University of Edinburgh

INFR11156: Algorithmic Foundations of Data Science (2019)

Lecture 6: Best-fit Subspaces and Singular Value Decomposition (2)

Let $A \in \mathbb{R}^{m\times n}$ be a matrix whose SVD is written as $\sum_i \sigma_i u_i v_i^\intercal$. We define $B = A^\intercal A$, i.e.,

$$\begin{aligned} B = A^\intercal A &= \left(\sum_i \sigma_i v_i u_i^\intercal\right)\left(\sum_i \sigma_i u_i v_i^\intercal\right) \\ &= \sum_i \sum_j \sigma_i \sigma_j v_i (u_i^\intercal u_j) v_j^\intercal \\ &= \sum_i \sigma_i^2 v_i v_i^\intercal. \end{aligned}$$

The matrix $B \in \mathbb{R}^{n\times n}$ is a square and symmetric, and has the same left and right-singular vectors. In particular, it holds for any v_j that

$$Bv_j = \left(\sum_i \sigma_i^2 v_i v_i^\intercal\right) v_j = \sigma_j^2 v_j,$$

meaning that v_j is an eigenvector of B with the corresponding eigenvalue σ_j^2. We write $\lambda_i = \sigma_i^2$ and v_i for the eigenvalues and their corresponding eigenvectors of B. Without loss of generality, we assume that $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$. Notice that all the eigenvalues $\lambda_i \geq 0$, i.e., matrix B is positive semi-definite (PSD).

Now we consider B^2. By definition, we have that

$$B^2 = \left(\sum_i \lambda_i v_i v_i^\intercal\right)\left(\sum_i \lambda_i v_i v_i^\intercal\right) = \sum_i \lambda_i^2 v_i v_i^\intercal.$$

By induction we have that

$$B^k = B^{k-1}B = \left(\sum_i \lambda_i^{k-1} v_i v_i^\intercal\right)\left(\sum_i \lambda_i v_i v_i^\intercal\right) = \sum_i \lambda_i^k v_i v_i^\intercal.$$

Hence, if $\lambda_1 > \lambda_2$, then the first term in the summation dominates, and $B^k \to \lambda_1^k v_1 v_1^\intercal$. However, this approach to approximate v_1 requires computing B^k for some k, which is inefficient as the matrix multiplication takes time $\Omega(n^2)$. Therefore, a more efficient approach is needed.

In this lecture, we study the *power method* for computing eigenvalues and eigenvectors, whose ideas are summarised as follows: instead of computing B^k, we select a random vector x and compute $B^k x$. To see why this approach works, we write $x = \sum_i c_i v_i$ for some constants $c_i \in \mathbb{R}$. Then, it holds that

$$B^k x = \left(\sum_i \lambda_i^k v_i v_i^\intercal\right)\cdot\left(\sum_i c_i v_i\right) = \sum_i c_i \lambda_i^k v_i.$$

For time complexity, notice that computing Bx for any vector x takes $O(n + \mathrm{nnz}(B))$ time if the non-zero entries of matrix B are stored by an adjacency list, where $\mathrm{nnz}(B)$ is the number of non-zero entries of matrix B. Hence, the total runtime for computing $B^k x$ is $O(k\cdot(n+\mathrm{nnz}(B)))$. For many applications where the matrix $B \in \mathbb{R}^{n\times n}$ is sparse, e.g., $\mathrm{nnz}(B) = O(n)$, the power method presents a vast speedup comparing with the naive algorithm that computes B^k directly. The formal description of the power method for computing λ_1 is shown in Algorithm 1.

Remark. It is important to notice that, even matrix A is spare, the matrix $B = A^\intercal A$ might not be a sparse matrix any more. In such case, to compute $B^k x$ it suffices to compute $\left(A^\intercal A\right)^k x$, which can be done in $O(k \cdot (n + \mathrm{nnz}(A)))$ time.

Algorithm 1 Power method for approximating λ_1

Input: a PSD symmetric matrix $B \in \mathbb{R}^{n\times n}$, and positive integer k
Choose x_0 uniformly at random from $\{-1,1\}^n$.
for $i = 1$ to k **do**
  $x_i = Bx_{i-1}$
end for
return x_k

To analyse the algorithm, by definition we have that $\sigma_1(A) = \max_{\|x\|=1} \|Ax\|$, and $\lambda_1(B) = \sigma_1^2(A)$. Hence, we can write the largest eigenvalue of B as

$$\lambda_1(B) = \max_{\|x\|=1} \|Ax\|^2 = \max_{\substack{x\in\mathbb{R}^n\\x\neq 0}} \frac{x^\intercal A^\intercal Ax}{\|x\|} = \max_{\substack{x\in\mathbb{R}^n\\x\neq 0}} \frac{x^\intercal Bx}{\|x\|}.$$

This is called the *Courant-Fischer Characterisation of Eigenvalues*. Hence, it suffices to study $(x_k^\intercal Bx_k) \cdot (x_k^\intercal x_k)^{-1}$.

Theorem 1. *For every PSD matrix B, positive integer k and parameter $\varepsilon > 0$, with probability $3/16$ over the initial choices of x_0, Algorithm 1 outputs a vector x_k such that*

$$\frac{x_k^\intercal Bx_k}{x_k^\intercal x_k} \geq (1-\varepsilon)\cdot\lambda_1\cdot\frac{1}{1+4n(1-\varepsilon)^{2k}}.$$

In particular, when setting $k = O(\log n/\varepsilon)$, we have that

$$\frac{x_k^\intercal Bx_k}{x_k^\intercal x_k} \geq (1-O(\varepsilon))\lambda_1.$$

The proof is based on the following two lemmas.

Lemma 2. *Let $v \in \mathbb{R}^n$ such that $\|v\| = 1$. Sample uniformly $x \in \{-1,1\}^n$. Then it holds that*

$$\mathbf{P}\left[|\langle x, v\rangle| \geq \frac{1}{2}\right] \geq \frac{3}{16}.$$

Lemma 3. *Let $x \in \mathbb{R}^n$ be a vector such that $|\langle x, v_1\rangle| \geq 1/2$. Then, for every positive integer k and positive $\varepsilon > 0$, if we define $y = B^k x$, then we have that*

$$\frac{y^\intercal By}{y^\intercal y} \geq (1-\varepsilon)\cdot\lambda_1\cdot\frac{1}{1+4\|x\|^2(1-\varepsilon)^{2k}}.$$

Proof of Theorem 1. By Lemma 2, with constant probability, a randomly sampled $x \in \{-1,1\}^n$ satisfies $|\langle x, v\rangle| \geq 1/2$ for any $\|v\| = 1$. Conditioning on this event, Lemma 3 states that

$$\frac{y^\intercal By}{y^\intercal y} \geq (1-\varepsilon)\cdot\lambda_1\cdot\frac{1}{1+4\|x\|^2(1-\varepsilon)^{2k}}.$$

Then, the theorem holds by the fact that $\|x\|^2 = n$. □

Proof of Lemma 2. Define a random variable $S = \langle x, v \rangle$. Then, it holds that $\mathbf{E}[S] = 0$, $\mathbf{E}\left[S^2\right] = \|v\|^2 = 1$, and[1]

$$\mathbf{E}\left[S^4\right] = 3\sum_{i=1}^{n} v_i^2 - 2\sum_{i=1}^{n} v_i^4 \leq 3.$$

Recall that the Paley-Zygmund inequality states that if Z is a non-negative random variable with finite variance, then it holds for every $0 \leq \delta \leq 1$ that

$$\mathbf{P}\left[Z \geq \delta \cdot \mathbf{E}Z\right] \geq (1-\delta)^2 \cdot \frac{(\mathbf{E}Z)^2}{\mathbf{E}[Z^2]},$$

which follows by noticing that

$$\begin{aligned}
\mathbf{E}[Z] &= \mathbf{E}\left[Z \cdot \mathbf{1}_{Z<\delta \mathbf{E}Z}\right] + \mathbf{E}\left[Z \cdot \mathbf{1}_{Z \geq \delta \mathbf{E}Z}\right] \\
&\leq \delta \mathbf{E}Z + \sqrt{\mathbf{E}Z^2} \cdot \sqrt{\mathbf{E}\mathbf{1}_{Z\geq\delta\mathbf{E}Z}} \\
&= \delta \mathbf{E}Z + \sqrt{\mathbf{E}Z^2} \cdot \sqrt{\mathbf{P}\left[Z \geq \delta \mathbf{E}Z\right]},
\end{aligned}$$

where the first inequality follows by Cauchy-Schwarz inequality. We apply the Paley-Zygmund inequality to the case $Z = S^2$ and $\delta = 1/4$ and have that

$$\mathbf{P}\left[S^2 \geq \delta \mathbf{E}[S^2]\right] = \mathbf{P}\left[S^2 \geq \frac{1}{4}\right] \geq \left(\frac{3}{4}\right)^2 \cdot \frac{1}{3} = \frac{3}{16}.$$

□

Proof of Lemma 3. We write x as a linear combination of the eigenvectors

$$x = a_1 v_1 + \cdots + a_n v_n$$

where the coefficients can be computed as $a_i = \langle x, v_i \rangle$. Then, we rewrite $y = B^k x$ as

$$y = a_1 \lambda_1^k v_1 + \cdots + a_n \lambda_n^k v_n,$$

and therefore

$$y^\intercal B y = \sum_{i=1}^{n} a_i^2 \lambda_i^{2k+1},$$

as well as

$$y^\intercal y = \sum_{i=1}^{n} a_i^2 \lambda_i^{2k}.$$

Without loss of generality let ℓ be the number of eigenvalues larger than $\lambda_1 \cdot (1-\varepsilon)$. Then, it holds that

$$y^\intercal B y \geq \sum_{i=1}^{\ell} a_i^2 \lambda_i^{2k+1} \geq \lambda_1 (1-\varepsilon) \sum_{i=1}^{\ell} a_i^2 \lambda_i^{2k}. \tag{1}$$

Since all the eigenvalues λ_i for $i \geq \ell + 1$ is at most $\lambda_1 \cdot (1-\varepsilon)$, we have that

$$\begin{aligned}
\sum_{i=\ell+1}^{n} a_i^2 \lambda_i^{2k} &\leq \lambda_1^{2k} \cdot (1-\varepsilon)^{2k} \sum_{i=\ell+1}^{n} a_i^2 \\
&\leq \lambda_1^{2k} \cdot (1-\varepsilon)^{2k} \|x\|^2 \\
&\leq 4 a_1^2 \lambda_1^{2k} \cdot (1-\varepsilon)^{2k} \|x\|^2 \qquad (2) \\
&\leq 4 \|x\|^2 (1-\varepsilon)^{2k} \sum_{i=1}^{\ell} a_i^2 \lambda_i^{2k},
\end{aligned}$$

[1]Obtaining the equality below is not straightforward, and involves some calculations. We leave this for homework.

where (2) follows from the fact that $a_1^2 = |\langle x, v_1 \rangle|^2 \geq 1/4$ by the assumption of the Lemma. Hence, we have that

$$y^\intercal y \leq \left(1 + 4\|x\|^2(1-\varepsilon)^{2k}\right) \cdot \sum_{i=1}^{\ell} a_i^2 \lambda_i^{2k}. \tag{3}$$

Combining (1) with (3) gives us that

$$\frac{y^\intercal B y}{y^\intercal y} \geq \lambda_1 \cdot (1-\varepsilon) \cdot \frac{1}{1 + 4\|x\|^2(1-\varepsilon)^{2k}}. \qquad \square$$

Sometimes, we know the eigenvector v_1 corresponding to λ_1, and we need to approximate v_2 and λ_2. Then a similar approach can be applied, but we only need to ensure that the initial vector used for the "power iterations" is perpendicular to v_1, see Algorithm 2 for formal description.

Algorithm 2 Power method for approximating λ_2

Input: a PSD symmetric matrix $B \in \mathbb{R}^{n \times n}$, and positive integer k
Choose x uniformly at random from $\{-1, 1\}^n$.
Let $x_0 = x - \langle v_1, x \rangle \cdot v_1$
for $i = 1$ to k **do**
    $x_i = Bx_{i-1}$
end for
return x_k

Now we briefly analyse this algorithm. We assume that $v_1, \ldots, v_n$ is an orthonormal basis of the eigenvectors for the eigenvalues $\lambda_1 \geq \ldots \geq \lambda_n$ of B. Then we write the initial random vector as

$$x = a_1 v_1 + \cdots + a_n v_n,$$

and with probability at least $3/16$ it holds that $|a_2| = |\langle x, v_2 \rangle| \geq 1/2$. Then, x_0 is the projection of x on the subspace orthogonal to v_1, i.e.,

$$x_0 = a_2 v_2 + \cdots + a_n v_n.$$

Notice that $\|x_0\| \leq n$. Furthermore, the output x_k can be written as

$$x_k = a_2 \lambda_2^k v_2 + \cdots + a_n \lambda_n^k v_n.$$

Then, we can apply the same analysis as before, and have the following result:

Theorem 4. *For every PSD matrix $B \in \mathbb{R}^{n \times n}$, positive integer k and parameter $\varepsilon > 0$, with constant probability over the choices of x, Algorithm 2 outputs a vector $y \perp v_1$ such that*

$$\frac{y^\intercal B y}{y^\intercal y} \geq \lambda_2 \cdot (1-\varepsilon) \cdot \frac{1}{1 + 4n(1-\varepsilon)^{2k}},$$

where λ_2 is the second largest eigenvalue of B, counting multiplicities.