Unsupervised Learning and Data Analysis
Homework 1

Question 1 - MLE Normal Distribution

Let A be a matrix, let x be a vector, and $|A|$ mean the determinant of a matrix. Let's denote

$$\nabla_x Ax = A^T \tag{1}$$

$$\nabla_x x^T A = A \tag{2}$$

$$\nabla_x x^T Ax = (A + A^T x \tag{3}$$

$$\frac{\partial}{\partial A} \ln |A| = A^{-T} \tag{4}$$

$$\frac{\partial}{\partial A} \operatorname{Tr}[AB] = B^T \tag{5}$$

$$\operatorname{Tr}[A^T B] = (\operatorname{vec} A)^T \operatorname{vec} B \tag{6}$$

Let $X_1, \ldots, X_N \sim \mathcal{N}(\mu, \Sigma)$ i.i.d, with $\mu \in \mathbb{R}^d, \Sigma \in \mathbb{R}^{d \times d}$. **Assume** Σ is positive definite.

The PDF of a d-multivariate normal random variable $X \sim \mathcal{N}(\mu, \Sigma)$ is:

$$f(x; \mu, \Sigma) = (2\pi)^{-d/2} |\Sigma|^{-1/2} \exp\big(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\big)$$

The likelihood function $L(\mu, \Sigma, x_1, \ldots x_d)$ of N iid RV is the product of their PDFs:

$$L = (2\pi)^{-Nd/2} |\Sigma|^{-N/2} \exp\left(-\frac{1}{2}\sum_{j=1}^{N}(x_j-\mu)^T \Sigma^{-1}(x_j-\mu)\right)$$

And therefore the log-likelihood is:

$$\begin{aligned}
\ell &= -\frac{Nd}{2}\ln(2\pi) - \frac{N}{2}\ln|\Sigma| - \frac{1}{2}\sum_{j=1}^{N}(x_j-\mu)^T\Sigma^{-1}(x_j-\mu) \\
&= -\frac{Nd}{2}\ln(2\pi) + \frac{N}{2}\ln|\Sigma^{-1}| - \frac{1}{2}\sum_{j=1}^{N}(x_j-\mu)^T\Sigma^{-1}(x_j-\mu)
\end{aligned}$$

Mean: We take the gradient wrt to μ:

$$\begin{aligned}
\nabla_\mu \ell &= -\frac{1}{2}\sum_{j=1}^{N}\nabla_\mu (x_j-\mu)^T\Sigma^{-1}(x_j-\mu) \\
&= -\frac{1}{2}\sum_{j=1}^{N}\nabla_\mu\left(x_j^T\Sigma^{-1}x_j - x_j^T\Sigma^{-1}\mu - \mu^T\Sigma^{-1}x_j + \mu^T\Sigma^{-1}\mu\right) \\
&= -\frac{1}{2}\sum_{j=1}^{N} -\Sigma^{-1}x_j - \Sigma^{-1}x_j + 2\Sigma^{-1}\mu \\
&= \Sigma^{-1}\sum_{j=1}^{N}(x_j-\mu)
\end{aligned}$$

where we've used properties Eq. (1), Eq. (2) and Eq. (3), and also the fact that $\Sigma = \Sigma^T$. We set $\nabla_\mu \ell = 0$, and get:

$$\sum_{j=1}^{N}(x_j-\mu) = 0$$

where we've used the fact that Σ is positive definite. Therefore we get:

$$\widehat{\mu} = \frac{1}{N}\sum_{j=1}^{N} x_j$$

Covariance: We take the gradient wrt to Σ^{-1}:

$$\begin{aligned}\nabla_{\Sigma^{-1}}\ell &= \frac{N}{2}\nabla_{\Sigma^{-1}}\ln|\Sigma^{-1}| - \frac{1}{2}\nabla_{\Sigma^{-1}}\sum_{j=1}^{N}\mathrm{Tr}\left[(x_j-\mu)^T\Sigma^{-1}(x_j-\mu)\right]\\ &= \frac{N}{2}\Sigma^T - \frac{1}{2}\nabla_{\Sigma^{-1}}\sum_{j=1}^{N}\mathrm{Tr}\left[(x_j-\mu)(x_j-\mu)^T\Sigma^{-1}\right]\\ &= \frac{N}{2}\Sigma^T - \frac{1}{2}\nabla_{\Sigma^{-1}}\mathrm{Tr}\left[\left(\sum_{j=1}^{N}(x_j-\mu)(x_j-\mu)^T\right)\Sigma^{-1}\right]\\ &= \frac{N}{2}\Sigma^T - \frac{1}{2}\left(\sum_{j=1}^{N}(x_j-\mu)(x_j-\mu)^T\right)^T\end{aligned}$$

we set to zero and transpose, and we get:

$$\begin{aligned}N\widehat{\Sigma} &= \sum_{j=1}^{N}(x_j-\mu)(x_j-\mu)^T\\ \Rightarrow \widehat{\Sigma} &= \frac{1}{N}\sum_{j=1}^{N}(x_j-\mu)(x_j-\mu)^T\end{aligned}$$

Question 4 - PCA

1) **Claim**. (Courant-Fischer): Let $S \in \mathbb{R}^{m\times n}$ be a collection of m samples $(s_1,\ldots,s_m)$, each $s_i \in \mathbb{R}^n$. Assume the samples are centered, i.e. $\mathbf{1}^T S = \mathbf{0}^T$. Let $C = S^T S$, and v_1 be the largest eigenvalue of C. Then v_1 represents the direction of the largest variance of S.

Proof. For a vector $u \in \mathbb{R}^n$,

$$\mathrm{Var}(Su) \equiv \frac{1}{m-1}\sum_{i=1}^{m}(s_i^T u - \mu)^2$$

where

$$\mu = \sum_{i=1}^{m} s_i^T u = \mathbf{1}^T S u$$

Since S is centered, we get that $\mu = 0$, and therefore:

$$\mathrm{Var}(Su) = \frac{1}{m-1}\sum_{i=1}^{m}(s_i^T u)^2 = \frac{1}{m-1}\|Su\|^2 = \frac{1}{m-1}u^T C u$$

Therefore, the unit vector that will maximize $\mathrm{Var}(Su)$ is:

$$\arg\max_{\|v\|^2=1} u^T C u$$

Since C is PSD, it has an orthonormal eigenbasis $\{v_i\}_{i=1}^n$, and we can write:

$$C = \sum_{i=1}^{n}\lambda_i v_i v_i^T$$

since C is PSD, we have that $\lambda_i \geq 0$ for all i. Let $u = \sum_{i=1}^n a_i v_i$ be a unit vector (such a representation exists and is unique because the eigenvectors form a basis). Then:

$$\begin{aligned}
u^TCu &= u^T\big(C\sum_{i=1}^n a_iv_i\big) = u^T\big(\sum_{j=1}^n \lambda_jv_jv_j^T(\sum_{i=1}^n a_iv_i)\big) \\
&= u^T\big(\sum_{j=1}^n \lambda_jv_j\langle v_j,\sum_{i=1}^n a_iv_i\rangle\big) = u^T\big(\sum_{j=1}^n \lambda_ja_jv_j\big) && (v_i^Tv_j=\delta_{ij}) \\
&= (\sum_{i=1}^n a_iv_i)^T\big(\sum_{j=1}^n a_j\lambda_jv_j\big) = \sum_{i=1}^n a_i\langle v_i,\sum_{j=1}^n a_j\lambda_jv_j\rangle \\
&= \sum_{i=1}^n a_ia_i\lambda_i && (\ v_i^Tv_j=\delta_{ij}) \\
&\leq \sum_{i=1}^n a_i^2\lambda_1 && (\lambda_1 \text{ is largest}) \\
&= \lambda_1\|u\|^2 = \lambda_1 && (\ \|u\|^2=1)
\end{aligned}$$

so for every $u \in \mathbb{R}^n$, $u^TCu \leq \lambda_1$ and in particular:

$$v_1^TCv_1 = v_1^T\lambda_1v_1 = \lambda_1\|v_1\|^2 = \lambda_1$$

i.e. the maximum is achieved by v_1, hence v_1 maximizes $\text{Var}(Su)$.

□

2) **Claim**. Let $X \in \mathbb{R}^{d\times n}$, with n samples $\{x_i\}_{i=1}^n$ and d features. Define $W := XX^T$ with orthonormal eigenvectors $u_1,\dots u_d$ and $\lambda_1,\dots\lambda_d$. Let $P_{u_i}:\mathbb{R}^d\to\mathbb{R}^d$ be the projection operator onto span$\{u_i\}$. Then:

$$\sum_{j=1}^n\|P_{u_i}x_j\|_2^2 = \lambda_i$$

Proof. The eigenvectors are orthogonal, hence: $u_i^Tu_j=\delta_{ij}$. Now:

$$\begin{aligned}
\sum_{j=1}^n \|P_{u_i}x_j\|^2 &= \sum_{j=1}^n \|u_iu_i^Tx_j\|^2 \\
&= \sum_{j=1}^n \langle u_iu_i^Tx_j, u_iu_i^Tx_j\rangle \\
&= \sum_{j=1}^n (x_j^Tu_iu_i^T)(u_iu_i^Tx_j) \\
&= \sum_{j=1}^n (x_j^Tu_i)(u_i^Tx_j) \\
&= \sum_{j=1}^n (u_i^Tx_j)^2
\end{aligned}$$

Additionally, note that:

$$\|u_i^TX\|^2 = \|\sum_{j=1}^n u_i^Tx_je_j\|^2 \overset{\text{Pythagoras}}{=} \sum_{j=1}^n(u_i^Tx_j)^2\|e_j\|^2 = \sum_{j=1}^n(u_i^Tx_j)^2$$

Therefore:

$$\sum_{j=1}^n \|P_{u_i}x_j\|^2 = \|u_i^TX\|^2 \tag{7}$$

via SVD, we have that $X = U\Sigma V^T$, with . In particular, $W = U\Lambda U^T$, and also

$$W = XX^T = U\Sigma\Sigma^TU^T$$

Therefore $\sigma_i^2 = \lambda_i^2$.
Additionally

$$u_i^T X = u_i^T U \Sigma V^T = e_i^T \Sigma V^T = \left[0, \dots, \overbrace{\sigma_i}^{i-th}, \dots 0\right] V^T = \sigma_i v_i^T$$

And therefore:

$$\begin{aligned} \|u_i^T X\|^2 &= \sigma_i^2 \|v_i\|^2 \\ &= \sigma_i^2 \qquad (V \text{ orthonormal}) \\ &= \lambda_i^2 \end{aligned} \tag{8}$$

Concluding,

$$\begin{aligned} \sum_{j=1}^{n} \|P_{u_i} x_j\|^2 &= \|u_i^T X\|^2 && \text{(Eq. 7)} \\ &= \lambda_i^2 && \text{(Eq. 8)} \end{aligned}$$

□

Question 5

Question 5.1

Let x be an $n \times 1$ random vector, and y be an $m \times 1$ random vector. We wish to find linear combinations $a^T x, b^T y$ of the random variables that will maximize their correlation.

We assume wlog that $\mu_x = \mu_y = \mathbf{0}$. Under these circumstance, the correlation between $a^T x$ and $b^T y$ is:

$$\begin{aligned} Corr[a^T x, b^T y] &= \frac{\text{Cov}[a^T x, b^T y]}{\sqrt{V[a^T x] V[b^T y]}} = \frac{a^T \, \text{Cov}[x, y] b}{\sqrt{E\left[(a^T x)^2\right] E\left[(b^T y)^2\right]}} \\ &= \frac{a^T \left(E[xy^T] - \mu_x \mu_y^T\right) b}{\sqrt{E\left[(a^T x)^2\right] E\left[(b^T y)^2\right]}} = \frac{a^T \left(E[xy^T]\right) b}{\sqrt{E\left[(a^T x)^2\right] E\left[(b^T y)^2\right]}} \\ &= \frac{E\left[(a^T x)(b^T y)\right]}{\sqrt{E\left[(a^T x)^2\right] E\left[(b^T y)^2\right]}} \end{aligned}$$

We notice that in the above expression, scaling a or b cancels out, hence we may assume that their scaling is such that $E\left[(a^T x)^2\right] = 1$ and $E\left[(b^T x)^2\right] = 1$. We put these as constraints. Then, the task of maximizing the correlation becomes:

$$\max_{a,b} \mathbb{E}\left[(a^T x)(b^T y)\right] \quad \text{s.t. } \mathbb{E}\left[(a^T x)^2\right] = 1, \mathbb{E}\left[(b^T y)^2\right] = 1$$

By choosing the linear combinations that have the highest correlation, we reduce the number of variables while retaining as much information as possible, just like PCA. Whereas PCA deals with random variables that come from a single set, CCA assumes the variables come from **two** sets. Just like PCA, we select an uncorrelated linear combinations $a^T x$ and $b^T y$, which are pairwise highly correlated. We can continue to do so, finding more and more principal directions, that are highly-correlated between the sets, and are uncorrelated to previously selected principal directions.

As shown in Question 2, the CCA optimization problem becomes the task of maximizing a bilinear form. In PCA, we maximize a quadratic form, which is a symmetric bilinear form.

Question 5.2: Comparing Statistical and Geometrical PCA

TL:DR skip over Statistical POV and Geometric POV, I wrote them just to introduce notation. Go to Difference in Optimization.

Statistical POV: We treat $\boldsymbol{x} \in \mathbb{R}^D$ as a random vector, aka a multivariate random variable. We assume features are scaled, i.e. $E[x_i] = 0$ and $\text{Var}[x_i] = 1$. Generally, $E[x_i x_j] \neq 0$, therefore the features are correlated. We wish to find a representation $\boldsymbol{y} \in \mathbb{R}^d$, $d << D$, given as a linear combination of $\boldsymbol{x}$, i.e. $\boldsymbol{y} = A\boldsymbol{x}, A \in \mathbb{R}^{d \times D}$, such that the features are uncorrelated, i.e.

$$Cov[y_i, y_j] = E[y_i y_j] - E[y_i]E[y_j] = 0$$

and since $E[y_i] = E[y_j] = 0$, this means we look for $E[y_i y_j] = 0$. In addition, we look for the most "meaningful" uncorrelated directions - those directions along which $\boldsymbol{x}$ varies the most. How do we capture this? Recall that with the random matrix:

$$P_x = \boldsymbol{x}\boldsymbol{x}^T$$

and $Cov[\boldsymbol{x}] = E[\boldsymbol{x}\boldsymbol{x}^T]$ we can measure the variance of the projection of $\boldsymbol{x}$ along a vector $\boldsymbol{u}$ by:

$$Var[\boldsymbol{u}^T\boldsymbol{x}] = \boldsymbol{u}^T E[\boldsymbol{x}\boldsymbol{x}^T]\boldsymbol{u} = E[\boldsymbol{u}^T\boldsymbol{x}\boldsymbol{x}^T\boldsymbol{u}] = E[(\boldsymbol{u}^T\boldsymbol{x})^2]$$

This gives the objective function:

$$\boldsymbol{u}_i^\star = \arg\max_{\boldsymbol{u}_i} E[(\boldsymbol{u}_i^T\boldsymbol{x})^2]$$
$$\text{s.t. } \boldsymbol{u}_i^T\boldsymbol{u}_j = \delta_{ij} \text{ for } j \leq i$$

Since $Var[\boldsymbol{u}^T\boldsymbol{x}] = \boldsymbol{u}^T P_x \boldsymbol{u}$, we have from Courant-Fischer that the largest eigenvectors of P_x are those that maximize it.

Geometrical POV: Let $X \in \mathbb{R}^{D\times n}$, where D is number of features and n is number of samples. Assume the data is centered, there are no redundant features, and $n \geq D$, i.e. $\text{rank}(X) = D$. Denote by $Y \in \mathbb{R}^{d\times n}$ an encoding of X, and denote by $\widetilde{X} \in \mathbb{R}^{D\times n}$ a reconstruction of X using Y.

MSE Objective: We wish, using linear operators, to find $\widetilde{X}$ s.t.:

$$\min_{\widetilde{X}} \|X - \widetilde{X}\|_F^2$$

Since both the encoding and decoding is done via linear operators, we can write:

$$\min_{\widetilde{X}} \|X - \widetilde{X}\|_F^2 \quad \text{s.t.} \quad \widetilde{X} = U_2 Y = U_2 U_1 X \quad U_1 \in \mathbb{R}^{d\times D}, U_2 \in \mathbb{R}^{D\times d}$$

Solution: PCA. Let $\widehat{P} = \frac{1}{n} X X^T$ be the covariance matrix, and let U be the first d largest eigenvectors of $\widehat{P}$. Then $U_1 = U^T, U_2 = U$.

Difference in Optimization: For Geometrical PCA, the objective function is Minimal MSE:

$$\min_{\widetilde{X}} \|X - \widetilde{X}\|_F^2$$
$$\text{s.t. } \widetilde{X} = U_2 U_1 X$$

whereas for Statistical PCA, the objective function is Variance Maximization:

$$\boldsymbol{u}_i^\star = \arg\max_{\boldsymbol{u}_i} E[(\boldsymbol{u}_i^T\boldsymbol{x})^2]$$
$$\text{s.t. } \boldsymbol{u}_i^T\boldsymbol{u}_j = \delta_{ij} \text{ for } j \leq i$$

this leads to the following differences:

- Goal: Statistical PCA aims to maximize variance, whereas Geometrical PCA aims to minimize reconstruction error. Each one achieves also the other's goals, but implicitly.
- Constraints and Uniqueness: Statistical MSE requires uncorrelated transformed features, whereas Geometrical PCA places no constraints (besides linearity, like Statistical PCA) on the transformed features. as we explain below, this causes Geometrical PCA to have redundant solutions, that Statistical PCA does not have.

Redundancy in the MSE solution: Geometric PCA is not-unique, statistical PCA is* (up to distinct eigenvalues and multiplication by -1). Notice that for every orthogonal matrix $R \in \mathbb{R}^{d\times d}$, we have that $U_1 = (UR)^T, U_2 = (UR)$ is also a minimizer. Denote $\widetilde{U} = UR$.

For example, if our data really only has entries in the first two entries, then we can encode it to $X - Y$ plane. However every rotation following this projection would also serve, provided the decoder also includes the inverse rotation.

I.e. the encoding Y isn't unique. However, statistical PCA is unique - how can this be? The answer is that in statistical PCA, we add a requirement that there is no correlation between the features (i.e. they are orthogonal), i.e. every row (feature) of Y is uncorrelated with every other row. f $Y = U^T X \in \mathbb{R}^{d\times n}$, then we have that $E[YY^T] = \Lambda = \text{diag}(\lambda_1, \dots, \lambda_d)$, i.e. the encodings are not correlated. However, for $\widetilde{Y} = \widetilde{U}^T X = (UR)^T X$, we have that:

$$E[\widetilde{Y}\widetilde{Y}^T] = R^T \Lambda R$$

i.e. there can be non-zero off-diagonal elements, and hence the features can be correlated, unlike (statistical) PCA.

Therefore, there are infinite solutions in the encoder-decoder framework, which are "redundant".

Question 5.3

Let $X \in \mathbb{R}^{d\times n}$, with n samples. Assume the data is centered, that is

$$X\mathbf{1}_n = \mathbf{0}_n$$

Let $k(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^t \boldsymbol{y}$. Define the kernel matrix (Gram matrix) $K \in \mathbb{R}^{n\times n}$:

$$K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^T \boldsymbol{x}_j$$

Therefore:

$$K = X^T X$$

We center the kernel matrix:

$$K_c := K - \frac{1}{n}\mathbf{1}_n^T K - \frac{1}{n} K \mathbf{1}_n + \frac{1}{n^2}\mathbf{1}_n^T K \mathbf{1}_n$$

and since the data is centered and $K = X^T X$, we get that $K_c = K = X^T X$.

Define the covariance matrix:

$$C = \frac{1}{n} X X^T \in \mathbb{R}^{d\times d}$$

Recalling the Kernel PCA algorithm, we now solve for eigenvectors $\boldsymbol{v}_j$ of K_c, with eigenvalues of $n\lambda_j$, i.e.:

$$K_c \boldsymbol{v}_j = n\lambda_j \boldsymbol{v}_j$$

We multiply by X and get:

$$\begin{aligned} XX^T(X\boldsymbol{v}_j) &= n\lambda_j (X\boldsymbol{v}_j) \\ \Rightarrow nC(\boldsymbol{u}_j) &= n\lambda_j \boldsymbol{u}_j \\ \Rightarrow C\boldsymbol{u}_j &= \lambda_j \boldsymbol{u}_j \end{aligned}$$

I.e. for $\boldsymbol{u}_j := X\boldsymbol{v}_j$, we have that $\boldsymbol{u}_j$ is an eigenvalue of C with eigenvalue λ_j.

We've shown that for Kernel PCA, the embedding $\widetilde{\boldsymbol{y}}_i$ of sample $\boldsymbol{x}_i$ is given by:

$$\begin{aligned} \widetilde{\boldsymbol{y}}_i^{(\ell)} &= \sum_{j=1}^{n} \boldsymbol{v}_\ell^{(j)} \overline{k}(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ &= \sum_{j=1}^{n} \boldsymbol{v}_\ell^{(j)} \boldsymbol{x}_i^T \boldsymbol{x}_j \\ &= \boldsymbol{x}_i^T \sum_{j=1}^{n} \boldsymbol{v}_\ell^{(j)} \boldsymbol{x}_j \\ &= \boldsymbol{x}_i^T X \boldsymbol{v}_\ell \end{aligned}$$

Recall that for linear PCA, the embedding y_i for sample x_i is given by:

$$\begin{aligned} \boldsymbol{y}_i^{(\ell)} &= \boldsymbol{x}_i^T \boldsymbol{u}_\ell \\ &= \boldsymbol{x}_i^T X \boldsymbol{v}_\ell \end{aligned}$$

And therefore we get that:

$$\widetilde{\boldsymbol{y}}_i = \boldsymbol{y}_i$$

i.e. kernel PCA with a linear kernel is exactly equivalent to linear PCA.