

Equation (16) can be seen as a set of equations written as:

$$(I - qA)p = (1-q)e \tag{17}$$

A well known iterative method for solving (17) is the Jacobi iteration which computes for each iteration k

$$p_i^{(k+1)} = (1-q) + q \sum_{(j,i)\in E} a_{ij} p_j^{(k)} \tag{18}$$

The method is guaranteed to converge if the matrix A is strictly or irreducibly diagonally dominant. Recently, a double-loop technique was introduced to force convergence of the Jacobi algorithm to the correct solution even when the sufficient conditions for convergence do not hold. The double loop technique works for either positive definite or column dependent matrices [29]. Thus, the PageRank computation converges with the rate of convergence determined by the Jacobi method.

The next result shows that the convergence rate of Graffiti is comparable to that of PageRank.

Lemma 3 *The error of the Jacobi iteration method decreases by powers of q.*

Proof The proof for Lemma 3 is derived from [23] where Haveliwala et al. show that for real-life networks, such as the Web, the convergence rate is q or better. This confirms that in practice, the convergence rate of Graffiti is the same as the one of PageRank.

In more detail, [23] shows that any matrix $A' = [qP' + (1-q)E']^T$, where P' and E' are $m \times m$ stochastic matrices and E' is rank-one row-stochastic, has a second eigenvalue of $|\lambda_2| \leq q$. It is easy to see that the matrix E from the proof of Theorem 1 is row-stochastic and has rank 1. Theorem 2 entails that the eigenvalue associated with the principal eigenvector of A (as defined in the proof of Theorem 1) is 1. Since the convergence rate of a stochastic matrix is given as $|\lambda_2|/|\lambda_1|$, it follows that $cr(A) \leq q$, where $cr(A)$ denotes the convergence rate of A. □

3.5 Implementation and efficiency analysis

Our Graffiti implementation is based on sparse matrix representations and optimized matrix operation strategies. We present the pseudo code in Algorithm 2.

Each row of the matrix $M_{|V|\times|C|}$ corresponds to the vector $\mu(v)$ for a node v. In order to capture two-step connections between nodes of the same type, as needed for the F^2-walk, we construct the matrices A and B for each type k. The second and third components of (7) are computed on line 18. Note that, we make use of matrix-vector multiplications and matrix summations, and never matrix-matrix multiplications. Finally, the last component of (7) is computed in line 20.

A straightforward iterative implementation of (7) would lead to a running time of $O(|C| \cdot |E_X| \cdot \sqrt{(|E_X|)}$ for a single iteration, which in a dense graph G_X would correspond to $O(|C| \cdot |V|^3)$. Our implementation guaranties a running time of $O(|C| \cdot |E_X|)$ for a single iteration, which is comparable to the running time of one PageRank-style power iteration as given by the Jacobi method [33].

The Graffiti algorithm is scalable and can be applied to real-world graphs containing millions of nodes and edges. Since the algorithm is represented in terms of matrix-vector multiplications (similar to PageRank computations), it can easily be implemented under the MapReduce framework [34]. The framework allows a distributed and highly scalable computation and guarantees the applicability of the proposed approach to big scale real-life problems.

4 Experiments

4.1 Data

In general, it is hard to freely obtain a heterogeneous dataset for research purposes as privacy, security and other legal issues prohibit sharing such data. Note that in our setting we need a ground truth for evaluating the classifier's performance. Therefore, for comparing the performance of the state-of-the-art approaches and the approach we propose, we constructed a dataset by using the *Flickr* photo sharing network (www.Flickr.com) and mapped users, photos and tags onto existing *Flickr* groups to obtain a ground truth for evaluation. As a second dataset, we crawled the *LibraryThing* social network (www.librarything.com), where users talk about the books they have read, their favorite authors, and so on.

Flickr allows users to share personal photos and attach tags to them (in the spirit of Web 2.0 "folksonomies"). We extracted a portion of *Flickr* with approx. 230,000 nodes belonging to one of three types: 123,000 *photos*, 104,900 *tags*, and 3,570 *users*. All nodes are linked by a total of about 2,700,000 edges; out of these 23,200

Algorithm 2 Graffiti Implementation.

```
Create $M_{|V| \times |\mathcal{C}|}$
for all $v \in V$ do
  for all $i \leq |\mathcal{C}|$ do
    $M[v, i] = (1 - q) \cdot \frac{\lambda_i(v)}{|V|}$
  end for
end for
while $|M^{(r+1)} - M^{(r)}| \geq \epsilon$ do
  for $k = 0$ to $m$ do
    Construct
      $B[v, u] = \begin{cases} \frac{1}{Out_{type(v)}(u)}, \text{ if } u \rightarrow v \\ 0 \text{ , otherwise.} \end{cases}$
    Create vectors:
    $\mathbf{X} = (1, ..., 1)$
    for $i = 0$ to $|\mathcal{C}|$ do
      for all $v \in V$ do
        $M^{(r+1)}[v, i] = (1 - q) \cdot \frac{\lambda_i(v)}{|V|}$
      end for
      $a = q \cdot (1 - \alpha)$
      $c = q \cdot \alpha \cdot (1 - \beta)$
      Construct
        $A[u, v'] = \begin{cases} \Big(a \cdot \frac{M[v', i]^{(r)}}{|N_{v'}|} + \\ c \cdot \sum_j \frac{M[v', j]^{(r)} \cdot \lambda_i(v')}{|N_{v'}|}\Big), \\ \qquad \text{if } v' \rightarrow u \\ 0 \text{ , otherwise.} \end{cases}$
      where $v, v' \in V_k$.
      $M[, i]^{(r+1)} = M[, i]^{(r+1)} + (B \cdot (A \cdot X^T))$
      Create a vector $\vec{b}$ which takes values $b(v)$ for each $v \in V_k$
      $b(v) = q \cdot \alpha \cdot \beta \cdot \sum_{u: u \rightarrow v} \sum_j \frac{M[u, j] \cdot \lambda_i(u)}{|N_u|}$
      $M[, i]^{(r+1)} = M[, i]^{(r+1)} + \mathbf{b}$
    end for
  end for
end while
```

S-edges. Edges express different binary relations among the three types of nodes. Edges between user-type and tag-type nodes entail which tags have been used by which users. Edges between tag- and photo-type nodes entail which tags have been used to describe which photos. Edges between user- and photo-type nodes express ownership of a photo by a certain user. We assume that the given graph is undirected, since these binary relations are naturally interpretable in both directions.

We use the same set of class labels for all three node types. Note that even if two nodes from different types have the same label, it is conceptually different,

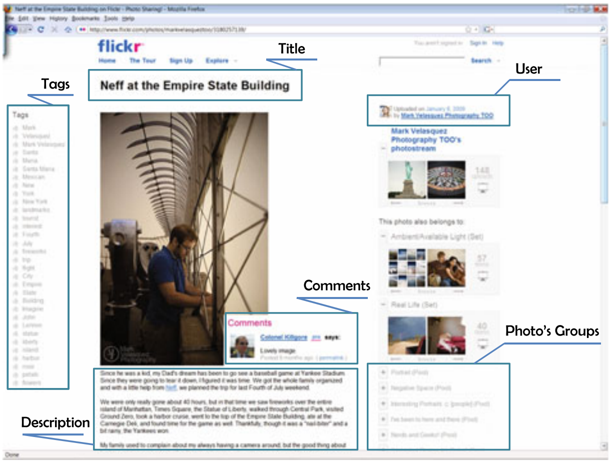
Table 2 Class label distribution for Flickr dataset.

Class	Size for type	
	Users	Photos
Animals	1,687	37,537
Flower	1,612	22,406
Architecture	870	23,159
Portrait	954	33,396
Nature	871	28,083

since the labels are type-specific (i.e. User.c_i is different from Photo.c_i). We use the following labels: (1) Animals, (2) Flower, (3) Architecture, (4) Portrait, and (5) Nature. Detailed overview of the label distribution is presented in Table 2.

These are explicit category names in *Flickr* itself, and the following describes how we obtained the ground-truth labels for our experimental data. Note that extracting more discriminative ground truth with possibly disjoint labels in a principled way in the *Flickr* setting is difficult if not impossible. User and photo nodes have a priori classes assigned by the users themselves: users can join thematically driven, explicitly named groups and the photos, users have pointed out as relevant, are assigned to groups as well. We use group names of some larger groups as class labels. For tag nodes, we collected statistics about how often a tag occurs together with a photo that belongs to a particular group, and then computed a probability/frequency vector of class memberships for each tag. This way we also reflect ambiguity of tags and statistically preferred meanings. For example, the tag "tiger" belongs to Animals, Nature, and Portrait (as people with nickname "tiger" exist, such as the singer Tom Jones), but the highest frequency of this tag would correspond to Animals. We computed analogous probability/frequency vectors for user and photo nodes, based on the number of postings per user in each of the groups/classes.

Figure 8 shows an example page found on *Flickr* which belongs to the "Portrait" group among many others, and depicts a photograph of a person taken at the Empire State Building. The photograph is associated with a number of tags e.g. "New York", "City", "Empire", "flowers", "rose", and so on, as well as a comment from another *Flickr* user saying that he liked the image. The user who took and posted



Since he was a kid, my Dad's dream has been to go see a baseball game at Yankee Stadium. Since they were going to tear it down, I figured it was time. We got the whole family organized and with a little help from Neff, we planned the trip for last Fourth of July weekend.

We were only really gone about 40 hours, but in that time we saw fireworks over the entire island of Manhattan, Times Square, the Statue of Liberty, walked through Central Park, visited Ground Zero, took a harbor cruise, went to the top of the Empire State Building, ate at the Carnegie Deli, and found time for the game as well. Thankfully, though it was a "nail-biter" and a bit rainy, the Yankees won.

My family used to complain about my always having a camera around, but the good thing about

Figure 8 An excerpt of the *Flickr* social network.

Table 3 Class label distribution for LibraryThing dataset for nodes of type User.

Class	Size
BBC Radio 3 listeners	241
FantasyFans	249
Graduate students	232
Historical fiction	348
Read young adult literature	356
Science fiction fans	294

this photograph is Mark Velasquez. She has shared on her profile few words on her occupation, the groups she is active member of (Travel Photography, Portrait, the Face, etc.) and some people who she has tagged as her friends have posted testimonials related to what kind of person she is or what do they like about her photography. The task of the classifier is to assign the user Mark Velasquez to the classes "Nature", "Flower", and "Portrait" (based on her online history), and the discussed here photograph to the class "Portait".

The crawl of the *LibraryThing* graph resulted in about 40,000 nodes distributed in four layers: users, books, authors, and tags. The graph consists of 1,500 users, 11,000 books, 2,500 authors and 23,000 tags. The nodes are connected with 420,000 links, out of which 640 are S-links. The ground truth for users was extracted directly from the existing user groups in *LibraryThing*. In our subgraph, users belong to *one or more* of the following groups: BBC Radio 3 Listeners, Fantasy Fans, Graduate Students, Historical Fiction, Read YA Literature, and Science Fiction Fans as shown in Table 3.

All books read by these users were mapped to their Amazon entries. Thus, we used the Amazon category system to derive ground truth for the nodes of type books. The resulting set of classes for books are: Children's Books, Science Fiction & Fantasy, Mystery & Thrillers, History, Religion & Spirituality, and Nonfictionbv. We show the class label distribution in Table 4. We used the Amazon top-layer categories. The type-specific classes for books were primarily dictated by the user profile and not by specially performed filtering on the Amazon categories.

Finally, the authors of these books were connected to their *Wikipedia* pages and the existing categories in Wikipedia were taken as ground truth for all nodes of type author. Their classes are: Children writers, Historians, Journalists, and Novelists. We have merged categories distinguishing authors only by nationality, e.g. British Historians and German Historians were mapped to Historians. The class label distribution for nodes of type Author are presented in Table 5.

The ground truth for nodes of type tag is a probability/frequency vector with an entry for each possible class of books. The value for class *c* is the ratio of books from this class tagged with the given tag to the total number of books tagged with this tag.

Table 4 Class label distribution for LibraryThing dataset for nodes of type Book.

Class	Size
Children's Books	3,034
History	1,044
Mystery & Thrillers	1,431
Nonfiction	1,344
Religion & Spirituality	1,014
Science Fiction & Fantasy	2,134

Table 5 Class label distribution for LibraryThing dataset for nodes of type Author.

Class	Size
Children writers	395
Historians	375
Journalists	422
Novelists	1,006

For the initial text-only classifier, we use node-type-specific features as follows. For the *Flickr* data, we used the set of tags that a user assigned for the *user* nodes, the title, tags and comments for the *photo* nodes, and the tag itself for the *tag* nodes. For the *LibraryThing* data, we used the profile and comments attached to the profile for each user as features for the *user* nodes, the book title and the Amazon editorial review for the *book* nodes, the first paragraph of the corresponding Wikipedia article for the *author* nodes, and the tag itself as a feature for the *tag* nodes. For all node types we performed two kinds of feature selection: (1) we removed all stopwords (prepositions, conjugations of auxiliary verbs, generic adjectives such as "nice", "however", etc. and common nouns such as "camera", "hardcover","somebody") as Bayesian classifiers are known to be critically sensitive to such words, and (2) we removed all strong cues for the ground-truth labels (words or phrases overlapping with the names of the classes) for we would otherwise have made the classification task almost trivial.

To demonstrate the *LibraryThing* data collection, we pick an example user from the network, namely, the user with a nickname "dovespirit". In her profile she has explicitly marked as friends two other *LibraryThing* users: "beachgirl66" and "dr.velma". Dovespirit is a registered member of the groups Christianity, Fantasy-Fans, Historical Fiction, Romance—from historical to contemporary, and Science Fiction Fans. Looking at her online library, we pick the book "Rapture in Death (In Death)" by J.D. Robb to illustrate the rest of the available data. The book is marked with tags "crime", "mystery", "futuristic", "romantic suspense", "science fiction series", "police", etc. The *LibraryThing* page of the book contains links to other users who also have this book in their online libraries. The Amazon.com Product Description of the book places it in the categories of "Books ⇒ Mystery & Thrillers ⇒ Police Procedurals" and "Books ⇒ Romance ⇒ Romantic Suspense". If we look at the *LibraryThing* page of the book's author J.D. Robb, we find a link to her Wikipedia author page. The latter gives details about the life and career of J.D. Robb whose real name is Nora Roberts. Some of the Wikipedia categories associated with this author are: American novelists; People from Washington County, Maryland; RITA Award Winners; and Female authors who wrote under male or gender-neutral pseudonyms. Based on her online activity in the *LibraryThing* network, the task of the classifier is to place user dovespirit into classes Science Fiction Fans, Historical Fiction and Read Young Adult Literature. It has to associate the book Rapture in Death with the labels Mystery & Thrillers and Science Fiction & Fantasy; and place the book's author J.D. Robb into the class Novelist.

4.2 Training set construction

To train the compared algorithms we apply the following training set extraction procedure. We collect a set of nodes—V_{train}, which are picked uniformly at random from the complete set of nodes V in the graph. V_{train} is used to train all compared algorithms, that is, estimate initial class prior distribution for the corpus, class/term distributions and other required parameters. All nodes in the set V_{train} carry a flag uniquely identifying them as training nodes and thus as nodes, which do not require classification. If during its classification phase an algorithm encounters a training node, it will discard it as a classification candidate and continue to the next eligible for classification node. Nevertheless, V_{train} is part of the total set of nodes V in order to preserve the link structure patterns characteristic for the given corpus. Analogously, the evaluation procedures ignore all training nodes and derive their estimates by looking at the set of nodes $V \setminus V_{train} = \{v \in V | v \notin V_{train}\}$.

4.3 Quality measure

We compare the performance of different classifiers based on the **precision/recall break-even point** (hence, also equal to the micro-averaged $F1$ **measure**) [28], which is a typical measure for the effectiveness of a multi-label classifier. We briefly explain its computation. For every test document d, we have a vector of predicted probabilities, $p(d)$, that d belongs to any class c, as well as a vector of true probabilities, $gt(d)$, for d being in any class c. Let k be the number of classes for which $gt(d, c) > 0$. We sort the prediction vector $p(d)$ in descending order of class probabilities $p(d, c)$ and take the top k of them. Now, we compute contingency tables for each class and determine the precision and recall values per class. **Precision** for class c is the fraction of items labeled c in the ground-truth out of these that are automatically classified into c; **recall** is the fraction of items that are classified into c out of those that bear ground-truth label c. The meeting point of the curves for micro-averaged precision and micro-averaged recall gives us the break-even point, which also equals the micro-averaged harmonic mean of precision and recall, namely F1.

As we produce a vector of predictions, we would also like to test to what extent the order of class probabilities in the ground truth is reflected in the prediction vector. Therefore, we compared the rankings provided by the ground truth vector $gt(d)$ and the prediction vector $p(d)$ by using the **normalized discounted cumulative gain** (NDCG) [25]. It assumes that $g(d)$ gives a “gain” to each class assignment and then measures how much“gain” is delivered by the prediction values of $p(d)$, giving higher weight to higher ranked classes. Both vectors, $gt(d)$ and $p(d)$, are sorted in decreasing order of their class probability values. We first define the measure of discounted cumulative gain DCG and then argue why normalization of DCG is needed to arrive at the definition of the normalized discounted cumulative gain $NDCG$. At a particular rank position r DCG is defined as:

$$DCG_r = \sum_{i=1}^{r} \frac{2^{rel_i} - 1}{\log_b(1+i)}, \tag{19}$$

where rel_i is the graded relevance of the result at position i. Here, the power distribution puts strong emphasis on retrieving relevant documents on higher ranks. The parameter b is a tuning parameter for the penalty associated with predicting a document at a lower rank position than its position in the ground truth vector.

To facilitate comparison of results across a document collection and have a cumulative gain at each position for a chosen value of r, the DCG measure has to be normalized across all documents. Thus, normalized discounted cumulative gain $NDCG$ is defined as:

$$NDCG_r = \frac{DCG_r}{IDCG_r} \tag{20}$$

where for the perfect ranking algorithm, the DCG_r will be the same as the ideal DCG at position r, abbreviated by $IDCG_r$, producing an $NDCG$ of 1.0.

We also compared both vectors in terms of ranking the class probabilities using **Kendall's tau** measure. Kendall's tau is a measure of correlation, and so measures the strength of the relationship between two variables. It varies from $[-1, 1]$ where -1 means that the correlation between the compared vectors is negative, while 1 means perfect match of the rankings produced by both vectors. Obtaining 1 on our corpus is impossible due to existence of ties, but the higher the value of Kendall's tau, the better the correlation among the prediction and ground truth probability vectors. Formally Kendall's tau is defined as:

$$\tau = \frac{n_c - n_d}{\frac{1}{2}n(n-1)} \tag{21}$$

where n_c is the number of concordant pairs, and n_d is the number of discordant pairs in the data set. The denominator in the definition of τ can be interpreted as the total number of pairs of items. So, a high value in the numerator means that most pairs are concordant, indicating that the two rankings are consistent. In our experiments, we use Kendall's tau implementation provided by the LAW project [9]. Note, that this implementation takes into account ties in the prediction as well as ground truth vector.

4.4 Methods under comparison

We compare the performance of five classifiers:

NB A text-only classifier, in our case a Naive Bayesian classifier, which serves as initializer of the topical distribution per node in the graph.

HC A hybrid classifier, that uses the links in a node's neighborhood to enhance its document representation and later on apply a weighted voting scheme to decide on the most probable class of a node [39]. The main idea is that the features of a given node (e.g., derived from the contents of a photo description or user profile) are enriched by superimposing them with features of the node's neighbors (along S-edges in our case). It uses a superposition weight α for the neighbors' feature values, to avoid undue dominance of neighbors. The value of α is tuned by cross-validation with training data (or held-out data in an experimental evaluation) for best performance. After using probabilistic reasoning on the class assignment of each document using its new set of features, the classifier uses a weighted voting scheme to promote classes with high support in the neighborhood.

TopicalPR A topical Page Rank approach [38]. This algorithm focuses on one class c—belonging to the label set of one node type—and performs a random walk over the entire graph with occasional random jumps biased towards nodes labeled c (e.g., all user nodes of class Nature). This way, it computes a PageRank-style confidence vector of nodes being assigned to class *c*. By doing this separately for each class *c*, we compute the multi-label assignment for all nodes, along with label likelihoods. The initial topical distribution per node is taken from the purely textual NB classifier. The algorithm has one tuning parameter, which in our experiments is also tuned to give best performance results for topical Page Rank.

Graffiti The proposed method Graffiti that takes as initial class distribution the predictions returned by the NB text-only classifier (the baseline).

RL An iterative Relaxation Labeling approach from the family of graphical models, that builds on, but extends and generalizes a model [1] which uses the theory of Markov Random Fields to derive a relaxation labeling technique for the graph labeling problem. The approach optimizes a global objective function on the graph whose nodes are being labeled. It starts with an initial node labeling (e.g. by a text-only classifier), and iteratively improves it by maximizing the likelihood of a node label assignment given the label assignments of the node's immediate neighbors (first-order MRF) until convergence. This approach has been shown be superior to other state-of-art algorithms [11, 35, 39], which currently are part of the NetKit [36]. Therefore, we pick this method as a representative of the class of graphical models.

Summary of all abbreviations denoting these classifiers is presented in Table 6.

4.5 Results for different training sizes

The most important feature of a classifier is how much training it requires to reach a certain performance. Tables 7 and 8 compare all five competitors by varying their training size. The best performing classifier is presented in bold. We train the classifiers by manually assigning the ground-truth labels to a set of randomly chosen nodes (uniformly chosen, in proportion to the cardinalities of the different node types). This process of sampling training nodes was repeated and averaged at least ten times for each measurement point. We validated the statistical significance for the superiority of Graffiti using a Student's t-test at level 0.05. We also applied the non-parametric Friedman's test using Gaussian Approximation and a Dunn's Multiple Comparison Test [44] as a post test with a significance level of 0.05. In the experiments, all gains of Graffiti are statistically significant.

Given enough training data, all classifiers can achieve high performance. The challenge is to maintain high performance with a small set of training data. Graffiti, outperforms all competitors in this regime as it takes advantage of the available cross-link structure. Topical Page Rank has difficulties interpreting the topic distribution within a type, because a simple random walk is not enough to capture the complexity of the label patterns and dependencies in heterogeneous graphs where different

Table 6 Summary of abbreviations of all compared methods.

Abbreviation	Method
NB	Naive Bayesian classifier
Graffiti	The proposed method as defined by (7)
TopicalPR	Topical Page Rank
HC	Hybrid classifier
RL	Relaxation Labeling classifier

Table 7 Performance comparison for different training sizes for the *Flickr* dataset with close to 230,000 test nodes.

System	*Flickr*			
	Tr.size	microF1	NDCG	KTau
NB	400	0.4727	0.7552	0.2646
Graffiti	400	**0.4957**	**0.7694**	**0.2880**
TopicalPR	400	0.4819	0.7612	0.2732
RL	400	0.2766	0.6686	0.1794
HC	400	0.4211	0.7255	0.2070
NB	1,000	0.5195	0.7797	0.3060
Graffiti	1,000	**0.5407**	**0.7930**	**0.3278**
TopicalPR	1,000	0.5293	0.7863	0.3160
RL	1,000	0.2830	0.6767	0.2051
HC	1,000	0.4781	0.7579	0.2685
NB	2,000	0.5480	0.7947	0.3294
Graffiti	2,000	**0.5651**	**0.8058**	**0.3473**
TopicalPR	2,000	0.5564	0.7999	0.3379
RL	2,000	0.2781	0.6781	0.2139
HC	2,000	0.5137	0.7779	0.3026
NB	4,000	0.5703	0.8064	0.3481
Graffiti	4,000	**0.5809**	**0.8140**	**0.3601**
TopicalPR	4,000	0.5744	0.8099	0.3532
RL	4,000	0.2795	0.6815	0.2254
HC	4,000	0.5409	0.7918	0.3248

Table 8 Performance comparison for different training sizes for the *LibraryThing* dataset with 38,000 test nodes.

System	*LibraryThing*			
	Tr.size	microF1	NDCG	KTau
NB	1,000	0.3766	0.6837	0.2262
Graffiti	1,000	**0.3858**	**0.6926**	**0.2362**
TopicalPR	1,000	0.3776	0.6857	0.2272
RL	1,000	0.3148	0.6401	0.1930
HC	1,000	0.3210	0.6446	0.1812
NB	2,000	0.3992	0.6534	0.2460
Graffiti	2,000	**0.4155**	**0.7153**	**0.2607**
TopicalPR	2,000	0.4063	0.7077	0.2490
RL	2,000	0.3123	0.6453	0.2035
HC	2,000	0.3293	0.7016	0.1930
NB	4,000	0.4258	0.7180	0.2640
Graffiti	4,000	**0.4422**	**0.7326**	**0.2782**
TopicalPR	4,000	0.4293	0.7227	0.2649
RL	4,000	0.3204	0.6529	0.2178
HC	4,000	0.3451	0.6637	0.2055
NB	10,000	0.4693	0.7439	0.2287
Graffiti	10,000	**0.4940**	**0.7647**	**0.3110**
TopicalPR	10,000	0.4804	0.7537	0.2946
RL	10,000	0.3343	0.6647	0.2374
HC	10,000	0.3756	0.6833	0.2287

Table 9 Micro-averaged F1 performance for the different node types in the *LibraryThing* set with training size 10,000.

System	Authors	Tags	Users	Books
	microF1	microF1	microF1	microF1
NB	0.4357	0.4310	0.3138	0.6137
Graffiti	0.4488	0.4672	0.3107	0.6168
TopicalPR	0.4293	0.4519	0.3078	0.6068
RL	0.3580	0.3399	0.2238	0.3362
HC	0.3818	0.3980	0.2918	0.3319

types of nodes have different topic distributions within their type-specific set of classes. The performance of the RL method is unexpectedly low. This is due to the extremely high average node degree in the graph and highly skewed ground truth label distribution, forming one "fat" class per node type. Both factors taken together skew the method toward a distribution where, for all X-edges, the nodes at their ends get assigned to the fat class characteristic for their type. This creates a feedback loop and the graph labeling stabilizes with every node being assigned to the fattest class for its type. In turn, this affects the break-even point where Precision meets Recall. Therefore, despite the relatively good initial classifier predictions (in this case—the NB classifier), the RL approach can not achieve high (micro-averaged F1) performance. Without thresholding to discover a "good" influential neighborhood around each node, the RL does not have a chance to realistically make use of the peculiarities of the graph structure in heterogeneous networks. On the other hand, such a thresholding is hard to define in order to prune edges across different types of nodes, as a similarity function to quantify the similarity between two nodes of different type is hard if not impossible to derive. Finally, the hybrid classifier HC performs relatively well, but fails to extract the correct class supporting scheme from a node's neighborhood, as here a node can have highly diverse neighbors, most of which do not belong to the type-specific class set as the node itself. The proposed method Graffiti is able to significantly enhance the classification result by taking advantage of the cross-type link structure in the graph.

We give further details on the classifiers' performance when evaluated separately for each node type in Tables 9 and 10. For the *flickr* data, the highest gains are obtained on the nodes of type Photo as they are best connected in the graph and have rich textual information for the NB baseline classifier. For the *librarything* data, the best results are achieved for the nodes of types Author and Tag.

Run time complexity analysis is given in Section 3.5. Typical run-times for the implementation of Graffiti were in order of 36 min, whereas NB took approximately 15 min, and TopicalPR and HC took 10 min on the LibraryThing dataset. The reader should note that the Graffiti code is a prototype version without any code

Table 10 Micro-averaged F1 performance for the different node types in the *Flickr* dataset.

System	Tr.Size	Photos	Tags	Users
		microF1	microF1	microF1
NB	1,000	0.6894	0.3126	0.5897
Graffiti	1,000	0.7279	0.3169	0.5857
TopicalPR	1,000	0.7058	0.3165	0.5854
RL	1,000	0.2636	0.3031	0.3394
HC	1,000	0.6155	0.3103	0.5456

Table 11 Per class results for nodes of type Photo.

System	Tr.size	c2	c5	c8	c11	c14
Precision						
NB	1,000	0.740	0.656	0.825	0.718	0.632
Graffiti	1,000	0.786	0.680	0.877	0.760	0.674
TopPR	1,000	0.764	0.660	0.851	0.740	0.651
RL	1,000	0.122	0.599	0.925	0.262	0.592
HC	1,000	0.684	0.525	0.916	0.745	0.696
Recall						
NB	1,000	0.724	0.849	0.481	0.759	0.568
Graffiti	1,000	0.801	0.901	0.470	0.813	0.578
TopPR	1,000	0.765	0.880	0.465	0.783	0.562
RL	1,000	0.004	0.464	0.049	0.602	0.122
HC	1,000	0.709	0.899	0.323	0.612	0.431

optimizations discussed in Section 4.9. The experiments were run on a machine with Intel Xeon CPU 2.80 GHz and 8 GB RAM.

4.6 Breakdown per type and class

We present the precision and recall values for all classes, specific for the *flickr* type Photo, in Table 11. Graffiti performs consistently better than the baseline classifier and superior to all methods in terms of statistically significant precision/recall gains.

The results obtained on the set of nodes of type Tag and User are comparable to the performance of the baseline method on the *flickr* dataset. We omit these class breakdown for lack of space. In the experiments on the *LibraryThing* dataset, significant improvements are obtained for node types Author and Tag. The types User and Book are comparable to the baseline method. In all experiments, Graffiti significantly improves the classification result for the bigger node types. The small ones, like User, contain few nodes and have almost uniformly distributed connections to classes of other node types. Therefore, for these node types Graffiti's performance is comparable to the baseline method.

4.7 Sensitivity of tunable parameters

We varied the tunable parameters α and β of the Graffiti method and show in Table 12 its micro-averaged F1 performance for a training set of 1,000 nodes and test set of 230,000 nodes. For general recommendations, broader studies and systematic tuning procedures are needed. However, the preferred way of finding the optimal parameters given data at hand is to use the well established n-fold cross validation [37]. This commonly used technique takes a set of m examples and

Table 12 Parameter sensitivity of Graffiti.

Micro-averaged F1 results on the *Flickr* dataset.

α	β				
	0.0	0.2	0.5	0.8	1.0
0.0	0.5342	0.5342	0.5342	0.5342	0.5342
0.2	0.5372	0.5366	0.5357	0.5345	0.5338
0.5	0.5395	0.5380	0.5353	0.5323	0.5297
0.8	0.5407	0.5383	0.5345	0.5293	0.5246
1.0	0.5407	0.5383	0.5339	0.5259	0.5200

Table 13 Micro-averaged F1 results on the *LibraryThing* dataset simulating lack of textual content for some percentage of the graph.

System	Percent p			
	5%	15%	25%	50%
NB	0.437	0.385	0.365	0.291
Graffiti	0.483	0.486	0.487	0.496
TopicalPR	0.464	0.470	0.474	0.486
RL	0.330	0.328	0.326	0.323
HC	0.350	0.318	0.304	0.252

The evaluation is done over all nodes that belong to the graph, irrespective of whether they had uniform or content-dependent class distribution initialization.

partitions them into n sets (“folds”) of size m/n. For each fold, a classifier is trained on the other folds and then tested on the fold. By applying this technique while varying parameters, the optimal set of parameters will be the one resulting in best performance of the classifier.

4.8 Sensitivity to initialization

Heterogeneous graphs can contain a wide range of node types. Many of the graph nodes might contain poor textual information or none at all. This will inevitably affect the initial class label distribution for these nodes. Therefore, we performed an experiment simulating lack of content for some percent p of the graph nodes. That is, p percent of all graph nodes were picked uniformly at random and were set to have a uniform initial class distribution vector λ. In the experiment we evaluated the performance of each of the algorithms on all graph nodes, irrespective of their initialization. These results are shown in Table 13. Then we looked only at the portion of the graph that contains all nodes initialized with a uniform class label distribution. The results of this evaluation are shown in Table 14. The training data size used in this experiment is 1,000. Test data size is 38,000 nodes.

In this experiment, classifiers relying on textual information such as NB and HC naturally fail to give good performance results with increase of the number of perturbed nodes. The hybrid classifier HC performs equally poor as its propagation scheme lacks reliable information onto which to base its decision for the most likely label per node. The relaxation labeling approach RL does not degrade its performance but its propagation scheme is not powerful enough to deliver good

Table 14 Micro-averaged F1 results on the *LibraryThing* dataset simulating lack of textual content for some percent of the graph p.

System	Percent p			
	5%	15%	25%	50%
NB	0.165	0.162	0.164	0.163
Graffiti	0.486	0.492	0.494	0.504
TopicalPR	0.476	0.485	0.489	0.500
RL	0.319	0.317	0.318	0.317
HC	0.165	0.163	0.164	0.164

The evaluation is done exclusively over the uniformly initialized set of nodes.

results. On the other hand, Topical PageRank (TopicalPR) shows good performance. However, Graffiti outperforms all competitors. With the growth of the percentage of nodes that are initialized with a uniform label probability, Graffiti's gains over Topical Page rank become smaller.

4.9 Discussion

Many factors influence the performance of a classifier: the size of the available training data, the quality of the content and link information, the relationships among the graph nodes that in turn dictate the dominant type of edges in the graph (e.g. cross-type edges vs. same-type edges) and the nature of label propagation schemes that can capture their mutual influence. Therefore, it is not possible to derive an upper bound on the best possible performance of a classification algorithm in a real-world heterogeneous environment. This limits our ability to realistically evaluate the gains of the competing algorithms.

The proposed classification algorithm, Graffiti, outperforms all state-of-the-art techniques for graph-based classification. We hypothesize that the more node types in the graph, the better the performance of Graffiti. It leverages the influence between same-type nodes connected trough a path of cross-type edges of *any length* by combining one- and two-hop walks. At the same time, the algorithm *avoids* exhaustive *bookkeeping* of propagation results related to neighbor labeling combinations that could influence a given node but come from arbitrary long paths and node types. It also incorporates same-type edge connections into the label propagation scheme.

4.10 Future work

As obtaining rich heterogeneous data is extremely hard if not impossible due to privacy issues, a challenging direction for future research would be to synthetically generate such data. This will facilitate the experimental evaluation of all heterogeneous graph classification algorithms under a realistic stress test case of increasing number of distinct node types in the graph.

Representing a heterogeneous network in its full generality also poses open research issues. Intuitively, the relations in the network are not binary but N-ary relations. They can model different properties for an individual relation instance as well as represent provenance information, e.g., information about the creator and the time of creation of the relation, certainty about the relation, strength of the relation, and so on. For example, the social network Flickr can be modeled as a hypergraph where nodes represent users, data items such as photos, and posted tags or comments. Hyperedges, connecting users, photos, and tags, represent the diverse information associated with any event, e.g., a user u posted photo p and annotated it with tag t. In the hyperlink graph such ternary relation can be expressed as a single hyperedge $(u; p; t)$. In contrast, if the network is modeled as a graph, then the event will be decomposed into two independent binary relations: userphoto and photo-tag, and the additional knowledge that these were interleaved will be lost. The binary relation assumption is typically made for tractability, scalability and noise reduction concerns. However, the most natural representation of these networks are hypergraphs. Current approaches dealing with hypergraphs are still in their infancy.

5 Conclusion

We have presented a principled approach to the challenging problem of classifying nodes in heterogeneous graphs, with several object types linked in a variety of ways. The model we propose is able to successfully leverage the prominent influence of cross-type links, as a powerful tool to facilitate class inference across different type nodes. Using the metaphor of a random surfer, walking through the relational graph, aiming at painting its nodes as a giant Graffiti in a flowing tinge sequence, we presented the *Graffiti* algorithm and its mathematical properties (unique solution, convergence rate). Our experiments on two real-life social networks *flickr* and *LibraryThing* provide evidence for the superiority of the proposed method.

References

1. Angelova, R., Weikum, G.: Graph-based text classification: learn from your neighbors. In: SIGIR ’06: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, New York (2006)
2. Angelova, R., Kasneci, G., Suchanek, F.M., Weikum, G.: Graffiti: node labeling in heterogeneous networks. In: WWW ’09: Proceedings of the 18th International Conference on World Wide Web. ACM, New York (2009)
3. Baeza-Yates, R.A., Boldi, P., Castillo, C.: Generalizing pagerank: damping functions for link-based ranking algorithms. In: SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 308–315. ACM, New York (2006)
4. Bartal, Y.: Probabilistic approximation of metric spaces and its algorithmic applications. In: Proceedings of the 37th IEEE Symposium on Foundations of Computer Science, pp. 184–193. IEEE, Piscataway (1996)
5. Berkhin, P.: Bookmark-coloring algorithm for personalized pagerank computing. Journal of Internet Mathematics **3**(1), 41–46 (2006)
6. Bharat, K., Henzinger, M.R.: Improved algorithms for topic distillation in a hyperlinked environment. In: SIGIR 1998: Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, New York (1998)
7. Blum, A., Chawla, S.: Learning from labeled and unlabeled data using graph mincuts. In: ICML: Proceedings of the 18th International Conference on Machine Learning, pp. 19–26. ICML (2001)
8. Blum, A., Lafferty, J.D., Rwebangira, M.R., Reddy, R.: Semi-supervised learning using randomized mincuts. In: ICML: Proceedings of the 21st International Conference on Machine Learning, pp. 97–104. ICML (2004)
9. Boldi, P., Vigna, S.: The webgraph framework I: compression techniques. In: Proceedings of the 18th International Conference on World Wide Web, pp. 595–601. WWW (2004)
10. Breslin, J.G., Passant, A., Decker, S.: The Social Semantic Web. Springer, New York (2009)
11. Chakrabarti, S., Dom, B., Indyk, P.: Enhanced hypertext categorization using hyperlinks. In: SIGMOD ’98: Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data. ACM, New York (1998)
12. Cohn, D., Hofmann, T.: The missing link—a probabilistic model of document content and hypertext connectivity. In: Neural Information Processing Systems 13 (2001)
13. Dhillon, I.S., Mallela, S., Modha, D.S.: Information-theoretic co-clustering. In: KDD: Proceedings of The Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, New York (2003)
14. Feldman, R., Shatkay, H.: Link analysis for bioinformatics: current state of the art. In: Pacific Symposium on Biocomputing. PSB (2003)
15. Feller, W.: An Introduction to Probability Theory and its Applications, 3rd edn. Wiley, New York (1968)

16. Gallagher, B., Tong, H., Eliassi-Rad, T., Faloutsos, C.: Using ghost edges for classification in sparsely labeled networks. In: KDD '08: Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM (2008)
17. Gao, B., Liu, T.-Y., Ma, W.-Y.: Star-structured high-order heterogeneous data co-clustering based on consistent information theory. In: ICDM '06: Proceedings of the 6th International Conference on Data Mining. IEEE Computer Society, Los Alamitos (2006)
18. Getoor, L.: Link mining: a new data mining challenge. SIGKDD Explor. Newsl. **5**(1), 84–89 (2003)
19. Getoor, L., Diehl, C.P.: Link mining: a survey. SIGKDD Explor. Newsl. **7**(2), 3–12 (2005)
20. Getoor, L., Taskar, B.: Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning). MIT Press, Cambridge (2007)
21. Haggstrom, O.: Finite markov chains and algorithmic applications. In: London Mathematical Society Student Texts. Cambridge University Press, Cambridge (2001)
22. Harshman, R.A.: Foundations of the parafac procedure: models and conditions for an explanatory multi-modal factor analysis. In: UCLA Working Papers in Phonetics, UMI Serials in Microform, pp. 1–84 (1970)
23. Haveliwala, T., Kamvar, S.: The Second Eigenvalue of the Google Matrix. Stanford University Technical Report (2003)
24. Haveliwala, T.H.: Topic-sensitive pagerank. In: WWW: Proceedings of the 11th International World Wide Web Conference. WWW (2002)
25. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of ir techniques. ACM Trans. Inf. Syst. Secur. (TISSEC) **20**(4), 422–446 (2002)
26. Jensen, D., Neville, J., Gallagher, B.: Why collective inference improves relational classification. In: ACM KDD: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2004)
27. Jensen, F.V.: Bayesian Networks and Decision Graphs. Springer, Secaucus (2001)
28. Joachims, T.: Transductive inference for text classification using support vector machines. In ICML: Proceedings of the 16th International Conference on Machine Learning, ICML. Morgan Kaufmann, San Mateo (1999)
29. Johnson, J.K., Bickson, D., Dolev, D.: Fixing convergence of Gaussian belief propagation. In: Proceedings of the 2009 IEEE International Conference on Symposium on Information Theory - Volume 3 (ISIT'09), vol. 3, pp. 1674–1678. IEEE Press, Piscataway (2009)
30. Kleinberg, J., Tardos, E.: Approximation algorithms for classification problems with pairwise relationships: metric labeling and markov random fields. In: FOCS: Proceedings of the 40th Annual Symposium on Foundations of Computer Science. IEEE Computer Society, Los Alamitos (1999)
31. Kolda, T.G., Bader, B.W., Kenny, J.P.: Higher-order web link analysis using multilinear algebra. In ICDM: Proceedings of the 5th IEEE International Conference on Data Mining, pp. 242–249 (2005)
32. Langville, A.N., Meyer, C.D.: Google's PageRank and Beyond. Princeton University Press, Princeton (2006)
33. Langville, A.N., Meyer, C.D.: Google's PageRank and Beyond: The Science of Search Engine Rankings. Princeton University Press, Princeton (2006)
34. Lin, J., Schatz, M.: Design patterns for efficient graph algorithms in MapReduce. In: Proceedings of the 2010 Workshop on Mining and Learning with Graphs Workshop (MLG-2010) (2010)
35. Lu, Q., Getoor, L.: Link-based classification. In: ICML, Proceedings of the Twentieth International Conference on Machine Learning. ICML (2003)
36. Macskassy, S.A., Macskassy, S.A., Macskassy, S.A., Provost, F., Provost, F.: Netkit-srl: a toolkit for network learning and inference. In: NAACSOS: Proceedings of the Annual Conference of the North American Association for Computational Social and Organizational Science (2005)
37. Nadeau, C., Bengio, Y.: Inference for the generalization error. J. Mach. Learn. **52**(3), 239–281 (2003)
38. Nie, L., Davison, B.D., Qi, X.: Topical link analysis for web search. In: SIGIR: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development on Information Retrieval, pp. 91–98. ACM, New York (2006)
39. Oh, H.-J., Myaeng, S.H., Lee, M.-H.: A practical hypertext catergorization method using links and incrementally available class information. In: SIGIR: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, New York (2000)

40. Page, L., Brin, S., Motwani, R., Winograd, T.: The Pagerank Citation Ranking: Bringing Order to the Web. Tech. rep., Stanford Digital Library Technologies Project (1998)
41. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Mateo (1988)
42. Richardson, M., Domingos, P.: The intelligent surfer: probabilistic combination of link and content information in PageRank. In: NIPS: Advances in Neural Information Processing Systems 14. MIT Press, Cambridge (2002)
43. Sen, P., Namata, G.M., Bilgic, M., Getoor, L., Gallagher, B., Eliassi-Rad, T.: Collective Classification in Network Data. Tech. Rep. CS-TR-4905, University of Maryland, College Park (2008)
44. Sheskin, D.: Handbook of Parametric and Nonparametric Statistical Procedures. CRC Press, Boca Raton (2007)
45. Shrager, J., Hogg, T., Huberman, B.A.: Observation of phase transitions in spreading activation networks. Science **236**, 1092–1094 (1987)
46. Stewart, W.: Introduction to the Numerical Solution of Markov Chains. Princeton University Press, Princeton (1994)
47. Wang, F., Zhang, C.: Label propagation through linear neighborhoods. In: ICML: Machine Learning, Proceedings of the Twenty-Third International Conference. ICML (2006)
48. Wang, X., Sun, J.-T., Chen, Z., Zhai, C.: Latent semantic analysis for multiple-type interrelated data objects. In: SIGIR: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, New York (2006)
49. Washio, T., Motoda, H.: State of the art of graph-based data mining. SIGKDD Explor. Newsl. **5**(1), 59–68 (2003)
50. Wu, T.-F., Lin, C.-J., Weng, R.C.: Probability estimates for multi-class classification by pairwise coupling. J. Mach. Learn. Res. **5**, 975–1005 (2004)
51. Yang, Y., Xu, D., Nie, F., Luo, J., Zhuang, Y.: Ranking with local regression and global alignment for cross media retrieval. In: MM: Proceedings of the 17th ACM International Conference on Multimedia, pp. 175–184. ACM, New York (2009)
52. Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Schölkopf, B.: Learning with local and global consistency. In: Advances in Neural Information Processing Systems, vol. 16, pp. 321–328 (2004)
53. Zhou, D., Weston, J., Gretton, A., Bousquet, O., Schölkopf, B.: Ranking on data manifolds. In: Proceedings of the 16th Conference on Advances in Neural Information Processing Systems, vol. 16, pp. 169–176 (2004)