A Multilevel Inference Mechanism for User Attributes over Social Networks

Hang Zhang[1,2], Yajun Yang[1,2(✉)], Xin Wang[1], Hong Gao[3], Qinghua Hu[1,2], and Dan Yin[4]

[1] College of Intelligence and Computing, Tianjin University, Tianjin, China
{aronzhang,yjyang,wangx,huqinghua}@tju.edu.cn
[2] State Key Laboratory of Communication Content Cognition, Beijing, China
[3] Harbin Institute of Technology, Harbin, China
honggao@hit.edu.cn
[4] Harbin Engineering University, Harbin, China
yindan@hrbeu.edu.cn

Abstract. In a real social network, each user has attributes for self-description called user attributes which are semantically hierarchical. With these attributes, we can implement personalized services such as user classification and targeted recommendations. Most traditional approaches mainly focus on the flat inference problem without considering the semantic hierarchy of user attributes which will cause serious inconsistency in multilevel tasks. To address these issues, in this paper, we propose a cross-level model called IWM. It is based on the theory of maximum entropy which collects attribute information by mining the global graph structure. Meanwhile, we propose a correction method based on the predefined hierarchy to realize the mutual correction between different layers of attributes. Finally, we conduct extensive verification experiments on the DBLP data set and it has been proved that compared with other algorithms, our method has a superior effect.

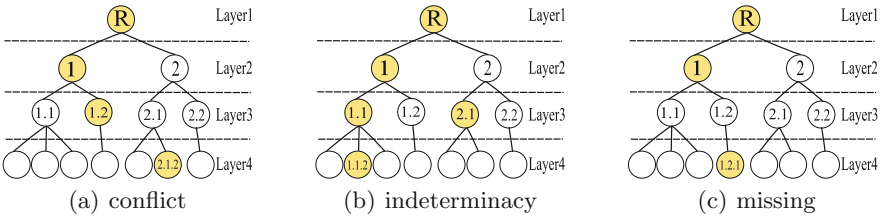
Keywords: Attribute inference · Multilevel inference · Social network

1 Introduction

In a social network, each user has a series of labels used to describe their characteristics called user attributes. However, for a certain type of attributes, they are not flat but hierarchical. The most existing methods [4,5] mainly focus on the single-level attribute inference and it will bring some problems for hierarchical structures as shown in Fig. 1. Even though utilizing the same method for every single-level, the attributes of different level may be conflicted for the same user, attributes at the same level may be indeterminate, and the results of a certain layer may be missing.

In this paper, we propose a multi-level inference model named IWM to solve the problems mentioned above. This model can infer hierarchical attributes for unknown users by collecting attributes from nearby users under maximum

C. S. Jensen et al. (Eds.): DASFAA 2021, LNCS 12682, pp. 710–718, 2021.
https://doi.org/10.1007/978-3-030-73197-7_49



(a) conflict (b) indeterminacy (c) missing

Fig. 1. Problems of labeling in real social networks

entropy random walk. Meanwhile, we propose a correction method based on the predefined hierarchy of attributes to revise the results. Finally, we conduct the experiments on real datasets to validate the effectiveness of our method.

The rest of the paper is organized as follows. Section 2 defines the problem. Section 3 proposes the multilevel inference model. Algorithm is given in Sect. 4. The experimental results and analysis are presented in Sect. 5. The related works are introduced in Sect. 6. Finally, we conclude this paper in Sect. 7.

2 Problem Definition

2.1 Semantic Tree

The semantic tree T is a predefined structure which is semantically exists used to describe the hierarchical relationship between different user attributes. We use T_g to represent the user attributes at T's gth layer.

2.2 Labeled Graph

Labeled graph is a simple undirected graph, denoted as $G = (V, E, T, L)$, where V is the set of vertices and E is the set of edges. T is the semantic tree of attributes in G. L is a function mapping V to a cartesian product of the attributes in T defined as $L : V \rightarrow T_1 \times T_2 \times \cdots \times T_m$, where m is the depth of T.

Problem Statement: Given a labeled graph $G(V, E, T, L)$ and labeled vertices set $V_s \subset V$, where V_s is the set of vertices with complete attributes. So for every vertex $v_s \in V_s, L(v_s) = \{l_1, l_2, \cdots, l_m\}$, where $l_1 \in T_1, l_2 \in T_2, \cdots, l_m \in T_m$. The input of the problem is $L(v_s)$ for every vertex $v_s \in V_s$ and the output is $L(v_u)$ for every vertex $v_u \in V_u$, where $V_u = V - V_s$.

3 Attribute Inference Model

Our attribute inference model can be divided into two parts. The first part is called the information propagation model. Based on the maximum entropy theory and one step random walk, vertices in V_s spread their own attributes to other vertices layer by layer. The second part is a correction model based on the semantic tree. This model realizes the mutual correction between different layers of attributes. These two models are described in detail below.

3.1 Information Propagation Model

The information propagation model is an extension of the model proposed in [7]. The main idea is that the higher the entropy value of the vertex, the stronger the uncertainty of its own user attributes, so more information should be collected. The attributes of v_j's each layer can be represented by $L_g(v_j) = \{l_x, w_x(v_j), l_x \in T_g\}$. Then the entropy value of v_j's gth layer $H_g(v_j)$ can be calculated as blow.

$$H_g(v_j) = -\sum_{l_x \in T_g} w_x(v_j) \times \ln w_x(v_j) \tag{1}$$

If v_i is a neighbor of v_j, then the transition probability $P_g(v_i, v_j)$ from v_i to v_j at gth layer is computed as follows.

$$P_g(v_i, v_j) = \frac{H_g(v_j)}{\sum_{v_j \in N(v_i)} H_g(v_j)} \tag{2}$$

Where $N(v_i)$ is the set of neighbors of v_j.

Next, we use the following equation to normalize the attribute probability obtained by different vertices.

$$w_x(v_j) = \frac{\sum_{v_i \in N(v_j)} P_g(v_i, v_j) \times w_x(v_i)}{\sum_{l_y \in T_g} \sum_{v_i \in N(v_j)} P_g(v_i, v_j) \times w_y(v_i)} \tag{3}$$

$L_g(v_j)$ will be updated through $w_x(v_j)$. In this way, the attribute information is spread hierarchically in the graph.

3.2 Attribute Correction Model

The formal definitions of the concepts involved in this section are given below.

Definition 1. *Define the following relationships in the semantic tree:*

(1) If x_2 is a child node of x_1, then x_1, x_2 have a relationship called $Child(x_1, x_2)$.
(2) Say that x_1, x_2 have a descendant relationship called $Descendant(x_1, x_2)$,if $Child(x_1, x_2) \cup \exists x_3(Child(x_1, x_3) \cap Descendant(x_3, x_2))$.
(3) If x_2 is a brother node of x_1, then x_1, x_2 have a relationship called $Brother(x_1, x_2)$.

Definition 2 ***(Descendant vertex set).*** *For a node x_1, its descendant node set is defined as $DesSet(x_1) = \{x|Descendant(x_1, x)\}$.*

Definition 3 ***(Brother vertex set).*** *For a node x_1, its brother node set is defined as $BroSet(x_1) = \{x|Brother(x_1, x)\}$.*

For the attribute l_x in the middle layer of the semantic tree, its existence depends on both $Parent(x)$ and $DesSet(x)$, so $w_x(v_j)$ can be corrected by Eq. (4).

$$w_x(v_j) = w_{Parent(x)}(v_j) \times \frac{(1-\alpha) \times w_x(v_j) + \alpha \times \sum_{y \in DesSet(x)} w_y(v_j)}{\sum_z (1-\alpha) \times w_z(v_j) + \alpha \times \sum_{y \in DesSet(z)} w_y(v_j)} \quad (4)$$

where $z \in BroSet(x)$ and α represents a correction strength. When the value of α is large, the result is inclined to the hierarchy of the semantic tree, otherwise, it is more inclined to the information collected by propagation.

There is another case that the highest layer attributes don't have any child node, so they can be corrected as follows.

$$w_x(v_j) = w_{Parent(x)}(v_j) \times \frac{w_x(v_j)}{\sum_{z \in BroSet(x)} w_z(v_j)} \quad (5)$$

4 Attribute Inference Algorithm

4.1 Algorithm Description

The detailed steps of the algorithm are shown in Algorithm 1. Firstly, we use Eq. (1) to calculate entropy $H_g(v_u)$ for all $v_u \in V_u$ layer by layer (line 1 to 3). Line 4 to 9 start inferring hierarchically. After all layers' information are collected, correction can be performed by Eq. (4) or Eq. (5) (line 10 to 11).

Algorithm 1. Cross-level Attribute Inference(G, V_s)

Input: $G(V, E, T, L)$ and V_s.
Output: $L(v_u)$ for every vertex $v_u \in V_u$.

1: **for** every layer g in T **do**
2: **for** every vertex $v_u \in V_u$ **do**
3: compute $H_g(v_u)$
4: **for** every vertex $v_u \in V_u$ **do**
5: **for** every layer g in T **do**
6: **for** every vertex $v_i \in N(v_u)$ **do**
7: compute $P_g(v_i, v_u)$
8: **for** every attribute $l_x \in T_g$ **do**
9: compute $w_x(v_u)$
10: **for** every attribute $l_x \in T$ **do**
11: correct $w_x(v_u)$
12: **if** $\sum_{v_u \in V_u} \sum_{l_x \in T} |diffw_x(v_j)| \leq |V_u| \times |T| \times \sigma$ **then**
13: **return** $L(v_u)$ for every vertex $v_u \in V_u$
14: **else**
15: **return** step 1

The algorithm terminates when the convergence is satisfied. The condition of convergence is given by the following equation.

$$\sum_{v_u \in V_u} \sum_{l_x \in T} |diffw_x(v_u)| \leq |V_u| \times |T| \times \sigma \tag{6}$$

where $diff(w_x(v_u))$ is the difference on $w_x(v_u)$ after the inference algorithm is executed, and σ is a threshold to control the number of iterations.

4.2 Time Complexity

We assume that the labeled graph G has n vertices and p attributes, the semantic tree has m layers. So the time complexity of information propagation is $O(m|V_u| + mnd + pnd) = O(mnd + pnd)$, where d is the average degree of all the vertices in G. After that, we need to modify every attribute for each user by the complexity of $O(pn)$. To sum up, the total time complexity of our algorithm for one iteration is $O(mnd + pn)$.

5 Experiment

The experiments are performed on a Windows 10 PC with Intel Core i5 CPU and 8 GB memory. Our algorithms are implemented in Python 3.7. The default parameter values in the experiment are $\alpha = 0.5$, $\sigma = 0.0001$.

5.1 Experimental Settings

Dataset. We will study the performance on DBLP dataset. DBLP is a computer literature database system. Each author is a vertex and their research field is used as the attributes to be inferred. We extract 63 representative attributes and define a 4-layer semantic tree in advance.

Baselines and Evaluation Metrics. We compare our method IWM with three classic attribute inference baselines which are SVM, Community Detection (CD) [6] and Traditional Random Walk (TRW) [7].

We use five commonly metrics to make a comprehensive evaluation of the inference results. The calculation method of these metrics are shown below.

$$Precison = \frac{\sum_{l \in T} |\{v_u | v_u \in V_u \wedge l \in Predict(v_u) \cap Real(v_u)\}|}{\sum_{l \in T} |\{v_u | v_u \in V_u \wedge l \in Predict(v_u)\}|} \tag{7}$$

$$Recall = \frac{\sum_{l \in T} |\{v_u | v_u \in V_u \wedge l \in Predict(v_u) \cap Real(v_u)\}|}{\sum_{l \in T} |\{v_u | v_u \in V_u \wedge l \in Real(v_u)\}|} \tag{8}$$

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{9}$$

$$Accuracy = \frac{1}{|V_u|} \times |\{v_u | v_u \in V_u \wedge Predict(v_u) = Real(v_u)\}| \tag{10}$$

$$Jaccard = \frac{1}{|V_u|} \times \sum_{v_u \in V_u} \frac{|Predict(v_u) \cap Real(v_u)|}{|Predict(v_u) \cup Real(v_u)|} \tag{11}$$

where $Predict(v_u)$ and $Real(v_u)$ respectively represent the inference result set and real original attribute set of v_u. For all metrics, the larger value means the better performance.

5.2 Results and Analysis

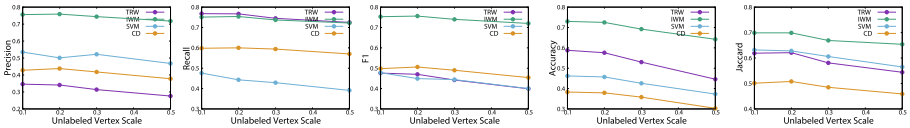
Exp1-Impact of Vertex Size. We conduct the first experiment in coauthor relationship networks with 5,000, 10,000, 20,000, and 40,000 vertices. The proportion of unknown vertices is 30% (Table 1).

Table 1. Inference performance on different vertex size.

Vertex size	Method	Precision				Recall				F1				Mean-Acc	Jaccard
		Layer2	Layer3	Layer4	Mean-Pre	Layer2	Layer3	Layer4	Mean-Rec	Layer2	Layer3	Layer4	Mean-F1		
5000	SVM	0.6410	0.5460	0.4700	0.5640	0.5630	0.5070	0.4450	0.5260	0.5810	0.5100	0.4300	0.5200	0.5021	0.6611
	CD	0.8428	0.6347	0.2117	0.4384	0.8307	0.6839	0.5718	0.6949	0.8364	0.6581	0.3078	0.5368	0.5180	0.5888
	TRW	0.8721	0.6423	0.6423	0.4099	**0.8754**	**0.7554**	**0.7317**	**0.7867**	0.8735	0.6931	0.3153	0.5377	0.6171	0.6446
	IWM	**0.9552**	**0.8310**	**0.8310**	**0.7773**	0.8629	0.7518	0.6867	0.7666	**0.9067**	**0.7892**	**0.6364**	**0.7718**	**0.7604**	**0.7187**
10000	SVM	0.8070	0.5800	0.4870	0.5180	0.6090	0.4860	0.4400	0.4480	0.6640	0.4990	0.4340	0.4490	0.4650	0.6314
	CD	0.7852	0.6427	0.2074	0.4488	0.7591	0.6106	0.4596	0.6103	0.7720	0.6259	0.2848	0.5164	0.3871	0.5109
	TRW	0.8309	0.6388	0.6388	0.3505	**0.8632**	0.7269	**0.7099**	**0.7653**	0.8466	0.6798	0.2583	0.4803	0.5815	0.6181
	IWM	**0.9492**	**0.8373**	**0.8373**	**0.7655**	0.8465	**0.7354**	0.6769	0.7526	**0.8949**	**0.7830**	**0.6170**	**0.7591**	**0.7288**	**0.7003**
20000	SVM	0.7400	0.5440	0.4460	0.5220	0.5320	0.4620	0.3920	0.4290	0.5820	0.4730	0.3980	0.4440	0.4260	0.6058
	CD	0.7602	0.6099	0.1888	0.4176	0.7332	0.6020	0.4423	0.5935	0.7463	0.6053	0.2634	0.4895	0.3579	0.4848
	TRW	0.8294	0.6063	0.6063	0.3143	**0.8392**	**0.7418**	**0.6817**	**0.7446**	0.8342	0.6561	0.2243	0.4418	0.5296	0.5810
	IWM	**0.9396**	**0.8170**	**0.8170**	**0.7436**	0.8372	0.7218	0.6526	0.7372	**0.8854**	**0.7664**	**0.5895**	**0.7403**	**0.6924**	**0.6688**
40000	SVM	0.7489	0.6167	0.4311	0.4850	0.4811	0.4522	0.3589	0.3950	0.5444	0.4911	0.3622	0.4050	0.3378	0.5473
	CD	0.7458	0.5333	0.1928	0.4547	0.6855	0.4669	0.2568	0.4710	0.7143	0.4979	0.2200	0.4626	0.1579	0.3797
	TRW	0.8093	0.5888	0.5888	0.2870	**0.8347**	0.7059	**0.6629**	**0.7340**	0.8214	0.6419	0.2006	0.4125	0.4652	0.5572
	IWM	**0.9360**	**0.8061**	**0.8061**	**0.7270**	0.8344	**0.7169**	0.6349	0.7284	**0.8817**	**0.7587**	**0.5667**	**0.7276**	**0.6561**	**0.6642**

It is obvious that our method shows the best performance on different evaluation indicators. For examplewhen it comes to a 20,000 vertices network, our model improves over the strongest baseline 22.2%, 35.1%, 16.3% and 6.3% on Precision, F1, Accuracy, and Jaccard index, separately. In terms of recall, our method does not have obvious advantages over TRW.

Exp2-Impact of the Proportion of Unknown Vertices. In Exp2 the vertex scale of the network is 20,000 and we set the unlabeled scale 10%, 20%, 30%, and 50% respectively.



(a) Precision (b) Recall (c) F1 (d) Accuracy (e) Jaccard

Fig. 2. Inference performance on different proportion of unknown vertices

We can analyze the results to get that as the proportion of unknown vertices increases, the decline tendency of our method is much slower than other methods. It is interesting to see that the five evaluate indicators of our method are 71.77%, 72.17%, 71.96%, 64.21% and 65.43% at the condition of 50% vertices lack of attributes which can show that it has a great value in practical applications.

Exp3-Real Case Study. In Table 2 we present partial results of the experiment which gives a clear comparison between our method and TRW. We use these examples to demonstrate the effectiveness of our method.

Table 2. Comparison of inference results by TRW and IWM.

Author	True label			TRW result			IWM result		
	Layer2	Layer3	Layer4	Layer2	Layer3	Layer4	Layer2	Layer3	Layer4
Chris Stolte	**Data**	Database	Query	**Unknown**	Database	Query	**Data**	Database	Query
Marcel Kyas	Network	**Wireless**	Localization	Network	**Databse**	Localization	Network	**Wireless**	Localization
William Deitrick	**Data**	Mining	Clusters	**Network, Data**	Unknown	Clusters	**Data**	Mining	Clusters
V. Dhanalakshmi	Learning	Language	Extraction	Unknown	Classification	Speech	Learning	Language	Speech

For Chris Stolte, IWM can complement the missing information which can't be inferred by TRW. For Marcel Kyas, our method modify the error information on Layer 3 and obtain the correct result. TRW causes indeterminacy problem on Layer2 of William Deitrick, while IWM can select more relevant attributes. However, for V. Dhanalakshmi, due to its special structure, when most of the collected information is interference, IWM can't make correct inference either.

6 Related Work

There has been an increasing interest in the inference of single-layer user attributes over the last several years.

Firstly, based on resource content there are [1,11] which utilize the user's text content for inference. [3] constructs a social-behavior-attribute network and design a vote distribution algorithm to perform inference. There are also methods based on the analysis of graph structure such as Local Community Detection [6] and Label Propagation [12]. [10] discovers the correlation between item recommendation and attribute reasoning, so they use an Adaptive Graph Convolutional Network to joint these two tasks. However, these methods don't explore

the relationship existing in the attribute hierarchy, which will greatly reduce the effectiveness in our multilevel problem.

Another method is to build a classifier to treat the inference problem as a multilevel classification problem. [2] trains a binary classifier for each attribute. [8] trains a multi-classifier for each parent node in the hierarchy. [9] trains a classifier for each layer in the hierarchical structure, and use it in combination with [8] to solve the inconsistency. However, classifier-based approaches have a high requirement for data quality. It will make the construction of the classifier complicated and the amount of calculation for training is huge.

7 Conclusion

In this paper, we study the multilevel user attribute inference problem. We first define the problem and propose the concept of semantic tree and labeled graph. We present a new method to solve this problem. The information propagation model is proposed to collect attributes for preliminary inference. The attribute correction model is proposed to conduct a cross-level correction. Experimental results on real-world data sets have demonstrated the superior performance of our new method. In future work, we will improve our method for multi-category attributes and do more works on optimizing the algorithm to save more time.

Acknowledgement. This work is supported by the National Key Research and Development Program of China No. 2019YFB2101903, the State Key Laboratory of Communication Content Cognition Funded Project No. A32003, the National Natural Science Foundation of China No. 61702132 and U1736103.

References

1. Choi, D., Lee, Y., Kim, S., Kang, P.: Private attribute inference from facebook's public text metadata: a case study of korean users. Ind. Manage. Data Syst. **117**(8), 1687–1706 (2017)
2. Fagni, T., Sebastiani, F.: Selecting negative examples for hierarchical text classification: an experimental comparison. J. Am. Soc. Inf. Sci. Technol. **61**(11), 2256–2265 (2010)
3. Gong, N.Z., Liu, B.: Attribute inference attacks in online social networks. ACM Trans. Priv. Secur. **21**(1), 3:1–3:30 (2018)
4. Krestel, R., Fankhauser, P., Nejdl, W.: Latent dirichlet allocation for tag recommendation. In: Proceedings of the 2009 ACM Conference on Recommender Systems, pp. 61–68. ACM (2009)
5. Lu, Y., Yu, S., Chang, T., Hsu, J.Y.: A content-based method to enhance tag recommendation. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence, pp. 2064–2069 (2009)
6. Mislove, A., Viswanath, B., Gummadi, P.K., Druschel, P.: You are who you know: inferring user profiles in online social networks. In: Proceedings of the Third International Conference on Web Search and Web Data Mining, pp. 251–260. ACM (2010)

7. Pan, J., Yang, Y., Hu, Q., Shi, H.: A label inference method based on maximal entropy random walk over graphs. In: Li, F., Shim, K., Zheng, K., Liu, G. (eds.) APWeb 2016. LNCS, vol. 9931, pp. 506–518. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45814-4_41
8. Secker, A.D., Davies, M.N., Freitas, A.A., Timmis, J., Mendao, M., Flower, D.R.: An experimental comparison of classification algorithms for hierarchical prediction of protein function. Expert Update (Mag. Br. Comput. Soc. Spec. Group AI) **9**(3), 17–22 (2007)
9. Taksa, I.: David Taniar: research and trends in data mining technologies and applications. Inf. Retr. **11**(2), 165–167 (2008)
10. Wu, L., Yang, Y., Zhang, K., Hong, R., Fu, Y., Wang, M.: Joint item recommendation and attribute inference: an adaptive graph convolutional network approach. In: Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, pp. 679–688. ACM (2020)
11. Yo, T., Sasahara, K.: Inference of personal attributes from tweets using machine learning. In: 2017 IEEE International Conference on Big Data, pp. 3168–3174. IEEE Computer Society (2017)
12. Zhu, X., Ghahramani, Z.: Learning from labeled and unlabeled data with label propagation (2003)