Lecture 9: Clustering, principal component analysis and autoencoders

2025–06–25

1.1 K-means clustering

Given a set of data points $\mathcal{D} = \{x_1, x_2, \ldots, x_n\}$ in $\mathbb{R}^d$, clustering aims to partition the n points into k sets $S = \{S_1, S_2, \ldots, S_k\}$, so that the *within-cluster sum-of-squares* is minimized. Precisely, the objective is to find:

$$\arg\min_S \sum_{i=1}^{k} \sum_{x \in S_i} |x - \mu_i|^2 = \arg\min_S \sum_{i=1}^{k} |S_i| \mathrm{Var}(S_i)\,, \tag{1.1}$$

where $|S_i|$ denotes the number of points in S_i and μ_i is the mean (also called centroid) of points in S_i, i.e.

$$\mu_i = \frac{1}{|S_i|} \sum_{x \in S_i} x\,. \tag{1.2}$$

Notice that the following identity holds

$$|S_i| \sum_{x \in S_i} |x - \mu_i|^2 = \frac{1}{2} \sum_{x,y \in S_i} |x - y|^2 = \sum_{x \in S_i} |x|^2 - \Big(\sum_{x \in S_i} x \Big)^2\,. \tag{1.3}$$

Therefore, the objective (1.1) is equivalent to

$$\arg\min_S \sum_{i=1}^{k} \frac{1}{|S_i|} \sum_{x,y \in S_i} |x - y|^2\,, \tag{1.4}$$

that is, finding partition so as to minimize the pairwise squared deviations of points within each cluster.

A common algorithm for solving the problem (1.1) is the so-called "k-means algorithm". Given the number of clusters, k, and an initial set of centers $m_1^{(1)}, \ldots, m_k^{(1)}$. The algorithm updates the partition and the centers by alternating between the following two steps.

1. Assignment step: update the partition by assigning each point x to the cluster corresponding to the nearest center.

$$S_i^{(l)} = \Big\{ x \in \mathcal{D} \;\Big|\; |x - m_i^{(l)}| \le |x - m_j^{(l)}|,\ \forall 1 \le j \le k \Big\}\,. \tag{1.5}$$

2. Update step: update the cluster centers.

$$m_i^{(l+1)} = \frac{1}{|S_i^{(l)}|} \sum_{x \in S_i^{(l)}} x\,. \tag{1.6}$$

We have the following result.

Proposition 1. The objective (1.1) is non-increasing in the k-means algorithm.

Proof. Let us consider the objective $\sum_{i=1}^{k}\sum_{x\in S_i^{(l)}}|x-m_i^{(l)}|^2$ at lth step, for $l\geq 1$. In the second step, we fix the partition $S_1^{(l)},\dots,S_k^{(l)}$ and update the centers according to (1.6). Using the fact

$$\frac{1}{|S_i^{(l)}|}\sum_{x\in S_i^{(l)}} x = \arg\min_{c\in\mathbb{R}^d}\sum_{x\in S_i^{(l)}}|x-c|^2,$$

we obtain

$$\sum_{i=1}^{k}\sum_{x\in S_i^{(l)}}|x-m_i^{(l+1)}|^2 \leq \sum_{i=1}^{k}\sum_{x\in S_i^{(l)}}|x-m_i^{(l)}|^2\,.$$

In the first step, we fix the centers and update the partition according to (1.5). Clearly, we have

$$\sum_{i=1}^{k}\sum_{x\in S_i^{(l+1)}}|x-m_i^{(l+1)}|^2 \leq \sum_{i=1}^{k}\sum_{x\in S_i^{(l)}}|x-m_i^{(l+1)}|^2\,.$$

Combining the two inequalities above, we have

$$\sum_{i=1}^{k}\sum_{x\in S_i^{(l+1)}}|x-m_i^{(l+1)}|^2 \leq \sum_{i=1}^{k}\sum_{x\in S_i^{(l)}}|x-m_i^{(l)}|^2\,,$$

which implies that the objective (1.1) is non-increasing. □

Remark.

1. Despite of Proposition 1, the k-means algorithm may not be able to find the optimal partition that solves (1.1).
2. The number of clusters k has to be chosen before applying k-means algorithm. In practice, the optimal number of clusters needs to be determined by comparing clustering results obtained with different numbers of clusters.
3. The total amount of operations required in each iteration is $\mathcal{O}(n\times k)$. Therefore, the computational complexity of the k-means algorithm is $\mathcal{O}(n\times k\times N)$, where N is the total number of iterations.

1.2 Principal component analysis

Given high dimensional data $\mathcal{D}=\{x_1,\dots,x_n\}$, principal component analysis (PCA) aims at identifying a linear and orthogonal projection

$$WW^\top x+b,\quad x\in\mathbb{R}^d\,, \tag{1.7}$$

where $W \in \mathbb{R}^{d\times k}$ satisfies $W^\top W = I_k$ and $b \in \mathbb{R}^d$, such that it minimizes the *reconstruction error*

$$L(W,b) = \frac{1}{n}\sum_{i=1}^{n} |x_i - (WW^\top x_i + b)|^2 \,. \tag{1.8}$$

It is straightforward to see that the optimal $b \in \mathbb{R}^d$ is given by

$$b = \frac{1}{n}\sum_{i=1}^{n}(x_i - WW^\top x_i) = \bar{x} - WW^\top \bar{x}\,, \tag{1.9}$$

where

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{1.10}$$

is the empirical mean of the data. Substituting (1.9) into (1.7), we see that the linear projection is

$$WW^\top(x-\bar{x}) + \bar{x}, \quad x \in \mathbb{R}^d \,. \tag{1.11}$$

Therefore, we only need to find the optimal matrix W.

Substituting (1.9) into (1.8), we see that finding the optimal $W \in \mathbb{R}^{d\times k}$ amounts to minimizing

$$L(W) = \frac{1}{n}\sum_{i=1}^{n} |(x_i - \bar{x}) - WW^\top(x_i - \bar{x})|^2 \,, \tag{1.12}$$

under the constraint $W^\top W = I_k$.

We have the following result.

Proposition 2. Define the empirical covariance matrix

$$\hat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})^\top \in \mathbb{R}^{d\times d} \,. \tag{1.13}$$

Then, for the objective in (1.12), we have

$$L(W) = \frac{1}{n}\sum_{i=1}^{n} |x_i - \bar{x}|^2 - \mathrm{tr}(W^\top \hat{\Sigma} W) \,. \tag{1.14}$$

Proof. From (1.12), using $W^\top W = I_k$, we can derive

$$\begin{aligned}
L(W) =& \frac{1}{n}\sum_{i=1}^n |(x_i-\bar{x}) - WW^\top(x_i-\bar{x})|^2 \\
=& \frac{1}{n}\sum_{i=1}^n \Big(|x_i-\bar{x}|^2 - 2(x_i-\bar{x})^\top WW^\top(x_i-\bar{x}) + (x_i-\bar{x})^\top WW^\top WW^\top(x_i-\bar{x})\Big) \\
=& \frac{1}{n}\sum_{i=1}^n \Big(|x_i-\bar{x}|^2 - 2(x_i-\bar{x})^\top WW^\top(x_i-\bar{x}) + (x_i-\bar{x})^\top WW^\top(x_i-\bar{x})\Big) \\
=& \frac{1}{n}\sum_{i=1}^n \Big(|x_i-\bar{x}|^2 - (x_i-\bar{x})^\top WW^\top(x_i-\bar{x})\Big)\,.
\end{aligned}$$

Noticing the identity

$$(x_i-\bar{x})^\top WW^\top(x_i-\bar{x}) = \mathrm{tr}\Big(W^\top(x_i-\bar{x})(x_i-\bar{x})^\top W\Big)\,,$$

and using (1.13), we can continue to derive

$$\begin{aligned}
L(W) =& \frac{1}{n}\sum_{i=1}^n \Big(|x_i-\bar{x}|^2 - (x_i-\bar{x})^\top WW^\top(x_i-\bar{x})\Big) \\
=& \frac{1}{n}\sum_{i=1}^n |x_i-\bar{x}|^2 - \frac{1}{n}\sum_{i=1}^n \mathrm{tr}\Big(W^\top(x_i-\bar{x})(x_i-\bar{x})^\top W\Big) \\
=& \frac{1}{n}\sum_{i=1}^n |x_i-\bar{x}|^2 - \mathrm{tr}(W^\top\hat{\Sigma}W)\,.
\end{aligned}$$

□

Proposition 2 implies that minimizing $L(W)$ in (1.12) is equivalent to solving

$$\max_{W\in\mathbb{R}^{d\times k}, W^\top W = I_k} \mathrm{tr}(W^\top\hat{\Sigma}W)\,. \tag{1.15}$$

Let us denote by $W_1,\dots,W_k \in \mathbb{R}^d$ the column vectors of W. Then, direct computation shows that

$$\begin{aligned}
&\mathrm{tr}(W^\top\hat{\Sigma}W) = \sum_{i=1}^k W_i^\top\hat{\Sigma}W_i\,, \\
&W^\top W = I_k \quad\Longleftrightarrow\quad W_i^\top W_j = \delta_{ij}, \quad \forall 1\le i,j\le k\,.
\end{aligned} \tag{1.16}$$

Therefore, (1.15) is equivalent to

$$\max_{W_1,\dots,W_k\in\mathbb{R}^d} \sum_{i=1}^k W_i^\top\hat{\Sigma}W_i\,, \quad \text{subject to} \quad W_i^\top W_j = \delta_{ij}, \quad \forall 1\le i,j\le k\,. \tag{1.17}$$

Note that the matrix $\hat{\Sigma}$ in (1.13) is clearly symmetric. Moreover, we have

$$v^\top\hat{\Sigma}v = \frac{1}{n}\sum_{i=1}^n v^\top(x_i-\bar{x})(x_i-\bar{x})^\top v = \frac{1}{n}\sum_{i=1}^n |(x_i-\bar{x})^\top v|^2 \ge 0\,, \quad \forall\, v\in\mathbb{R}^d\,.$$

Hence, $\hat{\Sigma}$ is semi-positive definite. As a result, all eigenvalues of $\hat{\Sigma}$ are non-negative real numbers.

Moreover, it can be shown (we omit the proof) that the maximum of (1.17) is achieved when $W_1, \dots, W_k$ are the k (pairwise orthogonal and normalized) eigenvectors of $\hat{\Sigma}$ corresponding to the largest k eigenvalues.

Based on the above analysis, we summarize the algorithm of PCA as follows.

1. Compute $\hat{\Sigma}$ in (1.13), where $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$.
2. Compute the (pairwise orthogonal and normalized) eigenvectors $W_1, \dots, W_k$ corresponding to the k largest eigenvalues of $\hat{\Sigma}$, so that $W_i^\top W_j = \delta_{ij}$ for $i, j = 1, \dots, k$.
3. Let $W \in \mathbb{R}^{d\times k}$ be the matrix whose column vectors are $W_1, \dots, W_k$.

Once W is computed, we obtain the linear projection in (1.11).

Remark (PCA by SVD decomposition). Let $X \in \mathbb{R}^{n\times d}$ be the data matrix, whose ith row is $x_i - \bar{x}$. Then, we can verify $\hat{\Sigma} = \frac{1}{n} X^\top X$. Consider the SVD decomposition of $X = U\Lambda V^\top$, where $U \in \mathbb{R}^{n\times n}$ satisfies $U^\top U = I_n$, $\Lambda \in \mathbb{R}^{n\times d}$ is a rectangular diagonal matrix with positive numbers, and $V \in \mathbb{R}^{d\times d}$ satisfies $V^\top V = I_d$. Then, $\hat{\Sigma} = \frac{1}{n} X^\top X = \frac{1}{n} V(\Lambda^\top \Lambda) V^\top$, where $\Lambda^\top \Lambda \in \mathbb{R}^{d\times d}$ is a diagonal matrix. Therefore, we can construct W by selecting column vectors of V corresponding to the k largest singular values.

1.3 Autoencoders

Different from PCA in the previous section which maps data in $\mathbb{R}^d$ to a low-dimensional space $\mathbb{R}^k$, where $1 \le k < d$, by a linear projection, autoencoders provide a low-dimensional representation of data using a nonlinear projection $\xi : \mathbb{R}^d \to \mathbb{R}^k$. The dimension k is often called latent dimension or bottleneck dimension. Precisely, an autoencoder is a function of the form $f = \varphi \circ \xi$, where $\xi : \mathbb{R}^d \to \mathbb{R}^k$, the encoder, maps an input x to the low-dimensional space $\mathbb{R}^k$, and $\varphi : \mathbb{R}^k \to \mathbb{R}^d$, the decoder, maps the "encoded" state $\xi(x)$ in $\mathbb{R}^k$ back to the high-dimensional space $\mathbb{R}^d$.

Assume that the data is sampled from a probability distribution μ. The low-dimensional representation is "good", if the reconstructed state $f(x)$, obtained by first encoding x and then decoding, remains close to x, in an average sense for data $x \sim \mu$. In other words, the autoencoder is optimized by minimizing the so-called reconstruction loss

$$\mathcal{L}(\xi, \varphi) = \int_{\mathbb{R}^d} |\varphi(\xi(x)) - x|^2 \, d\mu = \mathbb{E}_{x\sim\mu} |\varphi(\xi(x)) - x|^2 . \tag{1.18}$$

Assume that the data set is $\mathcal{D} = \{x_1, \dots, x_n\}$. Then, the autoencoder is learnt by minimizing the empirical loss function

$$\hat{\mathcal{L}}(\xi, \varphi) = \frac{1}{n} \sum_{i=1}^{n} |\varphi(\xi(x_i)) - x_i|^2 . \tag{1.19}$$

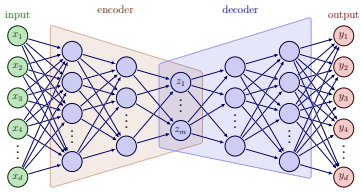

Figure 1.1: Illustration of autoencoders.

Clearly, if we choose a linear encoder and a linear decoder as

$$\begin{gathered} \xi(x) = W^\top x, \quad \varphi(z) = Wz + b, \\ \text{where } W \in \mathbb{R}^{d\times k},\ b \in \mathbb{R}^d,\ \text{and } W^\top W = I_k\,, \end{gathered} \tag{1.20}$$

the autoencoder $f = \varphi \circ \xi$ reduces to the linear map in (1.7) and the loss function (1.19) recovers the objective (1.8), respectively. Hence, in this setting, learning autoencoders recovers PCA. In general, autoencoders are represented by neural networks (see Figure 1.1) and learning autoencoders can be viewed as a nonlinear generalization of PCA.

In the following, we characterize the minimizer of the reconstruction loss (1.18). By optimizing the decoder first, we obtain the following result.

Proposition 3. Assume that $x \sim \mu$ and let $\widetilde{\mu}$ be the distribution of $z = \xi(x)$. We have

$$\min_{\xi}\min_{\varphi} \mathbb{E}_{x\sim\mu}|\varphi(\xi(x)) - x|^2 = \min_{\xi} \mathbb{E}_{z\sim\widetilde{\mu}}\big[\mathrm{Var}(x|\,\xi(x) = z)\big]\,, \tag{1.21}$$

where $\mathrm{Var}_{x\sim\mu}(x|\,\xi(x) = z)$ denotes the variance of x conditioned on the event that $\xi(x) = z$. Moreover, for a fixed encoder $\xi : \mathbb{R}^d \to \mathbb{R}^k$, the optimal decoder $\varphi : \mathbb{R}^k \to \mathbb{R}^d$ is given by

$$\varphi_\xi(z) = \mathbb{E}_{x\sim\mu}(x\,|\,\xi(x) = z)\,, \quad \forall\ z \in \mathbb{R}^k\,. \tag{1.22}$$

Proof. Recall that the law of total expectation implies

$$\mathbb{E}_{x\sim\mu}\big(|\varphi(\xi(x)) - x|^2\big) = \mathbb{E}_{z\sim\widetilde{\mu}}\Big[\mathbb{E}_{x\sim\mu}\Big(|\varphi(\xi(x)) - x|^2\,\Big|\,\xi(x) = z\Big)\Big]\,. \tag{1.23}$$

Also, for a fixed $z \in \mathbb{R}^k$, it is straightforward to verify that

$$\min_{x'\in\mathbb{R}^d} \mathbb{E}_{x\sim\mu}\Big(|x' - x|^2\,\Big|\,\xi(x) = z\Big) = \mathrm{Var}_{x\sim\mu}\big(x\,|\,\xi(x) = z\big)\,, \tag{1.24}$$

and the minimum is attained when $x' = \mathbb{E}_{x\sim\mu}(x\,|\,\xi(x) = z)$. Therefore,

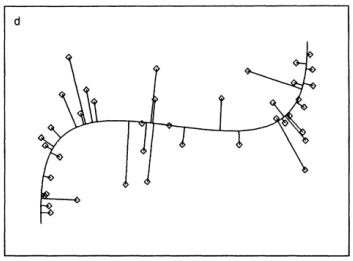

Figure 1.2: Illustration of principal curve.

using (1.23) and (1.24), we can derive

$$\begin{aligned}
&\min_{\xi}\min_{\varphi}\mathbb{E}_{x\sim\mu}|\varphi(\xi(x))-x|^2\\
=&\min_{\xi}\min_{\varphi}\mathbb{E}_{z\sim\widetilde{\mu}}\Big[\mathbb{E}_{x\sim\mu}\Big(|\varphi(\xi(x))-x|^2\,\Big|\,\xi(x)=z\Big)\Big]\\
=&\min_{\xi}\min_{\varphi}\mathbb{E}_{z\sim\widetilde{\mu}}\Big[\mathbb{E}_{x\sim\mu}\Big(|\varphi(z)-x|^2\,\Big|\,\xi(x)=z\Big)\Big]\\
=&\min_{\xi}\mathbb{E}_{z\sim\widetilde{\mu}}\Big\{\min_{\varphi(z)}\Big[\mathbb{E}_{x\sim\mu}\Big(|\varphi(z)-x|^2\,\Big|\,\xi(x)=z\Big)\Big]\Big\}\\
=&\min_{\xi}\mathbb{E}_{z\sim\widetilde{\mu}}\big[\mathrm{Var}_{x\sim\mu}(x|\,\xi(x)=z)\big]\,,
\end{aligned}$$

and the optimal decoder is given by (1.22). □

The expression (1.21) implies that the optimal encoder ξ minimizes the averaged conditional variances on its level-sets. The expression (1.22) states that, for a fixed encoder ξ, the optimal decoder maps z to the center of all points x that satisfy $\xi(x)=z$ (i.e. the pre-image of z under ξ).

Alternatively, by optimizing the encoder first, we immediately obtain the following characterization.

Proposition 4.

$$\min_{\varphi}\min_{\xi}\mathbb{E}_{x\sim\mu}|\varphi(\xi(x))-x|^2=\min_{\varphi}\mathbb{E}_{x\sim\mu}|\varphi(\xi_\varphi(x))-x|^2\,,$$

where

$$\xi_\varphi(x)=\arg\min_{z\in\mathbb{R}^k}|\varphi(z)-x|,\quad\forall\,x\in\mathbb{R}^d\,.\tag{1.25}$$

The expression (1.25) states that, given the decoder φ, the optimal encoder maps the data x to the index of the closest point (under the decoder) to x.

To summarize, the optimal autoencoder satisfies the self-consistent condition:

$$\begin{aligned}
\varphi(z)&=\mathbb{E}_{x\sim\mu}(x\,|\,\xi(x)=z),\quad z\in\mathbb{R}^k\,,\\
\xi(x)&=\arg\min_{z\in\mathbb{R}^k}|\varphi(z)-x|\,,\quad x\in\mathbb{R}^d\,.
\end{aligned}$$

In particular, when the bottleneck dimension $k=1$, autoencoders recover the so-called principal curves.