Principal component analysis

One very important application of the singular value decomposition of a matrix is a **principal component analysis (PCA)**, which is used to identify patterns in data. This can be efficiently used to reduce data to lower dimensional spaces (and store only these) or interpret the relationships previously not noticed.

Suppose we are given n data points $X_1, \ldots, X_n \in \mathbb{R}^d$, viewed as rows of a $n \times d$ matrix X. Each entry $x_{i,j}$ of $X_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,d})$ represents the value of some *feature* of X_i, i.e., if X_i represents a person, then $x_{i,j}$'s can represent his/her year of birth, the height, blood sugar level, blood presure, etc. The columns $C_1, \ldots, C_d$ of X are also called **feature vectors**.

Basic idea of PCA: *Determine the vectors $Y^{(1)}, \ldots, Y^{(d)} \in \mathbb{R}^n$, called **principal components (PCs)**, which are uncorrelated projections of centered data points $X_1, \ldots, X_n$ onto some unit vectors $v^{(1)}, \ldots, v^{(d)} \in \mathbb{R}^d$ such that the variances* $\operatorname{var}(Y^{(1)}), \ldots, \operatorname{var}(Y^{(d)})$ *are maximized.*

Algorithm for the computation of PCs of X:

1. **Centralization of data:**

   For each column C_j compute its mean value $\mu_j := \frac{1}{n}\sum_{i=1}^{n} x_{i,j}$ and subtract the **centroid** $\mu := (\mu_1, \mu_2, \ldots, \mu_d)$ from each row of X:

   $$X - \mathbf{1}_{n,d} \operatorname{diag}(\mu) = [x_{i,j} - \mu_j]_{i,j},$$

   where $\mathbf{1}_{n,d}$ stands for the $n \times d$ matrix with all entries equal to 1 and $\operatorname{diag}(\mu)$ is a diagonal matrix with j-th diagonal entry μ_j.

2. **Computation of the singular value decomposition (SVD) of $X - \mathbf{1}_{n,d} \operatorname{diag}(\mu)$:**

   Let $X - \mathbf{1}_{n,d} \operatorname{diag}(\mu) = UDV^T$ be the SVD of $X - \mathbf{1}_{n,d} \operatorname{diag}(\mu)$, where $U \in \mathbb{R}^{n \times n}$, $V \in \mathbb{R}^{d \times d}$ are orthogonal matrices and $D \in \mathbb{R}^{n \times d}$ is a diagonal matrix with the singular values $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_d \geq 0$ in decreasing order on the main diagonal.

3. **Computation of the PCs of X:**

   The PCs of X are points $Y^{(1)}, \ldots, Y^{(d)} \in \mathbb{R}^n$ obtained by

   $$Y^{(k)} = (X - \mathbf{1}_{n,d} \operatorname{diag}(\mu))v^{(k)} = \sigma_k u^{(k)}, \qquad k = 1, \ldots, d,$$

   where $v^{(k)}$ and $u^{(k)}$ are the k–th columns of V and U, respectively. The vectors $v^{(k)}$ and $u^{(k)}$ are called **right** (resp. **left**) **principal directions**.

Justification of the algorithm

Recall that the projection $\mathrm{pr}_v(x)$ of the vector $x \in \mathbb{R}^d$ onto the line ℓ_v spaned by a *unit* vector $v = (v_1, \ldots, v_d) \in \mathbb{R}^d$ is equal to $(x^T v)v$. So the column vector

$$Y := Xv = [X_i v]_i = v_1 C_1 + v_2 C_2 + \ldots + v_d C_d \in \mathbb{R}^n$$

consists of the lengths of projections of row vectors of X onto ℓ_v. Computing the mean $\overline{Y}$ of the vector Y we get

$$\overline{Y} = \overline{\sum_{j=1}^{n} v_j C_j} = \sum_{j=1}^{n} v_j \overline{C_j} = \sum_{j=1}^{n} v_j \mu_j \in \mathbb{R}.$$

So the centered vector $Y - \overline{Y} \cdot \mathbf{1}_{n,1}$ of Y is equal to

$$Y - \overline{Y} \cdot \mathbf{1}_{n,1} = v_1(C_1 - \mu_1) + \ldots + v_d(C_d - \mu_d) = (X - \mathbf{1}_{n,d}\,\mathrm{diag}(\mu))v. \tag{1}$$

Moreover, if $Y = Xv$ and $Z = Xw$ for some $v, w \in \mathbb{R}^d$, then their *sample covariance* is equal to

$$\begin{aligned} \mathrm{cov}(Y,Z) &= \frac{1}{n-1}(Y - \overline{Y} \cdot \mathbf{1}_{n,1})^T (Z - \overline{Z} \cdot \mathbf{1}_{n,1}) \\ &=^{(1)} \frac{1}{n-1} v^T (X - \mathbf{1}_{n,d}\,\mathrm{diag}(\mu))^T (X - \mathbf{1}_{n,d}\,\mathrm{diag}(\mu))w \\ &=^{(3)} v^T \Sigma w, \end{aligned} \tag{2}$$

where Σ stands for the *covariance matrix*:

$$\Sigma := \frac{1}{n-1}(X - \mathbf{1}_{n,d}\,\mathrm{diag}(\mu))^T (X - \mathbf{1}_{n,d}\,\mathrm{diag}(\mu)) = [\mathrm{cov}(C_i, C_j)]_{i,j}, \tag{3}$$

and $\mathrm{cov}(C_i, C_j)$ stands for the sample covariance of the columns C_i and C_j, i.e.,

$$\begin{aligned} \mathrm{cov}(C_i, C_j) &= \frac{1}{n-1}(C_i - \mu_i \cdot \mathbf{1}_{n,1})^T (C_j - \mu_j \cdot \mathbf{1}_{n,1}) \\ &= \frac{1}{n-1} \sum_{k=1}^{n} (x_{k,i} - \mu_i)(x_{k,j} - \mu_j), \end{aligned} \tag{4}$$

where $\mathbf{1}_{n,1} \in \mathbb{R}^n$ is a column of 1s. Using (2) notice that:

- The *variance* $\mathrm{var}(Y) := \mathrm{cov}(Y, Y)$ of Y is equal to $\mathrm{var}(Y) = v^T \Sigma v$.
- If v and w are such that $(X - \mathbf{1}_{n,d}\,\mathrm{diag}(\mu))v$ and $(X - \mathbf{1}_{n,d}\,\mathrm{diag}(\mu))w$ are orthogonal, then we have $\mathrm{cov}(Y,Z) = 0$.

Task

1. Derive the sample variance $\mathrm{var}(Y^{(k)})$ in terms of singular values of the matrix $X - \mathbf{1}_{n,d}\,\mathrm{diag}(\mu)$.

2. Prove that $Y^{(1)},\dots,Y^{(d)}$ are pairwise uncorrelated, i.e., $\mathrm{cov}(Y^{(i)},Y^{(j)}) = 0$ if $i \neq j$.

3. Write the following Matlab/Octave functions:

   (a) $[\mu,\mathsf{V_k},\mathsf{U_k},\mathsf{D_k}] = \mathsf{pca}(\mathsf{X},\mathsf{k})$: Given an input a matrix X with rows representing data points $X_1,\dots,X_n \in \mathbb{R}^d$ and an integer $k \leq \min(n,d)$, it returns the centroid μ, matrices V_k, U_k with columns being the first k left/right principal directions and a vector D_k of the sample variances of the first k PCs.

   (b) `[Z] = proj(X)`: Given an input X is as in (3a) above, it returns the matrix Z whose i-th row is a projection of $X_i - \mu$ to the the largest two right principal directions and plots both PCs and projections of data points on the same image.

   (c) `[r] = threshold(X,p)`: Given an input X is as in (3a) above and number $p \in (0,1)$, it returns the smallest number r such that $f(r) \geq p$, where

   $$f(k) := \frac{\mathrm{var}(Y^{(1)}) + \ldots + \mathrm{var}(Y^{(k)})}{\mathrm{var}(Y^{(1)}) + \ldots + \mathrm{var}(Y^{(d)})}$$

   and $Y^{(1)},\dots,Y^{(d)}$ are the PCs of X, and plots the graph of $f(k)$ with x-axis the value of k and y-axis the value of $f(k)$.

   Note: A common threshold for the number of PCs to take into account is $p = 0.9$.

   In all functions above stick to specifications: Inputs and outputs must be exactly as described above.

Submission

Use the online classroom to submit the following:

1. Files **pca.m**, **proj.m** and **threshold.m**, which should be well-commented and contain at least one test,

2. A report file **solution.pdf**, which contains the necessary derivations and answers to questions,

While you can discuss solutions of the problems with your colleagues, the programs and report must be your own creation. You can use all Octave functions from problem sessions.