Homework 10 Solutions

1. (10 points) *Statements (10 points).*

   a. Let us denote the dimensions of $\mathcal{S}$ and $\mathcal{S}^\perp$ by $0 \leq d, m \leq n$. Let $a_1, \ldots, a_d$ be a basis of $\mathcal{S}$ and $b_1, \ldots, b_m$ be a basis of $\mathcal{S}^\perp$. By Theorem 1.4 in Lecture Notes 9, we can write any vector in $\mathcal{V}$ as a linear combination of a vector in $\mathcal{S}$ and a vector in $\mathcal{S}^\perp$. In addition, all vectors are orthogonal to each other (a_i and b_j for $1 \leq i \leq d, 1 \leq j \leq m$ are orthogonal because they belong to orthogonal subspaces). This implies that $\{a_1, \ldots, a_d, b_1, \ldots, b_m\}$ is an orthonormal basis of $\mathcal{V}$ and consequently $d + m = n$.

   b. By Lemma 1.5 in Lecture Notes 10 $\text{null}\,(A) = \text{row}\,(A)^\perp$, so applying the result in the previous question we have

   $$\begin{aligned} n &= \dim\left(\text{null}\,(A)\right) + \dim\left(\text{row}\,(A)\right) && (1)\\ &= \dim\left(\text{null}\,(A)\right) + \dim\left(\text{col}\,(A)\right) \quad \text{by Theorem 6.2 in Lecture Notes 9} && (2)\\ &= \dim\left(\text{null}\,(A)\right) + \dim\left(\text{range}\,(A)\right) \quad \text{by equation (4) in Lecture Notes 10.} && (3)\end{aligned}$$

   c. Let $A = USV^T$ be the SVD of A. If the matrix is full rank, then all the singular values in the diagonal of S are nonzero, so S^{-1} exists (it just contains $1/\sigma_i$ $1 \leq i \leq n$ in the diagonal). In addition, since the matrix is fat, $V^TV = \text{I}$ (but not the other way round!). As a result

   $$\begin{aligned} A^T\left(AA^T\right)^{-1}A &= VSU^T\left(USV^TVSU^T\right)^{-1}USV^T && (4)\\ &= VSU^T\left(USSU\right)^{-1}USV^T && (5)\\ &= VSU^T\left(U^T\right)^{-1}S^{-1}S^{-1}U^{-1}USV^T && (6)\\ &= VSS^{-1}S^{-1}SV^T && (7)\\ &= VV^T. && (8)\end{aligned}$$

   By definition of the SVD, the columns of V are an orthonormal basis of the row space of A, so by Theorem 1.4 (6) the proof is complete.

   d. If the columns of A are orthonormal then $A^TA = \text{I}$, so the least-squares solution is just $\left(A^TA\right)^{-1}A^T = A^T$.

2. (10 points) *Global warming (10 points).*

   a. The code is

```
fit_matrix = np.zeros((n,4))
fit_matrix[:,0]=np.ones(n)
fit_matrix[:,1]=np.linspace(0.0,1.0,n)
fit_matrix[:,2]=np.cos(2 * np.pi * np.arange(1.0,n+1,1.)/ 12)
fit_matrix[:,3]=np.sin(2 * np.pi * np.arange(1.0,n+1,1.)/ 12)
coeffs_max = np.linalg.lstsq(fit_matrix,max_temp)[0]
coeffs_min = np.linalg.lstsq(fit_matrix,min_temp)[0]
reconstruction_max = np.dot(fit_matrix, coeffs_max)
reconstruction_min = np.dot(fit_matrix, coeffs_min)
trend_max =  np.dot(fit_matrix[:,:2], coeffs_max[:2]);
trend_min =  np.dot(fit_matrix[:,:2], coeffs_min[:2]);
```

b. The p-value is the probability of obtaining a greater or equal number of positive slopes under the null hypothesis. The distribution of positive slopes under the null hypothesis is binomial with parameters $n = 100$ and $p = 1/2$, so

$$p = \sum_{k=85}^{100} \binom{100}{k} \left(\frac{1}{2}\right)^k \left(\frac{1}{2}\right)^{100-k} \tag{9}$$

$$= \frac{1}{2^{100}} \sum_{k=85}^{100} \binom{100}{k} \tag{10}$$

$$= 0.75\%. \tag{11}$$

3. (10 points) *Weight prediction (10 points).*

a. The solution to an overdetermined least-squares problem with matrix A and data y is $\left(A^T A\right)^{-1} A^T y$. For α_1 we have $A = w$ and $y = h$ so

$$\alpha_1 = \left(w^T w\right)^{-1} w^T h \tag{12}$$

$$= \frac{w^T h}{w^T w}. \tag{13}$$

For α_2 and β we have $A = \begin{bmatrix} w & \mathbf{1} \end{bmatrix}$, where $\mathbf{1}$ is a vector of ones, and $y = h$ so

$$\begin{bmatrix} \alpha_2 \\ \beta \end{bmatrix} = \left(\begin{bmatrix} w^T \\ \mathbf{1}^T \end{bmatrix} \begin{bmatrix} w & \mathbf{1} \end{bmatrix} \right)^{-1} \begin{bmatrix} w^T \\ \mathbf{1}^T \end{bmatrix} h \tag{14}$$

$$= \begin{bmatrix} w^T w & w^T \mathbf{1} \\ \mathbf{1}^T w & n_{\text{train}} \end{bmatrix}^{-1} \begin{bmatrix} w^T h \\ \mathbf{1}^T h \end{bmatrix} \tag{15}$$

$$= \frac{1}{n w^T w - \mathbf{1}^T w^2} \begin{bmatrix} n_{\text{train}} & -w^T \mathbf{1} \\ -\mathbf{1}^T w & w^T w \end{bmatrix} \begin{bmatrix} w^T h \\ \mathbf{1}^T h \end{bmatrix} \tag{16}$$

$$= \frac{1}{n w^T w - \mathbf{1}^T w^2} \begin{bmatrix} n_{\text{train}} w^T h - w^T \mathbf{1}\mathbf{1}^T h & \\ & -\mathbf{1}^T w \qquad\qquad w^T w w^T h - \mathbf{1}^T w \mathbf{1}^T h \end{bmatrix} \tag{17}$$

$$= \frac{1}{n w^T w - \left(\sum_{i=1}^{n_{\text{train}}} w_i\right)^2} \begin{bmatrix} n_{\text{train}} w^T h - \left(\sum_{i=1}^{n_{\text{train}}} w_i\right)\left(\sum_{i=1}^{n_{\text{train}}} h_i\right) \\ w^T w w^T h - \left(\sum_{i=1}^{n_{\text{train}}} w_i\right)\left(\sum_{i=1}^{n_{\text{train}}} h_i\right) \end{bmatrix}. \tag{18}$$

b. Adding an intercept allows to account for linear structure that is displaced from the origin (a linear space that is displaced from the origin is called an affine space). An example would be data lying near a line that does not go through the origin.

c. The code is

```
coeff_model_1 = np.dot(heights_train,weights_train)/np.dot(heights_train, heights_train)
pred_model_1 = coeff_model_1 * heights_test
aux_matrix_train = np.ones((n_train,2))
aux_matrix_train[:,1] = heights_train
aux_matrix_test = np.ones((n_test,2))
aux_matrix_test[:,1] = heights_test
coeff_model_2 = np.linalg.lstsq(aux_matrix_train,weights_train)[0]
pred_model_2 = np.dot(aux_matrix_test, coeff_model_2)
```

The errors should be between 0.06 and 0.07 for both models.

d. The model produces similar results. Linear models with few parameters can be used with very little data, but they are not able to incorporate nonlinear structure that could be useful for prediction. As a result, they tend to be more appropriate in situations where data is scarce.

4. (10 points) *Noise amplification (20 points).*

a.

$$\begin{aligned} x_{\mathrm{LS}} &= \left(A^T A\right)^{-1} A^T y && (19)\\ &= \left(A^T A\right)^{-1} A^T \left(Ax + z\right) && (20)\\ &= x + V S^{-1} U^T z. && (21) \end{aligned}$$

b. The norm of the error is equal to

$$\begin{aligned} ||x - x_{\mathrm{LS}}||_2 &= \left|\left|V S^{-1} U^T z\right|\right|_2 && (22)\\ &= \left|\left|S^{-1} U^T z\right|\right|_2 \quad \text{because } V \text{ is orthogonal} && (23)\\ &= \sqrt{\sum_i^n \left(\frac{u_i^T z}{\sigma_i}\right)^2}. && (24) \end{aligned}$$

Note that if z has a component that is orthogonal to the left singular vectors $u_1, \ldots, u_n$ then it does not contribute to the error. Consequently, to maximize the error we can restrict ourselves to unit-norm vectors z that can be expressed as a linear combination of the left singular vectors,

$$\sum_{i=1}^n \alpha_i z[i], \quad \sum_{i=1}^n \alpha_i^2 = 1. \tag{25}$$

We have

$$\begin{aligned} ||x - x_{\mathrm{LS}}||_2 &= \sqrt{\sum_i^n \left(\frac{\alpha_i}{\sigma_i}\right)^2} && (26)\\ &\leq \frac{1}{\sigma_n}, \quad \text{where } \sigma_n \text{ is the smallest singular value.} && (27) \end{aligned}$$

The maximum is achieved by choosing z to be equal to the left singular vector u_n corresponding to the smallest singular value. The norm of the error is then equal to $1/\sigma_n$. In the case of A the maximum error is 25.3 (even though the noise has norm one!).

c.

$$\begin{aligned} B &= \begin{bmatrix} A \\ \gamma \, \mathrm{I} \end{bmatrix}, && (28)\\ c &= \begin{bmatrix} y \\ 0 \end{bmatrix}. && (29) \end{aligned}$$

B has dimensions $2m \times n$ and c has dimension $2m$.

d.

$$
\begin{aligned}
x_{\mathrm{RR}} &= \left(B^T B\right)^{-1} B^T c && (30)\\
&= \left(\begin{bmatrix} A^T & \gamma\,\mathrm{I} \end{bmatrix} \begin{bmatrix} A \\ \gamma\,\mathrm{I} \end{bmatrix}\right)^{-1} \begin{bmatrix} A^T & \gamma\,\mathrm{I} \end{bmatrix} \begin{bmatrix} Ax+z \\ 0 \end{bmatrix} && (31)\\
&= \left(A^T A + \gamma^2\,\mathrm{I}\right)^{-1} A^T \left(Ax+z\right) && (32)\\
&= \left(VS^2V^T + \gamma^2\,\mathrm{I}\right)^{-1} \left(VS^2V^T x + VSU^T z\right) && (33)\\
&= \left(VS^2V^T + \gamma^2 VV^T\right)^{-1} \left(VS^2V^T x + VSU^T z\right) && (34)\\
&= \left(V\left(S^2 + \gamma^2\,\mathrm{I}\right)V^T\right)^{-1} \left(VS^2V^T x + VSU^T z\right) && (35)\\
&= \left(V^T\right)^{-1} \left(S^2 + \gamma^2\,\mathrm{I}\right)^{-1} V^{-1} \left(VS^2V^T x + VSU^T z\right) && (36)\\
&= V \begin{bmatrix} \frac{1}{\sigma_1^2+\gamma^2} & 0 & \cdots & 0 \\ 0 & \frac{1}{\sigma_2^2+\gamma^2} & \cdots & 0 \\ & & \cdots & \\ 0 & 0 & \cdots & \frac{1}{\sigma_n^2+\gamma^2} \end{bmatrix} \left(S^2V^T x + SU^T z\right) && (37)\\
&= \sum_{i=1}^{n} \frac{\sigma_i}{\sigma_i^2+\gamma^2} \left(\sigma_i v_i^T x + u_i^T z\right) v_i. && (38)
\end{aligned}
$$

e. Using $x = \sum_{i=1}^{n} \left(v_i^T x\right) v_i$,

$$
x - x_{\mathrm{RR}} = \sum_{i=1}^{n} \frac{1}{\sigma_i^2+\gamma^2} \left(\gamma^2 v_i^T x + u_i^T z\right) v_i \tag{39}
$$

The first term is larger for larger γ and depends on the signal x, the second term is smaller for larger γ and depends on the noise. The best value of γ will be a tradeoff between controlling noise amplification due to small singular values (setting γ large enough) and not incorporating too much error proportional to the signal (setting γ not too large).

f. As shown in (b) when the smallest singular value of the matrix is very small, the noise can be amplified dramatically. By applying ridge regression this effect can be neutralized to some extent as explained in (e).