On the Utility of Abstraction in Labeling Actors in Social Networks

Ngot Bui
Computer Science Department
Iowa State University
Ames, Iowa 50010
Email: bpngot@cs.iastate.edu

Vasant Honavar
Computer Science Department
Iowa State University
Ames, Iowa 50010
Email: honavar@cs.iastate.edu

***Abstract*—Social networks are naturally represented as *heterogeneous* networks with multiple types of objects e.g., *actors*, *items* and multiple types of links e.g., links between *actors* that denote social ties e.g., friendship, and links that connect actors to *items* e.g., photo, video, articles, etc. that denote relationships between *actors* and *items*. In this paper, we consider the task of assigning labels to the unlabeled actors (individuals) in a large heterogeneous social network in which labels are available for a subset of actors. Specifically, we seek to learn a predictive model to label actors based on the attributes of the actors themselves and/or items that are linked to them in the network. Unfortunately, the number of distinct *items*, represented in real-world networks such as Facebook or Flickr is quite large (in the millions) although only a small subset of them are linked to specific actors. This leads to *data sparsity* which causes over-fitting and hence poor performance in predicting the labels of unlabeled actors. To address this problem, we induce hierarchical taxonomies over items and use the resulting taxonomies as a basis for selecting abstract and hence parsimonious representations of network data for learning the predictive models. Our experiments using three different predictors (Iterative classification Naïve Bayes, Iterative classification Logistic Regression, and EdgeCluster) on two real-world data sets, Last.fm and Flickr, show that the predictive models that take advantage of abstract representations of network data are competitive with, and in some cases, outperform those that do not.**

I. Introduction

The emergence of social networks e.g., Facebook, and social media e.g., Flickr has resulted in exponential increase in the amount of data that link diverse types of richly structured digital objects e.g., individuals, articles, images, videos, music, etc. Such data are naturally represented as a *heterogeneous* network with multiple types of objects e.g., *actors*, *items* and multiple types of links e.g., links between *actors* that denote social ties e.g., friendship, and links that connect actors to *items* e.g., photo, video, articles, etc. that denote relationships between individuals and items, e.g., the fact that a specific individual, say John Smith *likes* a particular painting, say Mona Lisa. This kind of data can be modeled in a way such that actors are associated with labels that can be in many forms: labels which denote political or religious point of views of actors; labels that come from many other characteristics that denote behaviors or preferences of actors in activities. There are several applications that can make use of labels of actors in a social network: social networking advertising systems that show the advertisements to actors in a particular topic which is closely relevant to their preferences; recommendation systems that are based on actors' interests to recommend objects (e.g., musics, movies). However, in real-world social networks, the labels that are associated with actors are not available due to many reasons such as privacy concern and/or out-of-date profile information.

In this paper, we seek to learn a predictive model to label actors based on the attributes of the actors themselves and/or items that are linked to them in the network. In contrast to traditional supervised learning scenarios, the actors in a network are not independently identically distributed (i.i.d) due to the presence of homophily [1], the propensity of actors with similar traits to be linked together and the resulting correlations [2] among their attributes (and labels). The *collective classification* approaches to labeling actors (or more generally, objects) in networks exploit homophily [3], [4], [2], [5] to their advantage. However, there is substantial room for improvement in the performance of the state-of-the-art approaches to labeling actors in heterogeneous real-world networks.

The number of distinct *items* (e.g., the movie "Back to the Future" or the play "Merchant of Venice"), represented in real-world networks such as Facebook or Flickr is quite large (in the millions) although only a small subset of them are linked to a specific actor, say, John Smith. This leads to data sparsity which in turn leads to over-fitting by overly complex models [6] and hence poor predictive performance in labeling the unlabeled actors in a network. To address this problem, some authors [5], [4], [2] have suggested replacing items by their attributes. Unfortunately, in many real-world scenarios, this results in loss of information essential for classifying or labeling actors. We note that the *items* linked to *actors* in real-world networks are often specified at varying levels of granularity or detail. What is needed is an effective means to choose an abstract yet sufficiently informative representation of network data to achieve accurate and reliable labeling. To address this problem, we propose an alternative by inducing hierarchical taxonomies over items and use the resulting taxonomies as a basis for selecting abstract and hence parsimonious representations of network data. This approach offers a means of striking a compromise between using the finest granularity identity-level representation of item objects of a social network on the one hand and the coarsest granularity type-level representation (in which objects are simply encoded by their types). Specifically, we induce from data, the item type abstraction hierarchies (ITAH) that group item objects into hierarchical taxonomies. We train classifiers using a subset

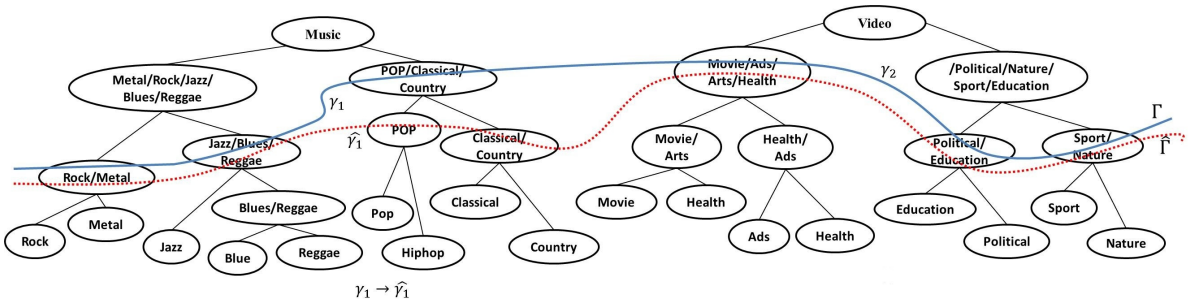


Fig. 1. Abstraction hierarchies of item type Music and Video, respectively.

of actors in the network and abstract representations of items linked to those actors. Our method starts with the most abstract representation and successively refines the representation.

We compare the Iterative classification Naïve Bayes classifier (INBC), the Iterative classification Logistic Regression Classifier (ILRC) [2] and the EdgeCluster [7] classifier that make use of the items' attributes with their counterparts (AINBC, AIRLC, and AEdgeCluster respectively) that make use of ITAH-induced abstract representations of network data. The results of our experiments with two real-world data sets, Last.fm and Flickr, show that the classifiers that make use of ITAH-induced abstract representations of network data competitive with, and in some cases, outperform those that does not.

The rest of the paper is organized as follows. Section 2 introduces the problem formulation. Section 3 describes our approach to inducing ITAH from network data and using the resulting ITAH to learn compact and accurate predictive models for labeling objects in a large social network. Section 4 presents results of experiments that demonstrate the utility of the proposed approach. Section 5 concludes with a summary and a discussion and an outline of some promising directions for further research.

II. Preliminaries

A heterogeneous social network with multiple types of nodes and links can be represented as a graph $G = (V, E)$ in which $V = \{A \cup I\}$ is a set of vertices where A denotes a set of *actors* and $I = I_1 \cup I_2 \cup \cdots \cup I_M$ (M is number of item type) is a set of items where I_i denotes a set of items of type *i;* and E is a set of edges that model the links e.g., those that link different *actors* or those that link *actors* with *items* e.g., books, paintings, movies. In the rest of the paper, we will use I_i to denote both a set of items of *i-th* type, say a set of books, as well as the name of the type of items represented in I_i, say, *Book.*

Definition 1. ***Item Type Abstractions.*** *An* item type abstraction hierarchy *ITAH T_i over a set of* items I_i*, is a dendrogram such that the root of T_i denotes I_i and the set of leaves of T_i form a partition of I_i. A subset γ_i of nodes of an ITAH T_i such that for any leaf $l \in T_i$, either $l \in \gamma_i$ or l is a descendent of some node $m \in \gamma_i$ defines a type abstraction of type I_i. The set of elements of any given type abstraction γ_i form a partition of I_i (and correspond to a cut through T_i). We define $size\,(\gamma_i)$, the* size *of a type abstraction to be $|\gamma_i|$, the number of elements in γ_i.*

Figure 1 shows an example of ITAHs for *music* and *video* items respectively. The type abstractions γ_1 for music and γ_2 for video correspond to cuts through the respective ITAHs. γ_2 contains three video subcategories (*Movie/Ads/Arts/Health*, *Political/Education*, and *Sport/Nature*) and hence $size\,(\gamma_i)$=3.

Definition 2. ***Network Abstractions.*** *Let $\Gamma = \{\gamma_1, \gamma_2, \cdots, \gamma_M\}$ be a global abstraction (GA) where each γ_i is a type abstraction over the corresponding ITAH T_i. A network abstraction $\nu\,(G, \Gamma)$ of a network G induced by Γ is simply a graph where each node of G is replaced by its corresponding type abstraction from Γ and each link between a pair of nodes in G is replaced by a link between the type abstractions of the corrresponding nodes specified by $\Gamma = \{\gamma_1, \gamma_2, \cdots, \gamma_M\}$. We refer to $\nu\,(G, \Gamma)$ as the Γ-induced abstraction of G. We define $size\,(\Gamma)$, the* size *of $\Gamma = \{\gamma_1, \gamma_2, \cdots, \gamma_M\}$ to be $size(\Gamma) = \sum_{i=1}^{M} |\gamma_i|$.*

Given an abstraction hierarchy T_i of the type I_i, we denote that a type abstraction $\hat{\gamma}_i$ is a *refinement* of an type abstraction γ_i if $\hat{\gamma}_i$ is obtained by replacing one node in γ_i by its descendents. In other words, γ_i is the abstraction of $\hat{\gamma}_i$. And we denotes that a global abstraction $\hat{\Gamma}$ is a *refinement* of a global abstraction Γ if a type abstraction in $\hat{\Gamma}$ is a refinement of some type abstraction in Γ.

Figure 1 shows an example of the global abstraction Γ and its refinement $\hat{\Gamma}$. The type abstraction $\gamma_1 = \{$*Rock/Metal, Jazz/Blues/Reggae, POP/Classical/Country*$\}$ in the *Music* hierarchy is substituted by its refinement $\hat{\gamma}_1 = \{$*Rock/Metal, Jazz/Blues/Reggae, POP, Classical/Country*$\}$.

Definition 3. ***Labeling Actors in Networks.*** *Consider a social network in which each actor $a \in A$ belongs to one or more categories in $C = \{C_1, C_2, \cdots, C_N\}$. Suppose the category membership of each actor $a \in A$ is denoted by a 3-valued label vector where the jth element is 1 if the actor a is assigned to category C_j; 0 if a is not assigned to category C_j; and u if a is not labeled with respect to its membership in category C_j. Given a social network in which only some of the actors are labeled with respect to their membership in any given category, the task is to complete the labeling.*

Algorithm 1 ITAH Learning Algorithm

Input: A set of item objects I_i; An integer K_i.
Output : An ITAH T_i, and an ordered set of type abstractions Δ_i of type I_i

1: Use k-means to partition I_i into K_i clusters to obtain $B_i = \{b_i^1 \cdots b_i^{K_i}\}$
2: $T_i = B_i$;
3: $\Delta_i = \{B_i\}$;
4: **while** $|B_i| > 1$ **do**
5: $\quad (x, y) = \arg\min_{B_i} \left\{ D\left(P_i^k || P_i^j\right) \right\}$
6: $\quad b_i^{xy} = b_i^x \cup b_i^y$
7: $\quad T_i = T_i \cup b_i^{xy}$ s.t. $Parent(b_i^x) = Parent(b_i^y) = b_i^{xy}$
8: $\quad B_i = B_i \setminus \{b_i^x \cup b_i^y\} \cup b_i^{xy}$
9: $\quad \Delta_i = \Delta_i \cup \{B_i\}$
10: **end while**

Algorithm 2 Global Abstraction Searching Algorithm

Input: $T_1, \cdots, T_M$, (training) data D_t, number of folds K
Output: An ordered set of global abstractions Λ.

1: Set current GA Γ to the most abstract GA.
2: **while** $|\Gamma| \leq \sum_{i=1}^{M} K_i$ **do** ▷ $K_i = leafnodes(T_i)$
3: $\quad$ Adding Γ to Λ
4: $\quad$ Induce M refinements $\hat{\Gamma}$'s of Γ based on $T_1, \cdots, T_M$
5: $\quad$ Train $M \times K$ classifiers using $\hat{\Gamma}$'s and K-fold cross-validation over D_t
6: $\quad$ Compute the $error\_rate(h_j(\hat{\Gamma}))$ of each classfier $h_j(\hat{\Gamma})$ using j^{th} fold from D_t
7: $\quad$ Set $\Gamma = \arg\min_{\hat{\Gamma}} \frac{1}{K} \sum_{j=1}^{K} error\_rate(h_j(\hat{\Gamma}))$
8: **end while**
9: Output Λ.

III. Labeling Actors in Social Networks

We proceed to describe our approach to labeling actors in social networks, i.e., solving the social network actor labeling (SNAL) problem: (i) Construct ITAH from social network data; (ii) Exploit the ITAH-induced network abstractions to induce compact yet accurate predictors of actor labels to solve the SNAL problem[1].

A. Constructing Item Type Abstraction Hierarchies

We generate an item type abstraction hierarchy for each type of items e.g., *Books, Movies*, by clustering item objects. Before we proceed, we need to select a *representation* of items and a similarity or distance measure for caculating the pair-wise similarity between items that can be used for successively grouping items into clusters. We exploit the user annotation of items in social networks (e.g., tags, title, description) [8] to represent each item by a *bag of words*. In what follows, we use subscripts and superscripts to denote indices of the *type* of the item and to index the items in a set of items of a given type respectively. Let $W_i = \left\{t_i^1, t_i^2, \ldots, t_i^{d_i}\right\}$ be the set of distinct terms occurring in the annotations of items in $I_i = \left\{o_i^1, o_i^2, \ldots, o_i^{m_i}\right\}$ (items of type *i*). We define the term distribution P_i^k of an item as the conditional probability distribution $\left(p_i^k(1), \cdots, p_i^k(d_i)\right)$ where $p_i^k(l) = P(t_i^l | o_i^k) = n(o_i^k, t_i^l) / \sum_{r=1}^{r=d_i} n(o_i^k, t_i^r)$ over the distinct term set W_i, where $n(o_i^k, t_i^l)$ is the number of occurences of term t_i^l in the annotation of the item o_i^k. To assess the distance between two items o_i^k and o_i^j, we use the Jensen-Shannon divergence [9] $D\left(P_i^k, P_i^j\right)$ between their term distributions P_i^k and P_i^j. The Jensen-Shannon divergence or information radius between two probability distributions P_i^k and P_i^j is defined as:

$$D(P_i^k || P_i^j) = \frac{1}{2}\left[\sum_l p_i^k(l) \log\left(\frac{2p_i^k(l)}{p_i^k(l) + p_i^j(l)}\right) + \sum_l p_i^j(l) \log\left(\frac{2p_i^j(l)}{p_i^k(l) + p_i^j(l)}\right)\right]$$

The smaller the divergence between the term distributions of two items the higher their similarity.

Algorithm 1 demonstrates how to learn an ITAH. Given the large number of items that need to be clustered, we make use of the two-level hybrid clustering algorithm [10] to generate ITAHs: first, for each item type *i,* we deploy *k*-means[2] to cluster the set I_i of items of that type into a set of K_i clusters. The resulting K_i clusters are then further grouped successively using standard hierarchical clustering algorithm to generate an ITAH T_i. The result of the clustering at each step of hierarchical clustering starting with the initial set of K_i clusters and ending with a single cluster are accumulated to obtain an ordered set $\triangle_i$ of type abstractions[3]. The type abstraction obtained by previous iteration is the refinement of the one obtained in the current iteration (i.e., line 9). Note that the leaves of T_i are the K_i clusters produced by the first stage (k-means) of the two-stage hybrid clustering procedure outlined above.

The clustering process outlined above is executed for each item type to obtain a collection of item type abstraction hierarchies $T_1 \cdots T_M$ and the corresponding ordered sets of type abstractions $\triangle_1 \cdots \triangle_M$. Actually, $\triangle_i$ stores all type abstractions of type I_i and it is equivalent to T_i. The purpose of introducing $\triangle_i$ is for convenience when accessing a type abstraction or the refinement of one type abstraction in item abstraction hierarchy T_i. The Cartesian product $\triangle = \times \triangle_i$ defines the space of global abstractions induced by the item type abstraction hierarchies $T_1 \cdots T_M$. Each choice of a global abstraction $\Gamma = \{\gamma_1, \gamma_2, \cdots, \gamma_M\} \in \triangle$ where $\gamma_i \in \triangle_i$

[1]Because the primary focus of this paper is on examining the power of network abstraction in labeling actors in a social networks, we focus on a simpler instance of the SNAL problem, wherein each actor belongs to exactly one category. However, the proposed approach naturally generalizes to the setting where each actor can simultaneously belong to multiple categories.

[2]To cluster large item object sets, we used the parallel k-means [11]

[3]Note that the cardinality of $\triangle_i$ is $O(K_i)$

induces a corresponding network abstraction $\nu(G, \Gamma)$ of the network G.

B. Representing Actors in Social Networks

For a given choice of network abstraction $\nu(G, \Gamma)$ of the network G induced by $\Gamma = \{\gamma_1, \gamma_2, \cdots, \gamma_M\}$, an item object o_i^j of type I_i is uniquely mapped to, and hence encoded by one of the nodes in γ_i, a cut through T_i. Each actor in the network is represented by the collection of (abstractions of) the item objects that link to it along with (the class labels of) the actors that link to it.

We consider two variants of actor representation: the *binary* encoding which simply indicates the presence or absence of a particular feature and the *count* (frequency) encoding [6], which provides the number of occurences of the feature. (analogous to to the multi-variate Bernoulli and multinomial Naïve Bayes in [12] models used for encoding documents).

C. Learning to Label Actors

Our basic approach to learning to label actors in a social network is to search the space of global abstractions of the network to find the ones that yield an accurate and compact predictor. We first use Algorithm 1 to generate collection of abstraction hierarchies. The procedure for searching optimal global abstractions is shown in Algorithm 2. We first start from most abstract global abstraction of the network. The size of this abstraction is equal to M (number of abstraction hierarchies). With a current global abstraction Γ, we induce M refinements of Γ (line 4) where each refinement is obtained by substituting a type abstraction (node) in Γ with its descendents in the abstraction hierarchy. Note that because the abstraction hierarchy is a binary tree, each refinement of Γ is of the same size i.e., $size(\Gamma) + 1$. We score each of these refinements by training and testing predictors (using K-fold cross-validation) based on the representations of actors induced by each refinement. The global abstraction that receives the highest score is chosen as the optimal one for the corresponding size (i.e., $size(\Gamma) + 1$) (line 7). The process is repeated until we reach the the least abstract global abstraction i.e., one that includes the leaf nodes of each of the M abstraction hierarchies. The output of Algorithm 2 is an ordered set of optimal global abstractions with sizes ranging from M to $\sum_i^M K_i$.

In summary, to learn classifiers for labeling actors, we create a collection of type abstraction hierarchies (i.e., $T_1, T_2, \ldots, T_M$) based on item objects of a network using Algorithm 1. Based on a set of abstraction hierarchies and labels of some of the actors (i.e., training data), we determine a set of optimal global asbtractions (i.e., Λ) using Algorithm 2. For each of global abstractions from the output of Algorithm 2, we build a classifier based on training data and then use the learned classifier to predict labels of unlabeled actors.

IV. Experiments and Results

A. Social Media Data

In order to obtain real-world network data sets that contain multiple types of objects and links (i.e., heterogeneous network data), we turn into crawling available social media networks and use them in our paper.

TABLE I
SUMMARIES OF DATA SETS

Last.fm		Flickr	
Groups	43	Groups	17
Users	45,139	Users	18,021
Tracks	430,976	Photos	1,478,174
Artists	50,983	User-User	178,720
User-User	239,479	User-Photo	1,478,174
User-Artist	433,752	-	-
User-Track	7,023,181	-	-

The first dataset is from Last.fm network. Last.fm is a music website where registered people can listen to tracks, join in favorite groups, and make friends with other music lovers. For our experiment, we focus on the network consisting of users as actors and tracks and artists as items which encode three kinds of relationships: user-user, user-track, and user-artist. We manually identified 43 disjoint groups in this network and crawled all users, and the items and artists that are linked to them along with the associated attributes e.g., biography of artists, and $tags$ of tracks. Each of groups in Last.fm refers to a set of users who have common characteristics/interests (e.g., http://www.last.fm/group/Metal denotes a group of users who are interested in Metal music).

Our second dataset is from Flickr, a network which allows registered users to upload photos and share photos. We selected users as actors and photos as items and two kind of relationships via user-user and user-photo connections. We manually identified 17 disjoint groups of actors in the Flickr network and crawled the associated information e.g., *tags*, *title*, and *description* of the photos. Likewise, each of groups in Flickr refers to a community of users who share the same taste in pictures (e.g., http://www.flickr.com/groups/iowa/ denotes a group of users who are interested in pictures about the state of Iowa).

In both cases, we use the group memberships of users as class labels for the actors for training and evaluating our predictors. We crawl the above two data sets by using Last.fm api[4] and Flickr api[5], discard users that have no links to any other users, and use the lucene[6] package to do preprocess the textual attributes of the item objects, i.e., remove stopwords, perform stemming, and remove infrequent terms. The result is a set of 3913, 1720, and 1408 terms (respectively) for representing photo, track, and artist items. The characteristics of the two data sets are shown in table 1.

B. Predictive Models

In order to examine the benefit of abstraction, we compare:

- Iterative classification Naïve Bayes Classifier (INBC) [2], a collective classification approach that uses naïve

[4]http://www.last.fm/api
[5]http://www.flickr.com/services/api/
[6]http://lucene.apache.org/

Bayes as the base classifier and the iterative classification algorithm as the collective inference technique and AINBC, a variant of INBC that is trained using abstract representations of the network data.

- Iterative classification Logistic Regression Classifier (ILRC) [2], [6], a collective classification approach that uses logistic regression as the base classifier and the iterative classification algorithm as the collective inference technique and AILRC, the variant of ILRC which is trained using abstract representations of network data.
- Edge-Centric Social Dimension (EdgeCluster) [7]: This method extracts the *social dimensions* and uses them as features to generate the discriminative model (see [7] for details) and AEdgeCluster, which uses abstract representations of network data.

In the cases of INBC, ILRC, and EdgeCluster, we extracted a bag of words representation of textual attributes of the items use as the features of a user (actor) whereas in the cases of AINBC, AILRC, and AEdgeCluster, we extract the abstract representation features as described in section 3. In all cases, we also used the neighbors' class labels and in the cases of EdgeCluster and AEdgeCluster, we also added the social dimension features. In each model, we compared the binary and the count based representations of features.

C. Experimental Setup

Our experiments are designed to explore the following question: How does the performance of AINBC, AILRC, and AEdgeCluster compare with that of INBC, ILRC, and EdgeCluster, respectively?

To answer this question, we trained AINBC, AILRC, and AEdgeCluster for values of z that range from M to $Z = \sum_i^M K_i$, where z is the size of the global abstraction Γ, and Z is the size of the largest global abstraction in Λ, and compared the performance of the above three classifiers with that of their counterparts, INBC, ILRC, and EdgeCluster, respectively, over the entire range from M to Z.

We evaluate the classifiers using 10-fold cross-validation. In each of the 10 cross-validation runs, we use 90% of data for training i.e., used by Algorithm 2 for identifying the ordered set of global abstractions Λ and for training the classifiers in Algorithm 3 and 10% of data for evaluating the performance of the classifiers trained by Algorithm 3. Since the data sets are unbalanced, we also use the macro-F1 measure which is influenced more by the classifier's performance on rare classes. We report the average accuracy and macro-F1 of 10-fold cross-validation experiment.

D. Results

We generated the photo ITAH in the case of Flickr and the track and artist ITAHs in the case of Last.fm. To ensure fairness of comparison, we set the maximum dimension of the features equal to the number of the features extracted from items in the absence of abstraction. Specifically, the numbers of leaf nodes of photo, track, and artist ITAHs are $K_1 = 3913$, $K_1 = 2800$, and $K_2 = 328$, respectively.

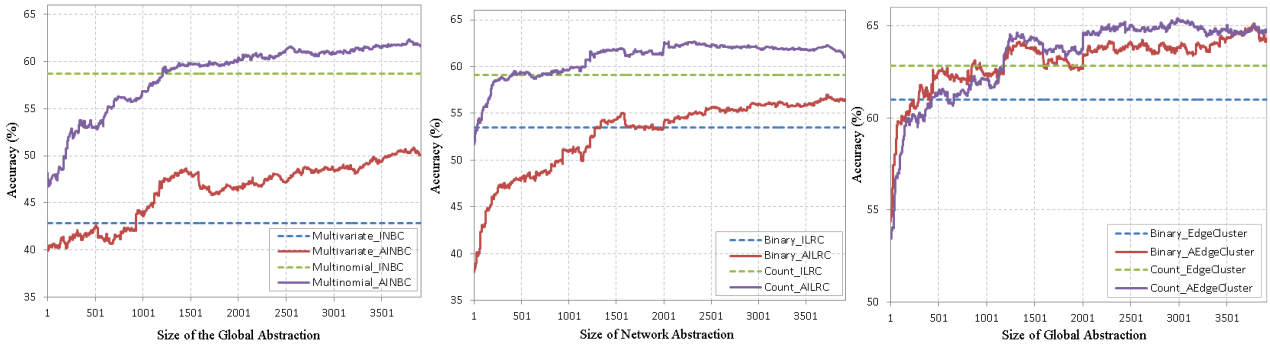
Figure 2 shows the results of the comparison of AINBC, AIRLC, and AEdgeCluster with INBC, ILRC, and EdgeCluster, respectively on two data sets considered in this work. As we can see in the figure, AINBC, AILRC, and AEdgeCluster approach the performance of INBC, ILRC, and EdgeCluster, respectively, with much smaller sizes of the global abstractions (compared to the maximum size of a global abstraction or to the dimension of items' attributes) with both binary and count-based feature representations of the data. In particular, with binary feature representation on Flickr data set, the performances of INBC, ILRC, and EdgeCluster are matched by those of AINBC, AILRC, and AEdgeCluster trained by using only network abstractions with far fewer features, 931, 1271, and 204, respectively. Likewise, with count-based feature representations on Last.fm data set, the performance of INBC, ILRC, and EdgeCluster are matched by those of AINBC, AILRC, and AEdgeCluster using global abstractions of sizes 1073, 801, and 1205, respectively. We saw qualitatively similar results in the case of binary representation except the case of AINBC on Last.fm. Not surprisingly, predictors that use count-based representation of features often outperform their counterparts that use binary representation.

Figures 3 shows the results of comparison of all six classifiers on both accuracy and macro-F1 measurements. We can see that AEdgeCluster performs the best in most of the cases except the ones on Last.fm with Macro-F1. The results of our experiments show that the abstraction representation makes it possible to construct preditive models, AINBC, AILRC, and AEdgeCluster, that are more compact than INBC, ILRC, and EdgeCluster. AINBC, AILRC, AEdgeCluster are competitive with, and in some cases, surpass the performance of INBC, ILRC, and EdgeCluster, respectively. We observe the same pattern of the classification performances on the rare classes of AINBC, AILRC, AEdgeCluster. Figures 3(e) to 3(h), show that AINBC, AILRC, and AEdgeCluster match, and in several cases, surpass the performances of INBC, ILRC, and EdgeCluster, respectively, with respect to the macro-F1 measure, with global abstractions that yield substantially more compact classifiers. For example, in the case of binary representation, using a global abstraction of size 756 (figure 3(e)), AEdgeCluster achieves its highest macro-F1 of 39.79% as compared to the EdgeCluster which achieves the macro-F1 of 35.38%.

The results of classifiers that exploit abstraction are in general competitive and sometime surpass the ones of classifiers that do not exploit abstraction. This can be explained that abstraction helps discover the latent prior knowledge (e.g., tracks in Last.fm can be classified into several types of tracks like rock music, classical music, etc.) from data. This latent prior knowledge are compact and distinguishable enough to improve the performance of the actor label predicting model.

V. Summary and Discussion

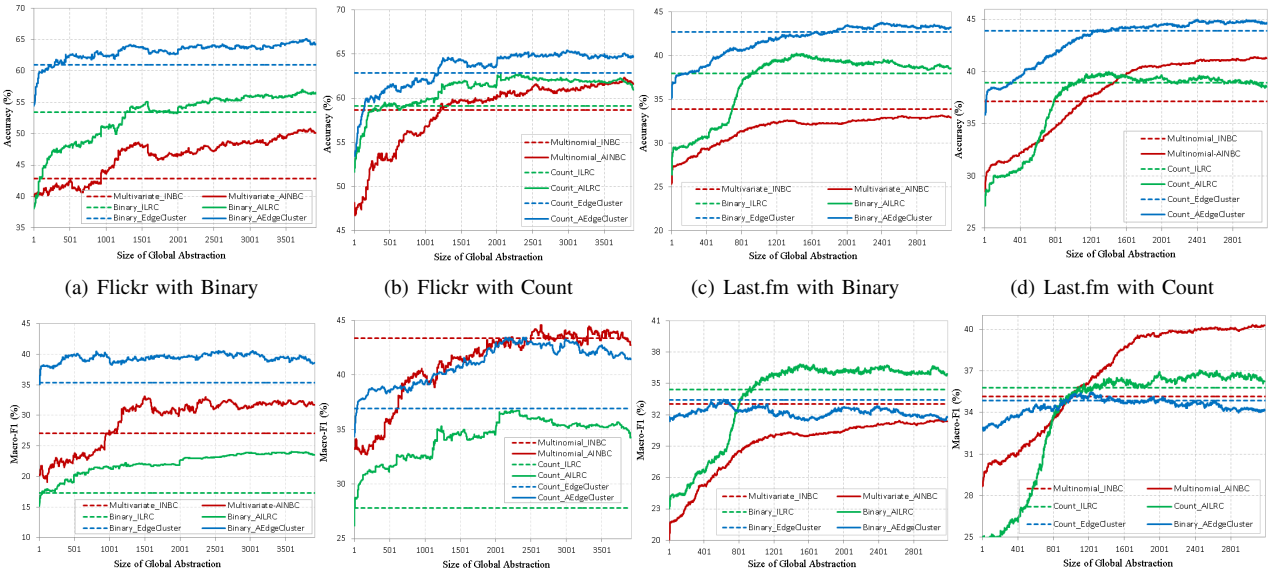
A variety of approaches to labeling nodes in networks have been explored in the literature. These include (i) Approaches



(a) INBC on Flickr (b) ILRC on Flickr (c) EdgeCluster on Flickr

(d) INBC on Last.fm (e) ILRC on Last.fm (f) EdgeCluster on Last.fm

Fig. 2. Accuracy of classifiers that exploit abstraction and classifiers that do not exploit abstraction on two real data sets Flickr (18,021 users) and Last.fm (45,139 users). Figures (a), (b), and (c): accuracies on Flickr of Iterative classification Naïve Bayes Classifier (INBC), Iterative classification Logistic Regression Classifier (ILRC), and EdgeCluster, respectively. Figures (d), (e), and (f): accuracies on Last.fm of INBC, ILRC, and EdgeCluster, respectively.



(a) Flickr with Binary (b) Flickr with Count (c) Last.fm with Binary (d) Last.fm with Count

(e) Flickr with Binary (f) Flickr with Count (g) Last.fm with Binary (h) Last.fm with Count

Fig. 3. Performance comparisons of six classifiers based on two variants of actor representation, Binary (Multivariate) and Count (Multinomial). Figures (a) and (b): accuracies on Flickr with Binary and Count. Figures (c) and (d): accuracies on Last.fm with Binary and Count. Figures (e) and (f): macro-F1 measurements on Flickr with Binary and Count. Figures (g) and (h): macro-F1 measurements on Last.fm with Binary and Count.

that develop a relational learner to classify an actor by iteratively labeling an actor to the majority class of its neighbors [13], [3]; (ii) Approaches that effectively exploit correlations among the labels and attributes of objects [2], [6], [4], [5] that distinguish between and make use of different types of correlations to assign labels to actor objects; (iii) Semi-supervised learning or transductive learning methods [14], [15] which include, among others, random-walk based methods [16], [17] that assign a label to an actor based on the known label(s) of objects represented by node(s) reachable via random walk(s) originating at the node representing the actor. With the exception of some approaches, e.g., RankClass [18], Graffiti [16], and EdgeCluster [7], most of the current approaches to classification of network data focus on $homogeneous$ networks, i.e., networks that consist of a single type of nodes and/or links. RankClass and Graffiti are the approaches that classify objects of all types in the network. In contrast, the focus of this paper is classifying the actor objects of the network based on the features extracted from heterogeneous networks with multiple types of nodes and links.

Using abstraction hierarchies on the problem of learning classifiers has been explored in several studies. Approaches in [19], [20] have made use of hierarchical taxonomies on *classes* to improve the accuracy of classifiers. Others [21], [22] have explored the use of abstract representations to train classifiers in the traditional supervised learning setting. In contrast, the focus of this paper is on learning and exploiting hierarchical groupings of items in social networks that induce abstract representations of social networks to obtain compact and accurate predictors.

Specifically, we have shown that predictors that make use of abstract representations of network data offer a means to learn predictive models that are competitive with, and in some cases, outperform those that do not make use of abstractions using three different prediction methods (INBC, ILRC, and EdgeCluster) with two different real-world data sets. Furthermore, abstract representation provides a way to learn compact predictive models and hence, helps minimize over-fitting.

One limitation of our abstraction-based predictive model is that, while it provides more compact models by reducing the input size when learning a model, the simplicity is achieved at the risk of some information loss due to abstraction. To trade off accuracy of the model against its complexity, it would be useful to augment the algorithm so that it can choose an optimal global abstraction over ITAHs. This can be achieved by designing a scoring function (based on a conditional minimum description length (CMDL) score), similar to [22] in the case of Naïve Bayes, to guide a top-down search for an optimal global abstraction.

Some promising directions for further research include: (i) extending the use of abstract representations to the object-object link model [23]; (ii) reducing the size of the feature space by hashing the very large numbers of items to be clustered to a lower dimensional space using hashing before generating abstract network representations [24]; (iii) incorporation of CMDL-like score for finding an optimal global abstraction.

VI. Acknowledgment.

This research was funded in part by an NSF grant IIS 0711356. The work of Vasant Honavar while working at the National Science Foundation, was supported by the Foundation. The conclusions reported here are those of the authors, and do not necessarily reflect the views of the Foundation.

References

[1] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annual Review of Sociology*, vol. 27, no. 1, pp. 415–444, 2001.

[2] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad, "Collective classification in network data." *AI Magazine.*, pp. 93–106, 2008.

[3] S. A. Macskassy and F. Provost, "Classification in networked data: A toolkit and a univariate case study," *JMLR*, vol. 8, pp. 935–983, 2007.

[4] X. Kong, X. Shi, and P. S. Yu, "Multi-label collective classification," in *SDM*, 2011, pp. 618–629.

[5] H. Eldardiry and J. Neville, "Across-model collective ensemble classification." in *AAAI*, 2011.

[6] Q. Lu and L. Getoor, "Link-based classification," in *ICML*, 2003, pp. 496–503.

[7] L. Tang and H. Liu, "Scalable learning of collective behavior based on sparse social dimensions," in *CIKM*, 2009, pp. 1107–1116.

[8] H. Becker, M. Naaman, and L. Gravano, "Learning similarity metrics for event identification in social media," in *WSDM*, 2010, pp. 291–300.

[9] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambrige, Massachusetts, 1999.

[10] E. Y. Cheu, C. K. Kwoh, and Z. Zhou, "On the two-level hybrid clustering algorithm," in *AISAT*, 2004, pp. 138–142.

[11] I. S. Dhillon and D. S. Modha, "A data-clustering algorithm on distributed memory multiprocessors," in *Large-Scale Parallel Data Mining*, 1999, pp. 245–260.

[12] A. McCallum and K. Nigam, "A comparison of event models for naïve bayes text classification," in *AAAI Workshop on Learning for Text Categorization*, 1998.

[13] S. A. Macskassy and F. Provost, "A simple relational classifier," in *MRDM Workshop at KDD-2003*, 2003, pp. 64–76.

[14] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schï¿œlkopf, "Learning with local and global consistency," in *NIPS*, 2004, pp. 321–328.

[15] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *ICML*, 2003, pp. 912–919.

[16] R. Angelova, G. Kasneci, and G. Weikum, "Graffiti: graph-based classification in heterogeneous networks," *World Wide Web Journal*, 2011.

[17] F. Lin and W. W. Cohen, "Semi-supervised classification of network data using very few labels," in *ASONAM*, 2010, pp. 192–199.

[18] M. Ji, J. Han, and M. Danilevsky, "Ranking-based classification of heterogeneous information networks," in *KDD*, 2011, pp. 1298–1306.

[19] S. Dumais and H. Chen, "Hierarchical classification of web content," in *SIGIR*, 2000, pp. 256–263.

[20] A. McCallum, R. Rosenfeld, T. M. Mitchell, and A. Y. Ng, "Improving text classification by shrinkage in a hierarchy of classes," in *ICML*, 1998, pp. 359–367.

[21] D. L. Baker and A. K. McCallum, "Distributional clustering of words for text classification," in *SIGIR*, 1998, pp. 96–103.

[22] J. Zhang, D.-K. Kang, A. Silvescu, and V. Honavar, "Learning accurate and concise naïve bayes classifiers from attribute value taxonomies and data," *Knowl. Inf. Syst.*, vol. 9, pp. 157–179, 2006.

[23] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, "Mixed membership stochastic blockmodels," *JMLR*, vol. 9, pp. 1981–2014, 2008.

[24] C. Caragea, A. Silvescu, and P. Mitra, "Combining hashing and abstraction in sparse high dimensional feature spaces," in *AAAI*, 2012.