CS 165: Foundations of Machine Learning **Winter 2019**

Lecture 5: January 22nd, 2019

Lecturer: Anima Anandkumar *Scribes: Akshay Vegesna, Kordag Mehmet Kilic*

Spectral methods are commonly used in ML applications. In this course, we will mostly deal with matrices and tensors. This lecture focuses on matrices.

5.1 Matrices

Suppose we have a matrix M. How do we express this matrix? We can think about this matrix M in terms of standard basis vectors: $\vec{e_1}, \vec{e_2}, \ldots, \vec{e_n}$. The vector $\vec{e_i} = [0 \ldots 0 \;\; 1 \;\; 0 \ldots 0]^T$ where the ith entry is 1 and other entries are 0.

Further, we can think of inner product as extracting information about the matrix. In particular, $M(i,j) = e_i^T M e_j$. The transformation of basis is the core of spectral analysis. By finding different bases, one can solve a problem in a simpler fashion.

Let's now consider a symmetric matrix $M \in \mathcal{S}^{n \times n}$. We can now ask what the most efficient basis is for M.

$$M = \sum_{i=1}^{n} \lambda_i u_i u_i^T \tag{5.1}$$

This leads to the notion of the *eigendecomposition* of a symmetric matrix. Since every symmetric matrix is similar to diagonal matrix, this decomposition always exists. Note that this expression above is preferable to the more straightforward representation $M(i,j) = \sum_{i,j=1}^{n} M(i,j) e_i e_j^T$ because we can impose conditions on the former expression to give favorable results. For example a low-rank (to rank r) approximation of M would take the following form:

$$\tilde{M} = \sum_{i=1}^{r} \lambda_i u_i u_i^T \tag{5.2}$$

This low-rank assumption has several applications for filtering in signal processing. In addition, there are n^2 terms in the $M(i,j) = \sum_{i,j=1}^{n} M(i,j) e_i e_j^T$ whereas it is reduced to n terms in Eq. 5.1, therefore it is a more compact form.

5.2 Variational Perspective

In this section, we study the optimization persepctive of eigenvalues and eigenvectors. The eigenvalues and eigenvectors are the solution to the following constrained optimization problem:

$$\min_{||u||_2 = 1} ||M - \lambda u u^T||_F \tag{5.3}$$

In the Eq.5.3, F denotes the *Frobenius norm.* For a m x n matrix, it is given by the following formula:

$$||A||_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|A(i,j)|^2} \tag{5.4}$$

If we restrict our attention to the maximum eigenvector, we have that this is the solution to the following optimization problem, which was justified in detail.

$$\max_{||u||_2=1} u^T M u \tag{5.5}$$

Note that this problem is a non-convex optimization problem because of the constraint since it is not a convex set (convex combination of them is not necessarily on the unit sphere even if it is inside of it). We convert it to an unconstrained optimization problem using the method of Lagrange multipliers, giving the following expression:

$$\max(u^T M u - \lambda(u^T u - 1)) \tag{5.6}$$

We can solve this problem by just looking for the first order and second-order conditions. In particular, the first-order solution, obtained by differentiating the argument of the maximization problem and setting it to zero, is given by:

$$Mu = \lambda u$$

This is the familiar eigenvector solution, and gives that the stationary points are given by the eigenvectors of the matrix M.

Assume that the eigenvalues are distinct. The second order condition for unconstrained maximization problems is given as follows:

$$\nabla^2 f(x^*) \prec 0 \tag{5.7}$$

Recall that $\prec$ denotes that we have strict negative definiteness. Hence, we have the following condition:

$$v^T \nabla^2 f(x^*) v < 0 \qquad \forall v \perp u \tag{5.8}$$

However, due to the constraint, we restrict the space of v to the tangent space of the constraints. Hence, the above equation does not hold for any selection of v.

Let us find the second derivative of Lagrangian. We had $Mu = \lambda u$. Then, write this as $(M - \lambda I)u = 0$. Taking the derivative with respect to u, we get the following Second-order condition for this maximization problem:

$$v^T (M - \lambda I) v < 0 \quad \forall v \perp u$$

Previously, we had the stationary points as the eigenvectors: $u_1^*, u_2^*, \ldots, u_n^*$ and the corresponding eigenvalues $\lambda_1^*, \lambda_2^*, \ldots, \lambda_n^*$. Assume that $\lambda_1^* > \lambda_2^* > \cdots > \lambda_n^*$ (recall that we assumed they are distinct). Take $u = u_1^*$ and $\lambda = \lambda_1^*$. Then,

$$\begin{aligned} v^T (M - \lambda_1^* I) v, &\qquad \forall v \perp u_1^* \\ => v^T M v - \lambda_1^* v^T v, &\qquad \forall v \perp u_1^* \end{aligned}$$

Without loss of generality, we can take $||v|| = 1$ since both M and λ_1^* are scaled by the same amount.

$$v^T M v - \lambda_1^* < 0 \qquad \forall v \perp u_1^*$$

since λ_1^* is the highest eigenvalue. Now, take $u = u_2^*$ and $\lambda = \lambda_2^*$. Then,

$$v^T M v - \lambda_2^* < 0 \qquad \forall v \perp u_2^*$$

will not hold since $u_1^* \in v \perp u_2^*$ and $\lambda_2^* < {u_1^*}^T M u_1^* = \lambda_1^*$ gives contradiction.

The only vector v satisfying this above equation is the eigenvector corresponding to the largest eigenvalue, an elegant solution to the constrained optimization problem. The other stationary points (other eigenvectors) are saddle points. If we initialize v on $\perp v_1^*$, we will not be able to escape saddle points using classical numerical methods.

5.2.1 Power Method

If we want to compute highest eigenvalue, power method can be used as a fast and reliable method. Basically, one should pick arbitrary vector of norm 1 and iterate the following:

$$u \leftarrow \frac{Mu}{||Mu||} \tag{5.9}$$

until convergence. This formula is a special case of gradient ascend with infinite learning rate.

5.2.2 Singular Value Decomposition (SVD)

One can observe that SVD is an extension of eigendecomposition to any size of matrices. Consider a matrix M with rank r.

$$M = \sum_{i=1}^{r} \sigma_i u_i v_i^T \tag{5.10}$$

In this case, finding the singular values can be given by the following optimization formula in bilinear form:

$$\max_{||u||_2=1, ||v||_2=1} u^T M v \tag{5.11}$$

5.3 Spectral Methods

5.3.1 Rayleigh Quotient

For matrix $S \in \mathbb{R}^{d \times d}$, assume that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$ with corresponding eigenvectors $u_1, \ldots, u_d$.

Rayleigh Quotient is defined as the following.

$$R(S, x) = \frac{x^T S x}{x^T x} \tag{5.12}$$

Maximizing or minimizing the Rayleigh quotient gives the maximum and minimum eigenvalues, respectively. This is done in the previous section in detail, by setting $||x|| = 1$ without loss of generality.

$$\max_{||z||=1} z^T S z = \lambda_1$$
$$\min_{||z||=1} z^T S z = \lambda_d$$

5.3.2 Optimal Projection

Assume that $S \in \mathbb{R}^{d \times d}$ with d eigenvalues.

$$\max_{P:P^2=I, Rank(P)=k} Tr(P^T S P) = \lambda_1 + ... + \lambda_k \text{ and } P \text{ spans } \{u_1, \ldots, u_k\} \tag{5.13}$$

5.3.3 Principal Component Analysis (PCA)

The main idea behind PCA is for centered points $x_i \in \mathbb{R}^d$ to find a projection P satisfying rank$(P) = k$ and

$$\min_{P \in \mathbb{R}^{d \times d}} \sum_{i \in [n]} ||x_i - Px_i||^2 \tag{5.14}$$

If we have $S = \text{Cov}(X)$ and $S = U \Lambda U^T$ is the resulting eigendecomposition, then one has $P = U_{(k)} U_{(k)}^T$ where $U_{(k)}$ are the top-k eigenvectors.

The formula for PCA is an optimal projection problem. To see this, one can apply Pythagorean theorem to vector norms since subspaces are orthogonal in Eq. 5.14.

$$\sum_i ||x_i - Px_i||^2 = \sum_i ||x_i||^2 - \sum_i ||Px_i||^2 \tag{5.15}$$

Then, we need to minimize this equation. Since we are in control of P only, the first term is not important for optimization. Due to negative sign, it is equivalent to maximization.

$$\max \frac{1}{n} \sum_i ||Px_i||^2 = \frac{1}{n} \sum_i Tr[P x_i x_i^T P^T] = Tr[PSP^T] \tag{5.16}$$

which gives optimal projection equation in Eq.5.13.

5.3.4 PCA on Gaussian Mixtures

Assume that we have k Gaussians.

- Let the data is obtained by $x = Ah + z$.
- We have $A \in \mathbb{R}^{d \times k}$ where columns are component means.

- Here, $h \in \{e_1, \ldots, e_k\}$ are basis vectors, they are called as *latent* variables because they are not observed. Here h selects a column from A, by Ah.
- Let $\mathbb{E}[h] = \omega$.
- $z \sim \mathcal{N}(0, \sigma^2 I)$ and z is uncorrelated with h.
- Let $\mu = A\omega$.

Then, the centralized covariance matrix of x will be the following.

$$\mathbb{E}[(x-\mu)(x-\mu)^T] = \sum_{i \in [k]} \omega_i (a_i - \mu)(a_i - \mu)^T + \sigma^2 I \tag{5.17}$$

The proof is given below.

$$\begin{aligned}
\mathbb{E}[xx^T] &= \mathbb{E}[(Ah+z)(Ah+z)^T] = A\mathbb{E}[hh^T]A^T + A\mathbb{E}[hz^T] + \mathbb{E}[hz]A^T + \mathbb{E}[zz^T] \\
&= A\mathbb{E}_[hh^T]A^T + \sigma^2 I \text{ by uncorrelatedness of h and z.} \\
\mathbb{E}[hh^T] &= \sum_i^n \omega_i e_i e_i^T \\
&= \mathcal{D}(\omega) \\
\mathbb{E}[xx^T] &= A\mathcal{D}(\omega)A^T + \sigma^2 I
\end{aligned}$$

We have the $Span(A)$ by Eq.5.17 and applying $(k-1)-PCA$ on $\mathbb{E}[xx^T]$ and taking union with μ. How can we learn A from $A\mathcal{D}(\omega)A^T$ or just $Span(A)$?

1. If A is orthogonal and $\omega \neq 0$, A is unique. Hence, we can get an eigenspace from $Span(A)$. In case $k << d$, low-rank approximation is useful.
2. To learn A through clustering, first project x to $Span(A)$ and then use distance-based clustering methods, such as *k-means*.

5.3.5 Canonical Correlation Analysis (CCA)

Suppose we have two collection of data points of the same number, $\{x_i\}_{i=1}^N$ and $\{y_i\}_{i=1}^N$. If we try to measure the correlation between them, we can calculate the correlation matrix $\mathbb{E}[XY^T]$.

However, there is a downside of this. Scales of data will be reflected to $\mathbb{E}[XY^T]$. That is to say, if X is of order 10^3 and Y is of order 0.01, then the correlations will be calculated wrongly.

Therefore, we first whiten each data set and then take the correlation between them.

We need to use *whitening transform.*

$$z_i = W^T x_i \tag{5.18}$$

$$\mathbb{E}[zz^T] = W^T Cov(X) W = I \tag{5.19}$$

By this transform, we need to calculate $X' = W_X^T X$ and $Y' = W_Y^T Y$. Then, we can calculate the correlation between X' and Y' to obtain the correct result.