Learning Latent Representations of Nodes for Classifying in Heterogeneous Social Networks

Yann Jacob, Ludovic Denoyer, Patrick Gallinari

Sorbonne Universites, UPMC Univ Paris 06, UMR 7606, LIP6, F-75005, Paris, France
CNRS, UMR 7606, LIP6, F-75005, Paris, France

firstname.lastname@lip6.fr

ABSTRACT

Social networks are heterogeneous systems composed of different types of nodes (e.g. users, content, groups, etc.) and relations (e.g. social or similarity relations). While learning and performing inference on homogeneous networks have motivated a large amount of research, few work exists on heterogeneous networks and there are open and challenging issues for existing methods that were previously developed for homogeneous networks. We address here the specific problem of nodes classification and tagging in heterogeneous social networks, where different types of nodes are considered, each type with its own label or tag set. We propose a new method for learning node representations onto a latent space, common to all the different node types. Inference is then performed in this latent space. In this framework, two nodes connected in the network will tend to share similar representations regardless of their types. This allows bypassing limitations of the methods based on direct extensions of homogenous frameworks and exploiting the dependencies and correlations between the different node types. The proposed method is tested on two representative datasets and compared to state-of-the-art methods and to baselines.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Learning; E.1 [**Data**]: Data Structures—*Graphs and networks*

Keywords

Machine learning; Classification; Social networks

1. INTRODUCTION

Social Media on the Web are most often complex heterogeneous networks with nodes and relations between nodes of different types, corresponding to different objects, concepts and relationships. Classical examples include user content networks like Flickr where nodes correspond to users, photos, comments, tags, and relations to friendship or authorship, or bibliographic networks like DBLP with nodes corresponding to authors, papers, venues, terms and relations to citations, co-authorship, proceedings, contained in, etc. Heterogeneous networks also frequently occur in other domains like e-commerce sites with users, items, comments and reviews, or recommendation sites such as TripAdvisor. Modeling and analyzing such complex networks has recently been explored for generic data mining tasks such as classification [2], clustering [22], link prediction [5] or influence analysis [16] and is still largely an open area.

WSDM'14, February 24–28, 2014, New York, New York, USA.

ACM 978-1-4503-2351-2/14/02 ...$15.00.
http://dx.doi.org/10.1145/2556195.2556225.

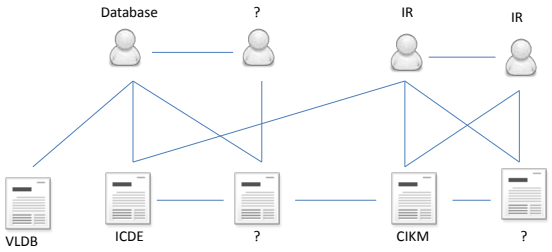


Figure 1: Bibliographic heterogeneous network with authors and articles connected. Authors are labeled with topic names (IR = Information Retrieval), while articles are labeled with conference venue names.

In this article, we consider the task of node classification in heterogeneous networks composed of different types of nodes, each node type beging associated with its own set of labels. For instance, in a bibliographic network, authors and papers might be labeled respectively by their research topic and conference name (see Figure 1), while in the Flickr, network users and photos might be respectively labeled by subscribed groups reflecting their topic interests and visual tags. Much work has been devoted to classification for homogeneous networks composed of a single node type, e.g. [3, 17, 19, 27] to cite a few. Classification in heterogeneous networks, representative of real world media, is much more recent with only a few attempts for now. Many of them rely on the idea of mapping an heterogeneous network onto an homogeneous network so that classical relational techniques are then used [8, 10, 14, 2]. They do not fully exploit the correlations between the different node labels or characteristics. Moreover, many of these methods also make strong

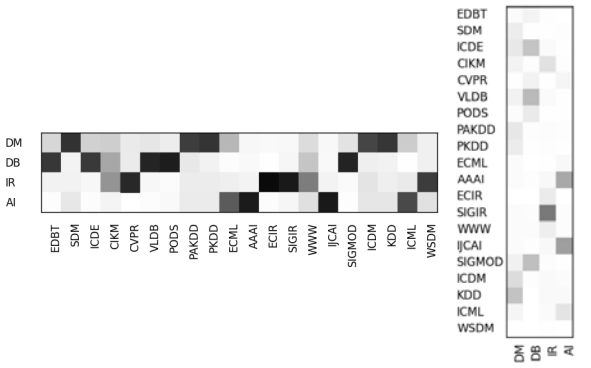


Figure 2: Two matrices illustrating the inter-dependencies between labels of connected nodes of different types for DBLP data. Left matrix corresponds to $P(Author-topic|Article-conference)$ while right matrix illustrates $P(Article-conference|Author-topic)$. The darker the square is, the higher the probability value. Both matrices show strong correlations between the *topics* of an author and the *conference* names.

hypothesis on the nature of the task that is not realistic for real world situations.

The assumption we make in the paper is **that nodes of different types do influence each other**, so that labels of the different node types will be inter-dependent; modeling this dependence between different node types is thus important for accurate node classification or labeling and cannot be achieved via classical homogeneous network formulations. For example, in a Flickr network, the groups a user belongs to will be related to the tags the user has been using or the comments he has posted and vice versa. This assumption is illustrated in Figure 2 which shows some statistics computed over a heterogeneous bibliographic network extracted from DBLP. The DBLP graph is composed of authors and scientific articles connected through an *authorship* relation[1]. Authors are labeled with 4 different topics — *IR = Information retrieval, DB = Database, DM = Data Mining, AI = Artificial Intelligence* — while articles are labeled with 20 possible conferences they have been published in. The left side of Figures 2 (left) shows the values of the probability $P(Author-topic|Article-conference)$ i.e. the probability that an author node has the label *topic* given that one of its neighboring article nodes has the label $conference$. Similarly, the right side of Figure 2 (right) illustrates the probability $P(Article-conference|Author-topic)$. Figure 2 exhibits strong correlations between the two label sets and confirms that the correlation between the labels of different nodes may be used for classification.

We propose a new algorithm for learning to label nodes in an heterogeneous network to exploit dependencies on label sets and node characteristics and relationships between different node types. The proposed algorithm operates in the transductive setting and makes use of a regularization framework. It can be used for labeling nodes of different types with different label sets, and with any graph structure.

[1]The dataset is described in Section 4.1.

It is designed to learn the dependencies between the label sets associated to the different node types, and to infer the labels associated to a node by exploiting both global graph properties and local node neighborhood characteristics.

It is important to mention that the proposed algorithm learns a latent representation of the network nodes so that all nodes, irrespectively of their type, will share a common latent space. Such representation will then be effectively used to infer the categories. Unlike unsupervised relational data projection methods [23, 4], our method is semi-supervised so that the projection space exploits both the relational structure and the labeled information. In other words, the algorithm makes use of the heterogeneous training labels to constrain the projection, by encouraging nodes with the same labels to have close latent representations. Moreover, it takes the correlations between the labels of different node types into account when learning the latent representation. Consequently, the model learns simultaneously a metric associated to the heterogeneous classification problem and a way to label the nodes given this metric space. Additionally, the same formulation can be naturally extended to modeling node attributes, such as content information associated to the nodes in our context, by simply considering these attributes as specific nodes.

The main contributions are: (i) (i) an effective method for mapping nodes represenntions onto a common lateent space: This mapping exploits the dependencies on the node label sets, and relationships between these labels sets and the node characteristics. This new method allows a simple formulation of the classication problem in an heterogeneous setting; (ii) An algorithm to solve the general classication problem in heterogeneous networks under a general assumption about the node classes or the relations between nodes. In particular it can handle the challenging case of non intersecting label sets between the different node types. All the dependencies are learned directly from the data without necessitating a user-defined or a heuristic pre-processing procedure to perform graph projection.

The paper is organized as follows: Section 2 presents the notations, and introduces some background concerning classical graph labeling models for homogeneous networks. Section 3 describes the proposed algorithm and an extension for incorporating node content information to the model. Section 4 presents experimental results on two datasets, and a qualitative analysis of the latent representations learned by the model. Section 5 provides a synthesis of the related literature.

2. BACKGROUND AND NOTATIONS

2.1 Notations

A heterogeneous network will be modeled as an undirected weighted graph with a node type associated to each node.

Let us denote by $\mathcal{T} = 1, 2, \ldots, T$ the set of T possible nodes types. Nodes are denoted x_i, and edges $w_{i,j} \in \mathbb{R}$, where $w_{i,j}$ is the weight of the relation between x_i and x_j. $w_{i,j} = 0$ if there is no link between x_i and x_j, the number of nodes is N. We will denote $t_i \in \mathcal{T}$ the node type x_i, $\mathcal{Y}^t$ the set of categories associated to the type of node t, and C^t the cardinality of $\mathcal{Y}^t$. We consider a transductive context where the network is composed of $\ell < N$ labeled nodes $(x_1, \ldots, x_\ell)$ with $\forall i \in \{1 \ldots \ell\}, y_i \in \mathbb{R}^{C^{t_i}}$, and $N - l$ unlabeled nodes.

The component j of y_i, denoted by y_i^j, is defined as $y_i^j = +1$ if x_i belongs to category j, and $y_i^j = -1$ otherwise.

2.2 Homogeneous Label Propagation Model

When there is only one type of node i.e. $T = 1$, classical transductive models exploit the manifold assumption where *two connected nodes tend to have the same labels*. Let us denote by $\hat{y}_i$ the vector of scores for the categories predicted at any node x_i in the network. Transductive learning in graph models often uses a loss function with the following form [1, 27, 28]:

$$(\hat{y_1}, \ldots, \hat{y_N}) = \underset{\tilde{y_1},\ldots,\tilde{y_N}}{\operatorname{argmin}} \sum_{i=1}^{\ell} \Delta(\tilde{y_i}, y_i) + \lambda \sum_{i,j} w_{i,j} ||\tilde{y_i} - \tilde{y_j}||^2 \quad (1)$$

where Δ corresponds to the cost of predicting scores $\tilde{y_i}$ instead of true categories y_i for labeled nodes and $||\tilde{y_i} - \tilde{y_j}||^2$ is a regularization term that encourages connected nodes to have the same score. λ is an hyper-parameter that corresponds to the smoothness one wants to obtain on the network structure. Many variations of this formulation have been proposed with different prediction and regularization losses. Different techniques can be used to minimize this function such as plain or stochastic gradient-descent algorithm, coordinate descent methods and random walks.

2.3 Extending the homogeneous setting to heterogeneous networks

The homogeneous framework can be extended in multiple ways to handle the case of heterogeneous networks [11, 8, 2]. Our claim is that such extensions do not allow one to fully benefit from the information present in the heterogeneous network and in particular from the correlations and dependencies between the different node labels.

As an illustration, let us consider the case of a graph composed of two types of nodes 1 and 2 with label sets $\mathcal{Y}^1$ and $\mathcal{Y}^2$ — see Figure 3 (a). It is not possible to make a direct propagation of labels from $\mathcal{Y}^1$ to $\mathcal{Y}^2$ since the two label sets are different and correspond to different label semantics. A common solution is to map the heterogeneous graph onto one or more homogeneous graphs as illustrated in Figure 3 (b) and (c) and to perform label propagation on the different graphs. This mapping may be defined either explicitly through graph projections or implicitly through the algorithm like in [2, 14] for example. It has two drawbacks: (i) Dependencies between the two types of labels are lost, since the labels predicted for one node are only dependent of the labels of its neighbors **of the same type**. For example, a relation like *users labeled as "old" are connected to pictures labeled as "flowers"* cannot be preserved by such projection. The more node types there are, the more this loss of information is important. (ii) Extending this idea to more complex graphs could be problematic, and the corresponding semantic can become complex. Consider for example the case of multiple (more than 2) node types. There are multiple ways to define such mappings, paths joining two nodes of the same type could be of different lengths with different node types on the path. What could then be the semantic of a link between two nodes of a given type? How to weight these links?

These are main reasons why such extensions are often limited by simplifying assumptions on the nature of the graphs, the label sets or the relations. A simple example is provided in Figure 3 (c), where there are multiple possible projections to an homogeneous graph for the photos. One can connect the photos published by the same user (giving a graph with multiple components). Or one can connect the photos published by the same user or published by friend users. Therefore, different homogeneous graphs can be obtained by projection for nodes of the same type. We promote here that the dependencies might be learned in an unified way and directly from the data under general and non restrictive hypothesis. This is made possible by a change of paradigm w.r.t. conventional approaches which is implemented here by using a latent representation space common to all the node types in the network.

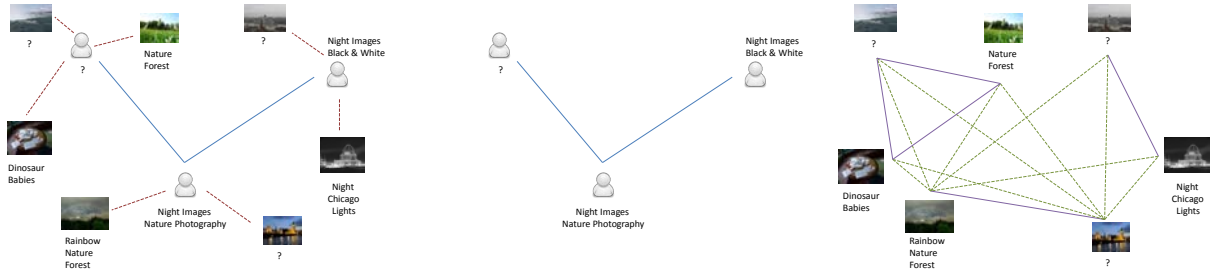
3. MODEL

3.1 Learning latent node representations

We first introduce the method for the case where no attribute (content) is associated to graph nodes and then describe in Section 3.2 a natural extension for incorporating node content. The underlying ideas of the method are as follows:

- Each node will be mapped onto a latent representation in a vectorial space $\mathbb{R}^Z$. The latent space is common to all node types.
- The new metric developed on this latent representation space $\mathbb{R}^Z$ is such that two connected nodes will tend to have similar representations (*this is an extension of the smoothness assumption to heterogeneous networks*).
- For each node type, a classification function will be learned. It takes as input a latent node representation and computes associated class label. A linear function will be used here to ensure smoothness in the classification decision while avoiding overfitting the new space.

The latent projection space is learned according to two constraints: smoothness and class separability of the projections. If one considers nodes of the same type, the smoothness constraint tends to group nodes that are linked by a path in the graph, whatever the nature of the path is (homogeneous or heterogeneous). The strength of this constraint decreases with the length of the path. This is an extension of the classical smoothness hypothesis on homogeneous graphs. On the other hand, if one considers nodes of different types, their projections will be close if they share multiple connections in the graph. This will happen for example if the classes in two node types are correlated (see for example Figure 5). This is also how dependencies or correlations between the different label sets are captured. Besides smoothness, the separability imposes an additional constraint on the latent projections. It is defined on the labeled nodes, but also operates on the unlabeled nodes representations through the diffusion mechanism. An alternative approach would be to learn a separate latent space for every pair of node types. This would be, however, too costly and not allow exploiting multiple dependencies between label sets. Note that exploiting node types pairs dependencies has been proposed by some authors [8, 11] for the case where all types of nodes share the same labels set.



(a) Original Heterogeneous graph

(b) Homogeneous graph of users

(c) Homogeneous graph of photos built using two different projection methods: connecting photos published by the same author (purple relation) and connecting photos published by friend authors (green relation)

Figure 3: (a) An heterogeneous network that can be mapped to different homogeneous networks (b), (c). Using Homogeneous models for classification of such a network is not trivial since several projection methods exist as shown in (c).

3.1.1 Transductive classification model

Let us denote by $z_i \in \mathbb{R}^Z$ the latent representation of node x_i. In order to capture the graph metric in the latent space, we impose the following loss which embodies the metric smoothness constraint and forces correlated nodes to have similar representations:

$$\sum_{i,j/w_{i,j}\neq 0} w_{i,j}||z_i - z_j||^2 \quad (2)$$

We are using an L_2 norm in the latent space, but other metrics could be used as well. This term is similar to the smoothness term for Equation 1 except that it is defined in the latent space and not in the label space as in Equation 1. The mapping onto the latent space is learned so that the labels for each type of node can be predicted from the latent representations. For that, we consider a linear classification function for each type of node k denoted by f_θ^k. This function takes as input a node representation and outputs the predicted label(s). The f-functions can be learned by minimizing a loss on labeled data as follows:

$$\sum_{i=1}^{\ell} \Delta(f_\theta^{t_i}(z_i), y_i) \quad (3)$$

where $\Delta(f_\theta^{t_i}(z_i), y_i)$ is the loss of predicting labels $f_\theta^{t_i}(z_i)$ instead of observed labels y_i. The final objective loss of our model combines the classification and regularization losses 3 and 2:

$$L(z,\theta) = \sum_{i=1}^{\ell} \Delta(f_\theta^{t_i}(z_i), y_i) + \lambda \sum_{i,j/w_{i,j}\neq 0} w_{i,j}||z_i - z_j||^2 \quad (4)$$

The minimization of this loss is performed both on the θ parameters and on the z_i representations. It aims at finding a trade-off between the smoothness over the latent representations of correlated nodes $\mathcal{Z}$ and the predicted observed labels in $\mathcal{Y}_k$. Optimizing this loss will allow us to learn:

- The projection z_i of each node x_i in the latent space.
- The classification functions f_θ^k for each nodes type k which transforms the latent space to category scores.

3.1.2 Learning

Learning consists in minimizing the loss function defined in Equation 4. Different optimization methods can be considered. We have used a Stochastic Gradient Descent Method. The algorithm is detailed in Algorithm 1.

Algorithm 1 Stochastic Gradient Descent Algorithm

1: **procedure** LEARNING(x, w, ϵ, λ)
2: **for** A fixed number of iterations **do**
3: Choose a (x_i, x_j) at random with $w_{i,j} > 0$.
4: **if** $i \leq \ell$ **then** ▷ if x_i is labeled
5: $\theta \leftarrow \theta + \epsilon\nabla_\theta \Delta(f_\theta^{t_i}(z_i), y_i)$
6: $z_i \leftarrow z_i + \epsilon\nabla_{z_i} \Delta(f_\theta^{t_i}(z_i), y_i)$
7: **end if**
8: **if** $j \leq \ell$ **then** ▷ if x_j is labeled
9: $\theta \leftarrow \theta + \epsilon\nabla_\theta \Delta(f_\theta^{t_j}(z_j), y_j)$
10: $z_j \leftarrow z_j + \epsilon\nabla_{z_j} \Delta(f_\theta^{t_j}(z_j), y_j)$
11: **end if**
12: $z_i \leftarrow z_i + \epsilon\lambda\nabla_{z_i}||z_i - z_j||^2$
13: $z_j \leftarrow z_j + \epsilon\lambda\nabla_{z_j}||z_i - z_j||^2$
14: **end for**
15: **end procedure**

The algorithm chooses iteratively a pair of connected nodes and then makes a gradient update over the parameters of the model. If one of the chosen nodes is labeled, the algorithm first performs an update according to the first term of Equation 3. This update – lines 5-6 and 9-10 – consists in successively modifying the parameters of the classification function θ and of the latent representations z_i and z_j so as to minimize the classification loss term. Note that this step not only modifies the classifier parameter θ, but also the node representation z. Then, the model updates node representation z w.r.t the smoothness term of Equation 2 – lines 12-13. If the node is unlabeled, only the later step will be performed. Nevertheless, the classification constraint will propagate to these unlabeled nodes via their connection to labeled ones.

Note that, while we use a stochastic gradient descent, other methods like minibatch gradients or batch algorithms could be used as well. Here, ϵ is the gradient step, and λ

is the trade-off between the classification and smoothness terms.

In our implementation, we have used an hinge-loss function for Δ:

$$\Delta(f_\theta^t(z), y) = \sum_{k=1} max(0; 1 - y^k f_\theta^{t,k}(z)) \quad (5)$$

where y^k is the desired score of category k for node x (-1 or $+1$) and $f_\theta^{t,k}(z)$ is the predicted score of category k by the model. Here again other loss functions, e.g. logistic could be used.

3.1.3 Complexity

At each step of the algorithm, one has to make two updates over the latent representation – lines 12-13 – and, zero, one or two additional updates if the chosen nodes are labeled. These updates have to be repeated several times, over all pairs of connected nodes. Let us denote by E the number of edges in the network, the complexity algorithm is thus $\mathcal{O}(E + \ell)$. The learning time thus scales linearly with regard to the number of edges which is the same complexity than classical homogeneous models.

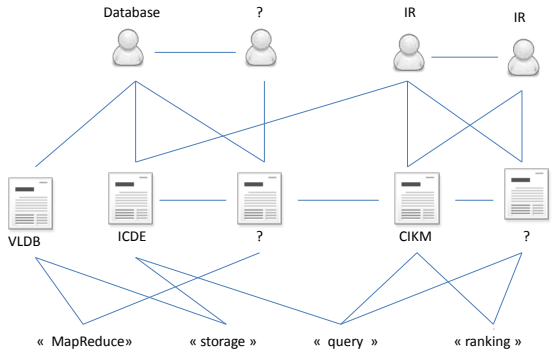


Figure 4: Bibliographic heterogeneous network with content modeled as word nodes.

3.2 Heterogeneous model and content

The objective function in Equation 4 only considers the graph structure through its weight matrix and node labels. Often, attributes are associated to nodes (e.g. textual content for comments or visual content for images). Let us consider as an example the case of textual attributes (the description is similar for any other discrete set of attributes or features associated to a node type). The above formulation allows to easily integrate this information into the model by associating each term to a new node like in [11] and a weighted link between this term and any node containing it. The weight of the relation may be defined according to any relevant measure (e.g. word frequency).

This solution is illustrated in Figure 4. In that case, the model jointly learns a latent representation of all the elements of the network – users, documents, words. Documents and their content words will then tend to have close latent representations, so will words appearing in the same documents or documents sharing similar words, even if they are not directly connected in the network. Nodes associated to words are special in the sense that they do not have associated labels and that in most real applications they do not share direct connections - except if one makes use of some term ontology for example, in which case this information could again be naturally incorporated in the model. Their representation does not appear in the classification loss term in Equation 4, but only in the smoothing term.

4. EXPERIMENTS

4.1 Datasets

Experiments have been performed on two datasets respectively extracted from DBLP and Flickr. For the former, the task is monolabel classification, and for the latter multilabel classification. For DBLP two families of experiments - without and with content words have been performed. They are respectively denoted by *DBLP No Content* and *DBLP With Content* hereafter. For the Flickr experiments, no additional node characteristics has been considered. The two datasets are introduced below.

The **DBLP** dataset is a bibliographic network composed of authors and papers. This dataset was published in [22], where the task was heterogeneous classification with the authors and papers sharing the same set of labels (their research domain). The task studied here considers two different sets of labels: authors are labeled with their research domain (4 different domains) while papers are labeled with the name of the conference where they were published (20 labels). Authors and papers are connected through a *authorship* relation. The version without content is composed of two types of nodes and the graph is bipartite with one relation type. The version with content has three types of nodes – authors, papers and words – where each paper is connected to the nodes of its title – as shown in Figure 4. Classification is monolabel on papers and authors. General statistics about this corpus are given in Table 1.

The **Flickr** corpus is a dataset of photos and users. The photo labels correspond to different possible tags while the user labels are their subscribed groups. The classification problem is multi-label: images and users may belong to more than one category. Photos are related to users through an *authorship* relation, while users are related to other users through a *friendship* relation. This dataset is illustrated in Figure 3 (a) and statistics are given in Table 1. We have kept the image tags that appear in at least 500 images, and user categories that appear also at least 500 times in the dataset resulting in 21 possible labels for photos and 42 for authors[2].

4.2 Models

We have compared the proposed approach denoted by **LSHM** (Latent Space Heterogeneous Model) with two other models representative of the main-stream families of algorithms. The first one transforms the heterogeneous classification problem into multiple homogeneous problems. The second one performs an unsupervised projection of the heterogeneous network onto a latent space and then applies a multi-class classification algotirhm on this space. Both models are state of the art approaches. Note that except for [2, 14] there does not exist any algorithm that handles the relational classification problem on heterogeneous networks with disjoint label sets. More precisely, the first model, de-

[2]The dataset is available at http://www-poleia.lip6.fr/~jacoby/Research/research_en.html

DBLP No content	Nodes	**Type**	**Nb. Nodes**	**Nb. Labeled Nodes**	**Nb. Labels**
		Paper	14,376	14,376	20
		Author	14,475	4,057	4
	Edges	**Type**	**Nb. Edges**		
		Author→Paper	41,794		
DBLP With Content	Nodes	Paper	14,376	14,376	20
		Author	14,475	4,057	4
		Words	8,920	0	
	Edges	Author→Paper	41,794		
		Paper→Word	114,624		
Flickr	Nodes	Photos	46,926	8,766	21
		User	4,760	3,476	42
	Edges	User←User	175,779		
		User←Photo	46,926		

Table 1: Statistics on the three datasets

Model	Type of node	Training Size		
		10%	30%	50%
MTH	Author	58.8	65.0	69.2
	Paper	28.2	32.7	34.8
ULS	Author	27.0	27.7	27.6
	Paper	11.0	12.4	12.5
LSHM (no Content)	Author	**64.9**	**75**	**83**
	Paper	**30.4**	**34.5**	**37.2**
LSHM (with Content)	Author	**67**	**82**	**87.1**
	Paper	**30.6**	**35.6**	**37.9**

Table 2: Accuracy over the DBLP datasets with a latent space of size 30.

Model	Type of node	Training Size		
		10%	30%	50%
MTH	User	42	**46.8**	48.7
	Photo (Same Author)	35.6	59.4	65.5
	Photo (Author are friends)	19.3	21.7	22
	Photo (Author are same or friends)	22.4	31.3	32.4
ULS	User	26.1	28.0	28.3
	Photo	9.8	9.8	9.9
LSHM	User	**42.9**	45.7	**49.1**
	Photo	**43.3**	**61.6**	**68.1**

Table 3: P@1 over the Flickr datasets with a latent space of size 200

noted by **Mapping to Homogeneous Model (MTH)**, uses multiple homogeneous graphs, one for each node type, to represent the heterogeneous network. For each homogeneous problem, we use the model proposed in [6] that minimizes the loss in Equation 1. It does not make use of the correlations between labels of nodes types.

The homogeneous graphs have been built as follows:

- For the DBLP corpus, the graph of authors is built by connecting two co-authors; the graph of papers is built by connecting two papers written by the same author.
- For Flickr, the graph of users is built by connecting two friends. Different graphs of photos are possible:
  - The **Same author** graph is built by connecting two photos that have the same author.
  - The **Author are friends** graph is built by connecting two photos that have been published by two friends.
  - The **Author are same or friends** graph is built by connecting two photos that have been published by the same user of by two friends.

Note that for these tests, the model "Same author" where two images are linked through an intermediate author node, shares similarities with the Graffiti model [2] which uses a 2 hops random walk for connecting nodes of the same type linked by a path of length 2 and separated by a node from another type.

The second model we have compared with, denoted by **ULS** (Unsupervised Latent Space), learns a latent representation for each node of the graph without using the labels and learns to predict the labels based on the latent representations. We use a variant of the model proposed in [4] for learning embeddings of knowledge databases. This model learns unsupervised projections of relational knowledge bases with a specific metric for each relation type. Here we used only one metric for all relations. Compared to our model, the main difference is that the latent space is built in an unsupervised way – the objective is to keep the graph structure in the latent space – while our method aims at finding "the best" representations for a particular classification problem.

Experiments were performed with 10 random training labeled sets and the reported performance was obtained by averaging those obtained over the 10 runs. Different configurations were tested:

- Experiments were performed with different training sizes: 10%, 30%, 50%. The training size refers to **the proportion of labeled nodes** used in the training set. For example, in the DBLP No Content dataset, a training set of 10% means that 1,437 papers over 14,376 and 405 users over 4,050 labeled users were considered. Note that for this dataset, all papers are labeled whereas only 4,050 users over 14,475 are labeled. Unlabeled nodes will only appear in the smoothing term of Equation 4. The evaluation was performed on the labeled nodes not used during training.
- The gradient descent step for LSHM was fixed to 0.1 – similar performance in terms of accuracy was obtained at the price of a lower convergence speed.

- The dimension of the projection spaces for ULS and LSHM is 30 was chosen that yields good performance for both models in our experiments.

- Since there are much more links among users than between users and photos in the Flickr dataset, the weights of the relations have been normalized in order to avoid that the $user \rightarrow user$ relations have more influence than the $user \rightarrow photo$ relations. Weight for $user \rightarrow photo$ relations is 1 while weight for $user \rightarrow user$ has been set to 0.1.

4.3 Evaluation measures

Mono-Label Datasets: . In the case of mono-label datasets (DBLP with and without content), we consider that the predicted category is the category with the highest score. We make use of an *accuracy* measure which is the ratio between the number of good classification and the number of total testing nodes.

Multi-Label Dataset: . For multi-label datasets, after predicting the scores of the different categories, one has to decide which subset of categories to assign to any node of the network. For the evaluation, we have considered two different evaluation measures: The **Precision at 1 (P@1)** measures the percentage of nodes for which the category with the highest score is a relevant category. The **Precision at k (P@k)** is the proportion of correct labels in the set of k labels with the highest predicted scores. For each evaluation node, k is artificially chosen to be the number of relevant categories. This measure is an optimistic measure of the capacity of a model to correctly rank the k relevant categories of any node.

4.4 Results

4.4.1 Quantitative evaluation

Tables 2 and 3 show the performance obtained on the different datasets for different training set sizes. The evaluation of the models is separately made on two different types of nodes – paper and author for DBLP, user and photo for Flickr. Globally, our model is able to outperform the other models. For example, for the DBLP dataset without content, at a training size of 10%, our model obtains 64.9% accuracy on the author nodes and 30.4% on the paper nodes while the MTH model obtains 58.8% for the authors and 28.2% for the papers. The ULS model only obtains 27% for the authors and 11% for the papers. This is due to the fact that the latent representations are learned independently of the classifiers, without using training labels[3]. When using the graph with content – i.e. when words are integrated into the heterogeneous network – our model obtains even better results and outperforms the heterogeneous model without content, showing the ability of our method to handle not only different types of nodes, but also content information. Compared to the MTH model performance increase is important for author nodes, and smaller for paper nodes. This is probably because there are far less labeled authors than papers so that the classification task on authors is more difficult and the simpler models fail. It is also more important when the proportion of labeled is increased (e.g. increase of 18% for 50% of labeled nodes). Here our more sophisticated LSHM model is able to take advantage of the additional information provided by these labels.

[3]The ULS results have been obtained in a latent space of size 30. We have tested many different sizes, and only the best results are reported here.

Training Size	Z	Acc. Author	Acc. Paper
10%	MTH	58.8	28.2
	ULS	27.0	11.0
	10	63.4	28.8
	20	62.6	29.1
	30	**64.9**	**30.4**
	50	62.5	29.7
	100	62.3	29.1
30%	MTH	65.0	32.7
	ULS	27.7	12.4
	10	74.3	33.9
	20	74.6	33.6
	30	**75**	**34.5**
	50	72.8	33.5
	100	72.7	33.4
50%	MTH	69.2	34.8
	ULS	27.6	12.5
	10	**83.6**	37
	20	82.6	**37.2**
	30	83	**37.2**
	50	80.6	36
	100	78.8	36

Table 4: (a) Accuracy on DBLP No Content depending on the representation size Z.

For Flickr, we compared LSHM model with different projected graphs for the photos in the MTH model, as described in section 2. The performance for $P@1$ and $P@k$ are given in Table 5. The performance increase is much smaller here than with DBLP, which is probably due to the pre-eminence of "friend" and "same author" relations respectively for users and photos. Said otherwise, for this Flickr dataset, much of the information is present in these two types of relations only. On the other hand, the proposed model allows us to reach best performance without testing for many graph projections.

4.4.2 Influence of the Representation size

Let us now examine how performance varies depending on the size of the latent space Z – Tables 4 and 5. For the DBLP corpus, using a space of 30 dimensions often gives the best results for all training sizes. When the number of dimensions is too small, the model is unable to find good representations, while when it is too large, the model is overfitting, resulting in a loss of performance. Concerning the Flickr corpus for which the graph topology is more complex, a latent space of 200 dimensions allows one to obtain good performance on both photos and users. When restricting the dimensions of this space to 20, the model seems to concentrate more on users than on photos. This is probably due to the fact that users are connected through direct relations, while photos are connected through paths of minimum size 2: when the model does not have enough representation capacity, it tends to better propagate on close nodes, and is not able to model longer propagations.

Training Size	Z	P@1/P@k Photo	P@1/P@k User
10%	MTH	35.6/35.4	42/33.4
	ULS	26.1/21.3	9.8/10.5
	20	38.4/33.8	**46.5**/**33.5**
	50	41.3/38	39.7/29.6
	100	**43.3**/40.4	39.8/30.5
	200	43.1/**41.7**	42.9/30.6
30%	MTH	59.4/56.3	46.8/**36.2**
	ULS	28.0/23.2	9.8/10.6
	20	48.8/46.4	**47.6**/33.7
	50	55/53.9	46.3/33.7
	100	60.4/58	45.6/33.5
	200	**61.6**/**58.2**	45.7/33.5
50%	MTH	65.5/63.7	48.7/36.0
	ULS	28.3/23.5	9.9/10.8
	20	50.3/48.7	**49.2**/30.5
	50	57.4/57.3	47.3/**36.4**
	100	67.6/**64.8**	48.3/35.1
	200	**68.1**/64.7	49.1/34.6

Table 5: P@1 and P@k on Flickr depending on the representation size Z w.r.t MTH and ULS.

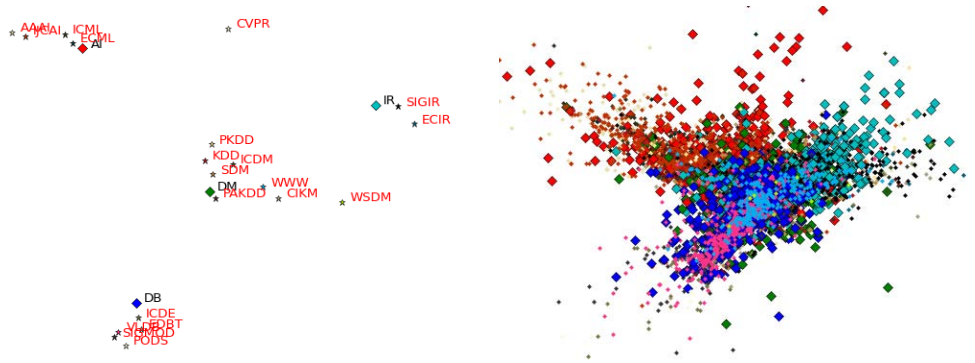
4.4.3 *Qualitative results*

In order to visualize the latent representations, we have performed a PCA over these representations for both articles and authors on the DBLP dataset for testing nodes. A 2 dimensional PCA projection corresponding to the first 2 axis is illustrated in Figure 5. The right figure shows all the labeled nodes in the latent space (after PCA dimension reduction) - different shapes correspond to different types of nodes (large diamond shapes for authors, small diamonds for conferences), and different colors correspond to different labels. PCA projection shows that the proposed model is able to map nodes with the same labels onto the same parts of the latent space. The left part of the figure shows the average latent representation projection for each label. The four diamonds correspond to the centroid of the four authors research domains, while the 20 smaller star points correspond to the centroids of the papers published at each of the 20 conferences. One can see that, for any particular research domain corresponding to the author classes, the model is able to find a latent representation which is close to the conferences that are related to this domain, showing the ability of the model to find correlation between the labels of the different types of nodes.

5. RELATED WORK

Graph node classification has motivated a lot of work during the last decade and different models have been proposed. Two main families of models can be distinguished:
(i) Collective classification techniques are extensions of inductive learning to relational data. They consider both node attributes, labels and their dependencies. The classification problem is formulated as an optimal assignment of labels to the vertices of a graph. Since exact algorithms cannot be used in general for this combinatorial problem, approximate iterative algorithms have been developed. Sen et al. [19] provide a general introduction and a comparison of these models. They distinguish between local and global models. The former such as *Iterative Classification* [19] and its variants like SICA [18], *Gibbs Sampling* [17], or *Stacked Learning* [15], make use of local classifiers taking as input the node attributes and statistics on the neighbors labels. The latter attempt to optimize a global function using graphical models, e.g. Markov Random Fields trained for example using loopy belief propagation. In practice, they advocate the use of simple local models which offer similar performance as more complex graphical models and do not suffer from convergence problems exhibited by the latter. All these methods have been proposed for homogeneous networks whereas heterogeneous classification was explicitly mentioned as an open problem.
(ii) The second family of models consists of semi-supervised and transductive regularized models – [28], [3] and [27] – which are based on the minimization of an objective function that encourages connected nodes to have the same labels. This family of models has been initially proposed for pure relational classification (no content associated to nodes) and for homogeneous networks. It has been extensively used in many different contexts. Extensions for handling node content information have been developed with applications to social network labeling [6] and Web-spam detection [1]. Several extensions have been proposed for dealing with more complex networks. For example, algorithms have been developed for multi-relational graphs, where nodes are all of the same type, but can be connected through multiple relations which will have different influences on the label propagation. These methods can learn from the data the weights of the different relations, and the multi-graph is then reduced to a simple graph by combining the different types of relations –[25, 12, 9]. Note that besides node classification, other methods have been proposed for other tasks like link-prediction [7] for example
Mining heterogeneous networks is a much more recent domain and different tasks have been addressed: classification of nodes [11, 10, 8, 2], link prediction [5],[24], influence analysis [16], clustering [20], entity similarity search [26], [21]. We will concentrate here on classification which has been addressed in a few papers.

Most of the proposed models are extensions of algorithms developed for homogeneous models. Typical of the transductive regularized models extensions is the work by Ji et al. [11] which is inspired by the homogeneous model of Zhou et al. [27]. For each couple of node types, they consider the weight matrix of the bipartite subgraph corresponding to the edges linking the two node types. The objective function is then a linear combination of the smoothness constraints defined over each of these matrices. For instance in an heterogeneous network of authors and papers like DBLP, this model will consider three relation types: author-author, paper-paper and author-paper and as many weight matrices. Instead of one smoothing term in the homogeneous case, the number of smoothing loss terms is proportional to the square of the number of node types ($n(n-1)/2$ if there are n node types) and of course there are as many regularization coefficients to set up which might become problematic for complex networks. Note that in these approaches, the authors do not make the choice of an a priori graph projections but consider all possible binary relationships between node types. An extension of this work [10] is proposed for a ranking problem: the objects in each class are ranked according to their class relevance. A very similar formulation is proposed in [8] which is also a direct extension of the algorithm in [27]. Again, multiple weight matrices between node type



(a) Centroid of different authors and papers labels (unlabeled nodes)

(b) All nodes

Figure 5: PCA on the DBLP corpus computed over the latent representation of the testing nodes projection when using 10% nodes for training.

sets are introduced and linearly combined in a loss function. All these extensions consider that the different types of nodes share the same set of categories. In our model, the latent space formulation allows to bypass this limitation.

Some authors have proposed extensions of the collective classification methods [19]. In a series of papers, Kong, Yu and colleagues [14, 13] make use of the notion of meta-path for extending the *Iterative Classification* technique to heterogeneous networks both for mono-label and multi-label classification. A meta-path is a path in an heterogeneous graph that links two nodes through an heterogeneous path. For instance in a bibliographic network, the meta-path "Paper $\xrightarrow{cite^{-1}}$ Author $\xrightarrow{cite}$ Paper" connects two papers with the same author. All the possible non-redundant meta-paths shorter than a chosen length are used. Their algorithm first extracts a certain number of relevant meta paths and considers the nodes linked by such paths as neighbors. Then it applies the ICA algorithm to the corresponding graph. Only one type of nodes is considered for a classification task since relevant meta path may differ from one type to another. In their experiments, they consider meta paths joining two nodes of the same type. Heuristics for limiting the number of meta paths are used. This is equivalent to defining a projection graph for each class based on these meta paths and then doing collective classification. These methods consider both the relational structure and node characteristics, and make use of the latter to initialize the node class probabilities. Here again the proposed method relies on heuristics for defining the paths, and does not directly exploit inter-node types dependencies.

To our knowledge, the only existing model addressing directly heterogeneous network classification with different label sets per node type is [2]. The algorithm is based on a random walk method, where, in addition to simple jumps to neighboring nodes which could be of any type, 2-hop jumps between nodes of the same type are allowed. Nodes of the same type can then be connected by a path of size 2 where the intermediate node is of a different type. Labels propagate among nodes of the same type. Paths joining nodes of the same type and longer than 2 are ignored. However the propagation mechanism allows the diffusion of labels on longer paths with a decreasing influence when the path length increases. This algorithm makes use of node characteristics in order to initialize node class distributions. Although the exact correspondence with our algorithm is difficult to analyze, this method intuitively corresponds to a random walk on an extended graph where connections corresponding to two hops between nodes of the same type define the connection graph. Compared to this approach, our model is able to consider the correlations between the labels of connected nodes of different types through the latent space projection which is not the case in their model. It also learns directly a discriminant function instead of propagating initial label distributions modulated by node class authority.

6. CONCLUSION

We have proposed a new model able to label nodes of heterogeneous networks where the nodes are of different types, each type corresponding to a particular set of possible categories. Our method is able to use the correlation between the labels of two connected nodes of different types, and thus to use the complex structure of the graph for labeling. Our algorithm is based on the idea of computing a latent representation of nodes in a space that is common to all the possible types of nodes, and to assume that two connected nodes tend to have close latent representations. The labels are then deduced from these representations. Our experiments on three datasets show that the proposed model outperforms classical approaches, and qualitative analysis show that the proposed method effectively captures the inter-dependencies between the labels of different types of nodes. Different extensions of this model are currently being investigated: the first one is to deal with multi-relational heterogeneous networks – i.e. heterogeneous networks where two nodes can be connected with more than one relation – and dynamic heterogeneous networks. We are also working on an extension which allows one to learn sparse representation of nodes in the graph.

7. ACKNOWLEDGMENTS

This work has been partially supported by the REMI FUI project and the ANR (French National Research Agency) MLVIS project

8. REFERENCES

[1] Jacob Abernethy, Olivier Chapelle, and Carlos Castillo. Witch: A new approach to web spam detection. In *In Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, 2008.

[2] Ralitsa Angelova, Gjergji Kasneci, and Gerhard Weikum. Graffiti: graph-based classification in heterogeneous networks. *World Wide Web*, 15(2):139–170, 2012.

[3] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *J. Mach. Learn. Res.*, 7:2399–2434, December 2006.

[4] A. Bordes, J. Weston, R. Collobert, and Y. Bengio. Learning structured embeddings of knowledge bases. In *AAAI*, 2011.

[5] Darcy Davis, Ryan Lichtenwalter, and N.V. Chawla. Multi-Relational Link Prediction in Heterogeneous Information Networks. In *ASONAM*, 2011.

[6] Ludovic Denoyer and Patrick Gallinari. A ranking based model for automatic image annotation in a social network. In *ICWSM*, 2010.

[7] Sheng Gao, Ludovic Denoyer, and Patrick Gallinari. Link pattern prediction with tensor decomposition in multi-relational networks. In *CIDM*, 2011.

[8] Taehyun Hwang and Rui Kuang. A heterogeneous label propagation algorithm for disease gene discovery. In *SDM*, page 12, 2010.

[9] Yann Jacob, Ludovic Denoyer, and Patrick Gallinari. Classification and annotation in social corpora using multiple relations. In *CIKM*, pages 1215–1220, 2011.

[10] Ming Ji, Jiawei Han, and Marina Danilevsky. Ranking-based classification of heterogeneous information networks. In *KDD*, pages 1298–1306. ACM, 2011.

[11] Ming Ji, Yizhou Sun, Marina Danilevsky, Jiawei Han, and Jing Gao. Graph regularized transductive classification on heterogeneous information networks. In *ECML PKDD*, volume 53, pages 570–586, 2010.

[12] T. Kato, H. Kashima, and M. Sugiyama. Integration of multiple networks for robust label propagation. In *SIAM Conf. on Data Mining*, pages 716–726, 2008.

[13] Xiangnan Kong, Bokai Cao, and Philip S. Yu. Multi-label classification by mining label and instance correlations from heterogeneous information networks. KDD '13, pages 614–622, 2013.

[14] Xiangnan Kong, Philip S. Yu, Ying Ding, and David J. Wild. Meta path-based collective classification in heterogeneous information networks. In *CIKM*, pages 1567–1571, 2012.

[15] Zhenzhen Kou. Stacked graphical models for efficient inference in markov random fields. In *In Proc. of the 2007 SIAM International Conf. on Data Mining*, 2007.

[16] Lu Liu, Jie Tang, Jiawei Han, and Meng Jiang. Mining topic-level influence in heterogeneous networks. In *CIKM*, 2010.

[17] Sofus A. Macskassy and Foster Provost. A simple relational classifier. In *Proceedings of the Second Workshop on Multi-Relational Data Mining (MRDM-2003) at KDD-2003*, pages 64–76, 2003.

[18] Francis Maes, Stephane Peters, Ludovic Denoyer, and Patrick Gallinari. Simulated iterative classification a new learning procedure for graph labeling. *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases Part II*, II:47–62, 2009.

[19] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.

[20] Yizhou Sun, Charu C. Aggarwal, and Jiawei Han. Relation strength-aware clustering of heterogeneous information networks with incomplete attributes. *PVLDB*, 5(5):394–405, 2012.

[21] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S. Yu, and Tianyi Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *PVLDB*, 4(11):992–1003, 2011.

[22] Yizhou Sun, Yintao Yu, and Jiawei Han. Ranking-based clustering of heterogeneous information networks with star network schema. In *KDD*, pages 797–806, 2009.

[23] Lei Tang, Xufei Wang, and Huan Liu. Community detection via heterogeneous interaction analysis. *Data Min. Knowl. Discov.*, 25(1):1–33, 2012.

[24] Chi Wang, Rajat Raina, David Fong, Ding Zhou, Jiawei Han, and Greg J. Badros. Learning relevance from heterogeneous social network and its application in online targeting. In *SIGIR*, pages 655–664, 2011.

[25] Meng Wang, Xian-Sheng Hua, Richang Hong, Jinhui Tang, Guo-Jun Qi, and Yan Song. Unified video annotation via multigraph learning. *Circuits and Systems for Video Technology, IEEE Transactions on*, 19(5):733 –746, may 2009.

[26] Xiao Yu, Yizhou Sun, Brandon Norick, Tiancheng Mao, and Jiawei Han. User guided entity similarity search using meta-path selection in heterogeneous information networks. CIKM '12, pages 2025–2029, New York, NY, USA, 2012. ACM.

[27] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Inform. Process. Systems 16*. 2004.

[28] Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning from labeled and unlabeled data on a directed graph. In *Proc. of the 22nd intern. conf. on Mach. learn.*, ICML '05, pages 1036–1043, 2005.