0368-3248-01-Algorithms in Data Mining **Fall 2012**

Lecture 6: SVD and PCA

Lecturer: Edo Liberty

Warning: *This note may contain typos and other inaccuracies which are usually discussed during class. Please do not cite this note as a reliable source. If you find mistakes, please inform me.*

---

1 Singular Value Decomposition (SVD)

We will see that any matrix $A \in \mathbb{R}^{m\times n}$ (w.l.o.g. $m \le n$) can be written as

$$A = \sum_{\ell=1}^{m} \sigma_\ell u_\ell v_\ell^T \quad (1)$$

$$\forall\ \ell \quad \sigma_\ell \in \mathbb{R},\ \ \sigma_\ell \ge 0 \quad (2)$$

$$\forall\ \ell,\ell' \quad \langle u_\ell, u_{\ell'}\rangle = \langle v_\ell, v_{\ell'}\rangle = \delta(\ell,\ell') \quad (3)$$

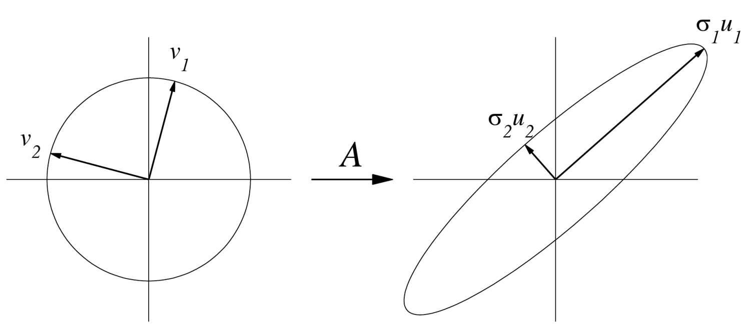
To prove this consider the matrix $AA^T \in \mathbb{R}^{m\times m}$. Set u_ℓ to be the ℓ'th eigenvector of AA^T. By definition we have that $AA^T u_\ell = \lambda_\ell u_\ell$. Since AA^T is positive semidefinite we have $\lambda_\ell \ge 0$. Since AA^T is symmetric we have that $\forall\ \ell,\ell'$ $\langle u_\ell, u_{\ell'}\rangle = \delta(\ell,\ell')$. Set $\sigma_\ell = \sqrt{\lambda_\ell}$ and $v_\ell = \frac{1}{\sigma_\ell}A^T u_\ell$. Now we can compute the following:

$$\langle v_\ell, v_{\ell'}\rangle = \frac{1}{\sigma_\ell^2} u_\ell^T A A^T u_\ell = \frac{1}{\sigma_\ell^2}\lambda_\ell \langle u_\ell, u_{\ell'}\rangle = \delta(\ell,\ell')$$

We are only left to show that $A = \sum_{\ell=1}^{m} \sigma_\ell u_\ell v_\ell^T$. To do that we examine the norm or the difference multiplied by a test vector $w = \sum_{i=1}^{m} \alpha_i u_i$.

$$\begin{aligned}
||w^T(A - \sum_{\ell=1}^{m} \sigma_\ell u_\ell v_\ell^T)|| &= ||(\sum_{i=1}^{m} \alpha_i u_i^T)(A - \sum_{\ell=1}^{m} \sigma_\ell u_\ell v_\ell^T)|| \\
&= ||(\sum_{i=1}^{m} \alpha_i u_i^T A - \sum_{i=1}^{m}\sum_{\ell=1}^{m} \delta(i,\ell)\alpha_i \sigma_\ell v_\ell^T|| \\
&= ||(\sum_{i=1}^{m} \alpha_i \sigma_i v_i^T - \sum_{i=1}^{m} \alpha_i \sigma_i v_i^T|| = 0
\end{aligned}$$

The vectors u_ℓ and v_ℓ are called the left and right singular vectors of A and σ_ℓ are the singular vectors of A. It is costumery to order the singular values in descending order $\sigma_1 \ge \sigma_2, \dots, \sigma_m \ge 0$.



2 Rank-k approximation in the spectral norm

The following will claim that the best approximation to A by a rank deficient matrix is obtained by the top singular values and vectors of A. More accurately:

Fact 2.1. *Set*

$$A_k = \sum_{j=1}^{k} \sigma_j u_j v_j^T.$$

Then,

$$\min_{\substack{B \in \mathbb{R}^{m \times n} \\ \operatorname{rank}(B) \leq k}} \|A - B\|_2 = \|A - A_k\|_2 = \sigma_{k+1}.$$

Proof.

$$\|A - A_k\| = \| \sum_{j=1}^{r} \sigma_j u_j v_j^T - \sum_{j=1}^{k} \sigma_j u_j v_j^T \| = \| \sum_{j=k+1}^{r} \sigma_j u_j v_j^T \| = \sigma_{k+1}$$

and thus σ_{k+1} is the largest singular value of $A - A_k$. Alternatively, look at $U^T A_k V = \operatorname{diag}(\sigma_1, \ldots, \sigma_k, 0, \ldots, 0)$, which means that $\operatorname{rank}(A_k) = k$, and that

$$\|A - A_k\|_2 = \|U^T (A - A_k) V\|_2 = \| \operatorname{diag}(0, \ldots, 0, \sigma_{k+1}, \ldots, \sigma_r)\|_2 = \sigma_{k+1}.$$

Let B be an arbitrary matrix with $\operatorname{rank}(B_k) = k$. Then, it has a null space of dimension $n - k$, that is,

$$\operatorname{null}(B) = \operatorname{span}(w_1, \ldots, w_{n-k}).$$

A dimension argument shows that

$$\operatorname{span}(w_1, \ldots, w_{n-k}) \cap \operatorname{span}(v_1, \ldots, v_{k+1}) \neq \{0\}.$$

Let w be a unit vector from the intersection. Since

$$Aw = \sum_{j=1}^{k+1} \sigma_j (v_j^T w) u_j,$$

we have

$$\|A - B\|_2^2 \geq \|(A - B)w\|_2^2 = \|Aw\|_2^2 = \sum_{j=1}^{k+1} \sigma_j^2 \left| v_j^T w \right|^2 \geq \sigma_{k+1}^2 \sum_{j=1}^{k+1} \left| v_j^T w \right|^2 = \sigma_{k+1}^2,$$

since $w \in \operatorname{span}\{v_1, \ldots, v_{n+1}\}$, and the v_j are orthogonal. □

3 Rank-k approximation in the Frobenius norm

The same theorem holds with the Frobenius norm.

Theorem 3.1. *Set*

$$A_k = \sum_{j=1}^{k} \sigma_j u_j v_j^T.$$

Then,

$$\min_{\substack{B \in \mathbb{R}^{m \times n} \\ \operatorname{rank}(B) \leq k}} \|A - B\|_F = \|A - A_k\|_F = \sqrt{\sum_{i=k+1}^{m} \sigma_i^2}.$$

Proof. Suppose $A = U\Sigma V^T$. Then

$$\min_{\text{rank}(B)\le k} \|A-B\|_F^2 = \min_{\text{rank}(B)\le k} \|U\Sigma V^T - UU^T BVV^T\|_F^2 = \min_{\text{rank}(B)\le k} \|\Sigma - U^T BV\|_F^2.$$

Now,

$$\|\Sigma - U^T BV\|_F^2 = \sum_{i=1}^n \left(\Sigma_{ii} - \left(U^T BV\right)_{ii}\right)^2 + \text{off-diagonal terms.}$$

If B is the best approximation matrix and $U^T BV$ is not diagonal, then write $U^T BV = D + O$, where D is diagonal and O contains the off-diagonal elements. Then the matrix $B = UDV^T$ is a better approximation, which is a contradiction.

Thus, $U^T BV$ must be diagonal. Hence,

$$\|\Sigma - D\|_F^2 = \sum_{i=1}^n (\sigma_i - d_i)^2 = \sum_{i=1}^k (\sigma_i - d_i)^2 + \sum_{i=k+1}^n \sigma_i^2,$$

and this is minimal when $d_i = \sigma_i$, $i = 1, \ldots, k$. The best approximating matrix is $A_k = UDV^T$, and the approximation error is $\sqrt{\sum_{i=k+1}^n \sigma_i^2}$. □

3.1 Closest orthogonal matrix

The SVD also allows to find the orthogonal matrix that is closest to a given matrix. Again, suppose that $A = U\Sigma V^T$ and W is an orthogonal matrix that minimizes $\|A - W\|_F^2$ among all orthogonal matrices. Now,

$$\|U\Sigma V^T - W\|_F^2 = \|U\Sigma V^T - UU^T WVV^T\| = \|\Sigma - \tilde{W}\|,$$

where $\tilde{W} = U^T WV$ is another orthogonal matrix. We need to find the orthogonal matrix $\tilde{W}$ that is closest to Σ. Alternatively, we need to minimize $\|\tilde{W}^T\Sigma - I\|_F^2$.

If U is orthogonal and D is diagonal and positive, then

$$\begin{aligned}\operatorname{trace}(UD) = \sum_{i,k} u_{ik} d_{ki} &\le \sum_i \left(\left(\sum_k u_{ik}^2\right)^{1/2} \left(\sum_k d_{ik}^2\right)^{1/2}\right) \\ &= \sum_i \left(\sum_k d_{ki}^2\right)^{1/2} = \sum_i \left(d_{ii}^2\right)^{1/2} = \sum_i d_{ii} = \operatorname{trace}(D).\end{aligned} \tag{4}$$

Now

$$\begin{aligned}\|\tilde{W}^T\Sigma - I\|_F^2 &= \operatorname{trace}\left(\left(\tilde{W}^T\Sigma - I\right)\left(\tilde{W}^T\Sigma - I\right)^T\right) \\ &= \operatorname{trace}\left(\left(\tilde{W}^T\Sigma - I\right)\left(\Sigma\tilde{W} - I\right)\right) \\ &= \operatorname{trace}\left(\tilde{W}^T\Sigma^2\tilde{W}\right) - \operatorname{trace}\left(\tilde{W}^T\Sigma\right) - \operatorname{trace}\left(\Sigma\tilde{W}\right) + n \\ &= \operatorname{trace}\left(\left(\Sigma\tilde{W}\right)^T\left(\Sigma\tilde{W}\right)\right) - 2\operatorname{trace}\left(\Sigma\tilde{W}\right) + n \\ &= \|\Sigma\tilde{W}\|_F^2 - 2\operatorname{trace}\left(\Sigma\tilde{W}\right) + n \\ &= \|\Sigma\|_F^2 - 2\operatorname{trace}\left(\Sigma\tilde{W}\right) + n.\end{aligned}$$

Thus, we need to maximize $\operatorname{trace}\left(\Sigma\tilde{W}\right)$. But this is maximized by $\tilde{W} = I$ by (4). Thus, the best approximating matrix is $W = UV^T$.

4 The “Thin” SVD

Also called “economy size” SVD. If $A \in \mathbb{C}^{m\times n}$, $A = U\Sigma V^T$, and $m \geq n$, then the “thin” SVD is $A = U_1\Sigma_1 V^T$ where

$$U_1 = [u_1, \ldots, u_n] \in \mathbb{C}^{m\times n}$$

and

$$\Sigma_1 = \operatorname{diag}(\sigma_1, \ldots, \sigma_n) \in \mathbb{R}^{n\times n}.$$

5 Applications of the SVD

1. Determining range, null space and rank (also numerical rank).
2. Matrix approximation.
3. Inverse and Pseudo-inverse: If $A = U\Sigma V^T$ and Σ is full rank, then $A^{-1} = V\Sigma^{-1}U^T$. If Σ is singular, then its pseudo-inverse is given by $A^\dagger = V\Sigma^\dagger U^T$, where $\Sigma^\dagger$ is formed by replacing every nonzero entry by its reciprocal.
4. Least squares: If we need to solve $Ax = b$ in the least-squares sense, then $x_{LS} = V\Sigma^\dagger U^T b$.
5. Denoising – Small singular values typically correspond to noise. Take the matrix whose columns are the signals, compute SVD, zero small singular values, and reconstruct.
6. Compression – We have signals as the columns of the matrix S, that is, the i signal is given by

$$S_i = \sum_{i=1}^{r} (\sigma_j v_{ij})\, u_j.$$

   If some of the σ_i are small, we can discard them with small error, thus obtaining a compressed representation of each signal. We have to keep the coefficients $\sigma_j v_{ij}$ for each signal and the dictionary, that is, the vectors u_i that correspond to the retained coefficients.

6 Differences between SVD and eigen-decomposition

1. Not every matrix has an eigen-decomposition (not even any square matrix). Any matrix (even rectangular) has an SVD.
2. In eigen-decomposition $A = X\Lambda X^{-1}$, that is, the eigen-basis is not always orthogonal. The basis of singular vectors is always orthogonal.
3. In SVD we have two singular-bases (right and left).
4. SVD tells everything on a matrix.
5. SVD as no numerical problems.
6. Relation to condition number; the numerical problems with eigen-decomposition; multiplication by an orthogonal matrix is perfectly conditioned.

7 Linear regression in the least-squared loss

In Linear regression we aim to find the best linear approximation to a set of observed data. For the m data points $\{x_1, \ldots, x_m\}$, $x_i \in \mathbb{R}^n$, each receiving the value y_i, we look for the weight vector w that minimizes:

$$\sum_{i=1}^{n}(x_i^T w - y_i)^2 = \|Aw - y\|_2^2$$

Where A is a matrix that holds the data points as rows $A_i = x_i^T$.

Proposition 7.1. *The vector w that minimizes $\|Aw - y\|_2^2$ is $w = A^\dagger y = V\Sigma^\dagger U^T y$ for $A = U\Sigma V^T$ and $\Sigma_{ii}^\dagger = 1/\Sigma_{ii}$ if $\Sigma_{ii} > 0$ and 0 else.*

Let us define $U_\parallel$ and $U_\perp$ as the parts of U corresponding to positive and zero singular values of A respectively. Also let $y_\parallel = 0$ and $y_\perp$ be two vectors such that $y = y_\parallel + y_\perp$ and $U_\parallel y_\perp = 0$ and $U_\perp y_\parallel = 0$.

Since $y_\parallel$ and $y_\perp$ are orthogonal we have that $\|Aw - y\|_2^2 = \|Aw - y_\parallel - y_\perp\|_2^2 = \|Aw - y_\parallel\|_2^2 + \|y_\perp\|_2^2$. Now, since $y_\parallel$ is in the range of A there is a solution w for which $\|Aw - y_\parallel\|_2^2 = 0$. Namely, $w = A^\dagger y = V\Sigma^\dagger U^T y$ for $A = U\Sigma V^T$. This is because $U\Sigma V^T V\Sigma^\dagger U^T y = y_\parallel$. Moreover, we get that the minimal cost is exactly $\|y_\perp\|_2^2$ which is independent of w.

8 PCA, Optimal squared loss dimension reduction

Given a set of n vectors $x_1, \ldots, x_n$ in $\mathbb{R}^m$. We look for a rank k projection matrix $P \in \mathbb{R}^{m\times m}$ that minimizes:

$$\sum_{i=1} ||Px_i - x_i||_2^2$$

If we denote by A the matrix whose i'th column is x_i then this is equivalent to minimizing $||PA - A||_{Fro}^2$ Since the best possible rank k approximation to the matrix A is $A_k = \sum_{i=1}^k \sigma_i u_i v_i^T$ the best possible solution would be a projection P for which $PA = A_k$. This is achieved by $P = U_k U_k^T$ where U_k is the matrix corresponding to the first k left singular vectors of A.

If we define $y_i = U_k^T x_i$ we see that the values of $y_i \in \mathbb{R}^k$ are optimally fitted to the set of points x_i in the sense that they minimize:

$$\min_{y_1,\ldots,y_n} \min_{\Psi\in\mathbb{R}^{k\times m}} \sum_{i=1} ||\Psi y_i - x_i||_2^2$$

The mapping of $x_i \to U_k^T x_i = y_i$ thus reduces the dimension of any set of points $x_1, \ldots, x_n$ in $\mathbb{R}^m$ to a set of points $y_1, \ldots, y_n$ in $\mathbb{R}^k$ optimally in the squared loss sense. This is commonly referred to as Principal Component Analysis (PCA).

9 The power method

We give a simple algorithm for computing the Singular Value Decomposition of a matrix $A \in \mathbb{R}^{m\times n}$. We start by computing the first singular value σ_1 and left and right singular vectors u_1 and v_1 of A, for which $min_{i<j} \log(\sigma_i/\sigma_j) \geq \lambda$:

1. Generate x_0 such that $x_0(i) \sim \mathcal{N}(0, 1)$.
2. $s \leftarrow \log(4\log(2n/\delta)/\varepsilon\delta)/2\lambda$
3. for i in $[1, \ldots, s]$:
4. $\quad x_i \leftarrow A^T A x_{i-1}$

5. $v_1 \leftarrow x_i/\|x_i\|$
6. $\sigma_1 \leftarrow \|Av_1\|$
7. $u_1 \leftarrow Av_1/\sigma_1$
8. return (σ_1, u_1, v_1)

Let us prove the correctness of this algorithm. First, write each vector x_i as a linear combination of the right singular values of A i.e. $x_i = \sum_j \alpha_j^i v_j$. From the fact that v_j are the eigenvectors of $A^T A$ corresponding to eigenvalues σ_j^2 we get that $\alpha_j^i = \alpha_j^{i-1}\sigma_j^2$. Thus, $\alpha_j^s = \alpha_j^0 \sigma_j^{2s}$. Looking at the ratio between the coefficients of v_1 and v_i for x_s we get that:

$$\frac{|<x_s, v_1>|}{|<x_s, v_i>|} = \frac{|\alpha_1^0|}{|\alpha_i^0|}\left(\frac{\sigma_1}{\sigma_i}\right)^{2s}$$

Demanding that the error in the estimation of σ_1 is less than ε gives the requirement on s.

$$\frac{|\alpha_1^0|}{|\alpha_i^0|}\left(\frac{\sigma_1}{\sigma_i}\right)^{2s} \geq \frac{n}{\varepsilon} \tag{5}$$

$$s \geq \frac{\log(n|\alpha_i^0|/\varepsilon|\alpha^0|_1)}{2\log(\sigma_1/\sigma_i)} \tag{6}$$

From the two-stability of the gaussian distribution we have that $\alpha_i^0 \sim \mathcal{N}(0,1)$. Therefore, $\Pr[\alpha_i^0 > t] \leq e^{-t^2}$ which gives that with probability at least $1-\delta/2$ we have for all i, $|\alpha_i^0| \leq \sqrt{\log(2n/\delta)}$. Also, $\Pr[|\alpha_1^0| \leq \delta/4] \leq \delta/2$ (this is because $\Pr[|z| < t] \leq max_r \Psi_z(r) \cdot 2t$ for any distribution and the normal distribution function at zero takes it maximal value which is less than 2) Thus, with probability at least $1-\delta$ we have that for all i, $\frac{|\alpha_1^0|}{|\alpha_i^0|} \leq \frac{\sqrt{\log(2n/\delta)}}{\delta/4}$. Combining all of the above we get that it is sufficient to set $s = \log(4n\log(2n/\delta)/\varepsilon\delta)/2\lambda = O(\log(n/\varepsilon\delta)/\lambda)$ in order to get ε precision with probability at least $1-\delta$.

We now describe how to extend this to a full SVD of A. Since we have computed (σ_1, u_1, v_1), we can repeat this procedure for $A - \sigma_1 u_1 v_1^T = \sum_{i=2}^n \sigma_i u_i v_i^T$. The top singular value and vectors of which are (σ_2, u_2, v_2). Thus, computing the rank-k approximation of A requires $O(mnks) = O(mnk\log(n/\varepsilon\delta))/\lambda)$ operations. This is because computing $A^T Ax$ requires $O(mn)$ operations and for each of the first k singular values and vectors this is performed s times.

The main problem with this algorithm is that its running time is heavily influenced by the value of λ. Other variants of this algorithm are much less sensitive to the value of this parameter, but are out of the scope of this class.