Overview of Unsupervised Learning

PS690 Computational Methods in Social Science

Jiawei Fu
Department of Political Science
Duke University

Sep 11, 2025

Overview

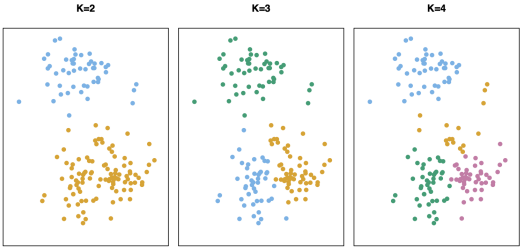
1. Unsupervised Learning
2. Clustering Methods
3. Dimension Reduction

Unsupervised Learning

- Previous lectures concern with predicting Y for a given set of covariate X.
- This is called supervised learning or "learn-ing with a teacher." The "student" presents an answer $\hat{y}_i$ for each x_i in the training sample, and the supervisor or "teacher" provides either the correct answer and/or an error associated with the student's answer, characterized by loss function $L(y, \hat{y})$.
- Unsupervised learning, learning without a teacher, is often much more challenging.
- Given N observations $X = (x_1, x_2, ..., x_n)$ from joint density $P(X)$, the goal is to directly infer the properties of this probability density without the help of a supervisor or teacher providing correct answers.
- The exercise tends to be more subjective, and there is no simple goal for the analysis, such as prediction of a response.

K-means clustering

- Cluster analysis aims to grouping or segmenting a collection of objects into subsets or "clusters," such that those within each cluster are more closely related to one another than objects assigned to different clusters.
- The K-means algorithm is one of the most popular iterative descent clustering methods.
- In K-means clustering, we seek to partition the observations into a pre-specified K number of clusters.



K-means clustering

- Let $C_1, ..., C_K$ denote sets containing the indices of the observations in each cluster.
- There sets satisfy
  1. $C_1 \cup ... \cup C_K = \{1, ..., N\}$; each observation belongs to at least one of the K clusters.
  2. $C_k \cap C_{k'} = \emptyset \ \ \forall k \neq k'$; no observation belongs to more than one cluster.
- A good clustering is one for which the within-cluster variation is as small as possible.
- Define the within-cluster variation for k-th cluster as $W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} ||x_i - x_{i'}||^2$.
- Therefore, the optimization problem that defines K-means clustering is

$$\min_{C_1,...,C_K} \sum_{k=1}^{K} W(C_k) = \frac{1}{2} \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} ||x_i - x_{i'}||^2$$

- It is very difficult to solve precisely because there are almost K^n ways to partition n observations into K clusters.

K-means clustering

- Observe that

$$\min_{C_1,\ldots,C_K} \sum_{k=1}^{K} W(C_k) = \frac{1}{2} \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} ||x_i - x_{i'}||^2$$

$$= \sum_{k=1}^{K} \sum_{i \in C_k} ||x_i - \overline{x}_k||^2$$

  where $x_k = (\overline{x}_{1k}, \ldots, \overline{x}_{pk})$ is the mean vector associated with the $k-$th cluster ($\overline{x}_{kj} = \frac{1}{|C_k|} \sum_{i \in C_k} x_{ij}$.)
- Therefore, the criterion is minimized by assigning the N observations to the K clusters in such a way that within each cluster the average dissimilarity of the observations from the cluster mean.

K-means clustering

Algorithm 10.1 *K-Means Clustering*

1. Randomly assign a number, from 1 to K, to each of the observations. These serve as initial cluster assignments for the observations.

2. Iterate until the cluster assignments stop changing:

   (a) For each of the K clusters, compute the cluster *centroid*. The kth cluster centroid is the vector of the p feature means for the observations in the kth cluster.

   (b) Assign each observation to the cluster whose centroid is closest (where *closest* is defined using Euclidean distance).

Figure: From [James et al., 2013]

K-means clustering

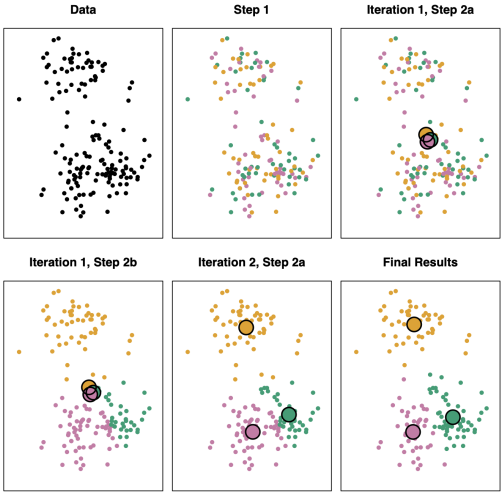
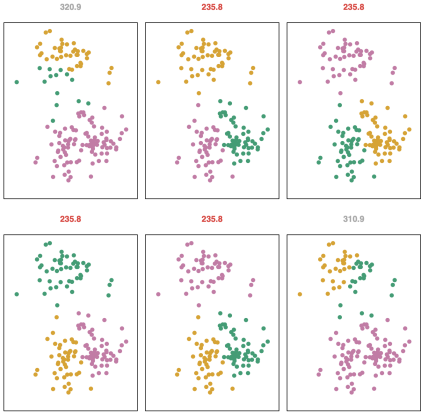


Figure: From [James et al., 2013]

K-means clustering

- K-means algorithm finds a local rather than a global optimum, the results obtained will depend on the initial (random) cluster assignment.
- It is important to run the algorithm multiple times from different random initial configurations. Then one selects the best solution.



How to choose K?

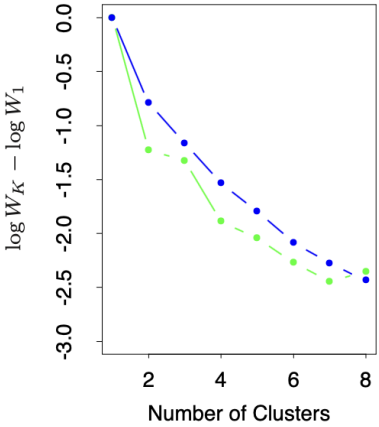
- A choice for the number of clusters K depends on the goal.
- A naive idea is to go through K from 1 to K_{max} and calculate the optimized value of the objective function.
- Typically, the optimized value decreases with increasing K.
- Thus cross-validation techniques, so useful for model selection in supervised learning, cannot be utilized in this context.

How to choose K?

- Suppose there are actually K^* distinct groups.
- Then for $K < K^*$, the clusters returned by the algorithm will each contain a subset of the true underlying groups.
- That is, the solution will not assign observations in the same naturally occurring group to different estimated clusters.
- To the extent that this is the case, the solution criterion value will tend to decrease substantially with each successive increase in the number of specified cluster.
- For $K > K^*$, one of the estimated clusters must partition at least one of the natural groups into two sub- groups. This will tend to provide a smaller decrease in the criterion as K is further increased.

How to choose K?

- Therefore, we are looking for a kink on the plot of optimized value of the objective function as a function of K.
- Note that this approach, usually called the Elbow Method, is somewhat heuristic.



How to choose K?

- Actually, this is still an active area of research and there are no definitive answer.
- Another idea is to treat choosing k as a hypothesis testing problem.
- The null hypothesis is H_k : the number of clusters is k, and the alternative is larger than k.
- We choose the first k that is not rejected.
- Other idea: we can compare the intracluster variability to the expected variability if the data were uniformly distributed on a rectangle. The number of clusters is then chosen based on the comparisons of these metrics. This is called Gap Statistic Method.

Principal Component Analysis

- Given a high-dimensional data set $X = (X_1, ...X_p)$, how can we find a low-dimensional representation of a data set that contains as much as possible of the variation?
- A central goal of deep learning is to discover representations of data that are useful for one or more subsequent applications.
- Before we go to deep learning, we will learn a simple autoencoder here: PCA.
- The idea is that each of the n observations lives in p-dimensional space, but not all of these dimensions are equally interesting.
- PCA seeks a small number of dimensions that are as interesting as possible, where the concept of interesting is measured by the amount that the observations vary along each dimension.

PCA

- Each of the dimensions, principal components, found by PCA is a linear combination of the p features.
- The first principal component of a set of features is the normalized linear combination fo the features

$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + ... + \phi_{p1}X_p$$

  that has the largest variance.
- By normalized, we mean loadings, $\phi_{11}, ..., \phi_{p1}$, such that $\sum_{j=1}^{p} \phi_{j1}^2 = 1$ because otherwise setting these elements to be arbitrarily large in absolute value could result in an arbitrarily large variance.
- Since we are only interested in variance, we assume that each of the variables in X has been centered to have mean zero.

PCA

- In practice, the first principal component loadings are solved by the optimization problem

$$\max_{\phi_{11},\ldots,\phi_{p1}} \frac{1}{n}\sum_{i=1}^{n}(\sum_{j=1}^{p}\phi_{j1}x_{ij})^2 = \frac{1}{n}\sum_{i=1}^{n} z_{i1}^2$$

$$s.t. \sum_{j=1}^{p}\phi_{j1}^2 = 1$$

- Note, it is just the sample variance of the n values of z_{i1} because the average of them is zero.
- We refer to $z_{11}, \ldots, z_{n1}$ as the scores of the first principal component.

PCA

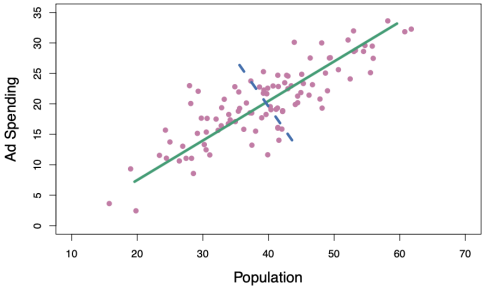


Figure: From [James et al., 2013]

PCA

- Write them into the matrix form:
  1. $\phi_1 = (\phi_{11}, ..., \phi_{p1})^T$
  2. $z_{i1} = \phi_{11}x_{i1} + ..., +\phi_{p1}x_{ip} = x_i'\phi_1$
  3. $\frac{1}{n}\sum_{i=1}^{n} z_{i1}^2 = \frac{1}{n}\sum_{i=1}^{n} \phi_1^T x_i x_i^T \phi_1 = \phi_1^T S \phi_1$, where $S = \frac{1}{n}\sum_{i=1}^{n} x_i x_i^T$ is the sample covariance matrix.
- Therefore, PCA solves

$$\max_{\phi_1} Var(Z_1) = \phi_1^T S \phi_1 \;\; s.t. ||\phi_1|| = 1$$

- We introduce a Lagrange multiplier that we will denote by λ_1; then we solve

$$\phi_1^T S \phi_1 + \lambda_1(1 - \phi_1^T \phi_1)$$

- Setting the derivative with respect to ϕ_1 equal to zero, we get $S\phi_1 = \lambda_1 \phi_1$
- Thus, ϕ_1 is the eigenvector of S.
- And $Var(Z_1) = \phi_1^T S \phi_1 = \lambda_1$; the variance is maximum when we take ϕ_1 equal to the eigenvector having the largest eigenvalue λ_1.

PCA

- The second principal component $Z_2 = \phi_{12}X_1 + \phi_{22}X_2 + ... + \phi_{p2}X_p$ is the linear combination of $X_1, ..., X_p$ that has maximal variance out of all linear combinations that are not correlated with Z_1.
- It turns out that constraining Z_2 to be uncorrelated with Z_1 is equivalent to constraining the direction $\phi_2 = (\phi_{12}, ..., \phi_{p2})$ to be orthogonal to the direction $\phi_1 = (\phi_{11}, ..., \phi_{p1})$: $\phi_2'\phi_1 = 0$.
- It also turns out that ϕ_2 will be the eigenvector of S with the second largest eigenvalue.

PCA

- In summary, PCA involves evaluating the mean x and the covariance matrix S of a data set and then finding the M eigenvectors of S corresponding to the M largest eigenvalues.
- Now, we can rephrase the original problem.
- Given the data with dimension p, our goal is to project the data onto a space having dimensionality $M < p$.
- In other words, we hope to find a matrix W such that $Z = XW$, has lower dimension but captures enough variance of data X.
- Then, $Z_M = X\phi_M$, $\phi_M = [\phi_1, ..., \phi_M]$ is a weight matrix $p \times M$ whose columns are the first M largest eigenvectors of $X^T X$; Z_M has a lower dimension $n \times M$, each row of Z_M is the compressed version of the original observation of dimensions p.

Another interpretation of PCA

- Suppose that we want to find an orthogonal set of M linear basis vectors $\phi_j \in \mathbb{R}^p$, and the corresponding scores (or say coefficient) $z_i \in \mathbb{R}^M$, such that we minimize the average reconstruction error

$$J(\phi, Z) = \frac{1}{N} \sum_{i=1}^{n} ||x_i - \hat{x}_i||^2,$$

  where $\hat{x}_i = \phi z_i = \sum_{j=1}^{M} z_{ij} \phi_j$, subject to ϕ (Dim: $p \times M$) is orthonormal.
- The optimal solution is obtained by setting ϕ, which contains the M eigenvectors with largest eigenvalues of S.

Another interpretation of PCA

- Again, start by estimating the first $\phi_1 \in \mathbb{R}^p$.

$$
\begin{aligned}
J(\phi_1, z_1) &= \frac{1}{N}\sum_{i=1}^{n} ||x_i - \hat{x}_i||^2 \\
&= \frac{1}{N}\sum_{i=1}^{n} [x_i^T x_i - 2z_{i1}\phi_1^T x_i + z_{i1}^2 \phi_i^T \phi_1] \\
&= \frac{1}{N}\sum_{i=1}^{n} [x_i^T x_i - 2z_{i1}\phi_1^T x_i + z_{i1}^2]
\end{aligned}
$$

- Take derivative wrt z_{i1} and set to zero, we get $z_{i1} = \phi_1^T x_i$.
- Plugging back: $J(\phi_1) = \frac{1}{N}\sum_{i=1}^{n}[x_i^T x_i - z_{i1}^2] = const - \frac{1}{N}\sum_{i=1}^{n} z_{i1}^2$
- Then we just minimize the second part; it is equivalent to maximize $\phi_1^T S \phi_1$. We have seen it before.

Another interpretation of PCA

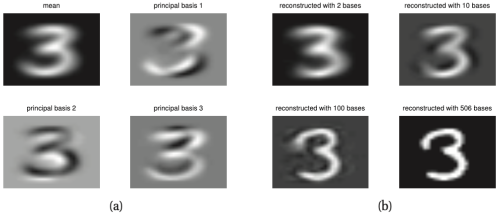


Figure: From [Murphy, 2012]

- PCA can let us use the lower dimension to represent the data and reconstruct the data. If we can add some random noise and probably can generate a new sample! We will learn generative models in DL.

References

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013).
An introduction to statistical learning: with applications in R, volume 103.
Springer.

Murphy, K. P. (2012).
Machine learning: a probabilistic perspective.
MIT press.