PRINCETON UNIV. F'16 COS 521: ADVANCED ALGORITHM DESIGN

Lecture 19: Tensor Methods

Lecturer: *Pravesh Kothari* Scribe:*Pravesh Kothari*

This lecture introduces tensors, a higher dimensional analog of matrices and their use in finding and intepreting patterns in data. The notes are based on the Rong Ge's excellent blog post on the topic and a similar lecture taught by Tim Roughgarden and Greg Valiant at Stanford (see references).

In earlier classes, we saw that the SVD gives us a method to write any rank r matrix $M \in \mathbb{R}^{m\times n}$ as

$$M = \sum_{i\leq r} \lambda_i u_i v_i^\top,$$

where $u_i \in \mathbb{R}^m$, $1 \leq i \leq r$ and $v_i \in \mathbb{R}^n$, $\leq i \leq r$ are mutually orthogonal sets of unit vectors. This can be rewritten as $M = UV^\top$ where U is the matrix with columns $u_1, u_2, \ldots, u_r$ and V, the matrix with columns $v_1, v_2, \ldots, v_r$.

However, such a representation of M is far from unique: for any orthogonal matrix $C \in \mathbb{R}^{r\times r}$, $M = (U \cdot C)(C^\top V^\top)$ gives a different decomposition of M. This is problematic in situations where we want to assign "meaning" to the rank 1 terms in a decomposition of a matrix. We start with the motivating example of a psychological study to measure intelligence proposed by Charles Spearman (borrowed from Rong Ge's excellent blog post article on the Off Convex blog, see references for a link) which illustrates such a situation.

We will see how higher dimensional tensors naturally come to our rescue here. Although most natural analogs of matrix problems are NP-hard in general, we will be able to give efficient algorithms in fairly general settings for the tensor decomposition problem, the analog to the matrix decomposition problem discussed above.

0.1 The Spearman Study

See Rong Ge's blogpost (linked in the references) for details of the Spearman study discussed in the class.

0.2 Formal Definitions

We now formally define tensors. A k tensor of dimensions is $n_1 \times n_2 \times \cdots \times n_k$ dimensional array of numbers. A 2 tensor is just a $n_1 \times n_2$ dimensional matrix.

How should we define a rank 1 tensor? Any $n_1 \times n_2$ rank 1 matrix can be written as $uv^\top$ for $u \in \mathbb{R}^{n_1}$ and $v \in \mathbb{R}^{n_2}$. While we have seen and used such a representation of a rank 1 matrix many times in this class, we now formally define an operation that takes two vectors $u \in \mathbb{R}^{n_1}$ and $v \in \mathbb{R}^{n_2}$ and produces a $N_1 \times n_2$ matrix $uv^\top$ as the *outer* product of u and v. Recall that the *inner* product of u and v (of matching dimensions) produces a scalar instead.

We now generalize the outer product to multiple vectors to define a rank 1 tensor. Specifically,

Definition 1 (Outer Product) *for $u_1, u_2, \ldots, u_k$ in $\mathbb{R}^{n_1}, \mathbb{R}^{n_2}, \ldots, \mathbb{R}^{n_k}$ respectively, the outer product $u_1 \otimes u_2 \otimes \cdots \otimes u_k$ of $u_1, u_2, \ldots, u_k$ is defined as the k-tensor with $i_1, i_2, \ldots, i_k$ entry given by $\prod_{j \leq k} u_j(i_j)$.*

We define a rank 1 k tensor as simply an outer product of k vectors.

Next, we must define a rank r tensor. For matrices M, observe that one can define the rank as the minimum integer r such that there are r pairs of vectors such that their outer products add up to M. We use a direct generalization of this to define a rank r tensor.

Definition 2 (Rank r tensor) *Given a k tensor $T \in \mathbb{R}^{n_1 \times n_2 \times \cdots n_k}$, the rank of T is defined as the minimum r such that T can be written as a sum of r rank 1 tensors of the same dimensions as T.*

Finally, we can define the problem of tensor decomposition.

Definition 3 (Tensor Decomposition) *Given a rank r, k-tensor T, the tensor decomposition problem asks for a decomposition of T as a sum of r rank 1 tensors (of appropriate dimensions).*

In the matrix world, we saw that low-rank decompositions are not generally unique. In contrast, tensor decompositions will be unique under very general conditions.

0.3 Examples of Tensors

We discuss two natural and widely used higher dimensional tensors.

For a probability distribution μ over $\mathbb{R}^n$, the third (multilinear) moment tensor of μ is defined as $T \in \mathbb{R}^{n \times n \times n}$ such that $T_{i,j,k} = \mathbb{E}_{x \sim \mu}[x_i x_j x_k]$. One can generalize this to higher moments naturally. Moment tensors appear naturally in the study of probability distributions especially because one can estimate them efficiently from samples drawn from a distribution and one can design various algorithms with distributions as inputs using the moment tensors.

Another example is a k-gram. Given a text corpus (such as the English language wikipedia) with n possible dictionary words, one can define a tensor $T \in \mathbb{R}^{n^{\otimes k}}$ so that for any $(i_1, i_2, \ldots, i_k)$, $T_{i_1, i_2, \ldots, i_k}$ is the number of times words $w_{i_1}, w_{i_2}, \ldots, w_{i_k}$ appeared consecutively in the corpus (where w_j denotes the j^{th} word in the dictionary.) As one can imagine, k-grams are useful in building models for languages.

0.4 Tensor Decomposition: Jennrich's Algorithm

We now give an algorithm for computing a low-rank decomposition of a tensor whenever it exists.

The main primitive is an algorithm for computing the eigendecomposition of a matrix $M = QSQ^{-1}$ whenever it exists where S is a diagonal matrix. We saw the special case when M is symmetric in an earlier class - however, such a decomposition can exist (and can then be computed) for non-symmetric matrices too.

We now describe Jennrich's algorithm when all components are of dimension n. Jennrich's algorithm continues to work whenever the components of the tensor are promised to be orthogonal (and doesn't need all components to be of same dimension).

1. **Input:** A $n \times n \times n$ tensor $T = \sum_{i=1}^{k} u_i \otimes v_i \otimes w_i$ - here u_i are unknown non-zero n dimensional vectors.
2. Choose random unit vectors $a, b \in \mathbb{R}^n$.
3. Define contractions of T using a and b: i.e. take matrix $M_a = \sum_{i=1}^{k} \langle w_i, a \rangle u_i v_i^\top$ and $M_b = \sum_{i=1}^{k} \langle w_i, b \rangle u_i v_i^\top = RTR^{-1}$.
4. Compute the eigen-decomposition of $M_a M_b^{-1} = QSQ^{-1}$ and $M_a^{-1} M_b$.
5. We can show that with high probability, the entries of the diagonal matrices S are distinct and inverses of the corresponding entries of T. Then, the columns of the matrix Q are the vectors (in some arbitrary permutation) $u_1, u_2, \ldots, u_k$ and columns of matrix R are the vectors $v_1, v_2, \ldots, v_k$ - notice that the eigenvalue corresponding to u_i is the reciprocal of the eigenvalue corresponding to v_i so we can match u_is and v_is correctly.
6. Given u_is and v_is,we can solve the linear system of equations to find w_is.

Observe that by definition, $M_a = UDV^\top$ and $M_b = UEV^\top$ where the columns of U are u_is and the columns of V are v_is and D, E are diagonal matrices with entries $\langle w_i, a \rangle$ and $\langle w_i, b \rangle$ respectively.

Thus, $M_a M_b^{-1} = UDV^\top (V^\top)^{-1} E^{-1} U^{-1} = U(DE^{-1})U^{-1}$ and similarly, $M_a^{-1} M_b = (V^\top)^{-1} D^{-1} U^{-1} UEV^\top = (V^\top)^{-1}()D^{-1}E)V^\top$.

The correctness of the algorithm now follows from the uniqueness of eigendecomposition of a matrix when the eigenvalues are distinct.

For a random choice of a and b, it is easy to show that with high probability (in fact with probability 1), given that us, vs and ws are orthogonal (linear independence is in fact enough), the eigenvalues are distinct.

The above argument can thus be used to prove the following theorem.

THEOREM 1
Given a 3 dimensional symmetric tensor

$$T = \sum_{i=1}^{r} u_i \otimes u_i \otimes u_i \tag{1}$$

for orthogonal vectors $u_1, u_2, \ldots, u_n$ or length at most 1 and any $\epsilon > 0$ there exists a polynomial time (in n and $1/\epsilon$) algorithm to recover $\tilde{u}_1, \tilde{u}_2, \ldots, \tilde{u}_r$ such that for each $i \leq r$ $\|u_i - \tilde{u}_i\|_2 \leq \epsilon$. As a consequence, we also get that a representation as in (1) is unique up to permutation of the r components. (Note: The orthogonality or symmetry conditions on components of T is not necessary for uniqueness or efficient recovery, linear independence suffices with a simple preprocessing step called as "whitening" before applying Jennrich's algorithm above.)

Bibliography

1. *Blog Post: "Tensor Methods in Machine Learning."* Rong Ge. [Online]. Available: http://www.offconvex.org/2015/12/17/tensor-decompositions/

2. *: The Modern Algorithmic Toolbox, Lecture 10: Tensors and Low-Rank Tensor Recovery* Tim Roughgarden and Gregory Valiant. [Online]. Available: http://theory.stanford.edu/~tim/s15/l/l10.pdf