Recovery Guarantees For Quadratic Tensors With Sparse Observations

Hongyang Zhang Stanford University
Vatsal Sharan Stanford University
Moses Charikar Stanford University
Yingyu Liang UW-Madison

Abstract

We consider the tensor completion problem of predicting the missing entries of a tensor. The commonly used CP model has a triple product form, but an alternate family of quadratic models which are the sum of pairwise products instead of a triple product have emerged from applications such as recommendation systems. Non-convex methods are the method of choice for learning quadratic models, and this work examines their sample complexity and error guarantee. Our main result is that with the number of samples being only linear in the dimension, all local minima of the mean squared error objective are global minima and recover the original tensor. We substantiate our theoretical results with experiments on synthetic and real-world data.

1 Introduction

Tensors provide a natural way to model higher order data [1, 2, 3, 4, 5]. They have applications in recommendation systems [6, 7, 8], knowledge base completion [9, 10, 11], predicting geo-location trajectories [12] and so on. Most tensor datasets encountered in the above settings are not fully observed. This leads to tensor completion, the problem of predicting the missing entries, given a small number of observations from the tensor [2, 13]. In order to recover the missing entries, it is important to take into account the data efficiency of the tensor completion model.

One of the most well known tensor models is the CANDECOMP/PARAFAC or CP decomposition [2]. For a third order tensor, the CP model will express the tensor as the sum of rank 1 tensors, i.e. tensor product of three vectors. The tensor completion problem of learning a CP decomposition has received a lot of attention recently [14, 15, 16]. It is commonly believed that reconstructing a third-order d dimensional tensor in polynomial time requires $\Theta(d^{3/2})$ samples [14, 17]. This is necessary even for low rank tensors, where $\Theta(d)$ samples are information theoretically sufficient for recovery. The sample requirement of CP decomposition limits its representational power for sparsely observed tensors in practice [12, 10, 18]. While regularization may be helpful when there are limited observations, adding strong regularization will also hurt the optimization performance.

On the other hand, an alternative family of quadratic tensor models have emerged from applications in recommendation systems [6] and knowledge base completion [11]. The *pairwise interaction* model has demonstrated strong performance for the personalized tag recommendation problem [6, 7, 8]. In this model, the (i, j, k) entry of a tensor is viewed as the sum of pairwise inner products: $\langle x_i, y_j\rangle + \langle x_i, z_k\rangle + \langle y_j, z_k\rangle$, where x_i, y_j, z_k correspond to the embedding of each coordinate. As another example, the translating embedding model [9] for knowledge base completion can be (implicitly) viewed as solving tensor completion with a quadratic model. Suppose that x, z are the embedding of two entities and y is the embedding

Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89.

of a relation. Then the smaller $\|x+y-z\|^2$ is, the more likely x, z are related by y. To be concrete, we formalize the notion of quadratic tensors as:

$$T_{i,j,k} = \sum_{l=1}^{r} \kappa(A_{i,l}, B_{j,l}, C_{k,l}), \forall\ 1 \le i, j, k, \le d.$$

where $A, B, C \subseteq \mathbb{R}^{d\times r}$ correspond to the embedding vectors, and $\kappa : \mathbb{R}^3 \to \mathbb{R}$ denotes a quadratic function. Both the pairwise interaction model and the translational embedding model correspond to specific choices of κ.

It is known that for the special case of pairwise interaction tensors, linear (in dimension) number of samples are enough to recover the tensor via convex relaxation [19]. However, in practice non-convex methods are the predominant method of choice for training quadratic models. This is because non-convex methods, such as alternating minimization and gradient descent, are more scalable to handle very large datasets. Despite the practical success, it has been a major challenge to theoretically analyze the performance of non-convex methods. In this work, we present the first recovery guarantee of non-convex methods for learning quadratic tensors. Besides the motivation of quadratic tensors, our work joins a line of recent work to further understand when local methods can lead to globally optimal solutions in non-convex low rank problems [20, 21, 22, 23, 24]. Our results show that quadratic tensor completion enjoys the property that all local minima are global minima in its non-convex formulation.

Main Results. Assume that we observe m entries of T uniformly at random. Denote the set of observed entries as Ω. Consider the natural least squares minimization problem.

$$f(X,Y,Z) = \sum_{(i,j,k)\in\Omega} \left(\sum_{l=1}^{R} \kappa(X_{i,l}, Y_{j,l}, Z_{k,l}) - T_{i,j,k} \right)^2 + Q(X,Y,Z),$$

where $Q(X, Y, Z)$ includes weight decay and other regularizers. (See Section 3 for the precise definition). Note that $f(X, Y, Z)$ is in general non-convex since it generalizes the matrix completion setting when $\kappa(X_{i,l}, Y_{j,l}) = X_{i,l}Y_{j,l}$. We show that as long as $R \ge 2\sqrt{m}$, all local minimum can reconstruct the ground truth T accurately.

Theorem 1 (informal)**.** *Assume that for all* $1 \le i \le d$, $\|e_i^\top A\|, \|e_i^\top B\|, \|e_i^\top C\| \le \sqrt{\mu r/d}$. *Let* ε *be the desired accuracy and* $m = \Theta(dr^4\mu^4(\log d)/\varepsilon^2)$. *For the regularized objective* f, *as long as* $R \ge 2\sqrt{m}$, *then all local minimum* V *of* f *can be used to reconstruct* $\hat{T} \subseteq \mathbb{R}^{d\times d\times d}$ *such that* $\frac{1}{d^3}\sum_{1\le i,j,k\le d} \left|\hat{T}_{i,j,k} - T_{i,j,k}\right| \lesssim \varepsilon/d$.

In the incoherent setting when μ is a small constant, the tensor entries are on the order of $1/d$. Our results imply that the average recovery error is on the order of ε/d. Hence we recover most tensor entries up to less than ε relative error. Our result applies to any quadratic tensor, whereas the previous result on convex relaxations only applies to pairwise interaction tensors [19]. An additional advantage is that our approach does not require the low rank assumption for recovery, we only need r in Theorem 1 to be small, where r is upper bounded by the rank R. We also note that the r^4 dependence of the sample complexity on r for our results is comparable to recent results for non-convex methods for matrix completion [24].

Our technique is based on over-parameterizing the search space to dimension $R = \Theta(\sqrt{m})$ (the $R = \Theta(\sqrt{m})$ dependence on over-parameterization is comparable to previous analyses for low-rank Burer-Monteiro formulations [21, 25]). We show that for the training objective, there is no bad local minimum after over-parameterization. Hence any local minima can achieve small training error. The regularizer Q is then used to ensure that the generalization error to the entire tensor is small, provided with just a linear number of samples from Ω. Since the result applies to any local minimum, it has implications for any non-convex method conceptually, such as alternating least squares and gradient descent.

Experiments. We substantiate our theoretical results with experiments on synthetic and real-world tensors. Our synthetic experiments validate our theory that non-convex methods can recover quadratic tensors with linear number of samples. Our real-world experiments compare the CP model and the quadratic model solved using non-convex methods on two real world datasets. The first dataset consists of 10 million movie ratings over time (Movielens-10M). The task is to predict movie ratings by completing the miss-

ing entries of the tensor. We found that the quadratic model outperforms CP-decomposition by 10%. The second dataset consists of a word tri-occurrence tensor comprising the most frequent 2000 English words. We learn word embeddings from the tensor using both the quadratic model and the CP model, and evaluate the embeddings on standard NLP tasks. The quadratic model is 20% more accurate than the CP model. These results indicate that the quadratic model is better suited to sparse, high-dimensional datasets than the CP model, and we hypothesize that this stems from its better data efficiency.

In conclusion, we show that provided with just linear number of samples from a quadratic tensor, we can recover the tensor accurately using any local minimum of the natural non-convex formulation. Empirically, the quadratic models enjoy superior performance when solved with the non-convex formulation, compared to the CP model. Together, they indicate that the quadratic model may be the right tensor model to use in practical settings with limited data.

Organization. The rest of the paper is organized as follows. In Section 2 we define the quadratic model more formally and review related work. In Section 3 we present our theoretical results. In Section 4 we experimentally evaluate the non-convex formulation for solving quadratic models and conclude in Section 5.

Notation. Given a positive integer d, let $[d]$ denote the set of integers from 1 to d. For a matrix $X \in \mathbb{R}^{d_1 \times d_2}$, let X_i denote the i-th row vector of X, for any $i \in [d_1]$. We use $X \succcurlyeq \mathbf{0}$ to denote that X is positive semi-definite. Denote by $\mathcal{S}_d$ as the set of symmetric matrices of size d by d. Denote by $\mathcal{S}_d^+$ as the set of d by d positive semidefinite matrices. Let $\|\cdot\|$ denote the Euclidean norm of a vector and spectral norm of a matrix. Let $\|\cdot\|_F$ denote the Euclidean norm of a matrix. Let $\|\cdot\|_1$ denote the ℓ_1 norm of a matrix or tensor, i.e. sum of absolute value of every entry. For two matrices A, B we define the inner product $\langle A, B\rangle = \mathrm{Tr}(AB^T)$. For three matrices $X, Y, Z \in \mathbb{R}^{d \times d'}$, denote by $[X; Y; Z] \in \mathbb{R}^{3d \times d'}$ as the three matrices stacked vertically.

Given an objective function $f : \mathbb{R}^d \to \mathbb{R}$, we use $\nabla f(U)$ to denote the gradient of $f(U)$, and $\nabla^2 f(U)$ to denote the Hessian matrix of $f(U)$, which is of size d by d.

We denote $f(x) \lesssim g(x)$ if there exists an absolute constant C such that $f(x) \le Cg(x)$.

2 Preliminaries

We now define the quadratic model more formally with examples. Recall that $T \in \mathbb{R}^{d \times d \times d}$ is a third order tensor, composed by a quadratic function over three factor matrices $A, B, C \subseteq \mathbb{R}^{d \times r}$.[1] In the introduction we defined κ as a function on real values, we now overload the notation and define $\kappa : \mathbb{R}^{d'} \times \mathbb{R}^{d'} \times \mathbb{R}^{d'} \to \mathbb{R}$ to work over vectors as well. More specifically,

$$\begin{aligned} T_{i,j,k} &= \kappa(A_i; B_j; C_k) \\ &= \langle [A_i; B_j; C_k], K \cdot [A_i; B_j; C_k] \rangle. \end{aligned}$$

Recall that $[A_i; B_j; C_k]$ is a $(3 \times d')$ matrix with the i, j, k rows of A, B, C stacked vertically. Here the kernel matrix $K \in \mathbb{R}^{3 \times 3}$ encodes the similarity/dissimilarity represented by κ between the input vectors. Different choices of K represent different quadratic models, for example when $K = I$, $T_{i,j,k} = \|A_i\|^2 + \|B_j\|^2 + \|C_k\|^2$. We assume that K is a symmetric matrix without loss of generality, since we can always symmetrize K without changing κ. We now describe two quadratic models which are commonly used in the literature.

Example 2. *The Pairwise Interaction Tensor Model [6] is proposed in the context of tag recommendation, e.g. suggesting a set of tags that a user is likely to use for an item. The Pairwise Model scores the triple (i, j, k) with the following measure:*

$$T_{i,j,k} = \langle A_i, B_j \rangle + \langle A_i, C_k \rangle + \langle B_j, C_k \rangle.$$

For this model, the kernel matrix K has $1/2$ on all off-diagonal entries, and 0 on the diagonal entries. In the tag recommendation setting, A_i, B_j and C_k correspond to embeddings for the ith user, jth

[1] We assume that the three dimensions all have size d in order to simplify the notations. It is not hard to extend our results to the more general case when different dimensions have different sizes. Also, we will focus on third order tensor for the ease of presentation – it is straightforward to extend the quadratic model to higher orders.

item, and kth tag respectively. The pairwise interaction model models two-way interactions between users, items and tags to predict if user i is likely to use tag k for item j.

Example 3. *The Translational Embedding Model (a.k.a TransE) [9] is well studied in the knowledge base completion problem, e.g. inferring relations between entities. The TransE model scores a triple (i, j, k) with*

$$T_{i,j,k} = \|A_i + B_j - C_k\|^2.$$

Intuitively, the smaller $T_{i,j,k}$ is, the more likely that entities i and k will be related by relation j.

The idea here is that if adding the embedding for *Italy* to the embedding for the *capital of* relationship results in a vector close to the embedding for *Rome*, then *Rome* and *Italy* are likely to be linked by the *capital of* relation.

2.1 Related Work

We first review existing approaches for analyzing non-convex low rank problems. One line of work focuses on the geometry of the non-convex problem, and show that as long as the current solution is not optimal, then a direction of improvement can be found [20, 21, 23, 24, 26]. There are a few technical difficulties in applying this line of approach to our setting. One difficulty is asymmetry — our setting requires recovering three set of different parameters. Existing analysis of alternating least squares does not seem to apply because of the asymmetry as well [27]. The second difficulty is that there exists multiple factor matrices which correspond to the same quadratic tensor in our setting. Hence it is not clear which factor matrices the gradient descent algorithms converges to. A second line of work builds on an interesting connection between SDPs and their Burer-Monteiro low-rank formulations [21, 22]. However, their results do not directly apply to our setting because the non-convex formulation is unconstrained. Recent work has applied this connection to analyzing over-parameterization in one hidden layer neural networks with quadratic activations [25]. Our techniques are inspired by this work, however our setting is fundamentally different from their setting. This is because we need to take into account the incoherence of the factor matrices. Hence we need to add the incoherence regularizer to our setting [20, 21]. We refer the reader to Section 3 for more technical details.

Next we review related works for tensor completion. One approach is to flatten the tensor into a matrix, or treat each slice of the tensor as a low rank matrix individually, and then apply matrix completion methods [28, 29, 30]. There are other models such as RESCAL [3], Tucker-based methods [2] etc. We refer the interested reader to a recent survey for more information [13].

3 Recovery Guarantees

In this section, we consider the recovery of quadratic tensors under partial observations. Recall that we observe m entries uniformly at random from an unknown tensor T. Let $\Omega \in [d]^3$ denote the indices of the observed entries. Given Ω, our goal is to recover T accurately. We first review the definition of local optimality conditions.

Definition 4. *(Local minimum) Suppose that U is a local minimum of $f(U)$, then we have that $\nabla f(U) = \mathbf{0}$ and $\nabla^2 f(U) \succcurlyeq \mathbf{0}$.*

We focus on the following non-convex least squares formulation with variables X, Y, Z, which model the true parameters A, B, C. In this setting, we assume that κ is already known. This is without loss of generality, since our approach also applies to the case when κ is unknown using the same proof technique.

$$\min_{X,Y,Z \subseteq \mathbb{R}^{d\times R}} g(X, Y, Z) =$$

$$\frac{1}{m} \sum_{(i,j,k)\in\Omega} \left(\sum_{l=1}^{R} \kappa(X_{i,l}, Y_{j,l}, Z_{k,l}) - T_{i,j,k} \right)^2$$

$$+\lambda_1(\|X\|_F^2 + \|Y\|_F^2 + \|Z\|_F^2) + \lambda_2 \sum_{i=1}^{d} q_\alpha(\|e_i^\top U\|)$$

$$+\langle [X; Y; Z], C\dot{[}X; Y; Z]\rangle.$$

Let us unpack the above function. The first term corresponds to the natural MSE over Ω. Next we have $q_\alpha(x) = (|x| - \sqrt{\alpha})^4 \mathbf{1}_{x \geq \sqrt{\alpha}}$. The role of $q_\alpha(x)$ is to penalize any row of X, Y, Z whose norm is larger than $\sqrt{\alpha}$, the desired amount from our assumption. It is not hard to verify that $q_\alpha(x)$ is

twice differentiable. Last, $C \subseteq \mathcal{S}^+_{3d}$ is a random PSD matrix with spectral norm at most λ_1. One can view C as a small perturbation on the loss surface. This perturbation will be important to smooth out unlikely cases in our analysis, as we will see later. Our main result is described below.

Theorem 5. *Let $T^\star \subseteq \mathbb{R}^{d\times d\times d}$ be a quadratic tensor defined by factors $A^\star, B^\star, C^\star \subseteq \mathbb{R}^{d\times r}$ and a quadratic function κ. Assume that*

$$\|e_i^\top A^\star\|, \|e_i^\top B^\star\|, \|e_i^\top C^\star\| \le \sqrt{\alpha}, \forall\ 1 \le i \le d.$$

We are given a uniformly random subset of m entries $\Omega \subseteq [d]^3$ from $T^\star$. Let $m \gtrsim d(\log d)/\varepsilon^2$ and $R \ge \sqrt{2m+2d}$. Under appropriate choices of λ_1 and λ_2, for any local minimum X, Y, Z of g, with high probability over the randomness of Ω and C, for $\hat{T}_{i,j,k} = \sum_{l=1}^R \kappa(X_{i,l}, Y_{j,l}, Z_{k,l})$, we have:

$$\frac{1}{d^3}\sum_{1\le i,j,k\le d}\left|\hat{T}_{i,j,k} - T^\star_{i,j,k}\right| \lesssim d\alpha^2\varepsilon.$$

Note that Theorem 1 simply follows from Theorem 5 by setting $\alpha = \sqrt{\mu r/d}$ as well as the corresponding value of m and R in Theorem 5.

For a concrete example of the recovery guarantee, suppose that $A^\star, B^\star, C^\star$ are all sampled independently from $\mathcal{N}(0, 1/\sqrt{d})$. In this case, it is easy to verify that $\alpha \lesssim r(\log d)/d$. Hence when $m \gtrsim dr^4 \log^3 d$, we have that the average recovery error is at most $O(\varepsilon/d)$. Note that every entry of $T^\star$ is on the order of $1/d$ based on the quadratic model. Hence our theorem shows that most tensor entries are accurately recovered up to a relative error of ε fraction.

Next we give an overview of the technical insight. The first technical complication of analyzing such a $g(X, Y, Z)$ is that the three factors are asymmetric. Therefore to simplify the analysis, we first reduce the problem to a symmetric problem, by viewing the search space as $[X; Y; Z] \in \mathbb{R}^{3d\times r}$ instead. We then show that all local minima of $g(X, Y, Z)$ are global minima. Here is where we crucially use the random perturbation matrix C – this is necessary to avoid a zero probability space which may contain non global minima. While this idea of adding a random perturbation is inspired by the work of Du and Lee [25] (and is also used in [31, 32]), the adaptation to our setting is novel and requires careful analysis. In the last part, we use the regularizer of g to argue that all local minima are incoherent, and their Frobenius norms are small. Based on these two facts, we use Rademacher complexity to bound the generalization error. We now go into the details of the proof.

Local optimality. Before proceeding, we introduce several notations. Denote by $U^\star = [A^\star; B^\star; C^\star] \subseteq \mathbb{R}^{3d\times r}$ as the three factors stacked vertically. Let $X^\star = U^\star U^{\star\top}$. For each triple $t = (i, j, k) \in [d]^3$, denote by $A_t \subseteq \mathbb{R}^{3d\times 3d}$ as a sensing matrix such that $\langle A_t, X^\star\rangle = T^\star_{i,j,k}$. Specifically, we have that A_t restricted to the row and column indices $i, j+d, k+2d$ is equal to K (the kernel matrix of κ), and 0 otherwise. We can rewrite $g(X, Y, Z)$ more concisely as follows.

$$\begin{aligned} f(U) =& \frac{1}{m}\sum_{t\in\Omega}\langle A_t, UU^\top - X^\star\rangle^2 + \lambda_1\|U\|_F^2 \\ &+ \lambda_2\sum_{i=1}^{3d} q_\alpha(\|e_i^\top U\|) + \langle C, UU^\top\rangle, \end{aligned}$$

where $U = [X; Y; Z] \subseteq \mathbb{R}^{3d\times R}$. We will use the following Proposition in the proof.

Proposition 6 (Proposition 4 in Bach et al. [33])**.** *Let g be a twice differentiable function convex function over $\mathcal{S}^+_d$. If the function $h : U \to g(UU^\top)$ defined over $U \subseteq d \times d'$ has a local minimum at a rank deficient matrix V, then $VV^\top$ is a global minimum of g.*

Now we are ready to show that there is no bad local minima in the landscape of $f(U)$.

Lemma 7. *In the setting of Theorem 5, with high probability any local minimum U of $f(\cdot)$ is a global minimum.*

Proof. We will show that $\mathrm{rank}(U) < R$, hence by Proposition 6, U is a global minimum of $f(U)$. Assume that $\mathrm{rank}(U) = R$. By local optimiality, $\nabla f(U) = \mathbf{0}$, we obtain that:

$$\left(\sum_{t\in\Omega} z_t A_t + \sum_{i=1}^d w_i e_i e_i^\top + \lambda_1\,\mathrm{Id} + C\right)U = \mathbf{0},$$

$$\text{where } w_i = \frac{4\lambda_2(\|e_i^\top U\| - \sqrt{\alpha})^3}{\|e_i^\top U\|}\mathbf{1}_{\|e_i^\top U\|\ge\sqrt{\alpha}},$$

$$\text{and } z_t = \frac{2}{m}\langle A_t, UU^\top - X^\star\rangle.$$

Denote by

$$M(w,z) = \sum_{i=1}^{d} w_i e_i e_i^\top + \sum_{t\in\Omega} z_t A_t, \text{ and}$$
$$\mathcal{A} = \Big\{ X - M(w,z) - \lambda_1 \mathrm{Id} : X \in \mathcal{S}_{3d}, XU = \mathbf{0}, \\ w \in \mathbb{R}^d, z \in \mathbb{R}^m \Big\}.$$

In the above definition, X is a symmetric matrix in the null space of U – recall that A_t is symmetric, for any $t \in [d]^3$. The set $\mathcal{A}$ is a manifold and clearly $C \in \mathcal{A}$ by the gradient condition.

Since the rank of the null space is $3d - R$, the dimension of such X is $\frac{3d(3d+1)}{2} - \frac{R(R+1)}{2}$. Together with w and z, we have that the dimension of $\mathcal{A}$ is

$$\frac{3d(3d+1)}{2} - \frac{R(R+1)}{2} + m + d.$$

We have assumed that $R \geq \sqrt{2m+2d}$. Hence the dimension of $\mathcal{A}$ is strictly less than $\frac{3d(3d+1)}{2}$. However, the probability that a random PSD matrix C falls in such a set $\mathcal{A}$ only happens with probability zero. Hence with high probability, the rank of V is less than R. The proof is complete. □

Rademacher complexity. Next we bound the generalization error using Rademacher complexity. We first introduce some notations. For any $S \subseteq [d]^3$, $X \subseteq \mathcal{S}_{3d}$, denote by

$$\mathcal{L}_S(X) = \frac{1}{|S|} \sum_{t\in S} |\langle A_t, X - X^\star \rangle| .$$

And let $\mathcal{G}$ denote the set of matrices as follows.

$$\mathcal{G} := \left\{ X \in \mathcal{S}_{3d}^+ : \mathrm{Tr}(X) \leq 2d\alpha, X_{i,i} \leq 2\alpha \ \forall\ i \in [d] \right\}.$$

We bound the Rademacher complexity of $\mathcal{G}$ in the following Lemma.

Lemma 8. *Let c be a fixed constant. In the setting of Theorem 5, with high probability over the randomness of Ω, we have that*

$$\sup_{X\in\mathcal{G}} \left| \mathcal{L}_\Omega(X) - \mathcal{L}_{[d]^3}(X) \right| \leq c\|K\|_1 d\alpha^2 \varepsilon.$$

The proofs for Lemma 8 as well as Theorem 5 are deferred to the Appendix – Theorem 5 follows by combining Lemma 7 and Lemma 8.

4 Experiments

In this section, we describe our experiments on synthetic data and real world data. For synthetic data, we validate our theoretical results and show that the number of samples needed to recover the tensor only grows linearly in the dimension using two non-convex methods – gradient descent and alternating least squares (ALS). We then evaluate the quadratic model solved using non-convex methods on real-world tasks in two diverse domains: a) predicting movie ratings in the Movielens-10M dataset. b) learning word embeddings using a tensor of word tri-occurrences. In the Movielens-10M dataset the quadratic model outperforms CP decomposition by more than 10%. In the word embedding experiment the quadratic model outperforms CP decomposition by more than 20% across NLP benchmarks for evaluating word embeddings.

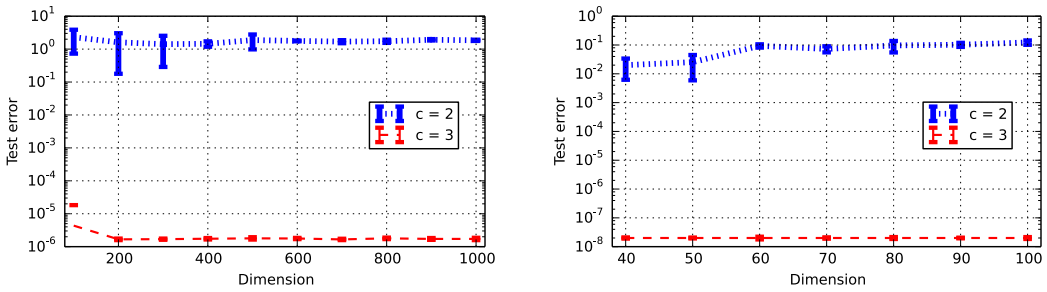
4.1 Synthetic Data

Both gradient descent and ALS are common paradigms for solving non-convex problems, and hence our goal in this section is to evaluate their performances on synthetic data. The ALS approach minimizes the mean squared error objective by iteratively fixing two sets of factors, and then solving the regularized least squares problem on the third factor. In addition, we also evaluate a semidefinite programming based approach which solves a trace minimization problem, similar to the approach in Chen et al. [19].

We now describe our setup. Let $A, B, C \in \mathbb{R}^{d\times r}$, where every entry is sampled independently from standard normal distribution. We sample a uniformly random subset of m entries from the quadratic tensor $T = \mathscr{T}(A,B,C)$. Let the set of observed entries be Ω, and the goal is to recover T given Ω. We measure test error of the reconstructed tensor $\hat{T}$ as follows:

$$\sqrt{\frac{\sum_{(i,j,k)\notin\Omega} (\hat{T}_{i,j,k} - T_{i,j,k})^2}{\sum_{(i,j,k)\notin\Omega} T_{i,j,k}^2}}.$$

Accuracy. We first examine how many samples ALS and the SDP require in order to recover T accurately. Let $m = c \times d \times r$, here m is the number of samples. We fix $r = 5$. For each value of d



(a) Test error vs. Dimension for ALS

(b) Test error vs. Dimension for SDP

Figure 1: ALS and SDP require $2dr$ to $3dr$ samples to recover a quadratic tensor with random factors, providing evidence that their sample complexity is $O(d)$. Here $r = 5$ and number of samples $m = cdr$.

and c, we repeat the experiment thrice, and report the median value with error bars. Because ALS is more scalable, we are able to test on much larger dimensions d. Fig. 1 shows that the sample complexity of both the SDP and ALS is between $2dr$ to $3dr$. When $m = 2dr$, both the SDP and ALS fail to recover T; but given $m = 3dr$ samples, they can recover T very accurately. ALS also converges within 30 iterations across our experiments (Fig. 2 in the Appendix shows how the error decays with the iteration). This makes ALS highly scalable for solving the problem on large tensors. We also repeat the same experiment for gradient descent (Section B.2 in the Appendix) and show that it also has linear sample complexity—though the constants seem to be worse than ALS.

4.2 Movie Ratings Prediction on Movielens-10M Dataset

The Movielens-10M dataset[2] contains about 10 million ratings (each between 0-5) given by $71,567$ users to $10,681$ movies, along with time stamps for each rating. We test both CP decomposition and the quadratic model on a tensor completion task of predict missing ratings given a subset of the ratings. We also compare with a matrix factorization based method which ignores the temporal information to evaluate if the temporal information in the time stamps is useful.

Methodology. We split the ratings into a training and test set with two different sampling rates: $p = 0.2$ and $p = 0.8$ corresponding to 20% and 80% of the entries being in the training set respectively, and repeat the experiment thrice for each p. The smaller $p = 0.2$ sampling rate is to evaluate the performance of the algorithm given very little data. To construct the tensor of ratings we bin the time window into 20 week long intervals, which gives a tensor of size $(71,567 \times 10,681 \times 37)$, where the third mode is the temporal mode. We then use CP decomposition and the quadratic model, both with ℓ_2 regularization to predict the missing ratings. For the matrix method we run matrix factorization with ℓ_2 regularization on the $(71,567 \times 10,681)$ dimensional matrix of ratings. We use alternating minimization with a random initialization and tune the regularization parameter for all algorithms. The evaluation metric is the mean squared error (MSE) on the test entries.

Results. The means and standard deviations of the MSE are reported in Table 1. There are two key takeaways. Firstly, we can see that the quadratic model consistently yields superior performance than the CP model for the choices of rank[3] and sampling rate we explored. The difference between the performances is also larger for the regime with the lower sampling rate, and we hypothesize that this is due to superior generalization ability of the quadratic model compared with the CP model. Another reason for the performance gap could be that the tensor is not a low-rank CP tensor since every user only rates a movie once. The quadratic model also gets a 4% improvement over the baseline which ignores the temporal information in the ratings and uses matrix factorization. This is expected—as a users

[2]https://grouplens.org/datasets/movielens/10m/

[3]We found that going to higher rank did not improve the performance of either model.

Algorithm	Sampling rate $p = 0.2$		Sampling rate $p = 0.8$	
	Rank r=10	Rank r=20	Rank r=10	Rank r=20
Matrix model	0.872 ± 0.004	0.947 ± 0.002	0.665 ± 0.003	0.667 ± 0.001
CP model	1.068 ± 0.087	1.141 ± 0.054	0.719 ± 0.010	0.705 ± 0.002
Quadratic model	$\mathbf{0.798 \pm 0.003}$	$\mathbf{0.772 \pm 0.003}$	$\mathbf{0.642 \pm 0.002}$	$\mathbf{0.638 \pm 0.002}$

Table 1: Results for the Movielens-10M dataset for varying sampling rates corresponding to different training and test splits, and different ranks of the factorization. The quadratic model yields the best results across all settings, with the gap being larger at lower sampling rates.

like or dislike for a genre of movies or a movie's rating may change over time.

4.3 Learning Word Embeddings

Word embeddings are vectors representations of words, where the vectors and their geometry encodes both syntactic and semantic information. We construct word embeddings using the factors obtained by doing tensor factorization on a suitably normalized tensor of word tri-occurrences, and compare the quality of word embeddings learned by the quadratic model and CP decomposition. This experiment tests if the quadratic model returns meaningful factors, in addition to accurately predicting the missing entries.

Methodology. We construct a 2000 dimensional cubic tensor T of word tri-occurrences of the 2000 most frequent words in English by using sliding window of length 3 on a 1.5 billion word Wikipedia corpus, hence the entry T_{ijk} of the tensor is the number of times word i, j and k occur in a window of length 3. As in previous work [34, 18], we construct a normalized tensor $\tilde{T}$ by applying an element-wise nonlinearity of $\tilde{T}_{ijk} = \log(1 + T_{ijk})$ for each entry of T. We then find the factors $\{A, B, C\}$ for a rank 100 factorization of $\tilde{T}$ for the quadratic model and CP decomposition using ALS. The embedding for the ith word is then obtained by concatenating the ith rows of A, B and C, and then normalizing each row to have unit norm.

Evaluation. In addition to reporting the MSE, we evaluate the learned embeddings on standard word analogy and word similarity tasks. The analogy tasks evaluate the percentage of word analogy questions which can be solved using the embeddings. The similarity tasks measure the correlation between word similarity scores determined from the embeddings and the true similarity scores. More details about these tasks can be found in Section C of the Appendix.

Metric	CP model	Quadratic model
MSE	0.5893	0.4253
Syntactic analogy	30.61%	46.14%
Semantic analogy	42.37%	54.76%
Word similarity	0.51	0.60

Table 2: Results for word embedding experiments. The quadratic model significantly outperforms the CP model across all tasks.

Results. The results are shown in Table 2. The quadratic model significantly outperforms the CP model on both the MSE metric, and on the NLP tasks which directly evaluate the embeddings.

4.4 Discussions

The quadratic model is a simplification and special case of the CP model, and hence has lesser representational power. This can lead to worse performance in certain tensor completion tasks, we discuss this more with an example of a hyperspectral image completion task in Section D.

5 Conclusions and Future Work

In this work, we showed that with a non-convex formulation we can recover quadratic tensors using a linear number of samples. Our results characterized the landscape of quadratic tensor models. There are several immediate open questions. Firstly, is it possible to show a convergence guarantee with a small number of iterations? Secondly, is it possible to achieve similar results with rank $O(r)$ as opposed to $\Theta(\sqrt{d})$? We believe that solving this may require novel techniques.

Acknowledgments

HZ and VS were partially supported by National Science Foundation award CCF-1704417 and Office of Naval Research award N00014-18-1-2295. MC was supported by NSF grant CCF-1617577 and a Simons Investigator Award.

References

[1] Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research*, 15(1):2773–2832, 2014.

[2] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[3] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *ICML*, volume 11, pages 809–816, 2011.

[4] Jimeng Sun, Dacheng Tao, and Christos Faloutsos. Beyond streams and graphs: dynamic tensor analysis. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 374–383. ACM, 2006.

[5] Ilya Sutskever, Joshua B Tenenbaum, and Ruslan R Salakhutdinov. Modelling relational data using bayesian clustered tensor factorization. In *Advances in neural information processing systems*, pages 1821–1828, 2009.

[6] Steffen Rendle and Lars Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 81–90. ACM, 2010.

[7] Steffen Rendle and Christoph Freudenthaler. Improving pairwise learning for item recommendation from implicit feedback. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 273–282. ACM, 2014.

[8] Yue Shi, Martha Larson, and Alan Hanjalic. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Computing Surveys (CSUR)*, 47(1):3, 2014.

[9] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795, 2013.

[10] Alberto García-Durán, Antoine Bordes, Nicolas Usunier, and Yves Grandvalet. Combining two and three-way embedding models for link prediction in knowledge bases. *Journal of Artificial Intelligence Research*, 55:715–742, 2016.

[11] Dat Quoc Nguyen. An overview of embedding models of entities and relationships for knowledge base completion. *arXiv preprint arXiv:1703.08098*, 2017.

[12] Yilun Wang, Yu Zheng, and Yexiang Xue. Travel time estimation of a path using sparse trajectories. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 25–34. ACM, 2014.

[13] Qingquan Song, Hancheng Ge, James Caverlee, and Xia Hu. Tensor completion algorithms in big data analytics. *arXiv preprint arXiv:1711.10105*, 2017.

[14] Prateek Jain and Sewoong Oh. Provable tensor factorization with missing data. In *NIPS*, pages 1431–1439, 2014.

[15] Andrea Montanari and Nike Sun. Spectral algorithms for tensor completion. *Communications on Pure and Applied Mathematics*, 2016.

[16] Aaron Potechin and David Steurer. Exact tensor completion with sum-of-squares. In *COLT*, pages 1619–1673, 2017.

[17] Boaz Barak and Ankur Moitra. Tensor prediction, rademacher complexity and random 3-XOR. In *COLT*, 2016.

[18] Vatsal Sharan and Gregory Valiant. Orthogonalized als: A theoretically principled tensor decomposition algorithm for practical use. In *ICML*, pages 3095–3104, 2017.

[19] Shouyuan Chen, Michael R Lyu, Irwin King, and Zenglin Xu. Exact and stable recovery of pairwise interaction tensors. In *Advances in Neural Information Processing Systems*, pages 1691–1699, 2013.

[20] Rong Ge, Jason D Lee, and Tengyu Ma. Matrix completion has no spurious local minimum. In *Advances in Neural Information Processing Systems*, pages 2973–2981, 2016.

[21] Nicolas Boumal, Vlad Voroninski, and Afonso Bandeira. The non-convex burer-monteiro approach works on smooth semidefinite programs. In *Advances in Neural Information Processing Systems*, pages 2757–2765, 2016.

[22] Nicolas Boumal, Vladislav Voroninski, and Afonso S Bandeira. Deterministic guarantees for burer-monteiro factorizations of smooth semidefinite programs. *arXiv preprint arXiv:1804.02008*, 2018.

[23] Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. Global optimality of local search for low rank matrix recovery. In *Advances in Neural Information Processing Systems*, pages 3873–3881, 2016.

[24] Rong Ge, Chi Jin, and Yi Zheng. No spurious local minima in nonconvex low rank problems: A unified geometric analysis. *arXiv preprint arXiv:1704.00708*, 2017.

[25] Simon S Du and Jason D Lee. On the power of over-parametrization in neural networks with quadratic activation. *arXiv preprint arXiv:1803.01206*, 2018.

[26] Xiao Zhang, Lingxiao Wang, Yaodong Yu, and Quanquan Gu. A primal-dual analysis of global optimality in nonconvex low-rank matrix recovery. In *International conference on machine learning*, pages 5857–5866, 2018.

[27] Yuanzhi Li, Yingyu Liang, and Andrej Risteski. Recovery guarantee of non-negative matrix factorization via alternating updates. In *Advances in neural information processing systems*, pages 4987–4995, 2016.

[28] Silvia Gandy, Benjamin Recht, and Isao Yamada. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems*, 27(2):025010, 2011.

[29] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *ICML*, volume 11, pages 809–816, 2011.

[30] Vivek F Farias and Andrew A Li. Learning preferences with side information. Technical report, 2017.

[31] Srinadh Bhojanapalli, Nicolas Boumal, Prateek Jain, and Praneeth Netrapalli. Smoothed analysis for low-rank solutions to semidefinite programs in quadratic penalty form. *arXiv preprint arXiv:1803.00186*, 2018.

[32] Thomas Pumir, Samy Jelassi, and Nicolas Boumal. Smoothed analysis of the low-rank approach for smooth semidefinite programs. In *Advances in Neural Information Processing Systems*, pages 2287–2296, 2018.

[33] Francis Bach, Julien Mairal, and Jean Ponce. Convex sparse matrix factorizations. *arXiv preprint arXiv:0812.1869*, 2008.

[34] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[35] Boaz Barak and Ankur Moitra. Noisy tensor completion via the sum-of-squares hierarchy. In *Conference on Learning Theory*, pages 417–445, 2016.

[36] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[37] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751, 2013.

[38] Elia Bruni, Gemma Boleda, Marco Baroni, and Nam-Khanh Tran. Distributional semantics in technicolor. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 136–145. Association for Computational Linguistics, 2012.

[39] Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, pages 406–414. ACM, 2001.

[40] David H Foster, Kinjiro Amano, Sérgio MC Nascimento, and Michael J Foster. Frequency of metamerism in natural scenes. *Josa a*, 23(10):2359–2372, 2006.

[41] Marco Signoretto, Raf Van de Plas, Bart De Moor, and Johan AK Suykens. Tensor versus matrix completion: a comparison with application to spectral data. *IEEE Signal Processing Letters*, 18(7):403–406, 2011.

[42] Hiroyuki Kasai and Bamdev Mishra. Riemannian preconditioning for tensor completion. *arXiv preprint arXiv:1506.02159*, 2015.

A Missing Proofs from Section 3

In this section, we fill in the missing proofs for Theorem 5. First, we present the proof of Lemma 8, which bounds the Rademacher complexity of $\mathcal{G}$, the set of quadratic tensors.

Proof of Lemma 8. We will prove the following inequality:

$$\mathop{\mathbb{E}}_{\Omega}\left[\sup_{X\in\mathcal{G}}\left|\mathcal{L}_{\Omega}(X)-\mathcal{L}_{[d]^3}(X)\right|\right]\leq c\|K\|_1 d\alpha^2\sqrt{\frac{d}{m}}\leq\frac{c\|K\|_1 d\alpha^2\varepsilon}{\sqrt{\log d}},\tag{1}$$

since $m\geq d\log d/\varepsilon^2$. Based on the above inequality, we can obtain that by Markov's inequality, the probability that

$$\sup_{X\in\mathcal{G}}\left|\mathcal{L}_{\Omega}(X)-\mathcal{L}_{[d]^3}(X)\right|>c\|K\|_1 d\alpha^2\varepsilon$$

happens with probability $1/\sqrt{\log d}$. In the following we will prove Equation (1). Let Ω' denote a set of m independent samples from $[d]^3$. Clearly,

$$\mathop{\mathbb{E}}_{\Omega}\left[\mathcal{L}_{\Omega}(X)\right]=\mathop{\mathbb{E}}_{\Omega'}\left[\mathcal{L}_{\Omega'}(X)\right].$$

By concavity, we have:

$$\begin{aligned}&\mathop{\mathbb{E}}_{\Omega}\left[\sup_{X\in\mathcal{G}}\left|\frac{1}{m}\sum_{t\in\Omega}|\langle A_t,X-X^\star\rangle|-\mathop{\mathbb{E}}_{\Omega'}\left[\frac{1}{m}\sum_{t'\in\Omega'}|\langle A_{t'},X-X^\star\rangle|\right]\right|\right]\\ \leq&\mathop{\mathbb{E}}_{\Omega,\Omega'}\left[\sup_{X\in\mathcal{G}}\left|\frac{1}{m}\sum_{t\in\Omega}|\langle A_t,X-X^\star\rangle|-\frac{1}{m}\sum_{t'\in\Omega'}|\langle A_{t'},X-X^\star\rangle|\right|\right]\end{aligned}\tag{2}$$

Let $\{\sigma_i\}_{i=1}^m$ denote m i.i.d. Rademacher random variables. By the symmetry of Ω and Ω', Equation (2) is equal to:

$$\begin{aligned}&\mathop{\mathbb{E}}_{\Omega,\Omega',\sigma}\left[\sup_{X\in\mathcal{G}}\left|\frac{1}{m}\sum_{l=1}^m\sigma_l\times\left(|\langle A_{t_l},X-X^\star\rangle|-\left|\langle A_{t'_l},X-X^\star\rangle\right|\right)\right|\right]\\ \leq&2\times\mathop{\mathbb{E}}_{\Omega,\sigma}\left[\sup_{X\in\mathcal{G}}\left|\frac{1}{m}\sum_{l=1}^m\sigma_l|\langle A_{t_l},X-X^\star\rangle|\right|\right]\\ =&2\times\mathop{\mathbb{E}}_{\Omega,\sigma}\left[\sup_{X\in\mathcal{G}}\left|\frac{1}{m}\sum_{l=1}^m\sigma_l\langle A_{t_l},X-X^\star\rangle\right|\right]\\ \leq&\frac{2}{m}\times\left(\mathop{\mathbb{E}}_{\Omega,\sigma}\left[\sup_{X\in\mathcal{G}}\langle\sum_{l=1}^m\sigma_{t_l}A_{t_l},X\rangle\right]+\mathop{\mathbb{E}}_{\Omega,\sigma}\left[\langle\sum_{l=1}^m\sigma_{t_l}A_{t_l},X^\star\rangle\right]\right).\end{aligned}\tag{3}$$

We focus on the first part of Equation (3), and it is not hard to bound the second part similarly since $X^\star\in\mathcal{G}$. It suffices to consider a random one matrix $xx^\top\in\mathcal{G}$. Specifically, since $tr(X)\leq d\alpha$ by $X\in\mathcal{G}$, we have

$$\begin{aligned}&\mathop{\mathbb{E}}_{\Omega,\sigma}\left[\sup_{X\in\mathcal{G}}\langle\sum_{l=1}^m\sigma_{t_l}A_{t_l},X\rangle\right]\\ \leq&d\alpha\mathop{\mathbb{E}}_{\Omega,\sigma}\left[\sup_{xx^\top\in\mathcal{G}:x\in\mathbb{R}^{3d}}\langle\sum_{l=1}^m\sigma_{t_l}A_{t_l},xx^\top\rangle\right]\\ \lesssim&\|K\|_1 d\alpha^2\sqrt{md}.\end{aligned}$$

The last step is via standard ϵ-net arguments (see Lemma 2.8 in [35] for an example). Hence the proof of Equation 1 is complete. □

Based on the above Lemma, we are ready to prove Theorem 5.

Proof of Theorem 5. By Lemma 7, we have that as long as U is a local minimum of $f(\cdot)$, then it is a global minimum. In particular, this implies that

$$f(U) \le f(U^\star) \le (\lambda_1 + \|C\|)\|U^\star\|_F^2 \le 2\lambda_1 \|U^\star\|_F,$$

since $\|C\| \le \lambda_1$. Recall that $\hat{T}$ is the reconstructed tensor. Hence we get that

$$\frac{1}{m}\sum_{(i,j,k)\in\Omega} (\hat{T}_{i,j,k} - T^\star_{i,j,k})^2 \le 2\lambda_1 \|U^\star\|_F^2 \lesssim \lambda_1 d\alpha$$

by the assumption on $U^\star$. By Cauchy-Schwarz inequality, this implies:

$$\frac{1}{m}\sum_{(i,j,k)\in\Omega} \left|\hat{T}_{i,j,k} - T^\star_{i,j,k}\right| \le \sqrt{\lambda_1 d\alpha} \lesssim d\alpha^2\varepsilon,$$

by setting $\lambda_1 = c^2\|K\|_1^2 d^2\alpha^3/m$ for a fixed constant c.

Next, it is not hard to see that $\|e_i^\top U\| \le \sqrt{2\alpha}$ by setting $\lambda_2 = 2d\lambda_1/\alpha$. Hence the (i,i)-th entry of $UU^\top$ is at most 2α and $\mathrm{Tr}(UU^\top) \le 2d\alpha$. This implies that $UU^\top \in \mathcal{G}$. By Lemma 8, the generalization error is also bounded by $c\|K\|_1 d\alpha^2\varepsilon \lesssim d\alpha^2\varepsilon$. The proof is complete. □

B Additional Synthetic Experiments

B.1 Convergence Rate of ALS

In Figure 2, we show that ALS can actually converge given a small number of iterations— we observe that within 30 iterations (each iteration requires solving a sparse d^2 by d least squares problems), ALS can achieve low test error.

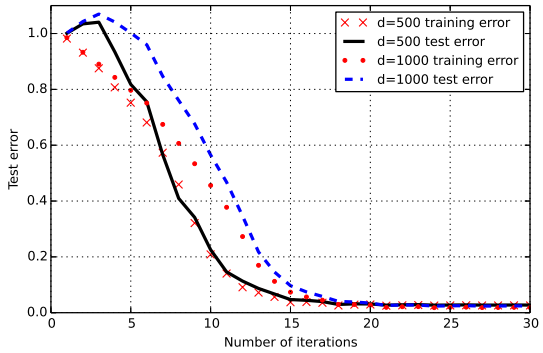


Figure 2: Training and test error for ALS vs the number of iterations. ALS achieves low test error within 30 iterations.

B.2 Sample Complexity for Gradient Descent

We also repeat the same experiment for gradient descent. We run gradient descent with rank $r = d$ for 20000 iterations. Recall that the number of samples $m = c \times d \times r$, and $r = 5$. Figure 3 shows that the sample complexity of gradient descent is between $5dr$ and $10dr$ samples. Our experiments suggest that the constants for the sample complexity are slightly better for ALS as compared to gradient descent, and ALS also seems to converge faster to a solution with small error.

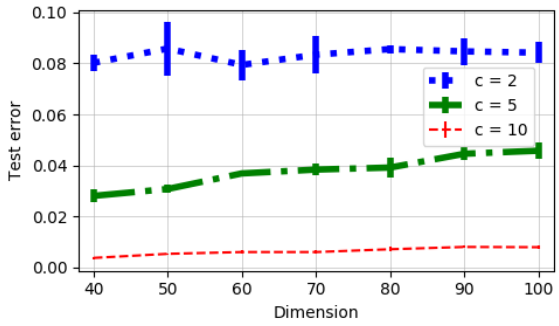


Figure 3: Gradient descent requires about $10dr$ samples to recover a quadratic tensor with random factors, providing evidence that the sample complexity for gradient descent is $O(d)$. Here rank $r = 5$ and number of samples is $m = cdr$.

C Evaluating Word Embeddings

We evaluate the word embeddings on standard word analogy and word similarity tasks. The word analogy tasks [36, 37] consist of analogy questions of the form *"cat is to kitten as dog is to __?"*, and can be answered by doing simple vector arithmetic on the word vectors. For example, to answer this particular analogy we take the vector for cat, subtract the vector for kitten, add the vector for dog, and then find the word with the closest vector to the resulting vector. Hence the analogy task tests how much the geometry in the vector space encodes meaningful syntactic and semantic information. There are two standard datasets for analogy questions, one of which has more syntactic analogies [37] and the other has more semantic analogies [36]. The metric here is the percentage of analogy questions which the algorithm gets correct. The other task we test is a word similarity task [38, 39] where the goal is to evaluate how semantically similar two words are, and this is done by taking the cosine similarity of the word vectors. The evaluation metric is the correlation between the similarity scores assigned by the algorithm and the similarity scores assigned by humans.

D Limitations of the Quadratic Model

In general, there exist tensors which can not be factorized exactly by any quadratic model. This is because if a tensor can be factorized using a quadratic model, then T can be written as the sum of at most $O(d)$ rank 1 tensors. To see this, consider the pairwise tensor model as an example – the same analysis can be applied to other quadratic models as well. Given three factors $x \in \mathbb{R}^{d_1}, y \in \mathbb{R}^{d_2}$ and $z \in \mathbb{R}^{d_3}$, it is not hard to see that the pairwise model defines the following tensor:

$$T(x, y, z) = x \otimes y \otimes e + x \otimes e \otimes z + e \otimes y \otimes z,$$

where $e \subseteq \mathbb{R}^d$ denotes the all one vector. Hence any tensor inside the span of $\{T(x, y, z) : x, y, z \subseteq \mathbb{R}^d\}$ can be factorized into at most $3d$ rank one tensors. This lack of representational power can lead to the

Rank r	CP model	Quadratic model
$r = 10$	0.247	0.265
$r = 20$	0.194	0.226
$r = 50$	0.128	0.205
$r = 100$	0.116	0.216

Table 3: Results for the hyperspectral image task. We see that the CP model outperforms the quadratic model, likely because the higher representational power of CP decomposition is useful in this task.

quadratic model performing worse than the CP model on certain tasks which require high representation ability—and we observe this on a hyperspectral image completion task.

D.1 Hyperspectral images

We evaluate CP decomposition and the quadratic model on a hyperspectral image "Riberia" [40] which has previously been considered in the context of tensor factorization [41, 42]. The image is a $1017 \times 1340 \times 33$ tensor T, where each slice of the image corresponds to the same scene being imaged at a different wavelength. As has been done in previous works [41, 42], we resize the image to $203 \times 268 \times 33$ by downsampling. We randomly sample 10% of the entries of the tensor, and the task is to estimate the remaining 90% of the entries. We then compare the test error of CP decomposition and the quadratic model for a given rank r factorization, measured in terms of the normalized Frobenius error of the recovered tensor $\hat{T}$ on the missing entries,

$$\sqrt{\frac{\sum_{(i,j,k)\notin\Omega}(\hat{T}_{i,j,k} - T_{i,j,k})^2}{\sum_{(i,j,k)\notin\Omega} T_{i,j,k}^2}}.$$

We report the result in Table 3. The CP model outperforms the quadratic model on this task, and the gap is larger for higher values of the rank r. We suspect this is because generalization to the test data is not the challenging aspect of this task, and performance only depends on the training error because a) we found that the best value of the ℓ_2 regularization parameter for both algorithms was actually 0, b) the models do not overfit even after setting the rank to be very high, such as $r = 100$.

Figure 4: One slice of the tensor for the hyperspectral image Riberia.