Relation Prediction in Multi-Relational Domains using Matrix Factorization

Christoph Lippert, Stefan Hagen Weber, Yi Huang, Volker Tresp LIPPERT@CIP.IFI.LMU.DE
Siemens AG, Corporate Technology, Information & Communications, Otto-Hahn-Ring 6, 81739 Munich Germany

Matthias Schubert, Hans-Peter Kriegel
Ludwig-Maximilians University Munich, Oettingenstraße 67, 80538 Munich, Germany

Keywords: relational-learning, relation prediction, matrix factorization, collaborative filtering, bioinformatics

Abstract

The paper is concerned with relation prediction in multi-relational domains using matrix factorization. While most past predictive models focussed on one single relation type between two entity types, in the paper a generalized model is presented that is able to deal with an arbitrary number of relation types and entity types in a domain of interest. The novel *multi-relational matrix factorization* is domain independent and highly scalable. We validate the performance of our approach using two real-world data sets, i.e. user-movie recommendations and gene function prediction.

1. Introduction

In recent years there has been an enormous increase in interest in the analysis of multi-relational data that contain several entity types and multiple relations. As an example consider social networks, such as Facebook where users tag other users as their friends and take part in various kinds of events. There are also various applications in other science fields as e.g. bioinformatics.

The paper aims at relation prediction based on a set of relation types. In this paper we consider relation types[1] that can be represented in matrix form $\mathbf{R} \in \mathbb{R}^{n \times m}$, where $(\mathbf{R})_{ij}$ either stands for the existence of a relation ($(\mathbf{R})_{ij} \in \{0, 1\}$) or for some attribute associated with a relation ($(\mathbf{R})_{ij} \in \mathbb{R}$). Matrix factorizations such as Singular Value Decomposition (SVD) have often been applied to relation prediction. Traditionally, SVD based approaches complete the original matrix by multiplying the two decomposed matrices $\mathbf{U} \in \mathbb{R}^{n \times k}$ and $\mathbf{V} \in \mathbb{R}^{m \times k}$, denoted by $\mathbf{R} \approx \mathbf{X} = \mathbf{U}\mathbf{V}^T$, where k is the rank maximum of $\mathbf{X}$. For example in the topic modeling approach pLSI (Hofmann, 1999) $(\mathbf{R})_{ij}$ stands for the frequency of word i in document j. PLSI then produces nonzero values for words not occuring in the document. The conventional approach would be to supplement a default value (e.g., zero) for missing entries. This is problematic for two reasons. First, the approach cannot distinguish between a zero rating and a missing rating. Second, the computational cost of the SVD scales cubic in the number of elements. Low-*norm* factorization such as the one described by (Takacs et al., 2007) specifies a matrix decomposition approach only based on known ratings. By doing this, improved scalability towards hundreds of millions observed entries was achieved.

We extend low-norm matrix factorization to multi-relational domains. The goal is to improve the performance of relation prediction by simultaneously utilizing all relation types in a domain of interest. Our *multi-relational matrix factorization* is able to deal with any number of entity types and relation types. The experiments described in the paper focus on two large scale multi-relational domains and demonstrate that the proposed approach is capable to significantly improve the accuracy of relation prediction while being well scalable. We use the approach to analyze the MovieLens data set, consisting of user-movie ratings,

[1] In this paper an *entity* is an instance of an *entity type* and a relation is an instance of a *relation type*.

NIPS 2008 workshop on "Structured Input, Structured Output" (SISO 2008)

and to perform gene function prediction in yeast.

2. Related Work

The maximum margin matrix factorization (MMMF) introduced by (Srebro et al., 2004) is a matrix factorization approach based only on the known matrix entries. Instead of a low-rank constraint the model applies a constraint on the Frobenius norm leading to a convex optimization problem which can be formulated as a semi-definite program (SDP). Unfortunately the MMMF model can only handle up to a few hundred entities. A way to make the model scalable is to minimize the objective by using gradient descent methods. In (Rennie & Srebro, 2005) the Polak-Ribière variant of Conjugate Gradients was utilized yielding matrix completion on the EachMovie data set with 2.6 million ratings and on the MovieLens data set containing 1 million ratings. In (Takacs et al., 2007), one of the leading approaches in the Netflix Prize[2], a simple gradient descent method was applied by cycling over nonzero values and iteratively updating $\mathbf{U}$ and $\mathbf{V}$ until a convergence criterium is met.

(Cohn & Hofmann, 2001) have presented a specific joint probabilistic model which attempts to explain both the content and the link structure of documents. (Yu et al., 2005) have proposed a supervised extension of LSI exploiting both the entity features and multiple (category) labels and applied the model to multi-label text classification. These models, however, are limited to handle two relation types. Recently some unsupervised approaches have been proposed to deal with graph clustering problems on multi-relational domains (Long et al., 2006b; Long et al., 2006a). (Singh & Gordon, 2008) have proposed a collective matrix factorization based on minimizing Bregman divergences between the model and the involved relation matrices.

3. Multi-Relational Matrix Factorization

Established matrix factorization models focused on a single relation type connecting two entity types. However, in many real world applications, a set of entity types are connected by multiple relation types which could be strongly correlated to each other. This paper proposes a novel approach called *multi-relational matrix factorization* (MRMF) which can handle an arbitrary number of entity types and relation types in a given domain and exploits multiple relation types simultaneously. Before presenting the approach some concepts and notions are defined.

[2] `http://www.netflixprize.com/`

Assume that there are N entity types $\{\mathcal{E}_1,\ldots,\mathcal{E}_N\}$ and M (binary) relation types $\{\mathcal{R}_1,\ldots,\mathcal{R}_M\}$ in a domain of interest. An entity type $\mathcal{E}$ consists of the indices of n entities, while $\mathcal{R}_f = \{(\mathcal{E}_{a_f};\mathcal{E}_{b_f})\}$ denotes the set of all observed relations of the f-th relation type, where $\mathcal{E}_{a_f}$ and $\mathcal{E}_{b_f}$ are the involved entity types with $a_f, b_f \in \{1,\ldots,N\}$. $\mathbf{E} \in \mathbb{R}^{n\times k}$ stands for the *entity factor matrix* of $\mathcal{E}$, where $k > 0$ is the number of the most informative factors. The relation values (zeros and non-zeros) of the f-th relation type are formed by a matrix $\mathbf{R}_f$ which is reconstructed via multiplying the corresponding entity factor matrices $\mathbf{E}_{a_f}$ and $\mathbf{E}_{b_f}$, denoted by $\mathbf{R}_f \approx \mathbf{E}_{a_f}\mathbf{E}_{b_f}^T$. Here we do not distinguish the factor and the loading matrix. Each entity type is considered as one real valued factor matrix, as it can be involved in reconstruction of multiple relation matrices. Note that in case of $a_f = b_f$ the relation type $\mathcal{R}_f$ is reflexive.

3.1. The Novel Approach

The factor matrices are trained by minimizing the following objective:

$$J = \lambda \sum_{i=1}^{N} \|\mathbf{E}_i\|_F^2 + \sum_{f=1}^{M} \sum_{(i,j)\in\mathcal{R}_f} \left((\mathbf{R}_f)_{ij} - \mathbf{e}_{a_f i}\mathbf{e}_{b_f j}^T\right)^2 \tag{1}$$

where $\lambda \in \mathbb{R}_+$ is a parameter making a tradeoff between the squared approximation error and the Frobenius norm of the model. M refers to the number of relation types. The objective is solved by gradient descent algorithm following the idea in (Takacs et al., 2007). The algorithm iteratively cycles over all relation matrices and updates the entity factor matrices.

$$\frac{\partial J_{ij}}{\partial \mathbf{e}_{a_f i}} = \lambda \mathbf{e}_{a_f i} - 2((\mathbf{R}_f)_{ij} - \mathbf{e}_{a_f i}\mathbf{e}_{b_f j}^T)\mathbf{e}_{b_f j} \tag{2}$$

$$\mathbf{e}_{a_f i}^{(t)} = \mathbf{e}_{a_f i}^{(t-1)} - \mu \frac{\partial J_{ij}}{\partial \mathbf{e}_{a_f i}^{(t-1)}} \tag{3}$$

where $\mu \in \mathbb{R}_+$ is the learning rate. Algorithm 1 shows the training process of the MRMF model as we have implemented it. It is important to note that Equation (1) is not the sum of independent terms as a factor matrix typically appears in several terms. Instead, in each iteration, every factor matrix is updated with respect to all relation types it involves until a common convergence is met as stated in Algorithm 1.

By optimizing a joint objective over all relation types the model reflects correlations between the relation types in a domain of interest. Intuitively MRMF finds the k most informative factors for each entity type that

have to fit all involved relation types. It prevents the model from over-fitting one relation type if this contradicted to other relation types in which the same entity types take part. Thus, by jointly decomposing all relation matrices as stated above the individual factor matrices regularize each other. This way our new method achieves a better generalization of prediction on unobserved relations and improves the prediction performance. Matrix completion is achieved by multiplying the two involved entity factor matrices.

Algorithm 1 MRMF Training

Require: entity types $\{\mathcal{E}_1, \ldots, \mathcal{E}_N\}$;
relation types $\{\mathcal{R}_1, \ldots, \mathcal{R}_M\} \ni \mathcal{R}_f = \{(\mathcal{E}_{a_f}; \mathcal{E}_{b_f})\}$;
$k \in \mathbb{N}$; $\mu \in \mathbb{R}_+$; $\lambda \in \mathbb{R}_+$; $\Delta_{min} \in \mathbb{R}_+$

for all $\mathcal{E}_i \in \{\mathcal{E}_1, \ldots, \mathcal{E}_N\}$ **do**
  $\mathbf{E_i} \leftarrow random^{|\mathcal{E}_i| \times k}$
end for
repeat
  for all $\mathcal{R}_f$ **do**
    for all $(i, j) \in \mathcal{R}_f$ **do**
      Calculate the gradients of the residual error for $\mathbf{E}_{a_f}$ and $\mathbf{E}_{b_f}$ as stated in Equation (2).
      Update $\mathbf{E}_{a_f}$ and $\mathbf{E}_{b_f}$ as stated in equation (3).
    end for
  end for
until $J^{(t-1)} - J^{(t)} < \Delta_{min}$

The time complexity of the approach is $O(k * \sum_{f=1}^{M} |\mathcal{R}_f|)$ per iteration, where k is the rank maximum of the entity type matrices and $|\cdot|$ denotes the number of the observed relations of each relation type.

4. Experiments

4.1. Movie Rating Prediction

Preference learning is a common task in machine learning where users give ratings for items. For this problem setting, matrix factorizations like the one stated in (Takacs et al., 2007) belong to the current state of the art. The advantage of our MRMF approach is that attribute information of users and items can be easily incorporated, which is difficult or impossible for state-of-the-art matrix decomposition approaches. In the experiment we used the MovieLens data set that contains 1,000,209 movie-ratings for 3,900 movies made by 6,040 users. The ratings range from one (worst) to five (best). The data set also contains information about the gender, age and occupation of the users. Additionally, the movies are categorized into twenty different genres. Figure 1 (a) shows the entity classes users and movies as well as their feature entities. The challenge

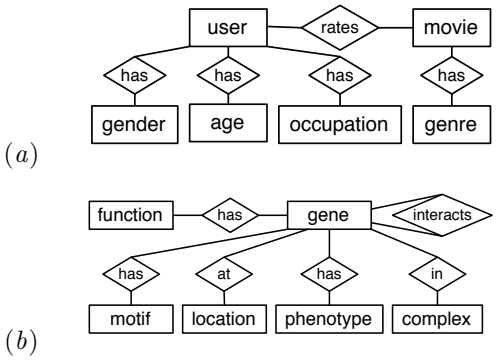


Figure 1. ER-diagrams showing (a) the MovieLens data and (b) an extract of the relations contained in the yeast gene data

	RMSE	MAE
MRMF	$\mathbf{0.8381} \pm 0.0012$	$\mathbf{0.6541} \pm 0.0014$
MF	0.8401 ± 0.0013	0.6583 ± 0.0015

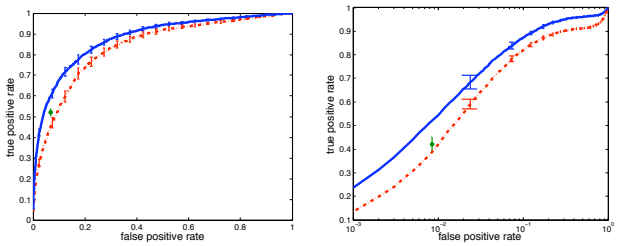
Table 1. 10 fold RMSE and MAE values and 95% confidence intervals for MovieLens for MRMF and MF

is the high sparsity of the rating matrix that is filled for about 4.2%. For validation purposes 10-fold cross validation over known ratings was applied. We compared the MF method re-implemented as proposed by (Takacs et al., 2007) and the MRMF. MRMF models were trained with parameters $k = 100$, $\lambda = 0.06$ and $\mu = 0.005$. The parameters of the MF method were thoroughly tuned to get meaningful results and are close to the parameters used for the MRMF. The RMSE values for the MRMF and MF are shown in Table 1. We proved the significance of the improvement using a paired t-test with $\alpha = 5\%$.

4.2. Gene Function Prediction

S. cerevisiae, the baker's yeast, has a genome of 6.275 predicted genes. The data set used in this paper is the version dated March 2007 of the Comprehensive Yeast Genome Database.[3] It gives a lot of relational information. Function denotes the functional role of the encoded protein in the organism. The annotations follow the FunCat annotation scheme. The FunCat categories are organized in a hierarchical structure. In the data 17 different FunCat terms are annotated on the most general level. Overall there are 506 terms annotated in yeast. The entity types are gene, function, disruption, chromosome, type, phenotype, motif, protein class, subcellular location, classification and complex. The entities are connected by relations as

[3] `http://mips.gsf.de/genre/proj/yeast/`



(a) (b)

Figure 2. Gene function prediction in yeast; ROC curve comparison of MRMF (solid blue line), MF (dotted red line) and SVM (large green point) with 95% confidence intervals for 17 FunCat terms on the top level (a) and for all 506 FunCat terms (b); the experiments follow the 5 repeats all-but-one sampling scheme described in the text.

shown Figure 1 (b) in a simplified manner. Because of the logical constraints implied by the hierarchical nature of the FunCat labels we adapted an all-but-one sampling. One of the 17 most general functions including its whole appended subtree was randomly removed from each gene and used in the test set. This sampling process was repeated five times. The MRMF was tested against the MF used above and SVM. For the latter we propositionalized the data to build a feature vector for each gene. Then a separate SVM was trained for each function. We experimented with a number of kernels (Gaussian, polynomial, linear) and found that the linear kernel was the most effective. Figures 2 (a) and (b) show ROC curves. The MRMF models used for gene function prediction were trained with parameters $k = 100$, $\lambda = 0.01$ and $\mu = 0.005$. The results for both experiments are presented in Table 2. For MRMF and MF F_1-measure and accuracy were obtained by introducing a threshold on the output values. As can be seen in Table 2 the MRMF is comparable to the SVM in terms of accuracy. Because of the imbalance of the data F_1, a combination of precision and recall, should give a better idea of the performance. MRMF significantly outperformes both the MF and the SVM in terms of F_1. The ROC curves of the MRMF and the MF in Figures 2 (a) and (b) are clearly separated which is also reflected by the area under curve. Interestingly, the MF gives quite competitive results even though it does not rely on any feature information at all. This shows that there are strong correlations between the functions of genes.

5. Conclusion

In this paper, a novel multi-relational learning method is introduced. The MRMF extends matrix factorization to multi-relational domains by stating an optimization criterion over all relation matrices. By using gradient descent to solve the MRMF optimization criterion, our solution stays well scalable with the number of updates per iteration being linear in the number of observed relations times the rank k. The experiments clearly show a performance gain over single relational matrix factorization. We showed that using multiple relations helps improve the performance of recommendation systems based on matrix factorization. When doing gene function prediction, MRMF outperforms the propositional SVM by exploiting correlations between the observed values in the target relation.

	#	F1	Acc.	AUC
MRMF	17	**63.1** $\pm$ 1.1	85.7 $\pm$ 0.5	**88.0** $\pm$ 0.7
MF		55.8 $\pm$ 1.6	79.7 $\pm$ 0.8	83.9 $\pm$ 1.1
SVM		56.7 $\pm$ 2.1	**85.9** $\pm$ 0.8	—
MRMF	506	**53.6** $\pm$ 1.8	**98.3** $\pm$ 0.1	**93.65** $\pm$ 0.5
MF		45.6 $\pm$ 1.5	97.6 $\pm$ 0.1	89.2 $\pm$ 0.5
SVM		46.4 $\pm$ 3.0	98.0 $\pm$ 0.1	—

Table 2. Results and 95% confidence intervals for gene function prediction in yeast, 5 repeats all-but-one. All values are in %.

Acknowledgments

The project was funded by means of the German Federal Ministry of Economy and Technology under the promotional reference 01MQ07012. The authors take the responsibility for the contents.

References

Cohn, D., & Hofmann, T. (2001). The missing link - a probabilistic model of document content and hypertext connectivity. *NIPS 13*.

Hofmann, T. (1999). Probabilistic Latent Semantic Indexing. *SIGIR '99*.

Long, B., Wu, X., Zhang, Z. M., & Yu, P. S. (2006a). Unsupervised learning on k-partite graphs. *KDD '06*.

Long, B., Zhang, Z. M., Wú, X., & Yu, P. S. (2006b). Spectral clustering for multi-type relational data. *ICML '06*.

Rennie, J. D. M., & Srebro, N. (2005). Fast maximum margin matrix factorization for collaborative prediction. *ICML '05*.

Singh, A. P., & Gordon, G. J. (2008). Relational learning via collective matrix factorization. *KDD '08*.

Srebro, N., Rennie, J. D. M., & Jaakkola, T. (2004). Maximum-margin matrix factorization. *NIPS 17*.

Takacs, G., Pilaszy, I., Nemeth, B., & Tikk, D. (2007). On the gravity recommendation system. *Proceedings of KDD Cup and Workshop*.

Yu, K., Yu, S., & Tresp, V. (2005). Multi-label informed latent semantic indexing. *SIGIR '05*.