Mandatory Assignment 3
MAT-INF4130

Ivar Stangeby

October 25, 2017

1 Introduction

Given a set of points $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\} \subseteq \mathbb{R}^m$ and an integer $k \leq m$, we are interested in finding the k-dimensional subspace W that *minimizes* the squared distance:

$$\sum_{i=1}^{m} \operatorname{dist}(\boldsymbol{x}_i - W)^2. \tag{1}$$

To this end we first establish some notation. Let $\boldsymbol{X} \coloneqq [\boldsymbol{x}_1, \dots, \boldsymbol{x}_n]$ be the matrix with ijth element denoted x_{ij}. If W is any k-dimensional subspace of $\mathbb{R}^m$ and $\{\boldsymbol{w}_j\}_{j=1}^{k}$ is an orthonormal basis for W, we may extend this basis to an orthonormal basis $\{\boldsymbol{w}_j\}_{j=1}^{m}$ for $\mathbb{R}^m$ by the Grahm-Schmidt algorithm. We can therefore decompose the space $\mathbb{R}^m$ into W and its orthogonal complement $W^\perp$:

$$\mathbb{R}^m = W \oplus W^\perp. \tag{2}$$

For a vector $\boldsymbol{x} \in \mathbb{R}^m$, denote by $\operatorname{proj}_{W^\perp}(\boldsymbol{x})$, its orthogonal projection onto $W^\perp$. Furthermore, recall that the Frobenius norm of a matrix $\boldsymbol{A} \in \mathbb{C}^{m \times n}$ is defined as:

$$\|\boldsymbol{A}\|_{\mathrm{F}} \coloneqq \left(\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2 \right)^{1/2} \tag{3}$$

and that this is equivalent to the 2-norm of the vector $\mathrm{Vec}(\boldsymbol{A})$ formed by stacking the columns of $\boldsymbol{A}$ on top of each other. We denote by $\langle \cdot, \cdot \rangle$ the standard inner product on $\mathbb{R}^m$ defined by $\langle \boldsymbol{x}, \boldsymbol{y} \rangle \coloneqq \boldsymbol{y}^T \boldsymbol{x}$.

Exercise 1

We start by showing that

$$\sum_{i=1}^{n} \|\mathrm{proj}_{W^\perp}(\boldsymbol{x}_i)\|_2^2 = \left\|\boldsymbol{X}^T \boldsymbol{W}\right\|_{\mathrm{F}}^2. \tag{4}$$

where $\boldsymbol{W}$ is the $m \times (m-k)$ matrix defined as $\boldsymbol{W} \coloneqq [\boldsymbol{w}_{k+1}, \dots, \boldsymbol{w}_m]$. Since $\{\boldsymbol{w}_{k+1}, \dots, \boldsymbol{w}_m\}$ is a basis for $W^\perp$, we can write $\mathrm{proj}_{W^\perp}(\boldsymbol{x}_i) = \sum_{j=k+1}^{m} c_i \boldsymbol{w}_i$ where $c_j = \langle \boldsymbol{x}_i, \boldsymbol{w}_j \rangle$. Expanding the left hand side of eq. (4) yields:

$$\begin{aligned}
\sum_{i=1}^{n} \|\mathrm{proj}_{W^\perp}(\boldsymbol{x}_i)\|_2^2 &= \sum_{i=1}^{n} \langle \mathrm{proj}_{W^\perp}(\boldsymbol{x}_i), \mathrm{proj}_{W^\perp}(\boldsymbol{x}_i) \rangle \\
&= \sum_{i=1}^{n} \sum_{j=k+1}^{m} \sum_{\ell=k+1}^{m} \langle \boldsymbol{x}_i, \boldsymbol{w}_j \rangle \langle \boldsymbol{x}_i, \boldsymbol{w}_\ell \rangle \delta_{j,\ell} \\
&= \sum_{i=1}^{n} \sum_{j=k+1}^{m} |\langle \boldsymbol{x}_i, \boldsymbol{w}_j \rangle|^2.
\end{aligned} \tag{5}$$

Since we are working over the reals, we may write $\langle \boldsymbol{x}_i, \boldsymbol{w}_j \rangle = \langle \boldsymbol{w}_j, \boldsymbol{x}_i \rangle$, hence the equality follows in eq. (4).

Exercise 2

Let $\boldsymbol{X} = \boldsymbol{U} \boldsymbol{\Sigma} \boldsymbol{V}^T$ be a singular value decomposition of $\boldsymbol{X}$.

a) We wish to show that

$$\left\|\boldsymbol{X}^T \boldsymbol{W}\right\|_{\mathrm{F}}^2 = \left\|\boldsymbol{\Sigma}^T \boldsymbol{U}^{\boldsymbol{T}} \boldsymbol{W}\right\|_{\mathrm{F}}^2. \tag{6}$$

This can easily be shown by noting that for arbitrary unitary matrix $\boldsymbol{B}$ and matrix $\boldsymbol{A}$ we have

$$\|\boldsymbol{B}\boldsymbol{A}\|_{\mathrm{F}} = \|\boldsymbol{A}\|_{\mathrm{F}}. \tag{7}$$

This follows from the fact that the Frobenius norm on $\mathbb{C}^{m\times n}$ is the same as the 2-norm on the vector space $\mathbb{C}^{mn}$ and that $\|\boldsymbol{Bx}\|_2 = \|\boldsymbol{x}\|_2$ for all $\boldsymbol{x}$. Indeed, we have

$$\|\boldsymbol{BA}\|_{\mathrm{F}} = \sum_{j=1}^{n} \|\boldsymbol{B}\boldsymbol{a}_{:j}\|_2 = \sum_{j=1}^{n} \|\boldsymbol{a}_{:j}\|_2 = \|\boldsymbol{A}\|_{\mathrm{F}}. \tag{8}$$

Setting $\boldsymbol{B} \coloneqq \boldsymbol{V}$ and $\boldsymbol{A} \coloneqq \boldsymbol{\Sigma^T U^T W}$ in eq. (7) and squaring proves the result.

b) We now wish to show that

$$\left\|\boldsymbol{X}^T\boldsymbol{W}\right\|_{\mathrm{F}} = \sum_{i=1}^{m} \sigma_i^2 \| \operatorname{proj}_{W^\perp}(\boldsymbol{u}_i)\|_2^2 \tag{9}$$

Expanding the matrix $\boldsymbol{U}^T\boldsymbol{W}$ tells us that the ijth entry is given by

$$(\boldsymbol{U}^T\boldsymbol{W})_{ij} = \langle \boldsymbol{u}_i, \boldsymbol{w}_{j+k}\rangle \tag{10}$$

for $i = 1, \ldots, m$ and $j = 1, \ldots, m-k$. Pre-multiplying by $\boldsymbol{\Sigma}^T$ gives

$$(\boldsymbol{\Sigma}^T\boldsymbol{U}^T\boldsymbol{W})_{ij} = \sigma_i \langle \boldsymbol{u}_i, \boldsymbol{w}_{j+k}\rangle \tag{11}$$

We can therefore write the Frobenius norm of this matrix as

$$\left\|\boldsymbol{\Sigma}^T\boldsymbol{U}^T\boldsymbol{W}\right\|_{\mathrm{F}}^2 = \sum_{i=1}^{m} \sigma_i^2 \sum_{j=1}^{m-k} \langle \boldsymbol{u}_i, \boldsymbol{w}_{j+k}\rangle = \sum_{i=1}^{m} \sigma_i^2 \|\operatorname{proj}_{W^\perp}(\boldsymbol{u}_i)\|_2^2$$

as we wanted to show. Note that the singular values are ordered by convention $\sigma_1 \geq \ldots \sigma_m \geq 0$.

c) We also have the property that since the columns of $\boldsymbol{W}$ are orthonormal, and since there are $m-k$ such columns, that

$$\|\boldsymbol{W}\|_{\mathrm{F}}^2 = m - k. \tag{12}$$

Since $\boldsymbol{U}^T$ is unitary, we have that

$$\|\boldsymbol{W}\|_{\mathrm{F}}^2 = \left\|\boldsymbol{U}^T\boldsymbol{W}\right\|_{\mathrm{F}}^2 = \sum_{i=1}^{m} \sum_{j=k+1}^{m} \langle \boldsymbol{u}_i, \boldsymbol{w}_j\rangle^2 = \sum_{i=1}^{m} \|\operatorname{proj}_{W^\perp}(\boldsymbol{u}_i)\|_2^2. \tag{13}$$

Exercise 3

a) We can now find the "distance to the best subspace" $\left\| \boldsymbol{X}^T \boldsymbol{W} \right\|_F^2$ by solving the minimization problem

$$\min \sum_{i=1}^{m} \sigma_i^2 \kappa_i \tag{14}$$

where $0 \leq \kappa_i \leq 1$ and $\sum_{i=1}^{m} \kappa_i = m - k$. Indeed, we have

$$\left\| \boldsymbol{X}^T \boldsymbol{W} \right\|_F^2 = \sum_{i=1}^{m} \sigma_i^2 \|\mathrm{proj}_{W^\perp}(\boldsymbol{u}_i)\|_2^2 \tag{15}$$

where setting $\kappa_i \coloneqq \|\mathrm{proj}_{W^\perp}(\boldsymbol{u}_i)\|_2^2$ yields the above sum. Furthermore, we have that $0 \leq \kappa_i \leq 1$ since the $\boldsymbol{u}_i$'s are orthonormal, and we showed above that $\sum_{i=1}^{m} \kappa_i = m - k$.

b) The solution to this problem is obtained by setting $\kappa_1 = \ldots = \kappa_k = 0$ and $\kappa_{k+1} = \ldots = \kappa_m = 1$, since the σ_i's are ordered by magnitude. Indeed, the sum is minimized if we chose the vectors such that the columns of $\boldsymbol{W}$ are perpendicular to $\mathrm{Span}(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k)$. Since $0 \leq \kappa_i \leq 1$, this means in order for $\sum_{i=1}^{m} = m - k$, we must have $\kappa_{k+1} = \ldots = \kappa_m = 1$.

c) We can therefore choose the matrix $\boldsymbol{W}$ to consist of the last $m - k$ vectors $\boldsymbol{u}_{k+1}, \ldots, \boldsymbol{u}_m$ and let $W^\perp = \mathrm{Span}(\boldsymbol{u}_{k+1}, \ldots, \boldsymbol{u}_m)$. This set of vectors satisfies the minimization problem discussed above.

d) We implement a routine for solving this minimization problem in Python.

```
def bestfit(X, k):
    U = np.linalg.svd(X)[0]
    B = U[:, :k]
    W = U[:, k:]
    dist = np.linalg.norm(X.T.dot(W))
    return B, dist
```

Exercise 4

We can alternatively choose to implement a "vanilla" least squares, where we view the first k components of each column in $\boldsymbol{X}$ as input data and the last $m-k$ components of each column as output data. By defining $\boldsymbol{Y} := \boldsymbol{X}_{1:k,1:n}$ and $\boldsymbol{Z} := \boldsymbol{X}_{k+1:m,1:n}$, we seek the matrix $\boldsymbol{A}$ that minimizes

$$\|\boldsymbol{A}\boldsymbol{Y} - \boldsymbol{Z}\|_{\mathrm{F}}. \tag{16}$$

a) It can be shown that if $\boldsymbol{A}$ solves the least squares problem, then it must satisfy the relation

$$\boldsymbol{A}^{T} = (\boldsymbol{Y}\boldsymbol{Y}^{\boldsymbol{T}})^{-1}\boldsymbol{Y}\boldsymbol{Z}^{T}, \tag{17}$$

however I was not successful in proving this.

b) We implement vanilla least squares in Python:

```
def vanleastsqr(X, k):
    Y = X[0:k, 0:]
    Z = X[k:, 0:]
    A = la.inv(Y.dot(Y.T)).dot(Y).dot(Z.T).T
    dist = la.norm(A.dot(Y) - Z)
    return A, dist
```

Exercise 5

The two algorithms previously discussed are tested on two test 2×21 matrices $\boldsymbol{X}_0$ and $\boldsymbol{X}_1$. The first row of matrix $\boldsymbol{X}_1$ contains 21 uniformly spaced values between -1 and 1, while the second row contains uniformly random numbers between -1 and 1. All the elements in $\boldsymbol{X}_2$ are random.

As we see from fig. 1, the "bestfit" algorithm provides a better fit in both cases. This might be due to numerical instability in the computation of the inverse in the vanilla least squares.

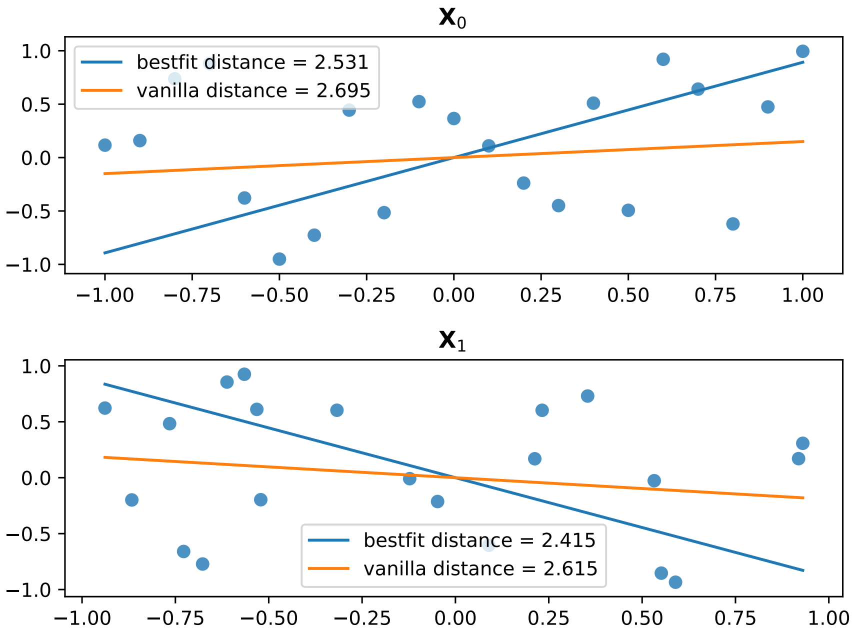

Figure 1: Two approaches to the least squares problem of finding the best subspace. As we see, the fit is better for the “bestfit” algorithm, in both cases.