CS 270 Algorithms **Spring 2021**

Lecture 6: Applications of PCA

Lecturer: Prasad Raghavendra Scribe: Cathy Chen, Chase Norman, Aaron (Louie) Putterman

6.1 Overview of SVD

Suppose we are given a matrix $A \in \mathbb{R}^{nxd}$. If we look at this matrix as a collection of n d-dimensional points, we have already described the problem of finding the "best" K-dimensional subspace for projecting this set, where "best" is defined to be the subspace which maximizes the variance of the projected points. Here, we continue this train of thought, and instead want the best rank-K matrix which approximates the matrix A. To this end, we introduce the SVD.

Consider again the matrix $A \in \mathbb{R}^{nxd}$. The singular value decomposition finds matrices $U \in \mathbb{R}^{nxn}$, $\Sigma \in \mathbb{R}^{nxd}$, and $V \in \mathbb{R}^{dxd}$, such that $A = U\Sigma V^T$. Furthermore, we require that U and V are orthogonal matrices, meaning $U^TU = I$ and $V^TV = I$, and that Σ is a diagonal matrix with non-negative entries.

Properties: 1) Since Σ is a diagonal matrix, but is not necessarily square, it can have at most $min(n, d)$ positive entries. Additionally, the rank of Σ will be the number of positive entries which it contains, and since U and V are orthogonal matrices, $A = U\Sigma V^T$ will have the same rank as Σ.

2) Denote by d the number of non-zero singular values of Σ. Then, by the properties of matrix multiplication, $A = U\Sigma V^T = \sum_{i=1}^{d} \sigma_i u_i v_i^T$.

3) The SVD can be computed efficiently, and even inductively in the decreasing order of singular values. This ability to top-k query the SVD is very important.

Claim: Consider the problem of finding the best rank-K approximation to a matrix $A \in \mathbb{R}^{nxd}$ under the l2-operator norm. Namely, $min_{M:rank(M)\leq k}\|A - M\|_2$. We claim that the optimal matrix $M = \sum_{i=1}^{k} \sigma_i u_i v_i^T = A_k$.

Proof: Assume that $k \leq min(n, d) - 1$, as otherwise we can trivially take the entire matrix A. For simplicity, assume $d \leq n$ (otherwise the same proof can be done with A^T). Now, consider the case when k = 0. Then, there is only one choice for our matrix M, which is the zero-matrix and is equivalent to A_0. From now on, we assume $0 \leq k \leq d - 1$.

Now, we observe that $A - A_k = A - \sum_{i=1}^{k} \sigma_i u_i v_i^T = \sum_{i=1}^{d} \sigma_i u_i v_i^T - \sum_{i=1}^{k} \sigma_i u_i v_i^T = \sum_{i=k+1}^{d} \sigma_i u_i v_i^T$. Clearly, under the l2-operator norm, $\| \sum_{i=k+1}^{d} \sigma_i u_i v_i^T \|_2 = \sigma_{k+1}$. Next, we prove that for any k-dimensional matrix M, $\|A - M\|_2 \geq \sigma_{k+1}$. Indeed, since M is at most rank K, the null-space of M is of dimension at least n - k (by the rank-nullity theorem). If we consider now the subspace V defined as the first k+1 right singular vectors of A (ie $\{v_i\}_{i=1}^{k+1}$), then, since $dim(N(M)) \geq n - k$ and $dim(V) \geq k + 1$, they must have a non-trivial intersection. Hence, there exists some vector $w \in V \cap N(M)$, such that $\|w\|_2 = 1$. Since the right

singular vectors of V are normalized, $w = \sum_{i=1}^{k+1} c_i v_i$, where $\sum_{i=1}^{k+1} c_i^2 = 1$. Bringing this together, we get that $\|(A-M)w\|_2^2 = \|Aw\|_2^2 = \|U\Sigma V^T w\|_2^2 = \sum_{i=1}^{k+1} c_i^2 \sigma_i^2 \geq \sigma_{k+1}^2$.

Hence, because A_k attains the minimum value, it is the best approximation.

Remark: The SVD will also reveal the best rank-k approximation under the Frobenius norm. The proof will follow a similar pattern to the operator norm proof.

6.2 Application of SVD: Learning mixtures of Gaussians

Presume we have a random variable that is the sum of Gaussians:

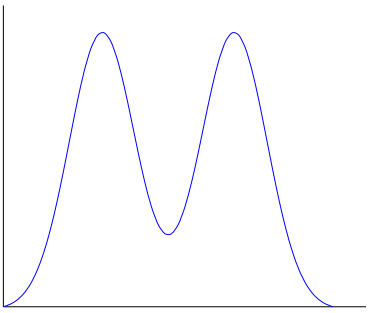
$$\theta = \frac{1}{2}\mathcal{N}(\mu_1, Id) + \frac{1}{2}\mathcal{N}(\mu_2, \mathrm{Id})$$

Where $\mathcal{N}(\mu, \mathrm{Id})$ is the standard Gaussian distribution with PDF $\phi(x) \propto e^{-\frac{\|x-\mu\|^2}{2}}$

Problem statement: Given samples from θ, find μ_1 and μ_2.

The nontrivial part of this problem is to identify which Gaussian each sample comes from. Given this information, μ_1 and μ_2 could be approximated directly.

In the one dimensional case, θ may look like the following:



The key observation is that for constant distance $\Delta = |\mu_1 - \mu_2|$, this problem can be directly solved. At worst, we can use a brute force algorithm to find optimal positions of μ_1 and μ_2 to fit the data.

6.2.1 Naive approach: Cluster Technique

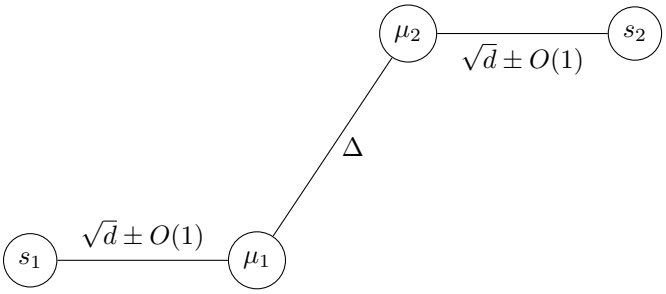
Let s_1 and s_2 be samples from the μ_1 and μ_2 mean distributions, respectively. We will operate in d dimensions.

Note that each component of s_1 and s_2 are independent. Since the covariance of the Gaussians are Identity,

the standard deviation is 1. Therefore we have:

$$\mathbf{E}[\|s_i - \mu_i\|] = \mathbf{E}\left[\sqrt{\sum_{j=1}^{d}(s_{ij} - \mu_{ij})^2}\right] \leq \sqrt{\sum_{j=1}^{d}\mathbf{E}[(s_{ij} - \mu_{ij})^2]} = \sqrt{\sum_{j=1}^{d} 1} = \sqrt{d}$$

With high probability, the distance from s_1 and s_2 to μ_1 and μ_2 is $\sqrt{d} \pm O(1)$. In essence, the deviation from $\sqrt{d}$ does not scale with dimension.



The key intuition for analyzing this is that **in high dimensions, any 2 vectors are orthogonal with high probability.** In general, we expect the dot product of two normalized vectors to have magnitude $O(\frac{1}{\sqrt{d}})$. As $d \to \infty$, this approximation becomes more accurate.

With this constraint, we expect:

$$\|s_1 - s_2\|^2 = \|s_1 - \mu_1 + \mu_1 - \mu_2 + \mu_2 - s_2\|^2 \approx \|s_1 - \mu_1\|^2 + \|\mu_1 - \mu_2\|^2 + \|\mu_2 - s_2\|^2 \approx (\sqrt{d})^2 + \Delta^2 + (\sqrt{d})^2 = 2d + \Delta^2$$

For any two samples s and s' from the same Gaussian with mean μ, we expect:

$$\|s - s'\|^2 = \|s - \mu + \mu - s'\|^2 \approx \|s - \mu\|^2 + \|\mu - s'\|^2 \approx (\sqrt{d})^2 + (\sqrt{d})^2 = 2d$$

Both of these are expected to vary by some amount $O(\sqrt{d})$. In order to reasonably distinguish the differences between the "intracluster" distances of $2d \pm \sqrt{d}$ and "intercluster" distances of $2d + \Delta^2 \pm \sqrt{d}$, we must have $\Delta = \Omega(d^{\frac{1}{4}})$ to overwhelm the random deviation of $O(\sqrt{d})$.

Recall that in the 1D case, $\Delta = O(1)$ could be solved. Since increasing dimension creates a constraint on Δ, we are unsatisfied with this approach. The cluster technique fails in higher dimensions.

6.2.2 Applying PCA to the problem

As before, let:

$$\theta = \frac{1}{2}\mathcal{N}(\mu_1, \mathrm{Id}) + \frac{1}{2}\mathcal{N}(\mu_2, \mathrm{Id})$$

Key: We will attempt to find a 2-dimensional subspace Π such that:

$$\Pi \leftarrow \max_{\Pi} \mathop{\mathbf{E}}_{x\sim\theta}\left[\|\mathrm{proj}_{\Pi} x\|^2\right] = \max_{\Pi}\left(\frac{1}{2}\mathop{\mathbf{E}}_{x\sim\mathcal{N}(\mu_1,\mathrm{Id})}\left[\|\mathrm{proj}_{\Pi} x\|^2\right] + \frac{1}{2}\mathop{\mathbf{E}}_{x\sim\mathcal{N}(\mu_2,\mathrm{Id})}\left[\|\mathrm{proj}_{\Pi} x\|^2\right]\right)$$

Claim: Π will be $\mathrm{Span}(\mu_1, \mu_2)$.

Proof: let v_1, v_2 be an orthonormal basis of Π. We have:

$$\Pi = \max_{\Pi} \left(\frac{1}{2} \mathop{\mathbf{E}}_{x \sim \mathcal{N}(\mu_1, \mathrm{Id})} \left[\langle v_1, x \rangle^2 + \langle v_2, x \rangle^2 \right] + \frac{1}{2} \mathop{\mathbf{E}}_{x \sim \mathcal{N}(\mu_2, \mathrm{Id})} \left[\langle v_1, x \rangle^2 + \langle v_2, x \rangle^2 \right] \right)$$

$$= \max_{\Pi} \left(\mathop{\mathbf{E}}_{y \sim \mathcal{N}(0, \mathrm{Id})} \left[\langle v_1, y + \mu_1 \rangle^2 + \langle v_2, y + \mu_1 \rangle^2 + \langle v_1, y + \mu_2 \rangle^2 + \langle v_2, y + \mu_2 \rangle^2 \right] \right)$$

By symmetry about 0, this is:

$$= \max_{\Pi} \left(\langle v_1, \mu_1 \rangle^2 + \langle v_2, \mu_1 \rangle^2 + \langle v_1, \mu_2 \rangle^2 + \langle v_2, \mu_2 \rangle^2 \right)$$

$$= \max_{\Pi} \left(\|\mathrm{proj}_{\Pi} \mu_1\|^2 + \|\mathrm{proj}_{\Pi} \mu_2\|^2 \right)$$

Which is maximized when $\Pi = \mathrm{Span}(\mu_1, \mu_2)$ □

By projecting our sample points onto Π, we can reduce the problem to 2 dimensions. The resulting distribution will be approximately the following:

$$\mathrm{proj}_{\Pi} \theta \approx \frac{1}{2} \mathcal{N}(\mu_1, \mathrm{Id}_{\Pi}) + \frac{1}{2} \mathcal{N}(\mu_2, \mathrm{Id}_{\Pi})$$

Since the dimension is constant, we can solve for μ_1 and μ_2 given they have a distance $\|\mu_1 - \mu_2\| \geq C$ for some constant C. Thus we remove our dependence on dimension using PCA.

6.3 Application of SVD: Robust Mean Estimation

One application of SVD is robust mean estimation. Suppose we have $x_1, ..., x_n \sim \mathcal{D}$.

Consider the problem of robust mean estimation: computing the mean of the distribution $\mathcal{D} \approx N(\mu, I_d)$ in the presence of outliers. Specifically, we are given $(1 - \epsilon)n$ samples from $\mathcal{D}$ (called "inliers" or "good" examples), and ϵn samples from an adversary (called "outliers" or "bad" examples). We want to estimate $\mu(\mathcal{D})$.

Observation: Note that if the adversary chooses examples that are very far from μ, then it will be relatively clear which samples are outliers, and we can simply remove these samples from the dataset. However, a more pernicious adversary could choose outliers that are harder to detect, and which shift the distribution by ϵ, from $N(\mu, 1)$ to $N(\mu - \epsilon, 1)$.

6.3.1 Approach 1 (Median)

One approach is to estimate μ by computing $median(X)$. This approach is reasonable if $x_i \in \mathbb{R}$ because in one dimension, $median(X)$ is within $O(\epsilon)$ of μ (We refer the reader to Section 2 of [2] for a proof of this fact). However, for $x_i \in \mathbb{R}^d$ the median in each coordinate is $O(\epsilon)$ from the median, and $||\hat{\mu} - \mu|| \approx \epsilon\sqrt{d}$. Ideally, we are looking for a dimension independent guarantee, so we are unsatisfied with this approach.

An alternative to using the median is to compute the Tukey median. The Tukey median is computed as follows: for a vector v and sample x, $pos_v(x)$ is the position of x in the direction of v and $depth(x) = \min_v pos_v(x)$,

and $tukey\_median(X) = \text{argmax}_x\, depth(x)$. However, there are no known efficient algorithms to find the Tukey median (computing the Tukey median is NP-hard).

As an intuition for why the Tukey median is preferred, notice that the geometric median depends on your choice of basis of X, but the Tukey median does not. This ad-hoc choice of basis is likely to cause inefficiencies.

6.3.2 Approach 2 (Filtering)

A better approach to robust mean estimation is to use information from the covariance matrix, which is $\Sigma_{\mathcal{X}} = cov[\mathcal{X}] = \mathbb{E}_x[(x-\mu)(x-\mu)^\top] \in \mathbb{R}^{n\times n}$. This matrix is PSD with eigenvalues that denote the length of axes of an ellipsoid which approximates the distribution.

To understand the intuition of this approach, first consider the one-dimensional case. If the adversary contributes ϵ-fraction of the samples to the dataset, then in order to shift the sample mean by Δ, the adversary must add points $\frac{\Delta}{\epsilon}$ away from μ. These points will add $\epsilon(\frac{\Delta}{\epsilon})^2 = \frac{\Delta^2}{\epsilon}$ to the variance. In higher dimensions, the adversary would similarly need to add ϵ-fraction of points at a distance $\frac{\Delta}{\epsilon}$ from μ in a certain direction in order to move μ by Δ in that direction. The variance around that direction will therefore be high (relative to the variance of 1 that we expect, since $\mathcal{D} \approx N(\mu, I_d)$), and the covariance matrix will have a large eigenvalue in the perturbed direction.

Therefore, we can use a filtering approach to estimate μ. First, we detect a direction in which the covariance matrix Σ_X has a large (relative to 1, since $\mathcal{D} \approx N(\mu, I_d)$) eigenvalue. Then we remove samples until the covariance matrix is spherical (the removed samples are more likely to be outliers than inliers). Finally, we estimate μ using the remaining samples. We refer the reader to Section 2.4 of [1] for additional details about this approach.

6.4 Median of Means

Many real world systems follow a power-law distribution ($\phi \propto x^{-c}$) rather than Gaussian. In these situations, averaging samples no longer provides an exponential convergence to the mean. The median of means algorithm is a more robust approach to finding the mean of both of these distributions.

Algorithm:

1) Split the dataset D into k random buckets

2) Let $\mu_i = mean(bucket_i)$

3) Let $\mu^* = median(\mu_1, ...\mu_k)$

Claim: To achieve a confidence of $1-\delta$ and within an accuracy ϵ, we require n = $O(\frac{log(1/\delta)}{\epsilon^2})$, meaning $K \approx log(1/\delta)$. In d-dimensions, Lugosi and Mendelson [5] proposed an estimator for the mean with a sub-Gaussian rate (up to constant factors), and Hopkins [4] showed how to compute it efficiently. More recently, Valiant [3] showed that the 1-dimensional mean can be computed with the optimal $n = 2 * \frac{log(1/\delta)}{\epsilon^2}$

samples.

Remarks: The approach of 1-dimensional median of means is particularly useful in streaming algorithms, as the median of means can be continuously updated.

Median of means as described is no longer as effective in higher dimensions. The Tukey median is required, making the algorithm intractable.

Open problem: Is this constant factor the same as in the Gaussian case in d-dimensions?

References

[1] I. Diakonikolas, D. Kane. Recent Advances in Algorithmic High-Dimensional Robust Statistics. Invited Book Chapter in *Beyond the Worst-Case Analysis of Algorithms*, Cambridge University Press, September 2020.

[2] Lai, Kevin A., Anup B. Rao, and Santosh Vempala. Agnostic Estimation of Mean and Covariance. 2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS). IEEE, 2016.

[3] Jasper C. H. Lee and Paul Valiant. Optimal Sub-Gaussian Mean Estimation in R, arxiv, Nov 2020.

[4] Samuel Hopkins. Mean Estimation with Sub-Gaussian Rates in Polynomial Time, 2019.

[5] Gábor Lugosi and Shahar Mendelson. Sub-Gaussian estimators of the mean of a random vector, 2017.