U.C. Berkeley — CS270: Algorithms
Professor Satish Rao

Lecture 18
Scribe: Anupam
Last revised

Lecture 18

1 Principal Components Analysis

The lecture will be in two parts, first we will discuss the singular value decomposition and low rank approximations for matrices, then we will discuss an application of spectral methods (PCA) to the Gaussian clustering problem.

2 SVD and least squares

2.1 The singular value decomposition

A matrix $A \in \mathbb{R}^{m\times n}$ defines a map $A : \mathbb{R}^n \rightarrow \mathbb{R}^m$ via left multiplication. The singular value decomposition of A is a factorization $A = VDU$ where U is an $n \times n$ orthogonal matrix, V is an $m \times m$ orthogonal matrix and D is an $m \times n$ diagonal matrix with positive entries. The map A is a composition of a rotation on $\mathbb{R}^n$, scaling by the singular values to obtain vectors in $\mathbb{R}^m$ followed by a rotation on $\mathbb{R}^m$. If we choose suitable bases for $\mathbb{R}^n$ and $\mathbb{R}^m$ then A is described by a diagonal matrix.

Denote the orthonormal bases for $\mathbb{R}^n, \mathbb{R}^m$ given by matrices U, V by $u_1, u_2, \cdots, u_n$ and $v_1, v_2, \cdots, v_m$. A convenient way to write the singular value decomposition for A is the following,

$$A = \sum_i \sigma_i v_i u_i^t \tag{1}$$

It can be verified that the above expression represents A as $Au_i = \sigma_i v_i$ by the singular value decomposition. For symmetric matrices, the matrices U and V are identical and SVD reduces to the spectral decomposition,

$$A = \sum_i \lambda_i v_i v_i^t \tag{2}$$

The spectral decomposition of the matrices AA^T and A^TA can be expressed in terms of the singular value decomposition of A,

$$\begin{aligned} AA^T &= \sum_i \sigma_i^2 v_i v_i^t \\ A^TA &= \sum_i \sigma_i^2 u_i u_i^t \end{aligned} \tag{3}$$

The singular vectors can therefore be computed by finding the eigenvectors of A^tA and AA^t respectively.

An intuitive proof of the SVD can be found in Gilbert Strang's paper on 'The fundamental theorem of linear algebra' [**?**].

2.2 Least squares

Consider the following algorithmic problem: we are given m data points $A = \{a_1, a_2, \cdots, a_m\}$ in $\mathbb{R}^n$ and wish to find a k dimensional subspace S such that the squared distance $\sum_{i \in [m]} d(a_i, S)^2$ is minimized.

One dimensional line fitting: Let us first consider the one dimensional problem of finding the line through the origin that minimizes the least squares error with respect to the data set. The data points $a_i \in \mathbb{R}^n$ are taken to be the rows of an $m \times n$ matrix A and the line is specified by the unit vector $u \in \mathbb{R}^n$.

$$\sum_{i \in [m]} |A|^2 = \sum_{i \in [m]} d(a_i, l)^2 + (a_i.u)^2 \tag{4}$$

Minimizing the least squares error is equivalent to finding the vector u maximizing $\sum_{i \in [m]} (a_i.u)^2 = |Au|^2 = u^T A^T A u$. The solution is the largest eigenvector of $A^T A$ which is equal to the singular vector u_1 by equation (3).

k dimensions: The k dimensional subspace minimizing the least squares error is the span of the k largest singular vectors. We prove this by induction, having proved the base case $k = 1$ above. Let V_k denote the k dimensional subspace minimizing the least squares error.

Select an orthonormal basis $w_1, w_2, \cdots, w_k$ for V_k such that w_k is orthogonal to V_{k-1}. Analogous to equation (4) we have,

$$\sum_{i \in [m]} |A|^2 = \sum_{i \in [m]} d(a_i, V_k)^2 + \sum_{i \in [m], j \in [k]} (a_i.w_j)^2 \tag{5}$$

The problem of finding the best k dimensional subspace is equivalent to maximizing,

$$\begin{aligned} \sum_{i \in [k]} w_i^T A^T A w &= \sum_{i \in [k-1]} w_i^T A^T A w + \max_{w \perp V_{k-1}} w^T A^T A w \\ &= \sum_{i \in [k-1]} \sigma_i^2 + \max_{w \perp V_{k-1}} w^T A^T A w \\ &= \sum_{i \in [k]} \sigma_i^2 \end{aligned} \tag{6}$$

In the second step we used the induction hypothesis and for the final step we used the fact that the maximum norm of $w^T A^T A w$ over the space orthogonal to the span of the first $k-1$ eigenvectors of $A^T A$ is $\lambda_k = \sigma_k^2$.

2.2.1 Gaussian clustering

Sequencing technology processes genomes to identify single nucleotide polymorphisms (SNPs) that account for most of the differences between individuals. The number of SNPs identified is proportional to the length $|x_i|$ of the processed sequence. The SNPs can be modeled as features $f_i : x_i \to \mathbb{R}$, the SNPs are far on the genome so we assume that the parameters f_i are mutually independent. The processing cost is proportional to the number of parameters k.

Suppose we have genetic data for a large number of people representing two distinct populations and the problem is to classify people into populations. The feature parameters f_i and f'_i for people from the two populations are distributed according to normal distributions with mean and variance (μ_i, σ_i) and (μ'_i, σ'_i) respectively.

The normal distribution: The reason that the normal distribution models population characteristics such as the heights of people or the lengths of tails of horses, is the central limit theorem which says that the sum of a large number of independent random variables tends to the normal distribution. The density function for the normal distribution with mean and variance (μ, σ) is $\phi(x) = \frac{1}{\sigma} e^{-\frac{\pi x^2}{\sigma^2}}$. The normal distribution is strongly concentrated around the mean,

$$\Pr_{x \sim N(\mu,\sigma)}[|x - \mu| \geq t\sigma] \leq e^{-t^2/2} \tag{7}$$

Example: For the purpose of illustrating the classification problem, we assume that the variances σ_i, σ'_i are all equal to σ. The measure of difference between two people represented by $x, y \in \mathbb{R}^d$ is the distance in the feature space $d(x, y) = \sum_i (f_i(x) - f_i(y))^2$. We next compute the expected distances between people belonging to the different populations.

Variables x_1, x_2 represent people from population 1 while y_1, y_2 are people from population 2. Using the concentration bounds (7) for the normal distribution,

$$\begin{aligned} Pr[|f_i(x_1) - f_i(x_2)| \geq O(\sqrt{\log k})\sigma] &\leq \frac{1}{\text{poly}(k)} \\ Pr[d(x_1, x_2)^2 \geq O(k \log k\sigma^2)] &\leq \frac{1}{\text{poly}(k)} \end{aligned} \tag{8}$$

The second inequality follows from the union bound, the expected distance between two people from the same population is $O(\sqrt{k \log k}\sigma)$. A similar bound holds for people y_1, y_2 drawn from the population 2.

Let μ_i and μ'_i be the mean values of f_i over the two populations, the distance between people from different populations can be bounded using the triangle inequality,

$$d(x_1, y_1) \leq d(x_1, \mu) + d(\mu, \mu') + d(y_1, \mu') \tag{9}$$

From the concentration bounds we know that $d(y_1, \mu')$ and $d(x_1, \mu)$ are both $O(\sqrt{k \log k}\sigma)$. The distances between points from the two different clusters can be expressed as,

$$\begin{aligned} d(x_1, x_2) &= d(y_1, y_2) = \widetilde{O}(\sqrt{k}\sigma) \\ d(x_1, y_2) &= d(\mu, \mu') + \widetilde{O}(\sqrt{k}\sigma) \end{aligned} \tag{10}$$

It is possible to separate the clusters if the difference between the means $d(\mu, \mu')$ is greater than the variance $\widetilde{O}(\sqrt{k}\sigma)$. Suppose there is an algorithm that classifies people according to the clusters if the following geometric separation condition holds, $d(\mu_i, \mu'_i) \geq \beta\sigma$. For example the threshold clustering algorithm that finds pairwise distances between points and outputs all points at a distance of at most t as one cluster, works for $\beta \geq \sqrt{k}$.

The idea is to project the input data onto a lower dimensional subspace so that the clustering problem is easier on the low dimensional data. We are looking for a projection that preserves the distance $d(\mu, \mu')$ while decreasing the variance of the data. A random projection onto an l dimensional subspace would compress both $d(\mu, \mu')$ and σ by a factor of $\sqrt{k}l$ as discussed in the previous lecture, this leaves the ratio between the variance and the distance between clusters unchanged and offers no advantage.

The means μ, μ' are unknown else we could project onto the line spanned by μ and μ'. The projection preserves $d(\mu, \mu')$ and reduces the variance σ by a factor of $\sqrt{k}$.

If the data consists of t Gaussian clusters, the span of the means of the clusters is the subspace spanned by the largest t singular vectors of the $n \times k$ matrix X. Computing the largest singular vectors and projecting the data onto this subspace effectively reduces the dimension of Gaussian clustering problem.