L1-Norm RESCAL Decomposition

Yorgos Tsitsikas*, Dimitris G. Chachlakis†, Evangelos E. Papalexakis* and Panos P. Markopoulos†‡

*University of California, Riverside, CA, USA
Email: gtsit001@ucr.edu, epapalex@cs.ucr.edu
†Rochester Institute of Technology, Rochester, NY, USA
Email: dimitris@mail.rit.edu, panos@rit.edu

2020 54th Asilomar Conference on Signals, Systems, and Computers | 978-0-7381-3126-9/20/$31.00 ©2020 IEEE | DOI: 10.1109/IEEECONF51394.2020.9443401

***Abstract*—Multi-way arrays (tensors) can naturally model multi-relational data. RESCAL is a popular tensor-based relational learning model. Despite its documented success, the original RESCAL solver exhibits sensitivity towards outliers, arguably, due to its L2-norm formulation. Absolute-projection RESCAL (A-RESCAL), an L1-norm reformulation of RESCAL, has been proposed as an outlier-resistant alternative. However, although in both cases efficient algorithms have been proposed, they were designed to optimize the factor matrices corresponding to the first and second modes of the data tensor independently. To our knowledge, no formal guarantees have been presented that this treatment can always lead to monotonic convergence of the original non-relaxed RESCAL formulation. To this end, in this work we propose a novel L1-norm based algorithm that solves this problem, which at the same time enjoys robustness features of the same nature as those in A-RESCAL. Additionally, we show that our proposed method is closely related to a heavily studied problem in the optimization literature, which enables it to be equipped with numerical stability and computational efficiency features. Lastly, we present a series of numerical studies on artificial and real-world datasets that corroborate the robustness advantages of the L1-norm formulation as compared to its L2-norm counterpart.**

***Index Terms*—RESCAL, tensor, decomposition, graph, outlier, L1-norm**

I. Introduction

Multi-Relational Learning (MRL) is a machine learning discipline with applications in diverse areas such as social network analytics, computational biology, and recommendation systems, to name a few [1], [2]. Numerous representation formulations have been developed for MRL, such as Probabilistic Relational Models, Relational Dependency Networks, and tensor-based models, among others [3], [4]. Similarly to other machine learning applications, tensor processing has successfully been employed in MRL [5], [6].

In this work, we focus on tensor-based models for learning with relational data, and, more specifically, the RESCAL model [7] which has been extensively used for collective classification, link-prediction, and link-based clustering, among other tasks [8]–[10]. RESCAL decomposes a 3-way tensor that models relations across entities into a smaller core tensor and a single factor-matrix modelling all entities in both sides of the relations [13]. It has been documented that, when dealing with relational datasets, RESCAL outperforms popular tensor factorization models such as DEDICOM [14], Canonical Polyadic (CP) decomposition [15], and TUCKER2 [16]. Motivated by the success of the original solver for RESCAL (L2-RESCAL) which was based on a least-squares loss function, variants with a logistic loss function and for non-negative decompositions have been proposed as well [11], [12].

That being said, as in TUCKER2, L2-RESCAL follows an L2-norm formulation which has been shown to exhibit sensitivity against outliers. Outliers often appear due to errors in data storage/transfer, sensor malfunctions, adversarial data contamination etc. To remedy their impact, decomposition models which take active precaution against them have been proposed. Arguably, absolute-projection/L1-norm reformulations of standard PCA and TUCKER2, L1-norm PCA (L1-PCA) [17], [18] and L1-norm TUCKER2 (L1-TUCKER2) [19]–[24], respectively, are the most straightforward ones.[1] Motivated by the success of L1-TUCKER2, Absolute-projection RESCAL (A-RESCAL) was recently proposed as a robust alternative to L2-RESCAL [26].

Despite the efficient design of L2-RESCAL and A-RESCAL, it is important to note that they are both essentially solving a relaxed version of RESCAL which attempts to independently model the latent spaces of the entities in each side of the given relations. With that in mind, they both employ a smart restructuring of the optimization problem, in an attempt to bias the two latent spaces in way that will hopefully lead to them being very similar or identical to each other by the end of the optimization process. However, to the best of our knowledge, no formal proof has been presented to date guaranteeing that this important property will be satisfied.

To this end, in this work our contributions are:

- **A novel L1-norm based RESCAL solver.** Our method enjoys the advantages offered by such a formulation, particularly with respect to robustness against outliers. We will refer to this method as L1-RESCAL.

‡Corresponding author.

This research is supported in part by the U.S. National Science Foundation, under grants OAC-1808582 and OAC-1808591, and by the U.S. Air Force Office of Scientific Research, under award number FA9550-20-1-0039. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding parties.

[1]The L1-norm in L1-PCA/L1-TUCKER2 should not be confused with the L1-norm regularization approach [25] that is often used to enforce sparsity.

- **Global theoretical monotonic convergence guarantees.** We prove that after each iteration of our proposed optimization scheme, the value of the objective function can only improve until it converges. Also, our method optimizes the two latent spaces jointly by construction.
- **A connection to an existing efficient optimization framework.** We show that our method can readily take advantage of an existing heavily studied problem in the optimization literature which endows it with various numerical stability and computational efficiency features.
- **Extensive experimental evaluation.** A series of experiments are carried out on both real-world and artificial data that showcase the advantages offered by L1-RESCAL over L2-RESCAL.

II. Problem Formulation

We consider multi-relational data tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{D\times D\times N}$ such that $\boldsymbol{\mathcal{X}}_{:,:,n} = \mathbf{X}_n \in \mathbb{R}^{D\times D}$ models weights of relations between entities across relation n, $\forall n \in [N]$, where $[N]$ is defined as the set of all integers from 1 to N. For $d \in [D]$, and given optimal $\boldsymbol{\mathcal{R}} \in \mathbb{R}^{d\times d\times N}$, L2-RESCAL [7] can be formulated as:

$$\begin{aligned}\mathbf{Q}_{opt} &= \underset{\substack{\mathbf{Q}\in\mathbb{R}^{D\times d}\\ \mathbf{Q}^T\mathbf{Q}=\mathbf{I}_d}}{\arg\inf} ||\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{R}} \times_1 \mathbf{Q} \times_2 \mathbf{Q}||_F \\ &= \underset{\substack{\mathbf{Q}\in\mathbb{R}^{D\times d}\\ \mathbf{Q}^T\mathbf{Q}=\mathbf{I}_d}}{\arg\sup} ||\boldsymbol{\mathcal{X}} \times_1 \mathbf{Q}^T \times_2 \mathbf{Q}^T||_F.\end{aligned}$$

Hence, RESCAL jointly analyzes $\{\mathbf{X}_n\}_{n\in[N]}$ and approximates $\boldsymbol{\mathcal{X}}$ by $\widehat{\boldsymbol{\mathcal{X}}}$ such that $\widehat{\boldsymbol{\mathcal{X}}}_{:,:,n} = \mathbf{Q}_{opt}\mathbf{R}_n\mathbf{Q}_{opt}^\top$, $\forall n \in [N]$. Despite its documented success, L2-RESCAL exhibits sensitivity against heavily corrupted entries in the processed tensor due to its L2-norm formulation.

A. Absolute-Projection RESCAL

The authors in [26] proposed an outlier-resistant reformulation of RESCAL, called A-RESCAL, which attempts to solve the L1-norm based variant

$$\mathbf{Q}_{opt} = \underset{\substack{\mathbf{Q}\in\mathbb{R}^{D\times d}\\ \mathbf{Q}^T\mathbf{Q}=\mathbf{I}_d}}{\arg\sup} ||\boldsymbol{\mathcal{X}} \times_1 \mathbf{Q}^T \times_2 \mathbf{Q}^T||_1, \qquad (P_{orig})$$

where $\|\cdot\|_1$ returns the sum of the absolute entries of its input argument. For $d = 1$ and under mild conditions, A-RESCAL can been solved both exactly and approximately. For general $d \in [D]$, a subspace-deflation based algorithm has been proposed which optimizes the columns of $\mathbf{Q}$ disjointly.

III. Proposed method

Note that, despite its efficiency, A-RESCAL does not provide a formal guarantee that $\mathbf{Q}_{opt}$ will jointly model the latent spaces of both the columns and the rows of $\mathbf{X}_n, \forall n \in [N]$. However, such a guarantee is crucial since it is the only property that separates RESCAL from TUCKER2. With this in mind, we now discuss a novel approach for solving (P_{orig}) which aims to overcome this issue.

Firstly, note that (P_{orig}) is equivalent to

$$\underset{\substack{\mathbf{Q}\in\mathbb{R}^{D\times d}\\ \mathbf{Q}^T\mathbf{Q}=\mathbf{I}_d}}{\arg\sup} \sum_{n=1}^{N}\sum_{i=1}^{d}\sum_{j=1}^{d} |\mathbf{q}_i^T\mathbf{X}_n\mathbf{q}_j|,$$

where $\mathbf{q}_i$ signifies the i-th column of $\mathbf{Q}$. By introducing the auxiliary tensor $\boldsymbol{\mathcal{B}} \in \{-1,+1\}^{d\times d\times N}$, we get

$$(\mathbf{Q}_{opt}, \boldsymbol{\mathcal{B}}_{opt}) = \underset{\substack{\forall i\ \mathbf{q}_i\in\mathbb{R}^D\\ \forall i\ ||\mathbf{q}_i||=1\\ \forall i\neq j\ \mathbf{q}_i^T\mathbf{q}_j=0\\ \boldsymbol{\mathcal{B}}\in\{-1,+1\}^{d\times d\times N}}}{\arg\sup} \sum_{i=1}^{d}\sum_{j=1}^{d} \mathbf{q}_i^T\mathbf{Y}(\boldsymbol{\mathcal{B}}_{i,j,:})\mathbf{q}_j, \quad (P_{eq})$$

where

$$\mathbf{Y}(\boldsymbol{\mathcal{B}}_{i,j,:}) = \sum_{n=1}^{N} \boldsymbol{\mathcal{B}}_{i,j,n}\mathbf{X}_n.$$

Consider now a relaxation of (P_{eq}), (P_{ineq}), which relaxes the constraint on the columns of $\mathbf{Q}$ allowing them to have norm less than one. We propose a tractable and robust way for approximating a solution for (P_{ineq}) by separately optimizing for $\boldsymbol{\mathcal{B}}$ and each column of $\mathbf{Q}$ in an alternating fashion. Specifically, we split (P_{ineq}) into the following $d+1$ simpler problems:

$$\forall i \in [d] \quad \mathbf{q}_i^* = \underset{\substack{\mathbf{q}_i\in\mathbb{R}^D\\ ||\mathbf{q}_i||\leq 1\\ \forall l\neq i\ \mathbf{q}_i^T\mathbf{q}_l=0}}{\arg\sup} \sum_{k=1}^{d}\sum_{j=1}^{d} \mathbf{q}_k^T\mathbf{Y}(\boldsymbol{\mathcal{B}}_{k,j,:})\mathbf{q}_j$$

with fixed $\boldsymbol{\mathcal{B}}$ and $\mathbf{q}_j$ $\forall j \neq i$, and

$$\boldsymbol{\mathcal{B}}^* = \underset{\boldsymbol{\mathcal{B}}\in\{-1,+1\}^{d\times d\times N}}{\arg\max} \sum_{i=1}^{d}\sum_{j=1}^{d} \mathbf{q}_i^T\mathbf{Y}(\boldsymbol{\mathcal{B}}_{i,j,:})\mathbf{q}_j$$

with fixed $\mathbf{Q}$. Also, by taking a closer look we see that

$$\boldsymbol{\mathcal{B}}^* = \operatorname{sgn}(\boldsymbol{\mathcal{X}} \times_1 \mathbf{Q}^T \times_2 \mathbf{Q}^T) \qquad (P_B)$$

and

$$\mathbf{q}_i^* = \underset{\substack{\mathbf{q}_i\in\mathbb{R}^D\\ ||\mathbf{q}_i||\leq 1\\ \forall j\neq i\ \mathbf{q}_i^T\mathbf{q}_j=0}}{\arg\sup} \mathbf{q}_i^T\mathbf{A}\mathbf{q}_i + \mathbf{c}^T\mathbf{q}_i, \qquad (P_{q_i})$$

where

$$\mathbf{A} = \frac{1}{2}\left(\mathbf{Y}(\boldsymbol{\mathcal{B}}_{i,i,:}) + \mathbf{Y}(\boldsymbol{\mathcal{B}}_{i,i,:})^T\right)$$

and

$$\mathbf{c} = \sum_{\substack{j=1\\ j\neq i}}^{d} \left(\mathbf{Y}(\boldsymbol{\mathcal{B}}_{j,i,:})^T + \mathbf{Y}(\boldsymbol{\mathcal{B}}_{i,j,:})\right)\mathbf{q}_j.$$

If we now define $\mathbf{N} \in \mathbb{R}^{D\times D}$ as the orthogonal projection matrix on the orthogonal complement of the space spanned by all columns of $\mathbf{Q}$ except $\mathbf{q}_i$, then we can calculate $\mathbf{q}_i^*$ as

$$\left\{\begin{array}{l}\mathbf{z}^* = \underset{\substack{\mathbf{z}\in\mathbb{R}^D\\ ||\mathbf{N}\mathbf{z}||\leq 1}}{\arg\inf} -\mathbf{z}^T\mathbf{N}\mathbf{A}\mathbf{N}\mathbf{z} - (\mathbf{N}\mathbf{c})^T\mathbf{z} \\ \\ \mathbf{q}_i^* = \mathbf{N}\mathbf{z}^*\end{array}\right\}. \qquad (P_{QCQP})$$

Algorithm 1: L1-RESCAL Solver

Input: $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{D\times D\times N}$ (data tensor), $d \in \mathbb{Z}^{+}$ (Number of components), $h \in \mathbb{Z}^{*}$ (Number of (P_{q_i}) iterations between two consecutive iterations of (P_B))

Output: $\mathbf{Q} \in \mathbb{R}^{D\times d}$

Initialize: $\boldsymbol{\mathcal{B}} \in \{-1,+1\}^{d\times d\times N}, \mathbf{Q} \in \mathbb{R}^{D\times d}$

$j \leftarrow 0$

repeat

  if $j \mod (h+1) == 0$ **then**

    $\boldsymbol{\mathcal{B}} \leftarrow \text{sgn}(\boldsymbol{\mathcal{X}} \times_1 \mathbf{Q}^T \times_2 \mathbf{Q}^T)$

  $i \leftarrow (j \mod d) + 1$

  $\mathbf{q}_i \leftarrow$ `TRS_Solver` $(\boldsymbol{\mathcal{X}}, \mathbf{Q}, \boldsymbol{\mathcal{B}}, i)$

  $j \leftarrow j + 1$

until *Convergence/Termination*

return $\mathbf{Q}$

Thus, solving each (P_{q_i}) boils down to solving a potentially non-convex Quadratically Constrainted Quadratic Program (QCQP). Interestingly, the optimization problem in (P_{QCQP}) has the form of a well studied problem in the optimization literature called the Trust Region Subproblem (TRS) [28]. Therefore, our method can readily benefit from the relevant numerical stability and computational efficiency results that have been published in the past few decades. At this point, our proposed optimization scheme is summarized in Algorithm 1.

An important aspect that we have not discussed yet, is what happens when our proposed method returns a $\mathbf{Q}$ that contains one or more columns that are not of unit norm. Firstly, it can be shown that the solution of TRS can have norm less than one, only when the TRS is convex. Therefore, in practice one can expect $\mathbf{Q}$ to have columns of unit norm virtually always. Even if the solution of TRS is not a unit vector, however, we show next that we can still easily obtain a valid solution.

Lemma 1. *If Algorithm 1 converges to a point* $(\mathbf{Q}, \boldsymbol{\mathcal{B}})$*, then* $\mathbf{q}_i^T \mathbf{Y}(\boldsymbol{\mathcal{B}}_{i,j,:})\mathbf{q}_j$ *is non-negative* $\forall i, j \in [d]$.

Proof. If we assume that $\exists i, j \in [d]$ such that $\mathbf{q}_i^T \mathbf{Y}(\boldsymbol{\mathcal{B}}_{i,j,:})\mathbf{q}_j < 0$, then we can always design a $\boldsymbol{\mathcal{B}}'$ such that $\mathbf{q}_i^T \mathbf{Y}(\boldsymbol{\mathcal{B}}'_{i,j,:})\mathbf{q}_j > 0$ by flipping the signs of all the elements of $\boldsymbol{\mathcal{B}}_{i,j,:}$. Since no other summand of (P_{ineq}) depends on $\boldsymbol{\mathcal{B}}_{i,j,:}$, we can see that the value of its objective function will be strictly greater at $(\mathbf{Q}, \boldsymbol{\mathcal{B}}')$. This implies that Algorithm 1 has not converged, which is a contradiction. □

Lemma 2. *If, upon convergence of Algorithm 1 to a point* $(\mathbf{Q}, \boldsymbol{\mathcal{B}})$*,* $\exists i \in [d]$ *such that* $||\mathbf{q}_i|| < 1$*, then removing* $\mathbf{q}_i$ *from* $\mathbf{Q}$ *or replacing it with any other vector is guaranteed to not reduce the value of the objective function of* (P_{orig}).

Proof. Firstly, we show that for such $\mathbf{q}_i$ all summands, $\mathbf{q}_k^T \mathbf{Y}(\boldsymbol{\mathcal{B}}_{k,j,:})\mathbf{q}_j$, of the objective function, $G(\mathbf{q}_i)$, of (P_{q_i}) are always zero. This is easy to see when $\mathbf{q}_i = \mathbf{0}$, while for $\mathbf{q}_i \neq \mathbf{0}$ notice that if there existed non-zero summands, then Lemma 1 would imply that all of them would be positive. Therefore, we would have that $G\left(\frac{\mathbf{q}_i}{||\mathbf{q}_i||}\right) > G(\mathbf{q}_i)$, and since $\frac{\mathbf{q}_i}{||\mathbf{q}_i||}$ is feasible for (P_{q_i}), we would contradict the fact that Algorithm 1 has converged. Thus, by observing that $\sum_{n=1}^{N} |\mathbf{q}_k^T \mathbf{X}_n \mathbf{q}_j| = \mathbf{q}_k^T \mathbf{Y}(\boldsymbol{\mathcal{B}}_{k,j,:})\mathbf{q}_j \ \forall k, j \in [d]$, we conclude that removing $\mathbf{q}_i$ from $\mathbf{Q}$ will have no effect on the value of the objective function of (P_{orig}). Also, note that replacing $\mathbf{q}_i$ with any $\mathbf{q}'_i \in \mathbb{R}^D$ cannot decrease the value of the objective function of (P_{orig}), since $\forall j \in [d]$ it holds that $\sum_{n=1}^{N} |\mathbf{q}_i'^T \mathbf{X}_n \mathbf{q}_j| \geq 0 = \sum_{n=1}^{N} |\mathbf{q}_i^T \mathbf{X}_n \mathbf{q}_j|$ and $\sum_{n=1}^{N} |\mathbf{q}_j^T \mathbf{X}_n \mathbf{q}'_i| \geq 0 = \sum_{n=1}^{N} |\mathbf{q}_j^T \mathbf{X}_n \mathbf{q}_i|$. □

Hence, Lemma 2 implies that we can just normalize non-zero columns of $\mathbf{Q}$ that are not of unit norm upon convergence of Algorithm 1, while zero columns can be replaced with any other vector provided that $\mathbf{Q}$ remains semi-orthogonal.

A. Convergence Analysis

Firstly, note that if we define

$$\mathbf{Z}(\boldsymbol{\mathcal{B}}) = \begin{bmatrix} \mathbf{Y}(\boldsymbol{\mathcal{B}}_{1,1,:}) & \mathbf{Y}(\boldsymbol{\mathcal{B}}_{1,2,:}) & \cdots & \mathbf{Y}(\boldsymbol{\mathcal{B}}_{1,d,:}) \\ \mathbf{Y}(\boldsymbol{\mathcal{B}}_{2,1,:}) & \mathbf{Y}(\boldsymbol{\mathcal{B}}_{2,2,:}) & & \\ \vdots & & \ddots & \\ \mathbf{Y}(\boldsymbol{\mathcal{B}}_{d,1,:}) & & & \mathbf{Y}(\boldsymbol{\mathcal{B}}_{d,d,:}) \end{bmatrix}$$

then, we have that

$$\sup_{\substack{\forall i\ \mathbf{q}_i \in \mathbb{R}^D \\ \forall i\ ||\mathbf{q}_i|| \leq 1 \\ \forall i \neq j\ \mathbf{q}_i^T \mathbf{q}_j = 0 \\ \boldsymbol{\mathcal{B}} \in \{-1,+1\}^{d\times d\times N}}} \sum_{i=1}^{d}\sum_{j=1}^{d} \mathbf{q}_i^T \mathbf{Y}(\boldsymbol{\mathcal{B}}_{i,j,:})\mathbf{q}_j \leq$$

$$\max_{\boldsymbol{\mathcal{B}} \in \{-1,+1\}^{d\times d\times N}} d\ \lambda_{max}\left(\frac{\mathbf{Z}(\boldsymbol{\mathcal{B}}) + \mathbf{Z}(\boldsymbol{\mathcal{B}})^T}{2}\right) < \infty,$$

where $\lambda_{max}(\cdot)$ returns the maximum eigenvalue of its input matrix. This implies that (P_{ineq}) is bounded above.

Secondly, observe that plugging $\mathbf{q}_i^*$ or $\boldsymbol{\mathcal{B}}^*$ into the objective function of (P_{ineq}) will never decrease its value. This holds because by definition $\mathbf{q}_i^*$ is maximizing the sum of the subset of summands in (P_{ineq}) that contain $\mathbf{q}_i$, without affecting the rest of the summands. It also holds for $\boldsymbol{\mathcal{B}}^*$ since it is essentially just flipping the signs of all negative summands. Therefore, Algorithm 1 is always guaranteed to converge monotonically. Finally, we can ensure that the value of the objective function in (P_{orig}) will not decrease after an update of a column of $\mathbf{Q}$, by including an iteration of (P_B) right after each iteration of (P_{q_i}). This can be achieved by setting $h = 0$ in Algorithm 1.

IV. Experimental data and results

To properly assess the performance of L1-RESCAL, we compared it against L2-RESCAL on link prediction and outlier robustness. For link prediction we first remove a number of links, and then we use the Area under the Precision-Recall curve (AUC_{PR}) to evaluate the quality of the predictions. Regarding outlier robustness, we introduce noise of large magnitude on a small number of entries in $\boldsymbol{\mathcal{X}}$, and then we use the reconstruction, $\tilde{\boldsymbol{\mathcal{Y}}}$, of the corrupted tensor, to assess

how close it is to $\boldsymbol{\mathcal{X}}$. All experiments were run for $h = 0$ and a range of number of components with multiple samples per number of components using random initialization for the optimization algorithms. Specifically, 50 and 5 samples per number of components were generated for the artificial and real-world datasets, respectively. Note that different noise was generated for each combination of samples and numbers of components, while for artificial data, a different data tensor was generated each time as well. Lastly, for a fair comparison, an iteration of L1-RESCAL was defined to include an update for all columns of the factor matrix, while its core tensors were calculated in the same way as its L2-RESCAL counterparts.

A. Data & Experiment types

a) Link Prediction: Here, we consider sparse binary-valued tensors where we zero out 20% of all ones to assess how well we can predict the existence of missing links.

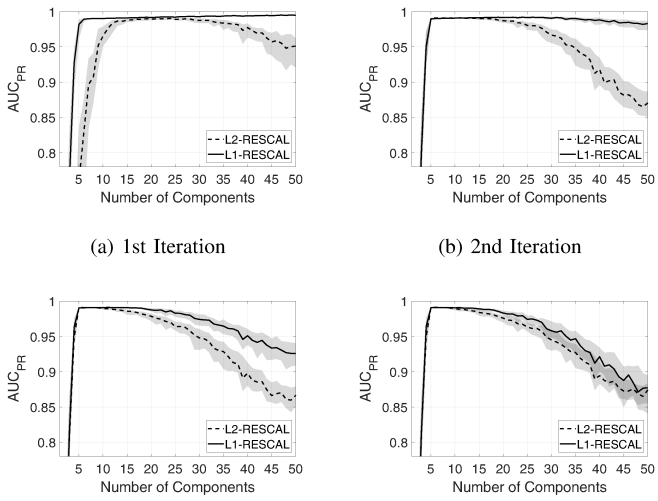
Firstly, note that we can construct such a tensor, $\boldsymbol{\mathcal{X}}$, with a predetermined number of components by randomly creating a binary-valued core, $\boldsymbol{\mathcal{G}}$, and a binary-valued factor matrix, $\mathbf{Q}$, which can then be used to define $\boldsymbol{\mathcal{X}}$ as $\boldsymbol{\mathcal{G}} \times_1 \mathbf{Q} \times_2 \mathbf{Q}$. Also, $\boldsymbol{\mathcal{G}}$ and $\mathbf{Q}$ are defined to contain only a small number of ones in order to produce a sparse $\boldsymbol{\mathcal{X}}$. Note that in this way, we also achieve a low probability of $\boldsymbol{\mathcal{X}}$ containing values other than zeros and ones. If $\boldsymbol{\mathcal{X}}$ does not turn out sparse enough or binary-valued, we can generate new cores and factor matrices, with potentially higher sparsity, until we get the desired outcome. In our experiments, we generated tensors of size $150 \times 150 \times 10$ with 5 number of components and at most 1% of all elements equal to 1. Also, to weaken the perfect RESCAL structure of $\boldsymbol{\mathcal{X}}$, we randomly flipped some zeros into ones. Specifically, the number of these flips was set approximately equal to 1% of the total number of the existing ones in $\boldsymbol{\mathcal{X}}$.

Next, we constructed a $225 \times 225 \times 29$ binary-valued tensor representing the kinship term sets gathered from 104 of 267 individuals from the Alyawarra speaking people of Central Australia [29]. Note that each of the 29 kinship terms is represented by a single frontal slice of the tensor.

b) Outlier Robustness: In this case, we first generate a core, $\boldsymbol{\mathcal{G}}$, of size $5 \times 5 \times 50$ with entries from the standard normal distribution, along with a random semi-orthogonal matrix $\mathbf{Q}$ of size 100×5. From these we create a tensor $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{G}} \times_1 \mathbf{Q} \times_2 \mathbf{Q}$ of size $100 \times 100 \times 50$ with 5 components. Then, in order to weaken the ideal low-rank structure of $\boldsymbol{\mathcal{X}}$, we generate a noise tensor, $\boldsymbol{\mathcal{N}}$, whose elements are also drawn from the standard normal distribution and we define $\boldsymbol{\mathcal{X}}' = \boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{N}} \cdot 0.01||\boldsymbol{\mathcal{X}}||_F/||\boldsymbol{\mathcal{N}}||_F$. Lastly, we add zero-mean Gaussian noise with standard deviation 20 to 10 random entries of $\boldsymbol{\mathcal{X}}'$, which in turn produces $\tilde{\boldsymbol{\mathcal{X}}}$. Our objective now is to assess how close to $\boldsymbol{\mathcal{X}}'$ a RESCAL approximation, $\tilde{\boldsymbol{\mathcal{Y}}}$, of $\tilde{\boldsymbol{\mathcal{X}}}$ can be in the presence of outliers of large magnitude. To quantify this approximation, we define the relative reconstruction error as $||\boldsymbol{\mathcal{X}}' - \tilde{\boldsymbol{\mathcal{Y}}}||_F/||\boldsymbol{\mathcal{X}}'||_F$.

B. L2-RESCAL vs L1-RESCAL

Starting the comparison with Fig. 1, we see that after only the first iteration both L2-RESCAL and L1-RESCAL show a dramatic improvement in their predictions at around 5 components, which is the actual number of components in the data. However, L1-RESCAL shows clearly a better overall prediction capability than L2-RESCAL at this point. Also notice that even though at the first iteration the performance of L2-RESCAL for more than 5 components starts to deteriorate non-trivially, it seems that L1-RESCAL is in fact capable of slightly improving the quality of its predictions as the number of components increase. Additionally, we observe that the performance of L2-RESCAL seems to be deteriorating at a much faster pace from iteration to iteration, while L1-RESCAL again seems to be more robust to overfactoring.



(a) 1st Iteration (b) 2nd Iteration

(c) 6th Iteration (d) 20th Iteration

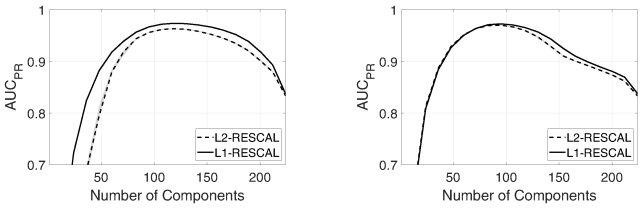
Fig. 1. Area under the Precision-Recall Curve at various stages of the optimization process for the artificial binary-valued data. Lines represent median values, while shaded regions illustrate the corresponding 1st and 3rd quartiles.

Similar observations, albeit less pronounced, can be made for the kinships dataset in Fig. 2. Specifically, notice how L1-RESCAL is offering again a stronger performance at the first iteration, while also being slightly more robust to overfactoring from iteration to iteration as compared to L2-RESCAL.

Lastly, in Fig. 3 it becomes clear that L1-RESCAL can provide considerably improved robustness against outliers as opposed to L2-RESCAL. Particularly, notice how the relative error of L1-RESCAL at the 20th iteration becomes as low as 0.1328, while for L2-RESCAL it tends to be greater than 1.5.

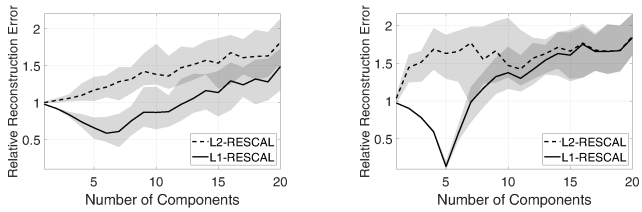
V. Conclusion

We proposed L1-RESCAL, a novel solver for the RESCAL decomposition based on the L1-norm. We proved that our method converges monotonically, and that it is also numerically stable due to its connection to the heavily studied Trust Region Subproblem. Further, our experiments confirmed that L1-RESCAL can provide robustness against outliers on data with low-rank structure. Lastly, we discovered that our method can offer better protection against overfitting due to either overfactoring or a larger than necessary number of iterations.



(a) 1st Iteration (b) 15th Iteration

Fig. 2. Area under the Precision-Recall Curve at various stages of the optimization process for the kinships dataset. Lines represent median values, while shaded regions illustrate the corresponding 1st and 3rd quartiles.



(a) 1st Iteration (b) 20th Iteration

Fig. 3. Outlier robustness at two stages of the optimization process for the artificial non-binary-valued data. Lines represent median values, while shaded regions illustrate the corresponding 1st and 3rd quartiles.

REFERENCES

[1] L. Getoor and B. Taskar, *Introduction to Statistical Relational Learning*. Cambride, MA, USA: MIT Press, 2007.

[2] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich, "A review of relational machine learning for knowledge graphs," in Proc. IEEE, vol. 104, no. 1, pp. 11–33, Jan. 2016.

[3] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer, "Learning probabilistic relational models," in Proc. IJCA/99, pp. 1300-1309, Stockholm, Sweden, 1999.

[4] J. Neville and D. Jensen, "Relational dependency networks," J. Mach. Learn. Res., vol. 8, pp. 637–652, May 2007.

[5] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Trans. Signal Process.*, vol. 65, pp. 3551-3582, 2017.

[6] E. E. Papalexakis, C. Faloutsos, and N. D. Sidiropoulos, "Tensors for data mining and data fusion: Models, applications, and scalable algorithms," *ACM Trans. Intell. Syst. Technol.*, vol. 8, pp. 16:1-16:44, Jan. 2017.

[7] M. Nickel, T. Volker, and H.-P. Kriegel, "A Three-Way Model for Collective Learning on Multi-Relational Data," in *Proc. Int. Conf. Machine Learn. (ICML 2011)*, Bellevue, WA, Jun. 2011, pp. 809-816.

[8] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, "Collective classification in network data," *AI Mag.*, vol. 29, no. 3, p. 93-106, 2008.

[9] B. Taskar, M.-F. Wong, P. Abbeel, and D. Koller, "Link prediction in relational data," in *Neural Inf. Process. Syst. (NIPS 2003)*, Vancouver, Canada, Dec. 2003, pp. 659-666.

[10] Y. Wang and M. Kitsuregawa, "Link based clustering of web search results," in *Proc. Int. Conf. Advances Web-Age Inf. Management*, Xi'an, China, Jul. 2001, pp. 225-236.

[11] M. Nickel and V. Tresp, "Logistic tensor-factorization for multi-relational data," in *Proc. Int. Conf. Machine Learn. Workshop: Structured Learn.: Inferring Graphs from Structured Unstructured Inputs*, Atlanta, GA, Jun. 2013.

[12] D. Krompaß, M. Nickel, X. Jiang, and V. Tresp, "Non-negative tensor factorization with RESCAL," in *Proc. European Conf. Machine Learn. Knowl. Discovery*, Prague, Czech Republic, Sep. 2013.

[13] M. Nickel and V. Tresp, "Tensor factorization for multirelational learning," in *Proc. Joint European Conf. Machine Learn. Knowl. Discovery*, Prague, Czech Republic, Sep. 2013, pp. 617-621.

[14] B.W. Bader, R.A. Harshman, and T.G. Kolda, "Temporal analysis of social networks using three-way dedicom," Sandia National Laboratories, TR-SAND2006-2161, 2006.

[15] F. L. Hitchcock, "The expression of a tensor or a polyadic as a sum of products," J. Math. Physics, vol. 6, n. 1-4, pp. 164-189, 1927.

[16] J. Ye, "Generalized low rank approximations of matrices," *Mach. Learn.,* vol. 61, pp. 167-191, Nov. 2005.

[17] P. P. Markopoulos, G. N. Karystinos and D. A. Pados, "Optimal algorithms for L1-subspace signal processing," *IEEE Trans. Signal Process.*, vol. 62, pp. 5046-5058, Oct. 2014.

[18] Y. Pang, X. Li, and Y. Yuan, "Robust tensor analysis with L1-norm," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 20, pp. 172-178, Feb. 2010.

[19] P. P. Markopoulos, D. G. Chachlakis, and E. E. Papalexakis, "The exact solution to rank-1 L1-norm TUCKER2 decomposition," *IEEE Trans. Signal Process. Lett.*, vol. 25, no. 4, pp. 511-515, Jan. 2018.

[20] D. G. Chachlakis and P. P. Markopoulos, "Robust decomposition of 3-way tensors based on L1-norm," in *Proc. SPIE Def. Comm. Sens.*, Orlando, FL, Apr. 2018, pp. 1065807:1–1065807:15.

[21] D. G. Chachlakis and P. P. Markopoulos, "Novel algorithms for exact and efficient L1-norm-based TUCKER2 decomposition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Calgary, Canada, Apr. 2018, pp. 6294-6298.

[22] P. P. Markopoulos, D. G. Chachlakis, and A. Prater-Bennette, "L1-norm Higher-Order Singular-value Decomposition," in *Proc. IEEE Global Conf. Signal Inf. Process. (IEEE GlobalSIP 2018)*, Anaheim, CA, Nov. 2018, pp. 1353-1357.

[23] D. G. Chachlakis, M. Dhanaraj, A. Prater-Bennette, and P. P. Markopoulos, "Options for multimodal classification based on L1-Tucker decomposition," in *Proc. SPIE Defense and Commercial Sens.*, Baltimore, MD, Apr. 2019, pp. 109 890O:1-109 890O–:13.

[24] D. G. Chachlakis, A. Prater-Bennette, and P. P. Markopoulos, "L1-norm Tucker Tensor Decomposition," in IEEE Access, vol. 7, pp. 178454-178465, 2019.

[25] M. Schmidt, "Least squares optimization with L1-norm regularization," Univ. British Columbia, Vancouver, BC, Canada, Project Rep. CS542B, 2005, pp. 195–221, vol. 504.

[26] D. G. Chachlakis, Y. Tsitsikas, E. E. Papalexakis and P. P. Markopoulos, "Robust Multi-Relational Learning With Absolute Projection RESCAL," Global Conf. Signal Inf. Process., Ottawa, ON, Canada, 2019, pp. 1-5.

[27] X. Fu, K. Huang, W. K. Ma, N. D. Sidiropoulos, and R. Bro, "Joint tensor factorization and outlying slab suppression with applications," *IEEE Trans. Signal Process.,* vol. 63, pp. 6315-6328, Dec. 2015.

[28] C. Fortin, "A survey of the trust region subproblem within a semidefinite framework," Master's thesis, University of Waterloo, 2000.

[29] W. W. Denham. (2016, Jan.) Alyawarra 1971 AU01 dataset. Alyawarra1971KinData.xls. [Online]. Available: https://www.kinsources.net/kidarep/dataset-49-alyawarra-1971-au01.xhtml