**COS 597C: New Directions in Theoretical Machine Learning** **Fall 2017**

# Lecture 6: 3 October 2017

*Lecturer: Sanjeev Arora* *Scribe: Woon Sang Cho*

**Note**: *LaTeX template courtesy of UC Berkeley EECS dept.*

## 6.1 Tensor Decomposition

We begin with Spearman's Hypothesis. He believed there are two types of intelligence: one is verbal and the other mathematical. To test this hypothesis with data $M$, we denote

$$M_{i,j} = \text{score of student } i \text{ on test } j$$

Now we denote the latent vectors of the hypothesis as follows:

$$u_i^v = \text{student } i\text{'s verbal intelligence}$$

$$u_i^m = \text{student } i\text{'s math intelligence}$$

We would expect

$$M = \begin{pmatrix} | \\ \mathbf{u}^v \\ | \end{pmatrix} \begin{pmatrix} \text{---} & \bar{\alpha} & \text{---} \end{pmatrix} + \begin{pmatrix} | \\ \mathbf{u}^m \\ | \end{pmatrix} \begin{pmatrix} \text{---} & \bar{\beta} & \text{---} \end{pmatrix}$$

This is called the "Latent Factor Analysis".

Now we assume that the test itself has two parts: *A and B*. We further assume that for part A of all test$_j$, there exist $\alpha_j, \beta j > 0$ such that

$$\text{score on test}_j = \alpha_j \times \text{verbal intelligence} + \beta_j \times \text{math intelligence}$$

Similarly, for part B of all test$_j$, there exist $\alpha_j^{'}, \beta j^{'} > 0$ such that

$$\text{score on test}_j = \alpha_j^{'} \times \text{verbal intelligence} + \beta_j^{'} \times \text{math intelligence}$$

Therefore we have data of form $M_{i,j,\sigma}$ where $\sigma = A, B$. This type is a tensor, a higher dimensional matrix, which element is indexed as $T_{i,j,k}$.

## 6.2 Other examples of linear models

We know that matrices can be factored in many ways. For example,

$$M = UV \implies M = URR^T V$$

, where $R$ is a rotation matrix. Singular value decomposition (SVD) gives us another representation, and rectangular matrices can be written in the form $\Sigma_i \sigma_i u_i v_i^T$.

### 6.2.1 Independent Component Analysis

Consider the cocktail party problem. You are at a crowded party and you have one ear that intakes sounds of various sources. One thing to note is that these sound signals are superimposed *linearly*. However, the cocktail party phenomenon is that, despite this environment where one receives a single signal composed of many source signals, one can filter and *focus* on one conversation, and discard all other sources embedded in the received signal.

One can achieve the same goal for this linear problem by using **Independent Component Analysis**. The assumption is that the mixed signal we receive ($S = Ax$) where $A$ is the mixing matrix, and $x$ is the source matrix which coordinates are independent random variables. The goal is to learn both $A$ and $x$.

### 6.2.2 Topic Models

Suppose we have a corpus of documents. Given many documents, the goal is to recover topics. It turns out that simply with *Bag of Words* vectors, one can easily solve this problem.

Let $A^{(1)}$ denote the distribution on words for topic 1. Similarly, $A^{(2)}$ denotes the distribution on words for topic 2. Sampling from these distributions results in a Bag of Words vector as follows.

$$\text{document}_j = \begin{pmatrix} | \\ i \\ | \end{pmatrix}$$

where $i$ denotes the number of times that $i^{th}$ word appears in this particular document $j$. In practice, since this bag of word vector contains indices to all words in the corpus, any realized or given document contains many more 0s. In other words, only a few are non-zero. Given this setting, specifically the goal is to recover the distribution matrix $\overline{A}$.

## 6.3 Jennrich's Algorithm

Suppose $T$ has a decomposition of the form:

$$T = \Sigma_{i=1}^{r} u_i \otimes v_i \otimes w_i$$

where $\{v_i\}$ are independent, $\{w_i\}$ are independent for all $i$. Furthermore, every pair of $\{u_i\}$s is independent.

**Lemma 1.** *This decomposition is unique and it can be found in time poly($n, \frac{1}{\epsilon}$) within accuracy $\epsilon$.*

The main idea is the **Matricize**. We pick random vectors $a$ and $b$, and denote the following matrices.

$$M_a = \Sigma_{i=1}^{r} \langle a, u_i \rangle \otimes v_i \otimes w_i$$

$$M_b = \Sigma_{i=1}^{r} \langle b, u_i \rangle \otimes v_i \otimes w_i$$

**Lemma 2.** $\{v_i\}$ = *eigenvectors of* $M_a M_b^{-1}$, *and* $\{w_i\}$ = *eigenvectors of* $(M_a^{-1} M_b)^T$ *and we can obtain the pairing information of the decomposition (which vector pairs with which, to be shown soon).*

With this lemma, it will be easy to find $u_i$s by solving linear equations.

*Remark* 1. Eigenvalues:

$$Av = \lambda v$$

where $A$ is $n \times n$ square matrix (not necessarily symmetric).

Suppose $A$ has $n$ (independent) eigenvectors, denoted as,

$$Q = \begin{bmatrix} | & | & | \\ q_1 & \cdots & q_n \\ | & | & | \end{bmatrix}$$

and

$$AQ = Q \cdot \text{diag}(\lambda)$$

Then,

$$A = Q \cdot \text{diag}(\lambda) \cdot Q^{-1}$$

Also we use the following property:

$$(AB)^{-1} = B^{-1}A^{-1}$$

### 6.3.1 Algorithm

$$M_a = VD_aW^T$$

where

$$D_a = \begin{pmatrix} \langle a, u_1 \rangle & & & \\ & \langle a, u_2 \rangle & & \\ & & \cdots & \\ & & & \langle a, u_n \rangle \end{pmatrix}$$

Also

$$M_b = VD_bW^T$$

with $D_b$ defined similarly as above.

Then we obtain,

$$\begin{aligned} M_a M_b^{-1} &= VD_aW^T(VD_bW^T)^{-1} && (6.1) \\ &= VD_aW^T(W^T)^{-1}D_b^{-1}V^{-1} && (6.2) \\ &= VD_aD_b^{-1}V^{-1} && (6.3) \\ & && (6.4) \end{aligned}$$

Note the eigenvalue matrix $D_aD_b^{-1}$, and the eigenvector matrix $V$.

We can follow the same procedure for the other matrix: $(M_a^{-1}M_b)^T$, and obtain an eigenvalue matrix $D_bD_a^{-1}$ (which is the reciprocal of diagonal entries in $D_aD_b^{-1}$) and their corresponding eigenvector matrix $W$.

The final step is to pair up $v_i$ and $w_i$ if and only if their eigenvalues are reciprocals and solve for each $u_i$ in $T = \Sigma_{i=1}^{r} u_i \otimes v_i \otimes w_i$.

This concludes Jennrich's algorithm for tensor decomposition.