Homework 3 of Statistical Machine Learning

Wang Yikai, 2017310740

June 4, 2018

1 Dimension Reduction, PCA

1.1 Minimum Error Formulation

Given a set of complete orthonormal basis,

$$\{\boldsymbol{\mu}_i\},\ i = 1, \cdots, p,\ \boldsymbol{\mu}_i^T \boldsymbol{\mu}_j = \delta_{ij}$$

Each data point can be represented as:

$$\boldsymbol{x}_n = \sum_{i=1}^{p} \alpha_{ni} \boldsymbol{\mu}_i$$

Due to the orthonormal property, we can get:

$$\boldsymbol{x}_n^T \boldsymbol{\mu}_i = \sum_{j=1}^{p} \alpha_{nj} \boldsymbol{\mu}_j^T \boldsymbol{\mu}_i = \alpha_{ni}$$

Consider a low-dimensional approximation:

$$\tilde{\boldsymbol{x}}_n = \sum_{i=1}^{d} z_{ni} \boldsymbol{\mu}_i + \sum_{i=d+1}^{p} b_i \boldsymbol{\mu}_i$$

where b_i are constants for all data points.

The best approximation is to minimize the error:

$$\min_{\boldsymbol{U}, \boldsymbol{z}, \boldsymbol{b}} J = \frac{1}{N} \sum_{n=1}^{N} \|\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n\|^2$$

According to the conclusion above, the error can be written as:

$$\begin{aligned} J &= \frac{1}{N} \sum_{n=1}^{N} \left\| \sum_{i=1}^{d} \alpha_{ni} \boldsymbol{\mu}_i - \sum_{i=1}^{d} z_{ni} \boldsymbol{\mu}_i + \sum_{i=d+1}^{p} \alpha_{ni} \boldsymbol{\mu}_i - \sum_{i=d+1}^{p} b_i \boldsymbol{\mu}_i \right\|^2 \\ &= \frac{1}{N} \sum_{n=1}^{N} \left\| \sum_{i=1}^{d} (\alpha_{ni} - z_{ni}) \boldsymbol{\mu}_i + \sum_{i=d+1}^{p} (\alpha_{ni} - b_i) \boldsymbol{\mu}_i \right\|^2 \end{aligned}$$

As z and b are independent, we can let $z_{ni} = \alpha_{ni} = \boldsymbol{x}_n^T \boldsymbol{\mu}_i$ to minimize error J with respect to z. Therefore the error is:

$$
\begin{aligned}
J &= \frac{1}{N} \sum_{n=1}^{N} \left\| \sum_{i=d+1}^{p} (\alpha_{ni} - b_i) \boldsymbol{\mu}_i \right\|^2 \\
&= \frac{1}{N} \sum_{n=1}^{N} \sum_{i=d+1}^{p} (\alpha_{ni} - b_i)^2 \boldsymbol{\mu}_i^T \boldsymbol{\mu}_i \\
&= \frac{1}{N} \sum_{i=d+1}^{p} \boldsymbol{\mu}_i^T \boldsymbol{\mu}_i \sum_{n=1}^{N} \left(\alpha_{ni}^2 - 2\alpha_{ni} b_i + b_i^2 \right)
\end{aligned}
$$

For each i, we need to minimize $\sum_{n=1}^{N} (\alpha_{ni}^2 - 2\alpha_{ni} b_i + b_i^2)$ with respect to b_i. Let the derivative equals to zero, there is:

$$
\sum_{n=1}^{N} (2b_i - 2\alpha_{ni}) = 0 \Rightarrow b_i = \frac{1}{N} \alpha_{ni} = \bar{\boldsymbol{x}}_n^T \boldsymbol{\mu}_i
$$

Therefore there are $z_{ni} = \boldsymbol{x}_n^T \boldsymbol{\mu}_i$ for $i = 1, \cdots, d$ and $b_i = \bar{\boldsymbol{x}}_n^T \boldsymbol{\mu}_i$ for $i = d+1, \cdots, p$.

1.2 MNIST

Guidance of the code, simply put the .gz files in the path data/ and run **python pca.py**.

PCA implementations with and without centering the dataset are shown below, and it seems that centering has little effects on the performance of the reconstructed figures.

Preserve 30% information:

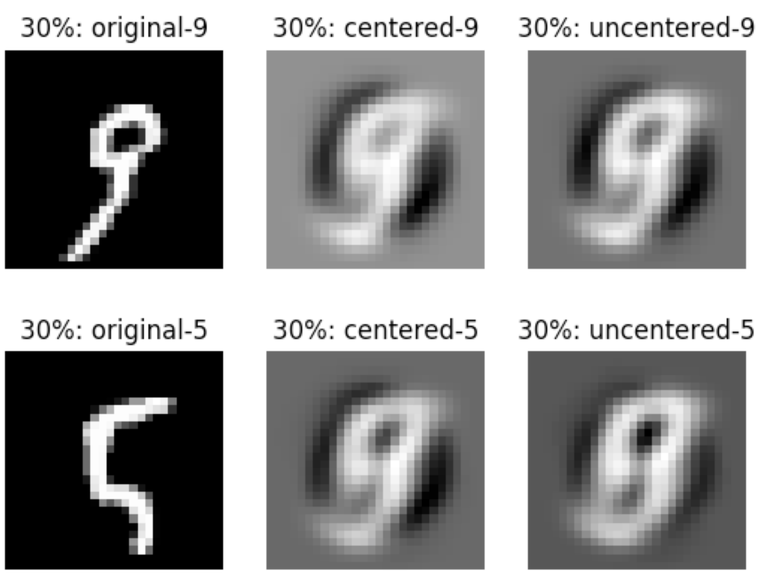

Figure 1: Original and reconstructed figures with 30% information

Preserve 60% information:

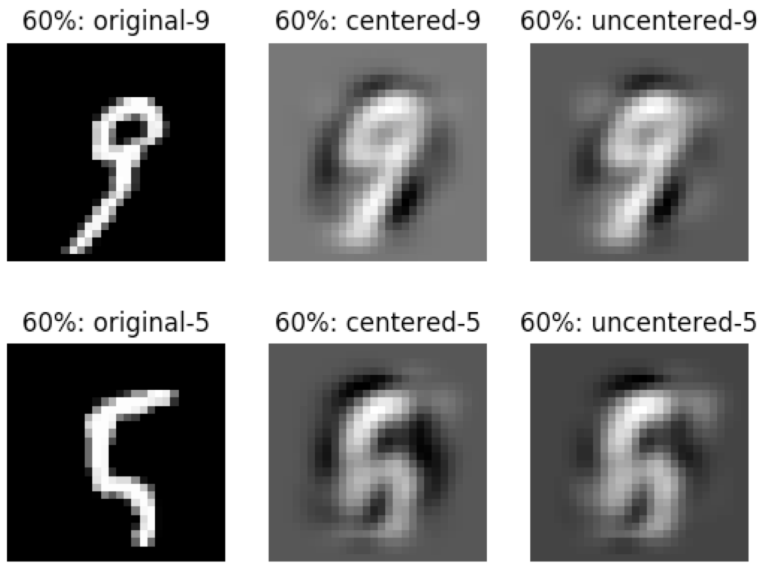

Figure 2: Original and reconstructed figures with 60% information

Preserve 90% information:

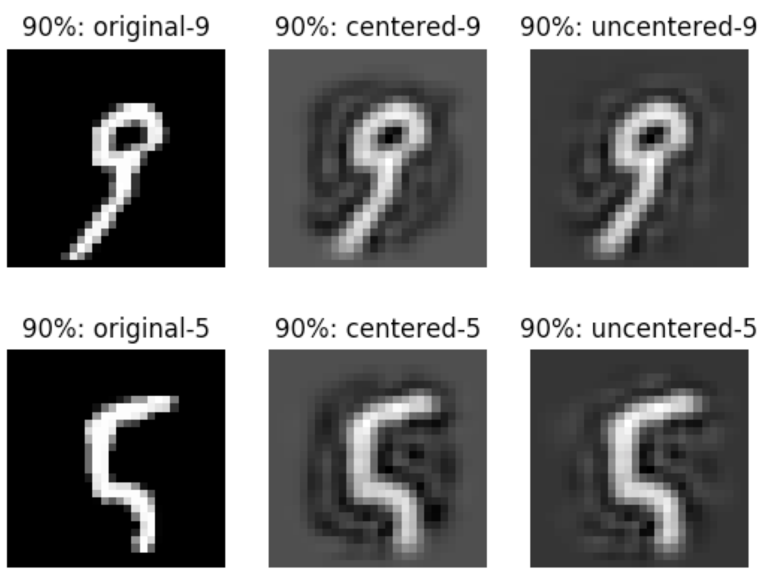

Figure 3: Original and reconstructed figures with 90% information

As shown in the figures, if we preserve 30% information, the reconstructed figures can hardly be recognized. The reconstruction with 90% information has the best performance.

2 Dimension Reduction, PCA

2.1 VC Dimension

For any three dots which are not in a line, the closure of the three dots must be a triangle. If there are no positive dots, draw H outside the the triangle; If there is one positive dot, draw a small H to cover this dot; If there are two, with some rotation of the dots, there exists a rectangle to cover the positive dots; If there are three, draw H to cover the triangle.

But for four dots, if there are three positive dots and one negative dot, and the negative dot locates inside the triangle of the three positive dots, then there is no rectangle to separate these dots.

For five dots, in the case below, the any rectangle cannot separate these dots.

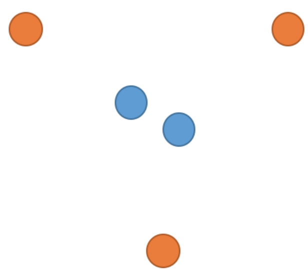

Figure 4: The case with 5 points that cannot be classified

2.2 Generalization Bound

Given the the hypothesis space $H = \{(a < x < b)|a, b \in \mathbb{R}\}$. The upper bound of the probability that a hypothesis $h \in H$ consistent with m instances $x_1, \cdots, x_m$ has an error of at least ϵ is:

$$P\left(|R(h) - R_n(h)| > \epsilon\right) \leq 2\exp\left[-\frac{2m\epsilon}{(b-a)^2}\right]$$

3 Reinforcement Learning

3.1 Value Iteration

$$\begin{aligned}
&0.72 = 0 + 0.9 \times [0.95 \times (-1) + 0.05 \times (35)] \\
&45.695 = 20 + 0.9 \times [0.2 \times (-1) + 0.75 \times (35) + 0.05 \times (50)] \\
&137.71 = 100 + 0.9 \times [0.1 \times (-1) + 0.2 \times (35) + 0.7 \times (50)] \\
&52.71 = 15 + 0.9 \times [0.1 \times (-1) + 0.2 \times (35) + 0.7 \times (50)] \\
&120.68 = 80 + 0.9 \times [0.05 \times (-1) + 0.15 \times (35) + 0.8 \times (50)]
\end{aligned}$$

3.2 Policy Improvement Theorem

Value Function:

$$V^\pi(s) = E_\pi\{r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots | s_t = s\} = E_\pi\{r_{t+1} + \gamma V^\pi(s_{t+1}) | s_t = s\}$$

State-action value function:

$$Q^\pi(s,a) = E_\pi\{r_{t+1} + \gamma V^\pi(s_{t+1}) | s_t = s, a_t = a\}$$

Policy π' is as good as or better than π means $V^{\pi'} \geq V^\pi(s)$ for any state s. We need to prove $\forall s, Q^\pi(s, \pi'(s)) \geq V^\pi(s) \Rightarrow V^{\pi'}(s) \geq V^\pi(s)$:

$$\begin{aligned}
V^\pi(s) &\leq Q^\pi(s, \pi'(s)) \\
&= E_{\pi'}\{r_{t+1} + \gamma V^\pi(s_{t+1}) | s_t = s\} \\
&\leq E_{\pi'}\{r_{t+1} + \gamma Q^\pi(s_{t+1}, \pi'(s_{t+1})) | s_t = s\} \\
&= E_{\pi'}\{r_{t+1} + \gamma r_{t+2} + \gamma^2 V^\pi(s_{t+2}) | s_t = s\} \\
&\leq E_{\pi'}\{r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 V^\pi(s_{t+3}) | s_t = s\} \\
&\vdots \\
&\leq E_{\pi'}\{r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 r_{t+4} + \cdots | s_t = s\} \\
&= V^{\pi'}(s)
\end{aligned}$$

Use a greedy way as the policy improvement strategy:

$$\pi'(s) = \arg\max_a Q^\pi(s,a) = \arg\max_a E_\pi\{r_{t+1} + \gamma V^\pi(s_{t+1}) | s_t = s, a_t = a\}$$

Then we can get a new policy that is as good as or better than the old policy.

If $\pi'(s)$ is as good as but not better than $\pi(s)$, then $V^{\pi'}(s) = V^\pi(s)$. At this time there is:

$$V^{\pi'}(s) = \max_a E_\pi\{r_{t+1} + \gamma V^\pi(s_{t+1}) | s_t = s, a_t = a\}$$

which equals to the formula of optimal value function in the slide 18.

Thus in PI, either π' is strictly better than π , or π is optimal.

3.3 TianShou

DQN is an off-policy, value-based algorithm.