Machine Learning 1 - Exercise 3

Fränz Beckius (374057)
Ivan David Aranzales Acero (399364)
Janek Tichy (584200)
Jeremias Eichelbaum (358685)
Johannes Krause (395469)

November 4, 2018

1 Lagrange Multipliers

1.a

$$\begin{aligned}
\nabla\theta L(\theta,\lambda) &= \nabla\theta(\sum_{k=1}^{n}||\theta - x_k||^2) - \lambda(\theta^T b) & (1)\\
&= (\sum_{k=1}^{N} 2(\theta - x_k)) - \lambda b & (2)\\
&= 2N\theta - 2\widehat{x} - \lambda b => \theta = \frac{\lambda b + 2\widehat{x}}{2N} & (3)\\
&=> \lambda = \frac{2N\theta - 2\widehat{x}}{b} & (4)\\
\nabla\theta L(\theta,\lambda) &= \nabla\theta(\sum_{k=1}^{n}||\theta - x_k||^2) - \frac{2N\theta - 2\widehat{x}}{b}(\theta^T b) & (5)\\
&= N\theta - 2\widehat{x} - 4N\theta + 2\widehat{x} & (6)\\
&= N\theta - 4N\theta = 0 & (7)
\end{aligned}$$

for $\widehat{x} = \frac{1}{N}\sum_{k=1}^{N} x_k$. Λ changes the slope of the tangent.

1.b

$$
\begin{align}
\nabla\theta L(\theta,\lambda) &= \nabla\theta(\sum_{k=1}^{n}||\theta - x_k||^2) - \lambda((||\theta - c||)^2 - 1) \tag{8}\\
&= (\sum_{k=1}^{N} 2(\theta - x_k)) - 2\lambda(\theta - c) \tag{9}\\
&= 2N\theta - 2\widehat{x} - 2\lambda(\theta - c) \tag{10}\\
&=> \theta = \frac{N\widehat{x} - \lambda c}{N - \lambda} \tag{11}\\
&=> \lambda = \frac{N\theta - \widehat{x}}{\theta - c} \tag{12}\\
\nabla\theta L(\theta,\lambda) &= \nabla\theta(\sum_{k=1}^{n}||\theta - x_k||^2) - \frac{N\theta - \widehat{x}}{\theta - c}((||\theta - c||)^2 - 1)) \tag{13}\\
&= 2N\theta - 2\widehat{x} - 2\frac{N\theta - \widehat{x}}{\theta - c}(\theta - c) - 2\frac{\widehat{x} - Nc}{(\theta - c)^2}((||\theta - c||)^2 - 1) \tag{14}\\
&= -2\frac{\widehat{x} - Nc}{(\theta - c)^2}((||\theta - c||)^2 - 1) \tag{15}
\end{align}
$$

Constrain is a line and the tangent can then be either on left or right side of $J(\theta)$

2 Bounds on Eigenvalue

2.a

From the Matrix Cookbook, page 6, we get the following 2 equations:

$$Tr(A) = \sum_{i}^{n} A_{ii}$$

$$Tr(A) = \sum_{i}^{n} \lambda_i$$

We apply this to our scatter matrix and we get:

$$\implies \sum_{i}^{n} S_{ii} = \sum_{i}^{n} \lambda_i$$

Since the largest eigenvalue cannot be larger than the sum of all eigenvalues (including itself):

$$\lambda_1 \leq \sum_{i}^{n} \lambda_i$$

$$\implies \lambda_1 \leq \sum_{i}^{n} S_{ii}$$

2.b

Applying the same equations from 2a, the sum of the diagonal values of the scatter matrix can only be a tight upper bound of the largest eigenvalue if:

$$\sum_{i}^{n} S_{ii} = \sum_{i}^{n} \lambda_i = \lambda_1$$

The largest eigenvalue can only be equal to the sum of all eigenvalues, if the remaining eigenvalues are zero. The data only shows variance in one direction.

2.c

λ_1 is by definition the largest variance of scatter matrix S and describes the variance along the first eigenvector w_1. The element S_{ii} can be rewritten as:

$$S_{ii} = \sum_{j=0}^{N} (x_{ij} - m_i)^2 = cov(x_i, x_i) = var(x_i) \tag{16}$$

and therefore $max(S_{ii})$ is defined as the largest variance along $axis_i$, which corresponds to the unit vector u_i. To calculate how the variance along $axis_i$ is related to the largest variance λ_1 we project u_i onto the first eigenvector w_1:

$$rel = abs((\lambda w_1) \cdot (max(S_{ii})u_i)) \tag{17}$$

Both vectors are normalized $\implies rel \in [0, 1]$. Therefore $max(S_{ii})$ can only be smaller or equal to λ_1.

2.d

$maxS_{ii}$ can only be a tight lower bound of λ_1 if the variance of the scatter matrix is aligned to the axis of the largest variance of our data (our largest eigenvalue), as shown in 2c. We have this case, e.g., when the scatter matrix is a diagonal matrix. Here, the diagonal values S_{ii} correspond exactly to the eigenvalues. Therefore, the largest diagonal value $max(S_{ii})$ corresponds to the largest eigenvalue λ_1

3 Iterative PCA

3.a

3.b