Convex Relaxation of $URU^\top$ Tensor Factorization

Raphaël Bailly
Heudiasyc, UTC/CNRS
Compiègne, France
baillyra@utc.fr

Antoine Bordes
Facebook AI Research
Paris, France
abordes@fb.com

Nicolas Usunier
Heudiasyc, UTC/CNRS
Compiègne, France
nicolas.usunier@hds.utc.fr

Abstract

We present in this paper a new learning method for relational data. The goal of our method is to perform link prediction in multi-relational data. Such data will be represented as a three-way *tensor*, each slice (matrix) representing one relation (not necessarily symmetric) between two nodes. We describe in this article a new approach for a particular low-rank tensor factorization problem, of the form $\boldsymbol{URU}^\top$. This kind of factorization typically occurs when the domains of the relations are identical. In such situations, we want to have one embedding matrix $\boldsymbol{U}$ for the entities, regardless of the relation R_k involved or the position of the entity, and we want the asymmetry of the relation to be managed by the low-rank matrix $\boldsymbol{R}_k$ associated to the relation R_k. The setting that we use is the same as the one used for the algorithm RESCAL [4]. We study the matrix built by stacking horizontally the slices of the tensor and their transposes. We show that the $\boldsymbol{URU}^\top$ tensor factorization problem can be reformulated as a low-rank factorization of this particular matrix. We then use the convex nuclear norm relaxation of the rank as a regularization term. Some experimental results show that our method performs better than RESCAL on the datasets *kinships* and *UMLS*.

1 Introduction

Learning relational data has been of a growing interest in fields as diverse as modeling social networks, semantic web, or bioinformatics. To some extent, a network can be seen as multi-relational data, where a particular relation represents a particular type of link between entities. Our goal in this paper is to be able to predict new links in a network by using all the information that it already contains and included in any type of relations.

A way to use the whole information at our disposal is to design embeddings for entities, and to ensure that these embeddings will be shared among all relations. This way, any relation of some type between two entities is likely to have an influence on the prediction of other types of relations.

We deal here with a particular type of modeling, where a set of relations between entities is represented by a *tensor* $\mathfrak{X}$. In such a model, the relation R_k between two entities e_i and f_j is given by the (i, j, k) coefficient $\mathfrak{X}_{ijk}$ of $\mathfrak{X}$.

The tensor factorization consists of a joint factorization of all the slices representing the different relations. The matrices $\mathfrak{X}_k$ of the relations R_k, which are defined by $\mathfrak{X}_k(i, j) = 1$ if $R_k(e_i, f_j)$ is satisfied, $\mathfrak{X}_k(i, j) = 0$ otherwise, are approximated by matrices of the form $\boldsymbol{U}\boldsymbol{R}_k\boldsymbol{V}^\top$, where $\boldsymbol{U}$ is an embedding matrix for the entities e_i , and $\boldsymbol{V}$ is an embedding matrix for the entities f_j.

In our case, the embeddings of e_i and f_j have to be identical, and the tensor $\mathcal{X}$ has dimension (n, n, p). The matrices $\boldsymbol{R}_k$ are (r, r) matrices, and the matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ are constrained to be equal, of dimension (n, r).

The usual methods of tensor factorization, as CANDECOMP / PARAFAC [3] or Tucker [6], do not readily take into account this equality constraint between embeddings $\boldsymbol{U}$ and $\boldsymbol{V}$. On the opposite, the DEDICOM [2] decomposition considers much stronger constraints than mere equality of embedding matrices: a core relation matrix has to be shared among all relation matrices. Finally, RESCAL [4] provides a non-convex factorization method based on an alternating least squares method. This method provides almost state-of-the-art performances for relation learning on some standard datasets such as *kinships* and *UMLS*.

In this paper, we formulate the $\boldsymbol{URU}^\top$ tensor factorization problem as a *single matrix factorization problem*. This single matrix is built by stacking horizontally all the slices $\mathcal{X}_k$ of the tensor and their transposes $\mathcal{X}_k^\top$ together, with an additionnal constraint: in the resulting factorized matrix, the blocks corresponding to $\mathcal{X}_k$ and $\mathcal{X}_k^\top$ must be transpose of each other. As a result, our formulation allows for any method or heuristic usually used to solve a low-rank *matrix* factorization under linear constraint to be used to solve the $\boldsymbol{URU}^\top$ *tensor* factorization problem. In particular, sub-gradient methods using nuclear norm regularization, a standard convex relaxation of the matrix rank, lead to a convex formulation of our problem.

We provide experimental results showing that our method achieves better results than RESCAL on the benchmarks *kinships* [7] and *UMLS* [5].

2 Model

Let $\{e_1 \dots e_n\}$ be a set of entities and let $\{R_1 \dots R_p\}$ be a set of binary relations between entities. The tensor $\mathcal{X} = \{\mathcal{X}_{ijk}\}$ is defined by $\mathcal{X}_{ijk} = 1$ if $R_k(e_i, e_j)$ is satisfied, $\mathcal{X}_{ijk} = 0$ otherwise. $\mathcal{X}_k$ refers to the k-th slice of $\mathcal{X}$ along the third mode.

Our goal is to find a collective factorization of all the slices $\mathcal{X}_k$, i.e. finding $\boldsymbol{U}$ and $\boldsymbol{R}_k$ such that

$$\forall k\,,\;\; \mathcal{X}_k \approx \boldsymbol{U R_k U}^\top \tag{1}$$

Conversely, given a factorization of all the slices $\boldsymbol{U R}_k \boldsymbol{U}^\top$, we will consider that between two assertions $R_k(e_i, e_j)$ and $R_{k'}(e_{i'}, e_{j'})$, the most likely will be the one with the highest score $(\boldsymbol{U R}_k \boldsymbol{U}^\top)_{ij}$ or $(\boldsymbol{U R}_{k'} \boldsymbol{U}^\top)_{i'j'}$.

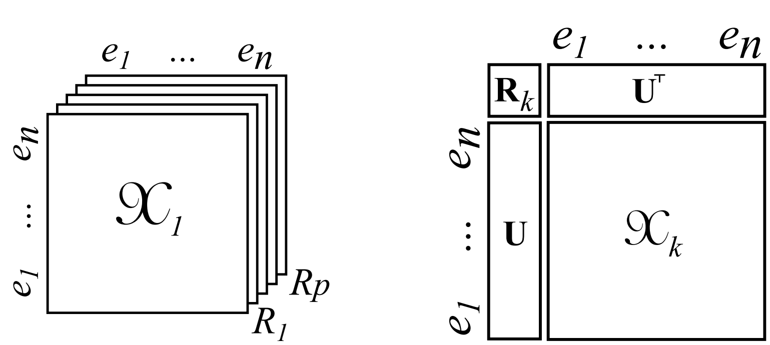

Figure 1: A tensor $\mathcal{X}$ represented as slices along the third mode. Each slice is a matrix $\mathcal{X}_k$ whose (i, j) entry represents the fact that $R_k(e_i, e_j)$ is satisfied. A factorization of the k-th slice $\mathcal{X}_k \approx \boldsymbol{U R_k U}^\top$.

3 Algorithm

3.1 Equivalence between optimization problems

For a given rank r, the optimization problem to be solved is the following: finding an (n, r) matrix $\boldsymbol{U}$ and a set of (r, r) matrices $\boldsymbol{R}_k$ minimizing

$$f(\boldsymbol{U}, \{\boldsymbol{R_k}\}) = \sum_k \|\mathfrak{X}_k - \boldsymbol{U}\boldsymbol{R_k}\boldsymbol{U}^\top\|_F^2 \tag{2}$$

For a given tensor $\mathfrak{X}$, the minimizer of (2) will be denoted $\mathfrak{X}^*$.

The algorithm RESCAL proposes to solve a regularized version of this problem, using an alternate least-square method with a regularizer of the form $\frac{1}{2}(\lambda_U \|\boldsymbol{U}\|_F^2 + \lambda_R \sum_k \|\boldsymbol{R_k}\|_F^2)$.

We now show that the previous problem can be formulated in the framework of minimization of the Frobenius norm of a single matrix under rank and linear constraints.

For a (n, p) matrix $\boldsymbol{M}$ and an (n, q) matrix $\boldsymbol{N}$, let $[\boldsymbol{M}|\boldsymbol{N}]$ be the matrix formed by stacking horizontally $\boldsymbol{M}$ and $\boldsymbol{N}$. More formally, $[\boldsymbol{M}|\boldsymbol{N}]_{i,j} = M_{i,j}$ for $j \leq p$, $[\boldsymbol{M}|\boldsymbol{N}]_{i,j+p} = N_{i,j}$ for $j \leq q$. More generally, for a set of matrices $\{\boldsymbol{X}_1, \dots, \boldsymbol{X}_p\}$ having the same number of rows, let $[\boldsymbol{X}_1|\dots|\boldsymbol{X}_p]$ be the matrix obtained by stacking horizontally the set of matrices $\boldsymbol{X}_1, \dots, \boldsymbol{X}_p$.

Proposition 1. *For a given tensor $\mathfrak{X}$, let $\boldsymbol{X}_1, \dots, \boldsymbol{X}_p, \boldsymbol{X}'_1, \dots, \boldsymbol{X}'_p$ be the set of matrices minimizing the objective*

$$\begin{gathered} \|[\boldsymbol{X}_1|\dots|\boldsymbol{X}_p|\boldsymbol{X}'_1|\dots,\boldsymbol{X}'_p] - [\mathfrak{X}_1|\dots|\mathfrak{X}_p|\mathfrak{X}_1^\top|\dots|\mathfrak{X}_p^\top]\|_F^2 \\ \textit{u. c. } \forall k\,,\, \boldsymbol{X}'_k = \boldsymbol{X}_k^\top\,,\, rk([\boldsymbol{X}_1|\dots|\boldsymbol{X}_p|\boldsymbol{X}'_1|\dots,\boldsymbol{X}'_p]) \leq r \end{gathered} \tag{3}$$

then the tensor $\mathfrak{X}^*$ defined by $\forall k, \mathfrak{X}_k^* = \boldsymbol{X}_k$ is a minimizer of (2).

Proof. First, it is clear that if $\boldsymbol{X}_k$ has the form $\boldsymbol{U}\boldsymbol{R_k}\boldsymbol{U}^\top$ with $rk(\boldsymbol{R_k}) \leq r$, then

$$rk([\boldsymbol{X}_1|\dots|\boldsymbol{X}_p|\boldsymbol{X}_1^\top|\dots,\boldsymbol{X}_p^\top]) \leq r$$

and the matrices $\boldsymbol{X}_k, \boldsymbol{X}_k^\top$ satisfy the constraints of equation (3).

Conversely, if the matrices $\boldsymbol{X}_k$ satisfy constraints of equation (3), let $\boldsymbol{U}\boldsymbol{D}[\boldsymbol{V}_1^\top|\dots|\boldsymbol{V}_p^\top|\boldsymbol{W}_1^\top|\dots,\boldsymbol{W}_p^\top]$ be the truncated SVD of $[\boldsymbol{X}_1|\dots|\boldsymbol{X}_p|\boldsymbol{X}_1^\top|\dots,\boldsymbol{X}_p^\top]$ to the rank r (i.e. $\boldsymbol{U}$ is an (n, r) matrix).

One has, for all k, $\boldsymbol{X}_k = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}_k^\top = \boldsymbol{W}_k\boldsymbol{D}\boldsymbol{U}^\top$. As $\boldsymbol{U}$ is a unitary matrix, one has $\boldsymbol{U}^\top\boldsymbol{U} = \boldsymbol{I}_r$, and $\boldsymbol{U}\boldsymbol{U}^\top\boldsymbol{X}_k = \boldsymbol{U}(\boldsymbol{U}^\top\boldsymbol{U})\boldsymbol{D}\boldsymbol{V}_k^\top = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}_k^\top = \boldsymbol{X}_k$, thus $\boldsymbol{X}_k = \boldsymbol{U}\boldsymbol{U}^\top\boldsymbol{W}_k\boldsymbol{D}\boldsymbol{U}^\top = \boldsymbol{U}\boldsymbol{R}_k\boldsymbol{U}^\top$ with $\boldsymbol{R}_k = \boldsymbol{U}^\top\boldsymbol{W}_k\boldsymbol{D}$.

Hence, $\boldsymbol{X}_k$ has the form $\boldsymbol{U}\boldsymbol{R_k}\boldsymbol{U}^\top$ with $rk(\boldsymbol{R_k}) \leq r$.

With the condition $\boldsymbol{X}'_k = \boldsymbol{X}_k^\top$, it is clear that the two minimization objectives (2) and (3) are equivalent, and that the solutions of (2) and (3) are equivalent too. $\square$

3.2 Convex relaxation

We present here our algorithm called *Tens2mat*.

The optimization problem (3) is not convex. It has the form of an SVD with linear constraints. We choose a standard convex relaxation of this type of problem, which is to consider regularizing using the nuclear norm $\|\|_*$ of a matrix instead of constraining its rank. The nuclear norm of a matrix is the absolute sum of its singular values, i.e. the $\|\|_1$ norm of the vector of its singular values. As a result, the optimization problem we solve is the minimization of:

$$\frac{1}{2}\sum_{k=1}^{p}(\|\boldsymbol{X}_k-\mathfrak{X}_k\|_F^2+\|\boldsymbol{X}'_k-\mathfrak{X}_k^\top\|_F^2)+\frac{\mu}{2}\sum_{k=1}^{p}\|\boldsymbol{X}'_k-\boldsymbol{X}_k^\top\|_F^2+\lambda\|[\boldsymbol{X}_1|\ldots|\boldsymbol{X}_p|\boldsymbol{X}'_1|\ldots,\boldsymbol{X}'_p]\|_* \quad (4)$$

This problem is a convex surrogate of (2), and can be efficiently solved by sub-gradient methods, even for large sparse matrices.

4 Experiments

We ran the algorithm on two benchmarks: *kinships* and *UMLS*.

Kinships. This dataset, created by Denham (1973) [7], describes the kinships of the Alyawarra, a tribe from Central Australia. Ther exists 26 types of kinship relations. This results in graph of 104 members of the tribe (entities) and 26 relation types, each of them depicting a different kinship term, such as 'Adiadya' or 'Umbaidya'. See [7] or [1] for more details.

UMLS. This dataset has been built by McCray (2003) [5]. It consists in a graph with 135 entities and 49 relation types. The entities are high-level concepts like 'Disease or Syndrome', 'Diagnostic Procedure', or 'Mammal'. The relations represent verbs depicting causal influence between concepts, like 'affect' or 'cause'.

We present here 10-fold cross-validation results, for both RESCAL and our algorithm Tens2mat. We used the RESCAL implementation available from: `github.com/mnick/rescal.py`. *Area Under the Precision-Recall Curve* (prAUC) is used to measure the quality of the solutions.

For the RESCAL algorithm, the considered hyper-parameters were the rank and the two other parameters of regularization λ_U and λ_R. For our method, the considered hyper-parameter were μ and λ. The 10-fold validation was made by randomly dividing the sample into 10 equal splits, using alternatively 8 splits for training, 1 split for validation of the hyper-parameters, and the last one for testing. Concerning *kinships*, the same post-processing step as Nickel et al. (see [4]) has been used for both methods. It consists of a renormalization step.

The results are summarized in the Table 1.

prAUC	*kinships*	*UMLS*
RESCAL	0.952 ($\sigma = 0.006$)	0.976 ($\sigma = 0.003$)
Tens2mat	$\mathbf{0.971}$ ($\sigma = 0.007$)	$\mathbf{0.980}$ ($\sigma = 0.004$)

Table 1: Precision-Recall AUC computed for both methods on *kinships* and *UMLS*.

Results obtained with our method improve those obtained with RESCAL. While for the *UMLS* dataset the results are roughly similar, the algorithm Tens2mat clearly outperforms RESCAL on the *kinships* dataset.

Hyper Parameters		*kinships*	*UMLS*
RESCAL	rank	100	72
	λ_U	0.75	0.95
	λ_R	1000	200
Tens2mat	λ	0.1	10
	μ	25	1.5

Table 2: Optimal hyper-parameter values for both methods and both datasets. Values are only indicative, as they may slightly vary depending on the experiment.

The average optimal hyper-parameters for both methods are listed in Table 2. The difference between the two algorithms RESCAL and Tens2mat relies essentially on the different regularization of the objective function.

For the *kinships* dataset, the rank of the matrice $\boldsymbol{U}$ obtained (both by RESCAL or Tens2mat) is roughly maximal, and the rank restriction plays no role in the regularization. The convex regularization we propose based on the nuclear norm $\|\|_*$ seems more efficient. The optimal μ parameter for Tens2mat is around 25.0, which means that sharing embeddings between right-hand side and left-hand side entities is an important factor in the learning process.

For the UMLS dataset, the optimal μ parameter for Tens2mat is around 1.5, which means that sharing embeddings is less important for this dataset. The matrix $\boldsymbol{U}$ obtained by the two algorithms has lower rank than the maximum rank, and most of the regularization seems to be supported by the rank restriction, regardless of the equality between left and right embedding matrices.

5 Future work

5.1 Multiple modes factorization

We adressed in this paper the case where a three-way tensor $\mathfrak{X}$ has two modes sharing their factorization. The method also applies in the case where the three modes (or more) share their factorization (e.g. n-gram tensor factorization), with a low-rank core tensor $\mathfrak{R}$.

5.2 Tensor completion

The method described in this paper applies also to tensor completion problems, where the tensor entries are real values (not only binary values), and some entries can be missing. Computing the gradient corresponding to the reconstruction error (limited to known entries) is even simpler in that case.

6 Conclusion

We presented in this paper an alternate insight of tensor factorization problem, in the case where two of its modes are sharing their embeddings. We describe a novel method to treat this problem, that reduces it to the standard problem of matrix factorization under linear constraints and its convex surrogates. The preliminary experiments that we have conducted show that our method reaches better performances than other comparable methods.

Acknowledgments

Raphaël Bailly is funded through a Google Faculty Research Award. This work was carried out in the framework of the Labex MS2T (ANR-11- IDEX-0004-02), and was also partly funded by the French National Agency for Research (EVEREST-12-JS02-005-01).

References

[1] Kemp C., Tenenbaum J. B, Griffiths T. L, Yamada T., and N. Ueda. Learning systems of concepts with an infinite relational model. In *Proceedings of the National Conference on Artificial Intelligence*, 2006.

[2] R. A. Harshman. Models for analysis of asymmetrical relation- ships among n objects or stimuli. In *First Joint Meeting of the Psychometric Society and the Society for Mathematical Psychology*, 1978.

[3] R. A. Harshman and M. E. Lundy. Parafac: Parallel factor analysis. In *Computational Statistics and Data Analysis*, 1994.

[4] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *ICML*, pages 809–816, 2011.

[5] McCray A. T. An upper level ontology for the biomedical domain. In *Comparative and Functional Genomics*, 2003.

[6] L. R. Tucker. Some mathematical notes on three-mode factor analysis. In *Psychometrika*, 1966.

[7] Denham W. The detection of patterns in alyawarra nonverbal behavior. In *Phd thesis*, 1973.