CS 329M: Topics in Artificial Intelligence **Spring 2017**

Lecture 2: April 10

Lecturer: James Zou *Scribe: Stephen Pfohl*

2.1 Spectral Methods 1: Tensor Methods

In these notes we will be investigating the usage of tensor decompositions for efficiently learning latent variable models.

2.1.1 Historical Context and Motivation

In 1904, Charles Spearman (most notably known for the Spearman Correlation and Factor Analysis) attempted to describe human intelligence in terms of a composition of unobserved latent factors. For purposes of a simplified explanation, consider a model in which intelligence may be described in terms of exactly two unobserved factors, namely *quantitative* and *verbal* intelligence. In particular, this model posits that each student may be described by a vector $\in \mathbb{R}^2$ where the first and second element of the vector correspond to the student's quantitative and verbal intelligence and that the difficulty of each school subject (e.g. Math, Physics, Literature, etc) may similarly be represented by a vector $\in \mathbb{R}^2$ where the first and second element correspond the the quantitative and verbal difficulty of that subject, respectively.

2.1.1.1 The Matrix Decomposition

More concretely, Spearman considered the case in which the scores of n_1 students were recorded for exams on n_2 school subjects such that these scores may be arranged in a matrix $M \in \mathbb{R}^{n_1 \times n_2}$. Given this observed data, the goal is to infer the values of two unobserved matrices: the *student matrix* $U \in \mathbb{R}^{n_1 \times 2}$ and the *subject matrix* $V \in \mathbb{R}^{n_2 \times 2}$ such that

$$M = UV^T$$

Framed in this way, each row of U corresponds to the quantitative and verbal scores for each student and each row of V corresponds to the quantitative and verbal difficulties of each school subject.

Let $U_q, U_v \in \mathbb{R}^{n_1}$ be the quantitative and verbal components (first and second columns) of U and define V_q, V_v analogously for the matrix V. Then M can be represented as follows

$$M = U_q V_q^T + U_v V_v^T$$

The above problem of determining U and V may be referred to as the problem of **matrix decomposition**. However, an arbitrary solution U and V to the matrix decomposition problem is not unique. This is problematic if one wishes to interpret the meaning of the learned latent factors, as the natural interpretation of one possible solution may directly contradict the conclusions that are naturally drawn from some other solution.

For intuition into why the solution is not unique, consider that an arbitrary rotation represented by the orthogonal rotation matrix $R \in \mathbb{R}^{2 \times 2}$ may be applied to both U and V without affecting M, since $RR^T = I$,

we have

$$UR(VR)^T = URR^TV^T = UV^T = M$$

As it turns out, the solution can be uniquely determined in two cases:

1. U **and** V **are orthogonal matrices**. Note that this means that the columns of U must each be orthogonal to each other and the columns of V must be orthogonal to each other, but puts no restriction on the relationship between the columns of U to the columns of V.
2. We have a tensor representation of the observed data allowing for a **tensor decomposition** to be applied.

The first case corresponds to the solutions one would obtain from singular value decomposition (SVD). However, this orthogonality restriction requires that each of the learned latent factors to be completely uncorrelated. This is an unsatisfactory modeling assumption for the intelligence example, as it is unlikely that quantitative and verbal intelligence are entirely uncorrelated. Given these restrictions, we now consider the extension of this model to the tensor case.

2.1.2 Introducing the Tensor

A tensor is a higher order generalization of the matrix. For the purposes of these notes, we will primarily be considering 3rd order tensors. For intuition, consider that if a matrix is an array of values arranged in a rectangular grid then a 3rd order tensor can be thought of as a collection of fixed size matrices arranged in a stack.

To extend the intelligence and test score example to the tensor case requires an additional dimension of measurement. For this, we can consider measuring the student's scores over time. If each of the tests are performed on each student in both the morning and the afternoon, M is now a tensor $\in \mathbb{R}^{n_1 \times n_2 \times 2}$. If U and V are now defined as before, it is now of interest to learn an additional matrix $W \in \mathbb{R}^{2\times 2}$ where the first row corresponds to the effect of the morning on the quantitative and verbal scores and the second row corresponds to the analogous effects of the afternoon. Define W_q and W_v to be the columns of W corresponding to the quantitative and verbal components of W, as was done before for matrices U and V.

Element i, j, k of the 3rd order tensor M is then defined as follows

$$M(i, j, k) = U_q(i)V_q(j)W_q(k) + U_v(i)V_v(j)W_v(k)$$

To generalize the above, for R latent factors such that $M \in \mathbb{R}^{n_1 \times n_2 \times n_3}, U \in R^{n_1 \times R}, V \in R^{n_2 \times R}, W \in R^{n_3 \times R}$,

Definition 1 $M(i, j, k) = \sum_{r=1}^{R} U(i, r)V(j, r)W(k, r)$

Definition 1 may be succinctly expressed with the tensor product

$$M = U \otimes V \otimes W$$

2.1.3 Properties of the Tensor Product

We now summarize some basic properties of the tensor product. Readers looking for more details on the tensor product are referred to [1]. Note that it is a generalization of the outer product and thus $U \otimes V = UV^T$ if U and V are matrices.

Theorem 2 (Scalar multiplication)

$$(aU) \otimes V \otimes W = a(U \otimes V \otimes W)$$

Proof: Follows trivially from Definition 1. ■

Theorem 3 (Addition)

$$(U_1 + U_2) \otimes V \otimes W = U_1 \otimes V \otimes W + U_2 \otimes V \otimes W$$

Proof: Each element of the sum $U_1 + U_2$ may be rewritten as the sum of the corresponding elements of U_1 and U_2. From Definition 1,

$$\begin{aligned}\sum_{r=1}^{R}(U_1(i,r) + U_2(i,r))V(j,r)W(k,r) &= \sum_{r=1}^{R} U_1(i,r)V(j,r)W(k,r) + \sum_{r=1}^{R} U_2(i,r)V(j,r)W(k,r) \\ &= U_1 \otimes V \otimes W + U_2 \otimes V \otimes W\end{aligned}$$

■

Theorem 4 (Order of terms matters)

$$U \otimes V \otimes W \neq V \otimes U \otimes W$$

Proof: Following from the intuition that $U \otimes V = UV^T$, note that the first two terms on each side of the equation can be represented by $LHS = UV^T$ and $RHS = VU^T = (UV^T)^T$. Given that we cannot assume that UV^T is symmetric, it follows that the two sides of the relation are not equal in general. ■

Theorem 5 (Two rotations cancel out) *Let R be an orthogonal rotation matrix, as before*

$$UR \otimes VR = U \otimes V$$

Proof:

$$\begin{aligned}UR \otimes VR &= (UR)(VR)^T \\ &= URR^TV^T \\ &= UV^T \\ &= U \otimes V\end{aligned}$$

■

Theorem 6 (Three rotations do not cancel out)

$$UR \otimes VR \otimes WR \neq U \otimes V \otimes W$$

Proof: Repetition of the process in the proof for Theorem 5 demonstrates that two of the rotation matrices will reduce to the identity matrix, but an additional rotation will remain in the expression for the final product. This implies that the results of the tensor decomposition are not invariant under an arbitrary coordinate rotation. ■

Theorem 7 (Tensor-vector multiplication) *If $M \in \mathbb{R}^{n_1 \times n_2 \times n_3} = U \otimes V \otimes W$ and $z \in \mathbb{R}^{n_3}$, then*

$$M \otimes z = U \otimes V \otimes (Wz)$$

2.1.4 The Tensor Decomposition

Given the prior background, we are now ready to define the tensor decomposition and consider an algorithm for finding a unique decomposition for any third order tensor that does not rely on orthogonality constraints and instead only relies on the linear independence of the columns of the matrices in the decomposition.

Theorem 8 (Efficient Tensor Decomposition) *For a third order tensor*

$$T = \sum_{r=1}^{R} \alpha_r U_r \otimes V_r \otimes W_r$$

where subscript r refers to the rth column of an arbitrary matrix, $\|U_r\|_2 = \|V_r\|_2 = \|W_r\|_2 = 1$, and scalar α_r, there exists an efficient algorithm to find the sets $\{\alpha_r\}, \{U_r\}, \{V_r\}, \{W_r\}$ provided that each set of vectors are linearly independent.

2.1.4.1 Jennrich's Algorithm

We now describe Jennrich's algorithm [2, 3] for performing efficient tensor decomposition.

Let $T \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ be a tensor that has some decomposition

$$T = \sum_{r=1}^{R} \alpha_r U_r \otimes V_r \otimes W_r$$

for $U \in \mathbb{R}^{n_1 \times R}, V \in \mathbb{R}^{n_2 \times R}, W \in \mathbb{R}^{n_3 \times R}$ where index r refers to the rth column.

Jennrich's algorithm

1. Generate vectors $a, b \in \mathbb{R}^{n_3}$ uniformly such that $\|a\| = \|b\| = 1$.
2. Define T_a and T_b to be the matrices that result from the projection of T onto vectors a and b.

   $$T_a = \sum_r \alpha_r (W_r^T a) U_r \otimes V_r$$

   $$T_b = \sum_r \alpha_r (W_r^T b) U_r \otimes V_r$$

   Define D_a to be a diagonal matrix $\in \mathbb{R}^{R \times R}$ such that the the ith element of the diagonal is given by $\alpha_i W_i^T a$ and define D_b analogously. Note that $T_a = U D_a V^T$ and $T_b = U D_b V^T$.
3. Finding the eigenvectors of $T_a T_b^+$ gives the matrix U, where here "+" denotes pseudo-inverse. As

   $$T_a T_b^+ = U D_a V^T (V^T)^+ D_b^+ U^+ = U D_a D_b^+ U^+$$

   therefore $T_a T_b^+ U = U D_{a/b}$, where $D_{a/b} = D_a D_b^+$ is the elementwise division of D_a with D_b.
4. Similar to the previous step, by symmetry, the eigenvectors of $T_b T_a^+$ give the matrix V.
5. Compute W and the vector α containing the set of α_r's by noting that $D_{a/b}$ and $D_{b/a}$ are the eigenvalues of $T_a T_b^+$ and $T_b T_a^+$ and are a function of the α, W, and the random vectors a, b. For example, $D_{a/b} = \alpha^T \operatorname{diag}(W^T a / b)$ and $D_{b/a} = \alpha^T \operatorname{diag}(W^T b / a)$ if a/b designates the elementwise division of vector a with vector b.

2.1.5 Applications

2.1.5.1 Topic Modeling

One application of tensor decomposition methods is topic modeling. In the topic modeling paradigm, we assume that natural language text is generated by some latent topics such as sports, politics, weather, etc. In the simplest formulation, for each topic, there is some vector $A_r \in \mathbb{R}^V$ representing a probability distribution over a vocabulary of size V that describes the probability of each word being generated by that topic. The generative model for producing text is then fully described by the latent topic vectors and some weights α_r that give the probability of observing each topic in the corpus. In this example, the goal of the topic modeling procedure is to simultaneously estimate the set of A_r vectors and the α_r weights.

Traditional approaches for approaching the topic modeling problem, such as Latent Dirichlet Allocation (LDA), have relied on computational intensive iterative procedures such as Gibbs Sampling and Markov Chain Monte Carlo (MCMC) that have no guarantees of convergence to a satisfactory solution. However, the tensor decomposition method previously described is capable of efficiently producing a robust and high quality solution. The algorithm proceeds by initializing a co-occurrence tensor $T \in \mathbb{R}^{V \times V \times V}$ with zeros. Word triplets are then randomly sampled from the corpus and $T(i, j, k)$ incremented by one if the triplet corresponding to the words in indices i, j, k are observed. The matrix T then takes on the following probabilistic interpretation where each word is conditionally independent given the topic

$$T(i, j, k) = \sum_r p(\text{topic} = r)p(i|r)p(j|r)p(k|r)$$

The co-occurrence tensor may be represented in expectation by the following, allowing for tensor decomposition to find the topic vectors A_r

$$T = \sum_r \alpha_r A_r \otimes A_r \otimes A_r$$

We can also consider learning the topic vectors for more complex graphical models, as the prior example was fairly limited as a language model due to the conditional independence assumption. For instance, we may consider a case in which we would like to model *ordered* consecutive word triplets are generated by some set of topics. If we assume that the words appear in the sequence a, b, c, we may then analogously initialize the model as before, but now take the first, second, and third dimensions of T to represent words that appear in the a, b, and c slots, respectively. Then, consecutive three word ordered samples are sampled from the corpus and T incremented. The tensor decomposition then allows for the set of ordered topic vectors to be learned as follows

$$T^{abc} = \sum_r \alpha_r A_{ra} \otimes A_{rb} \otimes A_{rc}$$

It is generally possible to consider the learning of more complex graphical structure by using this decomposition as a sub-routine. For instance, consider the model in which some latent state generates an observed variable d and an unobserved variable x that generates the observed variables a, b and c. Triplets of observed edge weights can be learned simultaneously, meaning A_d and $A_x A_a$ and $A_x A_b$ may be learned in one tensor decomposition and A_a, A_b, and A_c in another decomposition. A_x may then be calculated with linear algebra given that $A_x A_a, A_x A_b, A_a, A_b$ are known.

References

[1] Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M. Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *J. Mach. Learn. Res.*, 15(1):2773–2832, January 2014.

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

[3] Richard Harshman. Foundations of the parafac procedure: Models and conditions for an "explanatory" multi-modal factor analysis. *UCLA Working Papers in Phonetics*, 16, 1970.