Incremental learning with latent factor models for attribute prediction in social-attribute networks

Duc Kinh Le Tran*,** Cécile Bothorel*
Pascal Cheung Mon Chan**

*UMR CNRS 3192 Lab-STICC
Département LUSSI – Télécom Bretagne
{duc.letran, cecile.bothorel}@telecom-bretagne.eu
**Orange Labs
{duckinh.letran, pascal.cheungmonchan}@orange.com

Abstract. In this paper, we are interested in the problem of predicting attributes on the nodes in a social network. Most of the existing techniques addressing this problem are offline learning techniques and are not suitable in situations where massive data come in stream like social media. In this work, we use *latent factor models* to predict unknown attributes of the nodes in a social network and propose a method to incrementally update the prediction model on the arrivals of new data. Experiments on a real social media dataset show that our method is more rapid and can guarantee acceptable performances in comparison with state-of-the-art non-incremental techniques.

1 Introduction and problem statement

With the explosion of social media on the Internet in recent years, mining social media content has become more and more critical for many domains. One of the challenges of mining social media is how to leverage *relational information* (e.g friendships, interactions between social media users) and simultaneously *attributes* (e.g. users' interests, textual or any other additional information). Another challenge lies in the fact that these media provide vast and continuous streams of data. Using offline learning techniques, we have to aggregate all the data available from the past until the present. This approach is not suitable in this situation because (1) as new data come, the size of the dataset grows, it get more and more expensive to learn and to apply the model (2) this approach cannot capture the dynamic of the data stream: old data and recent data are treated uniformly.

In this paper, we address both challenges by introducing an incremental learning method for the task of predicting attributes of social actors in a social network. This problem has many real world applications, for example to predict users' interests or hobbies using social media. We build a graph of interactions among the social media users and enrich the graph with a set attributes on nodes. As the data (nodes, links) arrive as a permanent stream, we want to build models to periodically predict unknown attributes on the nodes.

To formulate our problem, we adopt the *social-attribute network* (Yin et al. (2010)). A *social-attribute network (SAN)* contains a social network G_s=(V_s, E_s) where V_s is the set of

nodes and E_s is the set of edges. The social graph is augmented with a bipartite graph $G_a = (V_s \cup V_a,\, E_a)$, called the attribute graph, connecting the *social nodes* in V_s with *attribute nodes* in V_a. The edges in E_s are *social links* and the edges in E_a (connecting social nodes and attribute nodes) are *attribute links.* There are 2 types of attribute link between a social node i and an attribute node k: a *positive link* if i has the attribute k; a *negative link* if i doesn't have k and in case we don't know whether i has k or not, there is no link between i and k (missing link). We are interested in the problem of predicting attributes of nodes (i.e the nature - positive or negative - of missing links in the attribute graph G_a) in the context of incremental learning. In this context, at each time step t, we have a snapshot $\mathcal{G}(t)$ of the SAN which represents all data (nodes and links) available from the past until t. In comparison with the previous snapshot $\mathcal{G}(t-1)$, $\mathcal{G}(t)$ has new nodes and new links. The new nodes can be social nodes or attribute nodes (in the experiments we only consider new social nodes due to limitation of the data set). We denote by $\Delta\mathcal{G}(t)$ the SAN constituted of the new links which have just been added at the time step t. The SAN $\mathcal{G}(t)$ contains two components (sub-networks), the first component is the snapshot $\mathcal{G}(t-1)$, and the second component is $\Delta\mathcal{G}(t)$. We formulate our incremental learning problem as follows: assume that we have built a model M_{t-1} to predict attributes of nodes at the time step $t-1$, *our problem is to update the model M_{t-1} with new data (nodes and links in $\Delta\mathcal{G}(t)$) to predict unknown attributes of nodes at the time step t.*

In the followings, we review *latent factor models* and *matrix factorization* in batch learning (Section 2) and then propose an approach for incremental learning based on these techniques (Section 3). We present some encouraging experimental results in Section 4. Finally in Section 5 we conclude and point out some promising directions in future work.

2 Latent factor model and matrix factorization

As stated earlier, the learning approach proposed in this paper is inspired from *latent factor models (LFM)* (Bartholomew et al. (2011)), which have long been used in statistics and machine learning. A LFM is a statistical model that represents each data instance by a set of latent variables. *Matrix factorization* (MF) can be considered as a method of latent factor modeling in which latent variables are continuous. The basic idea of MF is to decompose a high dimensional data matrix into lower dimensional matrices.

Techniques of MF have also been extended to handle multiple matrices at a time. Singh and Gordon (2008) introduced *collective matrix factorization (CMF)*. CMF can deal with relational data in which there are many types of entity and many types of relation between entities, each type of relation is represented by a relational matrix. CMF tries to map entities into a common latent space by factorizing simultaneously multiple relational matrices. In our problem setting, we have two matrices : the adjacent matrix of the social network (denoted by S) and the attribute matrix (denoted by A where A_{ik} is a binary value indicating whether the attribute link $(i,\, k)$ is positive or negative). Using CMF, we minimize:

$$\begin{aligned} Q_{CMF}(U,\, P,\, \mathcal{G}) =& \alpha \sum_{(i,\, j) \in E_s} \left(S_{ij} - u_i u_j^T\right)^2 + \sum_{(i,\, k) \in E_a} \left(A_{ik} - u_i p_k^T\right)^2 \\ &+ \lambda \left(\sum_{i=1}^{n_s} \|u_i\|^2 + \sum_{k=1}^{n_a} \|p_k\|^2 \right) \end{aligned} \tag{1}$$

where E_s is the set of social links, E_a is the set of attribute links; U is the matrix constituted of the latent vectors of all the social nodes and similarly, P is the matrix constituted of the latent vectors of all the attribute nodes of $\mathcal{G}$. The parameter α allows to adjust the relative importance of the social network in the model. The third term is a regularization term to penalize complex models with large magnitudes of latent vectors. λ is a regularization parameter.

Li and Yeung (2009) proposed another extension of MF, called *relation regularized matrix factorization* (*RRMF*). RRMF simultaneously exploits the social graph and the attribute graph by minimizing (with the same notations as in Equation 1):

$$\begin{aligned} Q_{RRMF}\left(U,\, P,\, \mathcal{G}\right) =& \alpha \sum_{(i,\,j)\in E_s} S_{ij} \left\| u_i - u_j \right\|^2 + \sum_{(i,\,k)\in E_a} \left(A_{ik} - u_i p_k^T\right)^2 \\ &+ \lambda \left(\sum_{i=1}^{n_s} \left\| u_i \right\|^2 + \sum_{k=1}^{n_a} \left\| p_k \right\|^2 \right) \end{aligned} \tag{2}$$

We can see that this is in fact the factorization of the attribute matrix A when adding regularization term $\alpha \sum_{(i,\,j)\in E_s} S_{ij} \left\| u_i - u_j \right\|^2$. This term is called the *relational regularization term* which allows to minimize the distances between connected social nodes in the latent space. The RRMF approach assumes that connected social actors tend to have similar profiles.

3 Incremental learning with latent factor models

In the incremental learning context defined in Section 1, we need to learn a prediction model (i.e the latent features of nodes) at each time step. The *batch learning* approach suggests that we learn the latent features at each time step using the whole snapshot $\mathcal{G}\left(t\right)$

$$U^\star\left(t\right),\, P^\star\left(t\right) = \underset{U,\,P}{\arg\min}\, Q\left(U,\, P,\, \mathcal{G}\left(t\right)\right) \tag{3}$$

where Q is one of the two objective functions defined above (Equation 1 and Equation 2). Different from the batch learning method, the incremental method learns a model only from new data (i.e SAN $\Delta\mathcal{G}\left(t\right)$) when reusing the old model, i.e latent features of nodes calculated in the previous time step. To do this, we minimize the following objective function:

$$\begin{aligned} Q_{inc}\left(U,\, P,\, t\right) =& Q\left(U,\, P,\, \Delta\mathcal{G}\left(t\right)\right) \\ &+ \mu \left(\sum_{i\in V_s(t-1)} \left\| u_i - u_i^\star\left(t-1\right) \right\|^2 + \sum_{k\in V_a(t-1)} \left\| p_k - p_k^\star\left(t-1\right) \right\|^2 \right) \end{aligned} \tag{4}$$

where $V_s\left(t-1\right)$ and $V_a\left(t-1\right)$ are respectively the set of social nodes and the set of attribute nodes in the previous time step; $u_i^\star\left(t-1\right)$ and $p_k^\star\left(t-1\right)$ are respectively the latent vectors of the social node i and the attribute node k learned in the previous time step and μ is a parameter of the model. This objective function consists of two terms. The first term is the objective function of MF on the incremental graph $\Delta\mathcal{G}\left(t\right)$. The second term is a regularization term for minimizing the shifts of latent features of the same nodes between time steps. By minimizing the two terms simultaneously, we learn latent features of nodes both from the new data and

from the latent features of existing nodes of the previous time step. We can easily see that the latent features of an existing node are updated if and only if there are new links connecting to it. The parameter μ allows to tune the contribution of the previous model to the current model.

In terms of optimization, we adapt the *Alternating Least Squared* (Zhou et al. (2008)) algorithm to minimize Q in Equation 3 for batch learning or Q_{inc} in Equation 4 for incremental learning. The basic idea of this algorithm is to solve the least square problem with respect to the latent features of one node at a time until convergence. The complexity of the algorithm linearly depends on the number of squared terms in the objective function, which is the total number of nodes and number of links in the SAN. In other words, the learning algorithm has linear complexity with respect to the size of the data. In case of incremental learning, when optimizing only on recent data ($\Delta\mathcal{G}(t)$), we can gain a lot in terms of computational cost.

4 Experiments

4.1 Experimental setup

The dataset used in these experiments is BlogCatalog, collected and used by Tang and Liu (2011). In BlogCatalog [1], a blogger can specify his connections with other bloggers. In addition, when submitting a new blog, a blogger specifies the categories of the blog among a set of pre-defined categories. A blogger's interests can be inferred from the categories of his blogs. The dataset contains only a small portions of the whole network: 10312 bloggers, 333983 connections between bloggers, 39 categories, and each blogger has on average 1.4 categories of interest. We can build a SAN out of this data set where bloggers are social actors and categories are attributes. Since we don't have a real data stream, we construct artificial SAN snapshots from this static data set to test our incremental learning method. We build SAN snapshots at 6 time steps in our experiment. We only consider adding new social nodes at each time step (the set of 39 attribute nodes is fixed). We initially pick 50% of the total social nodes and build the SAN snapshot $\mathcal{G}(0)$ from these nodes and all links (social links and attribute links) that involve them. At each time step $t \in \{1, 2, 3, 4, 5\}$, we randomly take 10% of the total social nodes (only nodes which have not been taken yet). We add these social nodes and all their social links to build a new snapshot $\mathcal{G}(t)$. About the attribute links, we assume that the attributes of new nodes at t are unknown until the next time step $t + 1$.

Our objective is to predict unknown attributes of nodes with our incremental methods (*Incremental CMF*, *Incremental RRMF*) at each time step. We compare our incremental learning methods with the *batch learning* approach (i.e using the whole snapshot $\mathcal{G}(t)$ at each time step t). Three batch learning methods are used to compare: batch learning with CMF, RRMF and another state-of-the-art method called *Social Dimension (SocialDim)* (Tang and Liu (2011)). The basic idea of this method is to transform the social network in to features of nodes using a graph clustering algorithm (where each cluster, also called a called a *social dimension*, corresponds to a feature) and then train a discriminative classifier (Support Vector Machine (Cortes and Vapnik (1995))) using these features. It has been shown that the SocialDim outperforms other well-known methods of classification in a network.

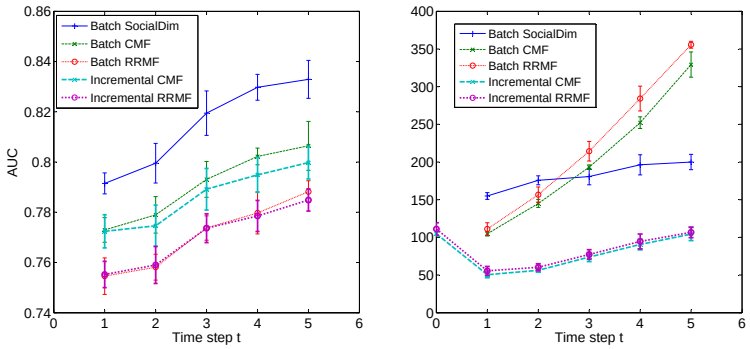
To measure the performances of the different prediction methods, we use *Area Under ROC Curve* (AUC) (Bradley (1997)). At each time step, the AUC is computed from the prediction

1. http://www.blogcatalog.com/

scores and the true labels of all missing links. We also measure the computational time of each method to show empirical gain in complexity of our incremental method.

About the choices of parameters, we have observed that the performances of LFM methods are relatively stable with changes of λ, α and μ in both batch learning and incremental learning. We have set $\lambda = 1.0, \alpha = 1.0, \mu = 100$ for CMF and $\lambda = 1.0, \alpha = 1.0, \mu = 10$ for RRMF to produce representative results for each methods in our experiment. The number of latent factors is set to 50, at which CMF and RRMF attain their maximal stable performances.

4.2 Experiment results



(a) AUC (b) Learning time

FIG. 1: Performance and learning time of incremental learning compared with batch learning

We perform 5 runs and plot the average AUC of each method in each time step in Figure 1a. We observe that the incremental learning techniques cannot give better performances than batch learning methods in all time steps. This is an expected observation: in this experimental setting, the "data stream" is not real. However, both CMF and RRMF give almost the same performance in batch learning and incremental learning (in all cases the difference is not more than 1%). In other words, with LFM, we can incrementally learn the prediction model instead of learning from scratch without any significant loss in performance. When comparing CMF and RRMF, we see clearly that CMF is better. We can also see that the performances of our incremental learning techniques are not too far from those of the reference method - SocialDim (difference of 4% in the worst cases).

Figure 1b shows the learning time (in seconds) of each tested method. To be fair, all the methods are implemented and executed in MATLAB on the same machine (CPU 2.5GHz and 4GB of RAM). The incremental learning techniques require to learn a model from the SAN $\mathcal{G}(0)$ (without prediction) at the time step 0, while the batch learning techniques don't need this step. But in the subsequent time steps (1 to 5), the incremental techniques always have much smaller learning time than that of the batch learning methods. In batch learning, the learning time of CMF and RRMF increases rapidly after each time step. Although the learning time of SocialDim increases less rapidly than that of CMF and RRMF, it is still very long compared to our incremental methods.

5 Conclusion

Motivated by the challenges of social media mining, we have proposed an incremental learning method based on LFM. Two alternatives (CMF and RRMF) inspired from LFM have been tested for the problem of incremental attribute prediction in a social network. Our learning algorithm can achieve relatively good performance compared to the reference method based on Social Dimension, a non-incremental classification method. In future work, we will test our incremental approach on real data streams. We expect that our incremental learning method can capture the dynamic of data stream and give better performances than batch learning. We also consider possible extensions of our models to deal with more complex data in social media, for example to consider other types of nodes and links in the SAN, to include attributes on edges, to handle directed links, etc.

References

Bartholomew, D. J., M. Knott, and I. Moustaki (2011). *Latent Variable Models and Factor Analysis: A Unified Approach*. Wiley Series in Probability and Statistics. Wiley.

Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recogn. 30*(7), 1145–1159.

Cortes, C. and V. Vapnik (1995). Support-vector networks. *Machine learning 20*(3), 273–297.

Li, W. and D. Yeung (2009). Relation regularized matrix factorization. In *IJCAI-09*, IJCAI'09, pp. 1126–1131. Morgan Kaufmann Publishers Inc.

Singh, A. and G. Gordon (2008). Relational learning via collective matrix factorization. In *Proceeding of the 14th ACM SIGKDD*, Number June, pp. 650–658. ACM.

Tang, L. and H. Liu (2011). Leveraging social media networks for classification. *Data Mining and Knowledge Discovery 23*(3), 447–478.

Yin, Z., M. Gupta, T. Weninger, and J. Han (2010). A Unified Framework for Link Recommendation Using Random Walks. ASONAM '10, pp. 152–159. IEEE Computer Society.

Zhou, Y., D. Wilkinson, R. Schreiber, and R. Pan (2008). Large-Scale Parallel Collaborative Filtering for the Netflix Prize. In *Proceedings of the 4th international conference on Algorithmic Aspects in Information and Management*, AAIM '08, pp. 337–348. Springer-Verlag.

Résumé

Dans ce travail, nous nous intéressons au problème de la prédiction d'attributs sur les nœuds dans un réseau social. La plupart des techniques sont hors ligne et ne sont pas adaptées à des situations où les données arrivent massivement en flux comme dans le cas des médias sociaux. Dans ce travail, nous utilisons les modèles de variables latentes pour prédire les attributs inconnus des nœuds dans un réseau social et proposer une méthode pour mettre à jour incrémentalement le modèle avec des nouvelles données. Des expérimentations sur un jeu de données issues des médias sociaux montrent que notre méthode est moins coûteuse en temps de calcul et peut garantir des performances acceptables en comparaison avec les techniques non-incrémentales de l'état de l'art.