1. SUMMARY OF PCA

Let Z be an $m \times n$ matrix. Each column $\mathbb{Z}_j$ is a sample from a multivariate normal distribution on $\mathbb{R}^n$.

(1) Compute means:

$$\mu = \frac{1}{n} Z * \text{ones}(n, 1)$$

(2) Subtract means:

$$X = Z - \mu * \text{ones}(1, n)$$

(3) Compute SVD

$$X = UDV^*$$

(4) Pick truncation rank

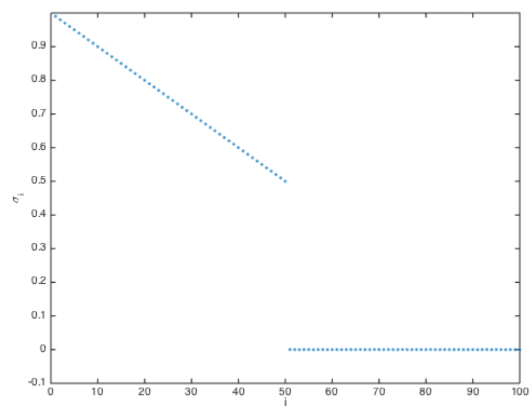

FIGURE 1. Lucky

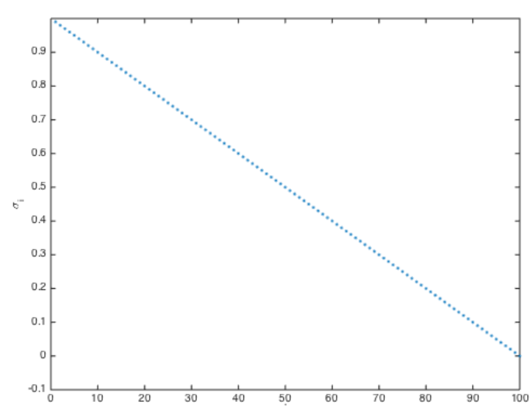

FIGURE 2. Typical

(5) Truncate $X \approx UDV^*$ $\{u_j\}_{j=1}^k$ are principle directions.
Set $Y = U^*X$, then

$$x_j \approx y_{1j}u_1 + y_{2j}u2 + \cdots + y_{kj}u_k$$

PCA crucially relies on the assumption that the data is drawn from a distribution that at least approximately multivariate normal.

2. LEVERAGE SCORES FOR FINDING SPANNING COLUMNS

Let A be $m \times n$ of approximate rank k. We seek to determine a set J_s of spanning columns such that

$$A \approx CX$$

where $C = A(:, J_s)$. We want elements of X to be small.

Options for finding J_s:

(1) draw k indices from $\{1, 2, \cdots, n\}$ at random, with equal probability weights. It turns out that this is bad.

Example:

$$A = \begin{bmatrix} I_k & 0 \\ 0 & 0 \end{bmatrix}$$

(2) Draw random rotation matrix Q of size $n \times n$(so $Q^*Q = I$). Set $\tilde{A} = AQ$. Drawing k columns at random from $\tilde{A}$ will work well. For instance, set $\tilde{J}_s = [1, 2, \cdots, k]$,

$$\tilde{A}(:, 1:k) = AQ(:, 1:k)$$

A Gaussian random matrix of size $n \times k$ does almost the same sampling.

Note 1. We only pick basis for col(A), we do not identify column of A.

Note 2. Slight oversampling is necessary.

Note 3. Recall SRFT

$$Y = ADFS$$

where D is a random diagonal matrix, F is discrete Fourier transform matrix, S is subsampling matrix and $Q = DF$.

Q is random rotations but it is not drawn uniformly from the set of all random rotations.

(3) Compute column norms of A and use these to compute sampling probabilities. Set

$$c_j = \|A(:, j)\|^2 \quad \text{for} \quad j = 1, 2, \cdots, n$$

Set

$$p_j = \frac{c_j}{c_1 + c_2 + \cdots + c_n} = \frac{c_j}{\|A\|_F}$$

Example:

$$A = \begin{bmatrix} I_k & 0 \\ 0 & 0 \end{bmatrix}$$

$$c_1 = c_2 = \cdots = c_k = 1, c_{k+1} = c_{k+2} = \cdots = c_n = 0$$

$$p_1 = p_2 = \cdots = p_k = \frac{1}{k}, p_{k+1} = p_{k+2} = \cdots = p_n = 0$$

You will draw $J_s = [1, 2, \cdots, k]$ with probability 1.

Example:

$$A = \begin{bmatrix} 10 & 20 & 0 \\ 10 & 20 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

So rank(A)=2 and

$$c_1 = 200, c_2 = 800, c_3 = 1$$

$$p_1 = \frac{200}{1001}, p_2 = \frac{800}{1001}, p_3 \approx 0$$

Very likely you will pick $J_s = [1, 2]$.

(4) Compute full SVD, then truncate to the top k dominant terms:

$$A \approx UDV^*$$

Let p_j be the square norm of jth column of V^*, then

$$p_j = \frac{1}{k}\sum_{i=1}^{k} |V(j,i)|^2$$

Observe:

$$\sum_{j=1}^{n} p_j = \sum_{j=1}^{n} \frac{1}{k}\sum_{i=1}^{k} |V(j,i)|^2 = \frac{1}{k}\sum_{i=1}^{k}\sum_{j=1}^{n} |V(j,i)|^2 = 1$$

One can prove that if J_s is drawn from $\{1, 2, \cdots, n\}$ without replacement, and with sampling probabilities $\{p_1, p_2, \cdots, p_n\}$, then the chosen J_s is with high probability good.

Issue 1. Compute SVD is expensive.

Issue 2. In practice, you may as well perform the second step deterministically. Simply execute CPQR on the matrix DV^*.