Idan Drori - 205983406

Theory Questions

1. Sparsity of Lasso estimator.

Let $\ell(a) = \frac{1}{2}\|y - Xa\|_2^2 + \lambda\|a\|_1$. Both the l_1 and l_2 norms are convex (all norms are convex), and $y - Xa$ is linear, and so ℓ is a convex function. Meaning it's sufficient to find a critical point and we can conclude that it's a minimum. We'll calculate the gradient of ℓ:

$$\nabla\ell(\boldsymbol{a}) = -X\boldsymbol{y} + X^T X\boldsymbol{a} + \lambda \cdot sign(\boldsymbol{a})$$

Where $sign(\boldsymbol{a}) = (sign(a_1), sign(a_2), ..., sign(a_d))^T$. And so:

$$\nabla\ell(\boldsymbol{a})_j = -z_j + \sigma_j a_j + \lambda \cdot sign(a_j)$$

We'll substitute $\hat{a}_j^{lasso} = \frac{sign(z_j)}{\sigma_j}\max(0, |z_j| - \lambda)$ and we get:

$$\begin{aligned}\nabla\ell(\hat{a}_j^{lasso})_j &= -z_j + \sigma_j \cdot \frac{sign(z_j)}{\sigma_j}\max(0, |z_j| - \lambda) + \lambda \cdot sign\left(\frac{sign(z_j)}{\sigma_j}\max(0, |z_j| - \lambda)\right) \\ &= -z_j + sign(z_j)\max(0, |z_j| - \lambda) + \lambda \cdot sign\left(\frac{sign(z_j)}{\sigma_j}\max(0, |z_j| - \lambda)\right)\end{aligned}$$

We'll show that $\nabla\ell(\hat{a}_j^{lasso})_j = 0$ for every $1 \le j \le d$. We first show that $|z_j| \ge \lambda$. Assume for the sake of contradiction that $|z_j| < \lambda$:

- If $\hat{a_j} > 0$ and $z_j > 0$ then because $\nabla\ell(\hat{\boldsymbol{a}})_j = -z_j + \sigma_j\hat{a}_j + \lambda = 0$, we get $\sigma_j\hat{a}_j = z_j - \lambda$, but $\sigma_j\hat{a}_j > 0$ and $z_j - \lambda < 0$ in contradiction.
- If $\hat{a}_j > 0$ and $z_j \le 0$, then $|z_j| < \lambda \Rightarrow z_j - \lambda < 0$. But once again $\sigma_j\hat{a}_j = z_j - \lambda$ and $\sigma_j\hat{a_j} > 0$, in contradiction.
- If $\hat{a}_j < 0$ and $z_j > 0$ then because $\nabla\ell(\hat{a})_j = -z_j + \sigma_j\hat{a}_j - \lambda = 0$, we get $\sigma_j\hat{a}_j = \lambda + z_j$. But $\sigma_j\hat{a}_j < 0$ and $\lambda + z_j > 0$ in contradiction.
- If $\hat{a}_j < 0$ and $z_j \le 0$ then $-z_j - \lambda < 0$ and $\sigma_j\hat{a}_j < 0$ but then $-z_j + \sigma_j\hat{a}_j - \lambda < 0$ in contradiction to the fact that $\nabla\ell(\hat{a})_j = -z_j + \sigma_j\hat{a}_j - \lambda = 0$.
- If $\hat{a}_j = 0$ then $\nabla\ell(\hat{a})_j = -z_j + \lambda = 0 \Rightarrow z_j = \lambda$ in contradiction to the assumption that $|z_j| < \lambda$.

Then we conclude that $|z_j| \ge \lambda$, and so:

$$\begin{aligned}\nabla\ell(\hat{a}_j^{lasso})_j &= -z_j + sign(z_j)\max(0, |z_j| - \lambda) + \lambda \cdot sign\left(\frac{sign(z_j)}{\sigma_j}\max(0, |z_j| - \lambda)\right) \\ &= -z_j + sign(z_j)(|z_j| - \lambda) + \lambda \cdot sign\left(\frac{sign(z_j)}{\sigma_j}(|z_j| - \lambda)\right) \\ &= -z_j + sign(z_j)(|z_j| - \lambda) + \lambda \cdot sign(z_j) \\ &= -z_j + sign(z_j)(|z_j| - \lambda + \lambda) = -z_j + sign(z_j)|z_j| = 0\end{aligned}$$

Meaning $\nabla \ell(\hat{a}_j^{lasso})_j = 0$ for every $1 \leq j \leq d$, and so $\hat{a}^{lasso} = \arg\min_{a \in \mathbb{R}^d} \left\{ \frac{1}{2}\|\boldsymbol{y} - X\boldsymbol{a}\|_2^2 + \lambda \|\boldsymbol{a}\|_1 \right\}$, as required.

2. Suboptimality of ID3.

(a) Let $\mathcal{X} = \{0,1\}^3, Y = \{0,1\}$

$$a = ((1,1,1),1), b = ((1,0,0),1), c = ((1,1,0),0), d = ((0,0,1),0)$$

We'll use the above training set to build a decision tree of depth 2. For each split we'll calculate the error gain as mutual information

$$G(S,i) = C(\mathbb{P}[Y=1]) - \mathbb{P}[X_i = 0]C(Y|X_i = 0) - \mathbb{P}[X_i = 1]C(Y|X_i = 1)$$

We'll define C as the entropy gain, so we get:

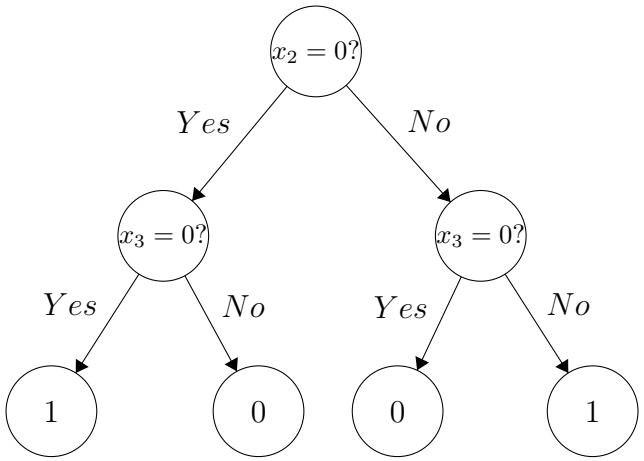
$$\begin{aligned} G(Y;X_1) &= H(Y) - \mathbb{P}[X_1 = 0]H(Y|X_1 = 0) - \mathbb{P}[X_1 = 1]H(Y|X_1 = 1) \\ &= 2(\frac{1}{2}\log\frac{1}{2}) - \frac{3}{4}H(Y|X_1 = 1) = 1 + \frac{3}{4}(\frac{2}{3}\log\frac{2}{3} + \frac{1}{3}\log\frac{1}{3}) \approx 0.31 \end{aligned}$$

$$G(Y;X_2) = 1 - \mathbb{P}[X_2 = 0]C(Y|X_2 = 0) - \mathbb{P}[X_2 = 1]C(Y|X_2 = 1) = 0$$

$$G(Y;X_3) = 1 - \mathbb{P}[X_3 = 0]C(Y|X_3 = 0) - \mathbb{P}[X_3 = 1]C(Y|X_3 = 1) = 0$$

So the first split will be on X_1 and a, b, c will be on the same branch. If we then ask about X_2, a, c will go on the same leaf, but if we ask about X_3 then b, c will go on the same leaf. It follows that regardless of which split ID3 will choose, we'll get at least one wrong classification and the training error will be at least $\frac{1}{4}$.

(b) Tree of depth 2 that achieves 0 training error:



2. Step-size Perceptron.

Let w^* be a separator for the data, where $\|w^*\| = 1$. We'll first show that if a mistake was made at iteration t then $w_{t+1} \cdot w^* \geq w_t \cdot w^* + \eta_t \gamma$:
We suppose x_t is positive then $y_t = 1$, then we get:

$$w_{t+1} \cdot w^* = (w_t + \eta_t x_t) \cdot w^* = w_t \cdot w^* + \eta_t \cdot (x_t \cdot w^*)$$

Because $x_t \cdot w^* \geq \gamma$, then:

$$w_{t+1} \cdot w^* = w_t \cdot w^* + \eta_t \cdot (x_t \cdot w^*) \geq w_t \cdot w^* + \eta_t \gamma$$

Similarly for $y_t = -1$.
We'll show that if a mistake was made at iteration t then $\|w_{t+1}\|^2 \leq \|w_t\|^2 + \eta_t^2$:
We suppose x_t is positive then $y_t = 1$, then we get:

$$\begin{aligned}\|w_{t+1}\|^2 = \|w_t + x_t\|^2 &= (w_t + \eta_t x_t)^T (w_t + \eta_t x_t) \\ &= \|w_t\|^2 + 2\eta_t \underbrace{(w_t \cdot x_t)}_{\leq 0} + \eta_t^2 \|x_t\|^2 \leq \|w_t\|^2 + \eta_t \|x\|^2 = \|w_t\|^2 + \eta_t^2\end{aligned}$$

Similarly for $y_t = -1$. From the first claim, after M mistakes we get:

$$w_M \cdot w^* \geq \gamma \sum_{t=1}^{M} (\frac{1}{\sqrt{t}})$$

From the second claim, after M mistakes we get:

$$\|w_M\| \leq \sqrt{\sum_{t=1}^{M} (\frac{1}{t})}$$

For large enough M we get:

$$\sum_{t=1}^{M} (\frac{1}{t}) \leq \ln(M+1) \leq \ln(M) + 1$$

And so:

$$\sqrt{\sum_{t=1}^{M} (\frac{1}{t})} \leq \sqrt{\ln(M) + 1} \leq \sqrt{2\ln(M)}$$

Then:

$$w_M \cdot w^* \geq \gamma \sum_{t=1}^{M} (\frac{1}{\sqrt{t}}) \geq \gamma \sqrt{M}$$

And from that:

$$\|w_M\| \leq \sqrt{\sum_{t=1}^{M} (\frac{1}{t})} \leq \sqrt{2\ln(M)}$$

From Cauchy-Schwartz we get:

$$\gamma\sqrt{M} \leq w_M \cdot w^* \leq \|w_M\| \cdot \|w^*\| = \|w_M\| \leq \sqrt{2\ln(M)}$$

And so $M \leq \frac{1}{\gamma^2} 2\ln(M)$, and using the hint:

$$M \leq \frac{4}{\gamma^2}\ln(\frac{2}{\gamma^2}) \leq \frac{4}{\gamma^2}\ln(\frac{4}{\gamma^2}) = \frac{4}{\gamma^2}\ln((\frac{2}{\gamma})^2) = \frac{8}{\gamma^2}\ln(\frac{2}{\gamma}) \Rightarrow M = O(\frac{1}{\gamma^2}\ln(\frac{1}{\gamma}))$$

4. Kernel k-means.

We assume that the initial centroids $\mu_1^{(1)}, ..., \mu_k^{(1)}$ are chosen as distinct data points $\phi(x_{c_1}), ..., \phi(x_{c_k})$ for some indices $c_1, ..., c_k \in \{1, ..., n\}$. We first compute the initial kernel matrix K where $K_{i,j} = \phi(x_i) \cdot \phi(x_j)$.

Assignment Step: As the algorithm runs, each centroid μ_j is the average of the points assigned to it, meaning $\mu_j = \frac{1}{|C_j|}\sum_{i \in C_j} \phi(x_i)$, where C_j is the set of indices of the points assigned to centroid j. The distance between $\phi(x_i)$ and centroid μ_j can be computed using the kernel matrix:

$$\begin{aligned}
\|\phi(x_i) - \mu_j\|^2 &= \phi(x_i)^2 - 2\phi(x_i) \cdot \mu_j + \mu_j^2 \\
&= K_{ii} - 2\phi(x_i) \cdot \left(\frac{1}{|C_j|}\sum_{l \in C_j} \phi(x_l)\right) + \frac{1}{|C_j|^2}\sum_{l \in C_j}\sum_{m \in C_j} K_{lm} \\
&= K_{ii} - 2 \cdot \left(\frac{1}{|C_j|}\sum_{l \in C_j} K_{il}\right) + \frac{1}{|C_j|^2}\sum_{l \in C_j}\sum_{m \in C_j} K_{lm}
\end{aligned}$$

Because the kernel values K_{ij} can be computed efficiently, the distance calculation doesn't depend on d'. Let $O(K)$ be the time complexity of computing K_{ij} for some indices i, j. Then the time complexity of this step for some μ_j would be:

$$O(K) + O(|C_j| \cdot O(K)) + O(|C_j|^2 \cdot O(K)) = O(|C_j|^2 \cdot O(K)) = O(n^2 \cdot O(K))$$

Re-Estimation Step: We update the centroids by updating the sets C_j, this way we don't need to make computations in the high-dimensional space. Similarly to the assignment step, the time complexity for this step would be:

$$O(n^2 \cdot O(K))$$

5. Maximum Likelihood.

(a) We want to maximize

$$f(\lambda) = \sum_{i=1}^{n} \log(\lambda e^{-\lambda x_i}) = n\log\lambda - \lambda\sum_{i=1}^{n} x_i$$

Computing the derivative:

$$f'(\lambda) = \frac{n}{\lambda} - \sum_{i=1}^{n} x_i$$

And the second derivative:

$$f''(\lambda) = -\frac{n}{\lambda^2} < 0$$

The second derivative is negative, and so for some λ^* such that $f'(\lambda^*) = 0$:

$$\arg\max_{\lambda} f(\lambda) = \lambda^*$$

So λ^* will be the ML estimate, finding its value:

$$\frac{n}{\lambda^*} - \sum_{i=1}^{n} x_i = 0 \Rightarrow \lambda^* = \frac{n}{\sum_{i=1}^{n} x_i}$$

(b) Let Y be the previous distribution. $Y \sim exp(1)$, so it follows that:

$$P(Y = \lambda | X_1 = x_1, ..., X_n = x_n) \propto P(Y = \lambda) \prod_i P(X_i = x_i | Y = \lambda) = e^{-\lambda} \prod_i \lambda e^{-\lambda x_i}$$
$$= \lambda^n exp(-\lambda(\sum_i x_i) - \lambda)$$

We want to find the λ that maximizes the above probability. Because $\lambda > 0$, and because the log function is rising monotonic, we want to find:

$$\arg\max_{\lambda} \left(\log \left(\lambda^n exp \left(-\lambda \left(\sum_i x_i \right) - \lambda \right) \right) \right) = \arg\max_{\lambda} \left(n \log \lambda - \lambda \left(\left(\sum_i x_i \right) + 1 \right) \right)$$

The above function is convex as a sum of concave function, and so if we find a critical point we can conclude that it'll be a maximal point. We'll derive the function and equate to 0:

$$\frac{n}{\lambda} - \left(\sum_i x_i \right) - 1 = 0 \Rightarrow \lambda = \frac{n}{\sum_i x_i + 1}$$

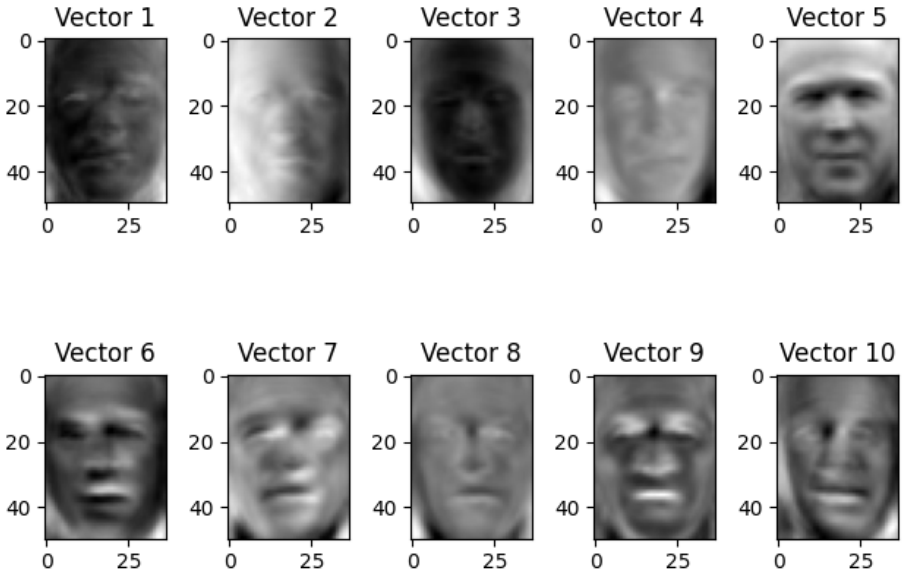
In summary, the MAP value of λ is $\lambda = \frac{n}{\sum_i x_i + 1}$

Programming Assignment

1. PCA

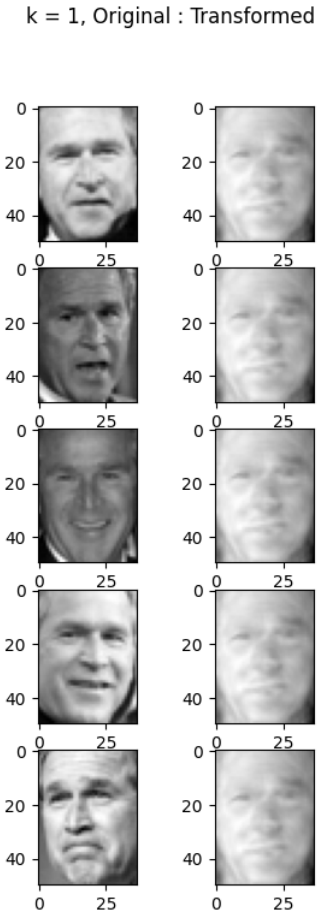
(a) Implemented in code file.

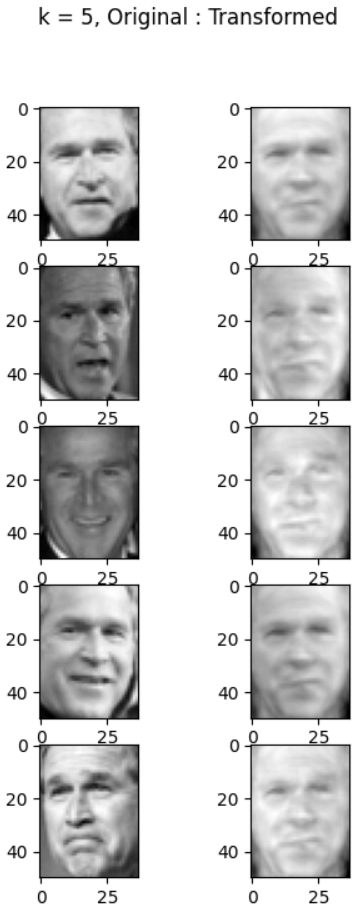
(b) I selected George W Bush:



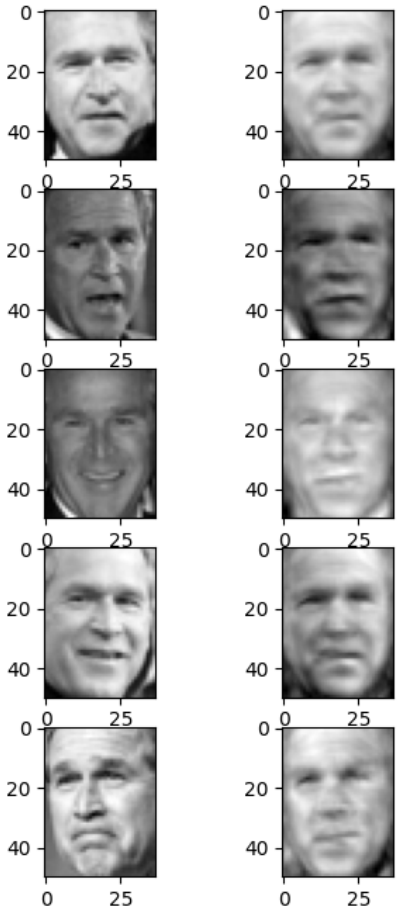
The above vectors represent "the most significant components" of the image data, a linear combination of these vectors best represents all the images.

(c) Plots for $k \in \{1, 5, 10, 30, 50, 100\}$:

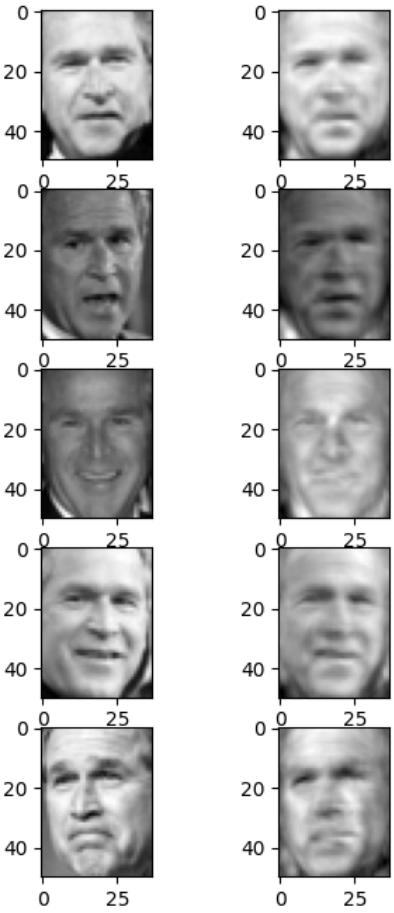




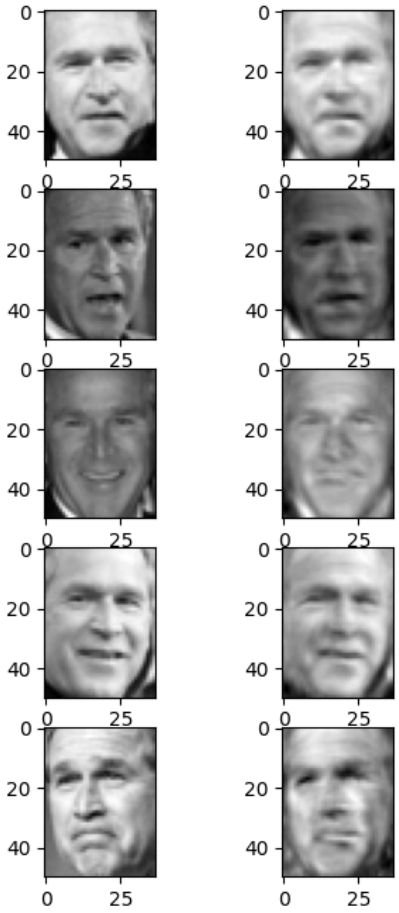
k = 10, Original : Transformed



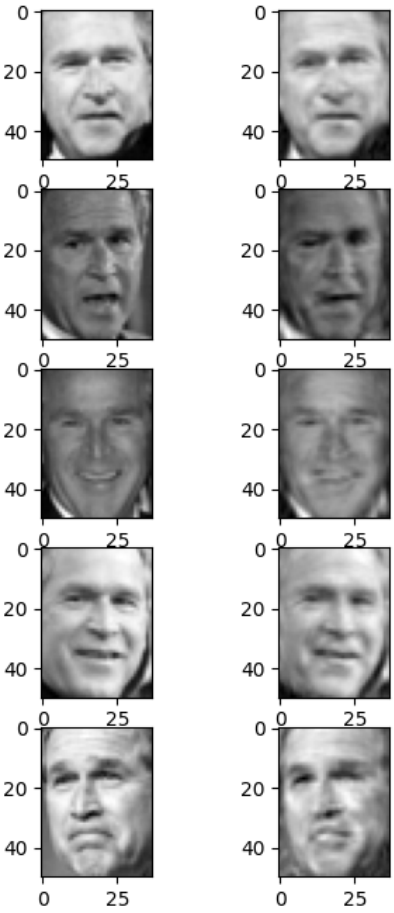
k = 30, Original : Transformed



k = 50, Original : Transformed



k = 100, Original : Transformed



We can see that as k increases, the difference between the original image and the transformed image decreases, meaning the pictures become more and more similar. This is expected, because as we increase the number of principal components, we project the components on a larger subspace, and so become more similar to the original data. By plotting the l_2-distances as a function of k we can see this more clearly: