Volume 4, Issue 5, May 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Tactical Implementation of Label Propagation in Social Corporate Network Using Graph Mining Technique

Dr.S.P.Victor*
HOD-CS-St.Xaviers College
India

S. A. AMALA NIRMAL DOSS
Research Scholar,MS University,Tirunelvel
India

Abstract-The Graph mining offers a convenient way to study structured datum with different level of implications. Our conventional setup initially focuses with dataset and its entity. This paper perform a detailed study of classified datum of label propagation towards variant clusters in the field of graph mining which can be carried out with unknown to known prediction strategies. We will implement our integrated graph mining techniques with real time implementation of Corporate Social network Domains. We will also perform survey analysis strategies for the successful implementation of our proposed research technique in several sampling domains with a maximum level of improvements. In near future we will implement the cluster mining techniques for predicting the Graph sub structure behaviors.

Keywords- Graph Mining, cluster, survey, prediction, corporate

I. INTRODUCTION

Classification Algorithms for Graph Data Classification is a central task in data mining and machine learning. As graphs are used to represent entities and their relationships in an increasing variety of applications, the topic of graph classification has attracted much attention in both academia and industry. For example, in pharmaceutics and drug design, we are interested to know the relationship between the activity of a chemical compound and the structure of the compound, which is represented by a graph. In social network analysis, we study the relationship between the health of a community (e.g., whether it is expanding or shrinking) and its structure, which again is represented by graphs. Graph classification is concerned with two different but related learning tasks.

Label Propagation. A subset of nodes in a graph is labeled. The task is to learn a model from the labeled nodes and use the model to classify the unlabeled nodes. Graph classification. A subset of graphs in a graph dataset is labeled. The task is to learn a model from the labeled graphs and use the model to classify the unlabeled graphs. The concept of label or belief propagation is a fundamental technique which is used in order to leverage graph structure in the context of classification in a number of relational domains. The scenario of label propagation occurs in many applications. As an example, social network analysis is being used as a mean for targeted marketing. Retailers track customers who have received promotions from them. Those customers who respond to the promotion (by making a purchase) are labeled as positive nodes in the graph representing the social network, and those who do not respond are labeled as negative.

The goal of target marketing is to send promotions to customers who are most likely to respond to promotions. It boils down to learning a model from customers who have received promotions and predicting the responses of other potential customers in the social network. Intuitively, we want to find out how existing positive and negative labels propagate in the graph to unlabeled nodes. Based on the assumption that “similar” nodes should have similar labels, the core challenge for label propagation lies in devising a distance function that measures the similarity between two nodes in the graph.

One common approach of defining the distance between two nodes is to count the average number of steps it takes to reach one node from the other using a random walk. However, it has a significant drawback: it takes O (n3) time to derive the distances and O (n2) space to store the distances between all pairs. However, many graphs in real life applications are sparse, which reduces the complexity of computing the distance. For example, Zhou et al introduces a method whose complexity is nearly linear to the number of non-zero entries of the sparse coefficient matrix.

II. PROPOSED METHODOLOGY

This proposed methodology focuses on the implementation of a node clustering algorithmic strategy to predict the unknown node behaviors by implementing the cluster computations. .

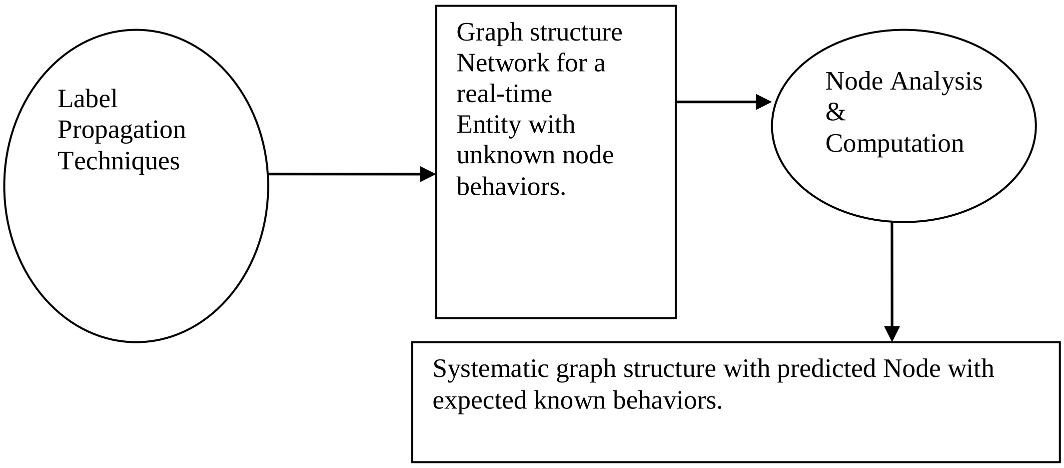

Fig 1.1: Proposed Graph mining structure

Implementation of Algorithmic strategies.

Consider the sample network with unknown node behaviors as follows, the network contains 28 nodes with 4 levels (0, 1, 2 and 3).Each node works well and earns their clients as child based on their promotional credit (P).But some nodes are not function well due to its Non promotional credit (NP) also with exceptions.

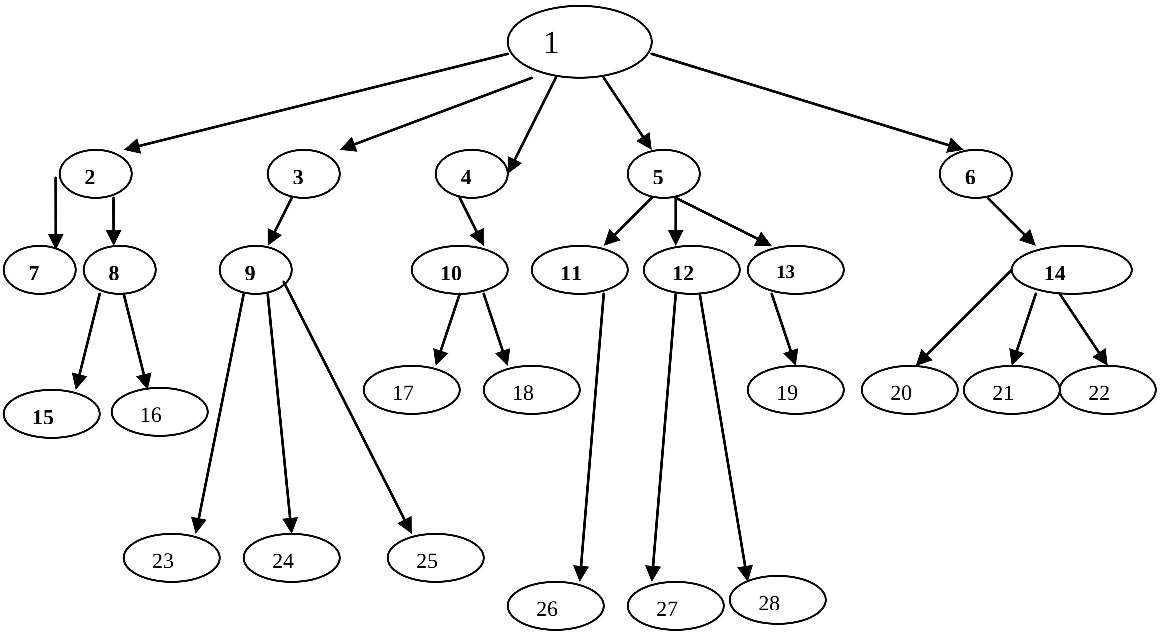

Fig 1.2: Social Corporate network graph with 4 levels

The following table illustrates the analysis of our graph in figure 1.2 where the degree represents the number of child .the promotional credit and non promotional credits are represented by P and NP towards eachnode.The response can be computed as Max=5 children,>Average=4 Children, Average=3 children,<Average=2,poor=1 child and worst = no progress. The level represents the node level in the graph. Sibling succession represents the maximum yield compared with same parent; cousin succession represents the maximum yield compared with other parent nodes. The future prediction of promotional credits for the default nodes are computed through degree responses but for the remaining nodes are unknown which will be computed by the proposed algorithmic strategy.

Table 1.1: Promotional Credit Table with unknown values

Node number	Degree	Promotion	Response	Level	Sibling Succession	Cousin Succession	Future Prediction
1	5	---	Max	0	---	---	5P
2	2	P	<Average	1	No	---	2P
3	1	NP	Poor	1	No	---	P
4	1	NP	Poor	1	No	---	P
5	3	P	Average	1	Yes	---	3P
6	1	NP	Poor	1	No	---	P

7	0	NP	Worst	2	No	No	NP
8	2	P	<Average	2	Yes	No	3P
9	3	P	Average	2	---	Yes	3P
10	2	P	<Average	2	---	No	2P
11	1	NP	Poor	2	No	No	P
12	2	P	<Average	2	Yes	No	2P
13	1	NP	Poor	2	No	No	P
14	3	P	Average	2	---	Yes	3P
15	0	NP	---	3	---	---	X15
16	0	NP	---	3	---	---	X16
17	0	NP	---	3	---	---	X17
18	0	NP	---	3	---	---	X18
19	0	NP	---	3	---	---	X19
20	0	NP	---	3	---	---	X20
21	0	NP	---	3	---	---	X21
22	0	NP	---	3	---	---	X22
23	0	NP	---	3	---	---	X23
24	0	NP	---	3	---	---	X24
25	0	NP	---	3	---	---	X25
26	0	NP	---	3	---	---	X26
27	0	NP	---	3	---	---	X27
28	0	NP	---	3	---	---	X28

Find the values for X15, X16, and X28 based on the following algorithmic strategies,

Start

Let X (i) =NP
1. for each node i, traverse to Parent (i),
If Degree(Parent(i)) >=3 then X(i) =X(i)+ 2P else X(i)=X(i)+ P

2. for each node i, traverse to Parent (i),
If Sibling Succession (Parent (i)) = Yes then X (i)-X (i) +2P else No change

3. for each node i, traverse to Parent (i),
If Cousin Succession (Parent (i)) = Yes then X (i)-X (i) +3P else No change

4. for each node i, traverse to Parent (i) and Grandparent (i)
If Grand Parent (i) =NP & Parent (i) =P and Degree (Parent (i))>=2 then X (i) =X (i) + 4P

Update X (i)

Stop

III. Computation and Evaluation

For the node “15”

X15=NP
1. Parent of “15” is node “8” degree of “8”=2 therefore X (15) =P
2. Sibling Succession (“8”) =Yes therefore X (15) =P+2P=3P
3. Cousin succession (“8”) =No Therefore X (15) =3P itself No change
4. Grandparent (“15”) =”2” =P only No change

Updation X (15) =3P

Similarly for the node “16” also X (16) =3P

For the node “17”

X17=NP
1. Parent of “17” is node “10” degree of “10”=2 therefore X (17) =P
2. Sibling Succession (“10”) =No therefore X (17) =P itself No change

3. Cousin succession (“10”) =No Therefore X (17) =P itself No change
4. Grandparent (“17”) =”4” =NP Parent (“17”) =”10” therefore X (17) =P+4P=5P

Updation X (17) =5P

Similarly for the node “18” also X (18) =5P

For the node “19”

X19=NP
1. Parent of “19” is node “13” degree of “13”=1 therefore X (19) =P
2. Sibling Succession (“13”) =No therefore X (19) =P itself No change
3. Cousin succession (“13”) =No Therefore X (19) =P itself No change
4. Grandparent (“19”) =”5” =P therefore X (19) =P

Updation X (19) =P

For the node “20”

X20=NP
1. Parent of “20” is node “14” degree of “14”=3 therefore X (20) =2P
2. Sibling Succession (“14”) =No therefore X (20) =P itself No change
3. Cousin succession (“14”) =Yes Therefore X (20) =2P+3P=5P
4. Grandparent (“20”) =”6” =NP, Parent (20) =14=P therefore X (20) =5P+4P=9P

Updation X (“20”) =9P

Similarly for the nodes 21, 22 X (21) =9P, X (22) =9P.

For the node “23”

X23=NP
1. Parent of “23” is node “9” degree of “9”=3 therefore X (23) =2P
2. Sibling Succession (“9”) =No therefore X (23) =P itself No change
3. Cousin succession (“9”) =Yes Therefore X (23) =2P+3P=5P
4. Grandparent (“23”) =”6” =NP, Parent (“23”) =”9”=P therefore X (20) =5P+4P=9P

Updation X (“23”) =9P

Similarly for the nodes 24, 25 X (24) =9P, X (25) =9P.

For the node “26”

X26=NP
1. Parent of “26” is node “11” degree of “11”=1 therefore X (26) =P
2. Sibling Succession (“11”) =No therefore X (26) =P itself No change
3. Cousin succession (“11”) =No Therefore X (26) =P itself No change
4. Grandparent (“26”) =”5” =P, therefore X (26) =P itself No change

Updation X (“26”) =P

For the node “27”

X27=NP
1. Parent of “27” is node “12” degree of “12”=2 therefore X (27) =P
2. Sibling Succession (“12”) =Yes therefore X (27) =P+2P=3P
3. Cousin succession (“12”) =No therefore X (27) =3P itself No change
4. Grandparent (“27”) =”5” =P, therefore X (27) =3P itself No change

Updation X (“27”) =3P

Similarly for the node 28 X (28) =3P.

Table 1.2: Promotinal credit table with predicted values using proposed algorithm.

Node number	Degree	Promotion	Response	Level	Sibling Succession	Cousin Succession	Future Prediction
1	5	---	Max	0	---	---	5P
2	2	P	<Average	1	No	---	3P
3	1	NP	Poor	1	No	---	P
4	1	NP	Poor	1	No	---	P
5	3	P	Average	1	Yes	---	3P
6	1	NP	Poor	1	No	---	P
7	0	NP	Worst	2	No	No	NP
8	2	P	<Average	2	Yes	No	2P
9	3	P	Average	2	---	Yes	3P
10	2	P	<Average	2	---	No	2P
11	1	NP	Poor	2	No	No	P
12	2	P	<Average	2	Yes	No	2P
13	1	NP	Poor	2	No	No	P
14	3	P	Average	2	---	Yes	3P
15	0	NP	---	3	---	---	3P
16	0	NP	---	3	---	---	3P
17	0	NP	---	3	---	---	5P
18	0	NP	---	3	---	---	5P
19	0	NP	---	3	---	---	P
20	0	NP	---	3	---	---	9P
21	0	NP	---	3	---	---	9P
22	0	NP	---	3	---	---	9P
23	0	NP	---	3	---	---	9P
24	0	NP	---	3	---	---	9P
25	0	NP	---	3	---	---	9P
26	0	NP	---	3	---	---	P
27	0	NP	---	3	---	---	3P
28	0	NP	---	3	---	---	3P

IV. Results and Discussion:

The implementation of our proposed methodology computes the expectation of node behaviors in a predictable way. The final net work may obtain the following structures if implemented in an optimistic approach as follows,

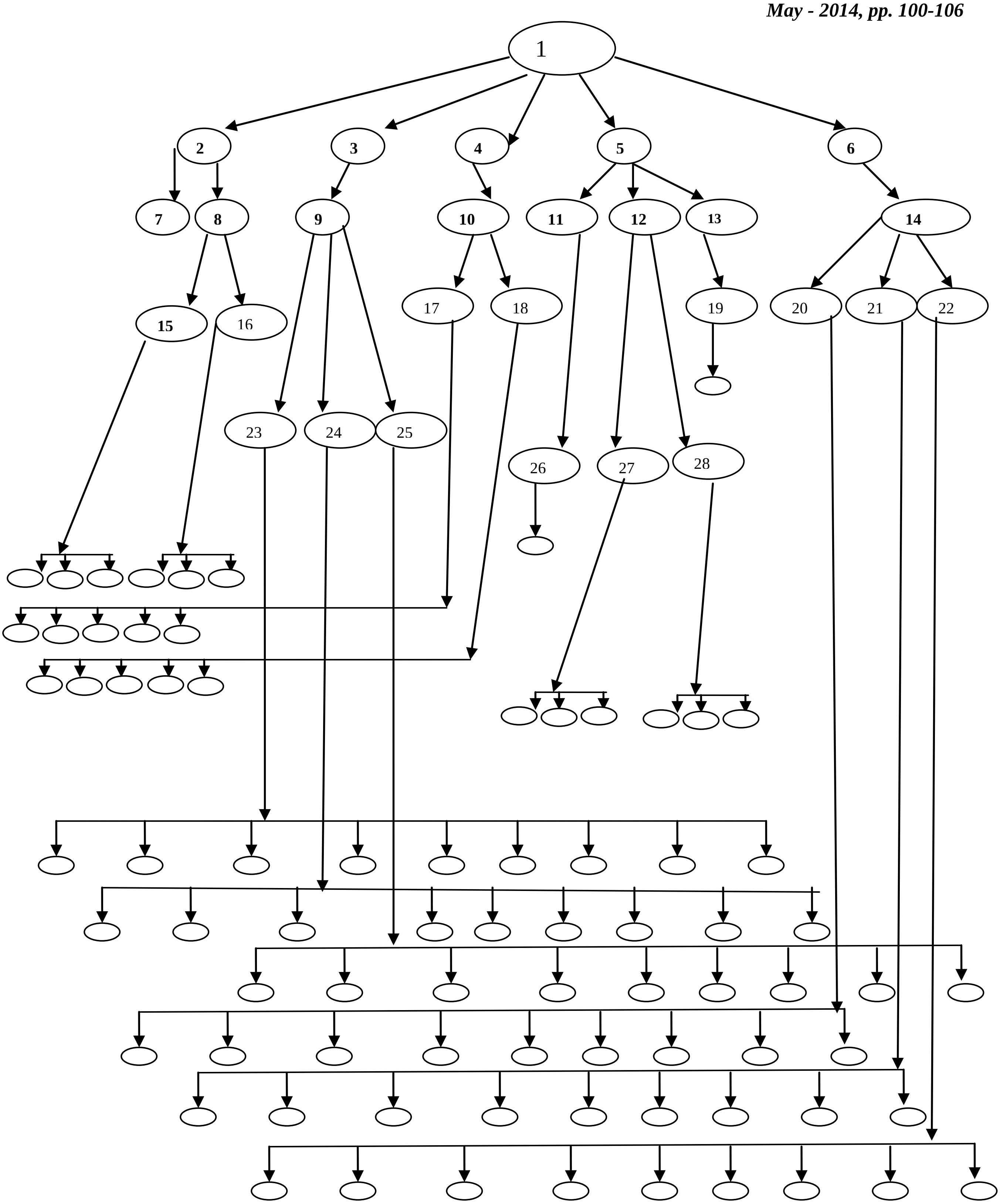

Fig 1.3: Social Corporate network graph with predicted node responses.

V. CONCLUSION:

In this paper, we implemented the graph mining technique of label propagation with our proposed algorithmic strategy. This graph mining techniques is based on the classification, clustering, decision tree approaches, which are the graph mining fundamentals. In addition, the strategies are supporting the optimistic way of stimulus response feature. We also have highlighted the research contributions and found out some limitations in different research works. Consequently, this work also depicts the critical evaluation in which prediction have been taken out to show the similarities and differences among different node responsibilities equilant to social network clients. The spatiality of this work is that it reveals the literature review of different graph mining techniques and provides a vast amount of information under a single paper. In our future work, we have planned to propose a cluster mining method based on graph mining technique, provide its implementation and compare its results with the different existing classification based graph mining algorithms.

REFERENCES

[1] G. Di Fatta, S. Leue, E. Stegantova. "Discriminative Pattern Mining in Software Fault Detection."Workshop on Software Quality Assurance, 2006.

[2] E. W. Dijkstra. "A note on two problems in connection with graphs. Numerische Mathematik" ,269-271.

[3] T. Falkowski, J. Bartelheimer, M. Spilopoulou. "Mining and Visualizing the Evolution of Subgroups in Social Networks", ACM International Conferenceon Web Intelligence, 2006.

[4] M. Fiedler, C. Borgelt. "Support computation for mining frequent sub graphs in a single graph." Workshop on Mining and Learning with Graphs(MLG'07), 2007.

[5] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Ma-honey. "Statistical properties of community structure in large social and information networks." In WWW, pages 695-704, 2008.

[6] J. Feigenbaum, S. Kannan, A. McGregor, S. Suri, J. Zhang. "Graph Distances in the Data-Stream Model". SIAM Journal on Computing, 38(5): pp.1709–1727, 2008.

[7] J. Ferlez, C. Faloutsos, J. Leskovec, D. Mladenic, M. Grobelnik. "Monitoring Network Evolution using MDL". IEEE ICDE Conference, 2008.

[8] F. Eichinger, K. B-ohm, M. Huber. "Improved Software Fault Detection with Graph Mining". Workshop on Mining and Learning with Graphs,2008.

[9] F. Eichinger, K. B-ohm, M. Huber. "Mining Edge-Weighted Call Graphs to Localize Software Bugs". PKDD Conference, 2008.

[10] W. Fan, K. Zhang, H. Cheng, J. Gao. X. Yan, J. Han, P. S. Yu O. Verscheure.Direct "Mining of Discriminative and Essential Frequent Patterns via Model-based Search Tree". ACM KDD Conference, 2008.

[11] C. Liu, F. Guo, and C. Faloutsos. Bbm: "Bayesian browsing model from petabyte-scale data. In KDD", pages 537-546, 2009.

[12] Y. Low, J. Gonzalez, A. Kyrola, D. Bick son, C. Guestrin, and J. M. Heller stein. "Graph lab: A framework for parallel machine learning". In UAI, pages 340-349, 2010.

[13] R. Gemulla, E. Nijkamp, P. Haas, and Y. Sisma-nis. "Large-scale matrix factorization with distributed stochastic gradient descent". In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 69-77. ACM, 2011.

[14] A. Ghoting, R. Krishnamurthy, E. P. D. Pednault,B. Reinwald, V. Sindhwani, S. Tatikonda, Y. Tian,and S. Vaithyanathan. "System: Declarative machine learning on map reduce". In ICDE, pages 231-242, 2011

[15] U. Kang, H. Tong, J. Sun, C.-Y. Lin and C. Faloutsos. "Gbase: an ancient analysis platform for large graphs".VLDB J., 21(5):637-650, 2012.

