Tensor decomposition + Another method for topic modeling

Lecturer: Sanjeev Arora
Scribe: Holden Lee

April 13, 2015

Today we talk about *tensor decomposition*, a general purpose tool for learning latent variable models. Then we switch gears and talk about a recent improvement of the topic modeling algorithm we saw in an earlier lecture.

0.1 Tensor decomposition

Tensor decomposition is the analog of spectral decomposition for tensors.

The nice thing about eigenvalues/eigenvectors is that they exist (ok, singular values/vectors in case of nonsymmetric matrices) and you can efficiently compute them. For M a symmetric $n \times n$ matrix, we can write

$$M = \sum \lambda_i u_i u_i^T.$$

A 3-D tensor M is a $n \times n \times n$ array. Extending linear algebra to tensors is nontrivial. Many problems regarding tensors are NP-hard, like rank (which is not straightforward to define).

Today we are interested in tensors that we are guaranteed have a representation like $M = \sum \lambda_i u_i^{\otimes 3}$, where the u_i are orthogonal. We don't know the u_i's and are trying to recover them. We can actually recover these similarly to the **power method**. (Recall that the power method repeatedly sets $x \hookleftarrow \frac{Mx}{\|Mx\|_2}$; it gives the top eigenvector if there is a gap between the top 2 eigenvalues. The running time is inversely proportional to this gap.)

Definition 0.1: The **tensor-vector product** (aka *flattening* by x) is defined as follows: Mx is the matrix where

$$(Mx)_{ij} = \sum_k M_{ijk} x_k.$$

Now

$$Mx = \sum \lambda_i (u_i \cdot x) u_i^{\otimes 2}.$$

This looks like a spectral decomposition: it takes the orthogonal directions u_i and boosts them by $\lambda_i(u_i \cdot x)$. (Under the isomorphism $V \otimes V \cong V \otimes V^*$, $u_i^{\otimes 2}$ corresponds to $u_i u_i^T$.)

Why does this work? From inspection, the eigenvalues of Mx are $u_i \cdot x$ since the u_i's are orthonormal and spectral decomposition is unique. The Mx's are approximately Gaussian, and there is a good chance that Mx has a top eigenvalue, with a significant gap to the next eigenvalue.

0.1.1 Method of moments

In topic modeling, etc., what is really going on is that we are using the method of moments. The general setup is that we sample

$$x \sim D := D(A)$$

where A is the matrix of hidden parameters; given observed X we try to recover A. We can consider the moments

$$\begin{aligned} \mathbb{E}X &= f_1(A) \\ \mathbb{E}(X^{\otimes 2}) &= f_2(A) \\ \mathbb{E}(X^{\otimes 3}) &= f_3(A) \\ &\vdots \end{aligned}$$

Then we try to solve this nonlinear system of equations. A lot of machine learning can be thought of in this way.

Mathematicians and statisticians have studied questions like: What distributions can we identify from the third moments, or up to the kth moments?

Recall that in topic modelling, under the separability assumption, a document is sampled from A with $w \in \mathrm{Dir}(\alpha)$. We considered

$$XX^T = A \underbrace{\mathbb{E}[ww^T]}_{R} A^T$$

and used separable matrix factorization. We were exactly using second moments to recover the distribution.

See [AGH$^+$14] for more on this framework.

Dictionary learning was not method of moments; we drew edges between X, X' when $|\langle X, X' \rangle| \geq \frac{1}{2}$ and used community detection on the resulting graph.

0.1.2 Example: Mixtures of identical spherical gaussians

Consider k Gaussians $N(\mu_i, \sigma^2)$ in n dimensions ($\mu_i \in \mathbb{R}^n$) where σ^2 *is known*. Let the mixing weights w_i be such that $\sum_{i=1}^k w_i = 1$. To pick a sample, pick i with probability w_i, and output a sample from $N(\mu_i, \sigma^2)$. We have

$$\begin{aligned}\mathbb{E}[X] &= \sum_{i=1}^k w_i\mu_i \\ \mathbb{E}[X^{\otimes 2}] &= \sum_{i=1}^k w_i\mu_i^{\otimes 2} + \sigma^2 I \\ \mathbb{E}[X^{\otimes 3}] &= \sum_{i=1}^k w_i\mu_i^{\otimes 3}\end{aligned}$$

Assume we shift coordinates so that $\mathbb{E}[X] = 0$, and that the μ_i are linearly independent. If we can do a tensor decomposition of $\mathbb{E}[X^{\otimes 3}]$ then we will obtain the μ_i and weights w_i. However, we can't do tensor decomposition yet because the μ_i are in general not orthogonal. We must first whiten the vectors.

0.1.3 Whitening

The idea of **whitening** is to change tensors of the form $\sum w_i\mu_i^{\otimes 3}$ to $\sum w_i\nu_i^{\otimes 3}$ where the ν_i's are orthogonal. Letting $U = (\mu_1, \ldots, \mu_n)$, we have

$$P = \sum_{i=1}^k w_i\mu_i^{\otimes 2} = U \operatorname{diag}(w_i) U^T.$$

(This is not the spectral decomposition, because U is not orthogonal.) The spectral decomposition is, say

$$P = VDV^T$$

where V is orthogonal. *Assume U, V are full rank.* We would like to find a matrix A such that the vectors $\nu_i := A\sqrt{w_i}\mu_i$ are orthogonal, i.e., $AU\operatorname{diag}(\sqrt{w_i})$ are orthogonal. This is equivalent to

$$[AU\operatorname{diag}(\sqrt{w_i})][\operatorname{diag}(\sqrt{w_i})U^T A^T] = 1 \iff APA^T = 1.$$

Thus, take $A = W^T$ where $W = VD^{-\frac{1}{2}}$. Then

$$APA^T = D^{-\frac{1}{2}}V(VDV^T)V^T D^{-\frac{1}{2}} = I$$

as needed.

In the Gaussian case, if we applied W to $\sum_{i=1}^k w_i\mu_i^{\otimes 3}$, we would get

$$\sum w_i(W^T\mu_i)^{\otimes 3} = \sum \frac{1}{\sqrt{w_i}}\nu_i^{\otimes 3}.$$

(Of course, we actually get the noisy versions of $\sum_{i=1}^k w_i\mu_i^{\otimes 2}$, $\sum_{i=1}^k w_i\mu_i^{\otimes 3}$, so if we want to do proper analysis we'll have to take error into account.)

(See also [BCMV13] for a somewhat different setting, overcomplete tensor decomposition.)

0.2 SVD-based approaches for topic models (presentation by Andrej Risteski)

We explain a paper by Bansal, Bhattacharyya, and Kannan [BBK], which uses SVD plus some other tricks. They develop and prove a SVD-based algorithm that learns topic models with L^1 error under certain assumptions including the catch words assumption (a weakening of the anchor words assumption).

We set up notation. Let k be the number of topics and n be the number of words. Let A be the words×topics matrix, giving the distribution of words for each topic, and W be the topics×documents matrix. Let $M = AW$. If $W_{\bullet i}$ is a column of W, then $\widetilde{M_{\bullet i}}$ is generated according to m draws on the distribution given by $M_{\bullet i}$. (m is the number of words in each document.)

The goal is to recover A with L^1 error. Previous works such as Arora et al. recovered with L^2 error. Note that L^2 error ignores words with small frequency, and empirically, a lot of words have small frequency. Moreover, columns are distributions so the natural norm is L^1.

0.2.1 Assumptions

We make the following assumptions. See the paper for the precise parameters.

1. (Dominating topic) We assume there is a **dominating topic** in each document:
   (a) for each document d there exists a topic $t(d)$ such that $W_{t(d),d} > \alpha$. For all other topics $t' \neq t(d)$, $W_{t',d} \leq \beta$, where $\beta - \alpha$ is large enough.
   (b) (Each topic appears as a dominating topic enough times) For each topic t there are $\geq \varepsilon_0 w_0 s$ documents d in which $W_{t,d} \geq 1 - \delta$.

2. **Definition 0.2:** w is a **catch word** for topic t if for all $t' \neq t$, $A_{wt'} \leq \rho A_{wt}$, and the probability of appearing is not too small, $A_{wt} \geq \frac{8}{m\delta^2\alpha} \ln\left(\frac{20}{\varepsilon w_0}\right)$.

   The catch words for t occupy a significant proportion of the words for topic t

$$\sum_{w \text{ is catch word for topic } t} A_{wt} > \frac{1}{2}.$$

   (You can replace $\frac{1}{2}$ by p_0, and get dependence on p_0 in later parameters. For simplicity we don't do this. There is some absolute lower bound on p_0.).

3. (Almost pure documents) There is a small fraction of almost pure documents. For all i, $\geq \varepsilon_0 w_0 D$ of the documents are such that $W_{td} > 1 - \delta$.

4. (No-local-minima assumption) Let $p_j(\zeta, t)$ be the probability that t is the dominant topic in the document, word j appears ζ time, i.e., with proportion $\frac{\zeta}{m}$. Then

$$p_j(\zeta, t) > \min(p_j(\zeta - 1, t), p_j(\zeta + 1, t)).$$

The motivation is that there are two possibilities: either the probability of the word appearing ζ times decays as ζ gets larger (e.g. as a power law), or it's a catch word, and it keeps rising until some frequency, and then decays.

5. (Dominant admixture) The proportion of documents where topic i is dominant is $\frac{D}{k}$, where k is the total number of documents.

0.2.2 Algorithm

The intuition is that topic models is like soft clustering, soft because each document doesn't belong to 1 cluster exclusively.

Intuitively, what is the obstacle? Suppose the frequency of a certain word in cluster 1 is in $[0, \sigma]$ and in cluster 2 is $[\mu, 1]$, with the spread much larger in cluster 2. Then clustering could split the second cluster into two.

This is solvable with the trick of *thresholding before clustering*. If μ is known, threshold by μ: if a coordinate is $> \mu$, then set it to be 1, and 0 otherwise. If you directly apply SVD, you can handle less noise than if you threshold first.

Consider the following problem.

Problem 0.3: Given a random $n \times n$ matrix A where some $m \times m$ submatrix has $\mathbb{P}(A_{ij} \geq \mu) \geq \frac{1}{2}$, and the other entries are $N(0, \sigma)$, find the submatrix (planted clique).

Solution. First consider the naive SVD solution.

The idea is that the spectral norm of the $m \times m$ matrix is significantly larger than the spectral norm of the rest of the matrix.

1. Let C be the subset (clique); let $\mathbb{1}_C$ be the characteristic vector. Then (assuming there is not a significant negative contribution)

$$\frac{\|A\mathbb{1}_C\|}{\|\mathbb{1}_C\|} \sim \frac{\sqrt{K(K\frac{\mu}{2})^2}}{\sqrt{K}} = O(K\mu)$$

2. The spectral norm of the random part is $\sqrt{n}\sigma$.

SVD will work whenever $K\mu \gg \sqrt{n}\sigma$,

$$\frac{\mu}{\sigma} \gg \frac{\sqrt{n}}{k}. \tag{1}$$

Now consider thresholding first:

1. If $A_{ij} > \mu$ then set $\widetilde{A}_{ij} = 1$; if $A_{ij} < \mu$ set $\widetilde{A}_{ij} = 0$. In the planted clique the entries are 1 with probability $\frac{1}{2}$; away from it entries are 1 with probability $\sim e^{-\frac{\mu^2}{s\sigma^2}}$.

2. Now we shift back so the mean on the non-clique part is 0. Set $\widetilde{\widetilde{A}} = \widetilde{A} - e^{-\frac{\mu^2}{2\sigma^2}} J$, where J the all 1's matrix.

The planted part has spectral norm $\left(\frac{1}{\sqrt{k}}\right)^2 k^2 = k$. The random part has spectral norm $\lesssim \sqrt{n}e^{-\frac{\mu^2}{2\sigma^2}}$.

Thus, after thresholding, we can solve the problem whenever $k \gg \sqrt{r}e^{-\frac{\mu^2}{2\sigma^2}}$, i.e.

$$e^{\frac{\mu^2}{\sigma^2}} \gg \frac{\sqrt{n}}{k}.$$

which is a larger range than in (1). □

The algorithm is the following (informally).

1. (Pick thresholds) For all words j, pick a threshold ζ_j as follows. Take $\zeta_j \in \{0, 1, \dots, m\}$,

   $$\zeta_j = \text{argmax}_j \left\{ \left| \left\{ d : \widetilde{M_{wd}} > \frac{\zeta}{m} \right\} \right| \geq \frac{D}{k} \text{ and } \left| \left\{ d : \widetilde{f_{jd}} = \frac{\zeta}{m} \right\} \right| \leq \varepsilon \frac{D}{k} \right\}.$$

   Then define the threshold matrix

   $$T_{wd} := \begin{cases} \sqrt{\zeta_w}, & \text{if } \widetilde{A_{wd}} > \frac{\zeta_w}{m} \text{ and } \zeta_w \text{ is not too small} \\ 0, & \text{otherwise.} \end{cases}$$

2. Now use the Swiss army knife [KK10].[1]
   (a) Take T, do a rank k-SVD, and produce $T^{(k)}$.
   (b) Run a 2-approximation for k-means to get tentative cluster centers.
   (c) Run Lloyd's algorithm on columns S of B, with starting points and centers above.
3. Determine catchwords. (See the paper for details.)
4. Determine the $(1-\delta)$-pure documents and get the topic-word mix.

A key point in the analysis is to show that the thresholding doesn't break the clusters. We need to use the non-local-min assumption.

Proposition 0.4 (Lemma A1 in [BBK])**:** If $\sum_{\zeta \geq \zeta_0} p_j(\zeta, i) \geq \nu$ and $\sum_{\zeta \leq \zeta_0} p_j(\zeta, i) \geq \nu$, then $p_j(\zeta_0, i) \geq \frac{\nu}{m}$.

Proof. Let $f(\zeta) := p_j(\zeta, i)$. One of the following happens.

1. $f(\zeta) \geq f(\zeta - 1)$ for all $n \leq \zeta_i \leq \zeta_0$
2. $f(\zeta + 1) \leq f(\zeta)$ for all $m - 1 \geq \zeta \geq \zeta_0$.

[1] The theorem says that the algorithm works when $> (1-\varepsilon)$ of points satisfy the proximity condition. M_i in cluster T_r satisfies the proximity condition if for any $s \neq r$, the projection of A_i onto the μ_r-to-μ_s line is at least Δ_{rs} closer to μ_r than μ_s. Here $\Delta_{rs} = ck\left(\frac{1}{\sqrt{n_r}+\sqrt{n_s}}\right)\|M - C\|$ where C consists of the cluster centers.

Let's assume (1). Then

$$\zeta_0 p_j(\zeta_0, i) \geq \sum_{\zeta \geq \zeta_0} p_j(\zeta, i) \geq \nu \implies p_j(\zeta_0, i) \geq \frac{\nu}{m}.$$

The other case is similar. □

Lemma 0.5 (Thresholding does not separate dominating topics, Lemma A3 in [BBK])**:** With high probability, for a fixed word w and topic t,

$$\min(\mathbb{P}(\widetilde{A_{wd}} \leq \frac{\zeta_w}{m}; d \in T_t), \mathbb{P}(\widetilde{A_{wd}} > \frac{\zeta_w}{m}, d \in T_t) \leq O(m\varepsilon w_0).$$

where T_t consists of the documents with dominant topic t.

References

[AGH+14] Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *arXiv preprint arXiv:1210.7559*, 15:1–55, 2014.

[BBK] Trapit Bansal, C Bhattacharyya, and Ravindran Kannan. A provable SVD-based algorithm for learning topics in dominant admixture corpus. pages 1–22.

[BCMV13] Aditya Bhaskara, Moses Charikar, Ankur Moitra, and Aravindan Vijayaraghavan. Smoothed Analysis of Tensor Decompositions. *arXiv:1311.3651 [cs, stat]*, 2013.

[KK10] Amit Kumar and Ravindran Kannan. Clustering with spectral norm and the k-means algorithm. *Proceedings - Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 299–308, 2010.