Topic XI:
Unsupervised Learning

Wei You

香港科技大學
THE HONG KONG UNIVERSITY OF
SCIENCE AND TECHNOLOGY

Fall, 2023

Introduction

Most of the statistical learning method we encountered in this course involves not only the predictors/features/independent variables, but also the response variable.

- The goal is to predict the response variables using the predictors.
- The response can be viewed as the “correct” answer, i.e., a supervisory signal.
- The task of learning a function that maps an input to an output based on the training data is called supervised learning.

What if we do not have access to the response variables?

We need unsupervised learning.

- Involves no response variable, i.e., just data, no label.
- Learn some underlying hidden structure of the data.
- We shall look at two most important examples: dimensionality reduction and clustering.

Dimensionality Reduction

Principal component analysis (PCA) is a popular dimensionality reduction technique.

- In PCA, we wish to find linear combinations of the predictor that differentiate the individuals as much as possible.
- Mathematically, we wish to find orthogonal transformations of the original variables $\boldsymbol{x} \in \mathbb{R}^p$ such that the transformed variables have the greatest variance.

Principal Component Analysis

Consider the transformation

$$Z = \boldsymbol{\xi}^T \boldsymbol{x}.$$

Assuming that the covariance matrix of $\boldsymbol{x}$ is $\boldsymbol{\Sigma}$. Then

$$\mathsf{Var}(Z) = \mathsf{Var}(\boldsymbol{\xi}^T \boldsymbol{x}) = \boldsymbol{\xi}^T \boldsymbol{\Sigma} \boldsymbol{\xi}.$$

First principal component

$$Z_1 = \boldsymbol{\xi}_1^T \boldsymbol{x}, \quad \text{where} \quad \boldsymbol{\xi}_1 = \arg\max_{\boldsymbol{\xi}} \boldsymbol{\xi}^T \boldsymbol{\Sigma} \boldsymbol{\xi} \quad \text{subject to } \|\boldsymbol{\xi}\|_2 = 1$$

- Normalization $\|\boldsymbol{\xi}\|_2 = 1$ removes variance's dependence on the norm of $\boldsymbol{\xi}$.

Suppose we want to find another transformation that is

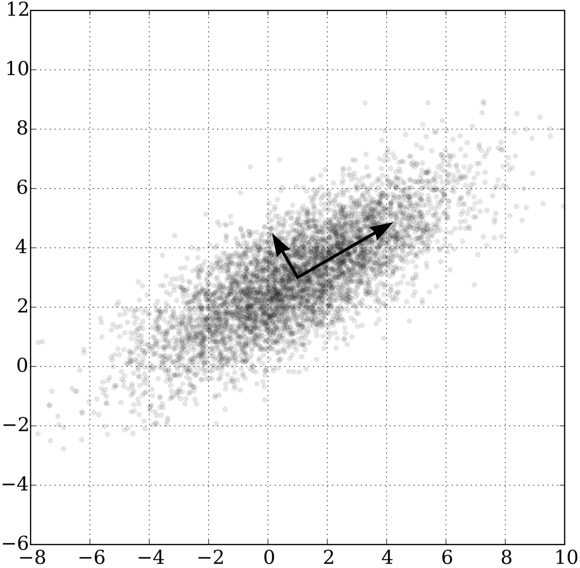
- uncorrelated to what we already have (1st PC);
- maximizes the variance subject to normalization.

$(k+1)$-th pricinpal component

Given the first k principal component $Z_l = \boldsymbol{\xi}_l^T \boldsymbol{x}, l = 1, 2, \ldots, k$. The $(k+1)$-th pricinpal component is solved by

$$\boldsymbol{\xi}_{k+1} = \arg\max_{\boldsymbol{\xi}} \boldsymbol{\xi}^T \boldsymbol{\Sigma} \boldsymbol{\xi} \quad \text{subject to } \|\boldsymbol{\xi}\|_2 = 1 \quad \text{and} \quad \boldsymbol{\xi}^T \boldsymbol{\xi}_l = 0, l = 1, 2, \ldots, k.$$

- Note that the sign of the principal components cannot be determined – both $\pm\boldsymbol{\xi}_{k+1}$ are the solution.
- We shall see that the orthogonality constraint is indeed the uncorrelatedness constraint.


12
10
8
6
4
2
0
−2
−4
−6
−8
−6
−4
−2
0
2
4
6
8
10

PCA is usually solved using eigendecomposition.

- Suppose that $\mathbf{\Sigma}$ has distinct eigenvalues. By induction, it is easy to see that the k-th principal component is exactly the k-th eigenvector of $\mathbf{\Sigma}$.
- Therefore, we apply eigendecomposition of $\mathbf{\Sigma}$

$$\mathbf{\Sigma} = \sum_{j=1}^{p} \lambda_j \boldsymbol{\xi}_j \boldsymbol{\xi}_j^T = \mathbf{\Gamma}\mathbf{\Lambda}\mathbf{\Gamma}^T,$$

  where $\mathbf{\Gamma} = (\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \ldots, \boldsymbol{\xi}_p) \in \mathbb{R}^{n \times p}$ and $\mathbf{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_p)$ with $\lambda_1 \geq \cdots \geq \lambda_k$.
- The k-th principal component is then $Z_k = \boldsymbol{\xi}_k^T \boldsymbol{x}$.
- Orthogonal = uncorrelated. Note that

$$\text{cov}(Z_j, Z_k) = \boldsymbol{\xi}_j^T \mathbf{\Sigma} \boldsymbol{\xi}_k = \lambda_k \boldsymbol{\xi}_j^T \boldsymbol{\xi}_k = \begin{cases} 0 & \text{if } j \neq k, \\ \lambda_k & \text{if } j = k. \end{cases}$$

In practice, one replaces $\boldsymbol{\Sigma}$ by an estimate such as the sample covariance matrix

$$\boldsymbol{S} = n^{-1}\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}^T.$$

- $\boldsymbol{X} \in \mathbb{R}^{n\times p}$ is the data matrix, whose rows are the observations.
- $\tilde{\boldsymbol{X}} \in \mathbb{R}^{n\times p}$ is $\boldsymbol{X}$ but with column means removed.
- n^{-1} or $(n-1)^{-1}$ does not affect PCA.

Apply singular value decomposition (SVD) for $\tilde{\boldsymbol{X}}$

$$\tilde{\boldsymbol{X}} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T.$$

Then

$$\boldsymbol{S} = n^{-1}\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}^T = n^{-1}\boldsymbol{V}\boldsymbol{D}^2\boldsymbol{V}^T$$

- $\boldsymbol{D}^2$ is a diagonal matrix with elements d_i^2, in descending order.
- $\boldsymbol{U} \in \mathbb{R}^{n\times p}$ and $\boldsymbol{V} \in \mathbb{R}^{p\times p}$ are orthogonal matrices.
- The columns $\boldsymbol{v}_k$ of $\boldsymbol{V}$ are the eigenvectors of $\boldsymbol{S}$.
- The k-th (estimated) pricipal component is $\boldsymbol{Z}_k = \boldsymbol{v}_k^T\boldsymbol{X}^T \in \mathbb{R}^n$.

Remarks

- $\boldsymbol{v}_k$ is called the loading of the k-th PC.
- d_i^2 measures the importance of the k-th PC.
- $d_i^2/\sum_{i=1}^p d_i^2$ is interpreted as proportion of the total variation explained by z_k.
- **Dimensionality reduction:** Usually retain the first few PCs.
- PCs are uncorrelated with each other.

Cluster Analysis

Suppose we have a set of data, where each of the observation is assigned a class label. However, the labels are unobservable (these are called latent variables).

- Cluster ananlysis: The goal is to group the data into several clusters such that each cluster is considered as a homogeneous subpopulation.
- Let $\mathcal{C}_k$ denote the k-th cluster. Suppose we have defined a function L that measures the "dissimilarity" of a cluster, then

$$\mathcal{C}^* = \arg\min_{\mathcal{C}} \sum_k L(\mathcal{C}_k).$$

Example: Dissimilarity measure.

- The dissimilarity between $\boldsymbol{x}_i$ and $\boldsymbol{x}_l$ can be measured by

$$D(\boldsymbol{x}_i, \boldsymbol{x}_l) = \sum_{j=1}^{p} w_j d_j(x_{ij}, x_{lj}),$$

  - w_j is the weight assigned to the j-th variable.
  - For continuous variables, we may use the squared distance $d_j(x_{ij}, x_{lj}) = (x_{ij} - x_{lj})^2$.
  - If the variable is not continuous, we may use the Hamming distance $d_j(x_{ij}, x_{lj}) = \mathbb{1}(x_{ij} \neq x_{lj})$.

- The following is an example of a dissimilarity measure

$$L(\mathcal{C}_k) = \frac{1}{2} \sum_{i \neq i' \in \mathcal{C}_k} D(\boldsymbol{x}_i, \boldsymbol{x}_{i'}).$$

Half here because $D(\boldsymbol{x}_i, \boldsymbol{x}_{i'})$ and $D(\boldsymbol{x}_{i'}, \boldsymbol{x}_i)$ are both counted.

Ideally, we would search for the clusters $\mathcal{C}_k$ such that the dissimilarity measure is minimized.

- However, finding such an optimal set of clusters is feasible only for small dataset.
- The number of possible assignment is prohibitively large due to the combinatorial nature. E.g. for $n = 19$ and $K = 4$, there are $\approx 10^{10}$ of them!

As a remedy, one may rely on iterative greedy descent to obtain an approximation to the optimal clustering.

k-Means Clustering

Consider the following dissimilarity measure

$$L(\mathcal{C}) \overset{\text{def.}}{=} \sum_{k} L(\mathcal{C}_k) = \sum_{k=1}^{K} \frac{1}{2|\mathcal{C}_k|} \sum_{i \neq i' \in \mathcal{C}_k} \|\boldsymbol{x}_i - \boldsymbol{x}_{i'}\|_2^2 = \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k} \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k\|_2^2,$$

where $\bar{\boldsymbol{x}}_k = \frac{1}{|\mathcal{C}_k|} \sum_{i \in \mathcal{C}_k} \boldsymbol{x}_i$.

- The last equality can be proved by the identity

$$\sum_{i \in \mathcal{C}_k} \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k\|_2^2 = \sum_{i \neq i' \in \mathcal{C}_k} (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k)^T (\bar{\boldsymbol{x}}_k - \boldsymbol{x}_{i'}).$$

Note that, for any given $\mathcal{C}_k$,

$$\widehat{\boldsymbol{\mu}}_k \overset{\text{def.}}{=} \bar{\boldsymbol{x}}_k = \arg\min_{\boldsymbol{\mu}} \sum_{i \in \mathcal{C}_k} \|\boldsymbol{x}_i - \boldsymbol{\mu}\|_2^2$$

Hence we can enlarge the optimization problem

$$\left(\mathcal{C}^*, \{\boldsymbol{\mu}_k^*\}_{k=1}^K\right) = \arg\min_{\mathcal{C}, \{\boldsymbol{\mu}_k\}} \sum_{k=1}^K \sum_{i \in \mathcal{C}_k} \|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|_2^2,$$

This is naturally solved by an alternating optimization procedure.

k-means clustering

1. Randomly choose K initial centroids $\{\boldsymbol{\mu}_k\}_{k=1}^K$.
2. For a given $\{\boldsymbol{\mu}_k\}_{k=1}^K$, minimize $\min_{\mathcal{C}} \sum_{k=1}^K \sum_{i:\mathcal{C}(i)=k} \|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|_2^2$ yields
$$\mathcal{C}(i) = \arg\min_k \|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|_2^2.$$
3. For a given $\mathcal{C}$, update the cluster centroids by the mean of the current cluster
$$\widehat{\boldsymbol{\mu}}_k = \arg\min_{\boldsymbol{\mu}} \sum_{i:\mathcal{C}(i)=k} \|\boldsymbol{x}_i - \boldsymbol{\mu}\|_2^2 = \bar{\boldsymbol{x}}_k.$$
4. Repeat step 2 and 3 until assignment do not change.
5. Try multiple initial values, pick the solution with the best objective value.

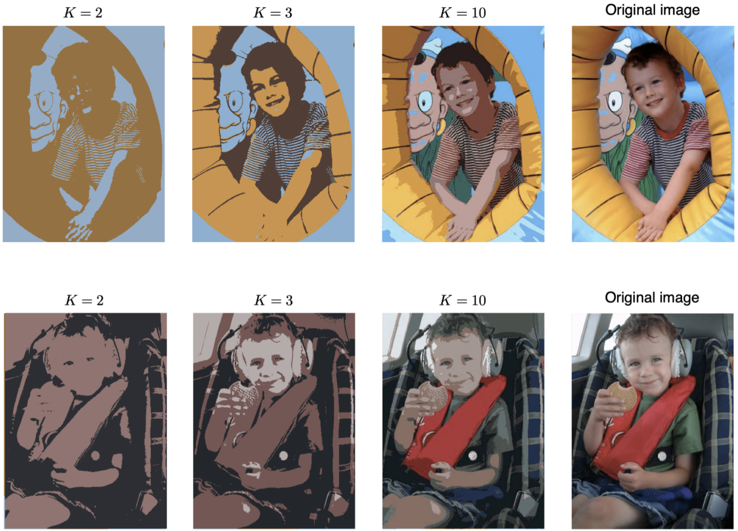
Remark on Section 14.3.6 of ESL.

- ESL claim that the k-means algorithm minimizes (14.31), which in our notation is

$$L(\mathcal{C}) = \frac{1}{2}\sum_{k=1}^{K}\sum_{i\neq i'\in\mathcal{C}_k}\|\boldsymbol{x}_i - \boldsymbol{x}_{i'}\|_2^2 = \sum_{k=1}^{K}|\mathcal{C}_k|\sum_{i\in\mathcal{C}_k}\|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k\|_2^2$$

- The equality still holds without problem, but it does not lead to the k-mean algorithm.

Example: Image compression. Each observation is a pixel with RGB channels.



Remarks – k-Mean

- The objective function is non-increasing in each iteration.
- The convergence is very fast.
- However, it converges only to a local minima. The algorithm depends sensitively on the choice of initial values.
- Hence, it is recommended to try multiple initial values, pick the solution with the best objective value.
- There are also sophisticated initialization methods, e.g., k-means++ from (Arthur and Vassilvitskii, 2007).

k-means++ clustering

1. Randomly choose the first initial centroids $\boldsymbol{\mu}_1$.
2. For each data point $\boldsymbol{x}$ not chosen yet, compute $D(\boldsymbol{x})$, the distance between $\boldsymbol{x}$ and the nearest center that has already been chosen.
3. Choose one new data point as a new center, with probability proportional to $D^2(\boldsymbol{x})$.
4. Repeat step 2 and 3 until k initial centroids have been chosen.
5. Proceed with k-means clustering with the chosen centroids.

Model-Based Clustering

We can take a probabilistic approach to clustering. To this end, we assume that the observations are independent and identically distributed according to some finite mixture model with K components

$$f(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k f_k(\boldsymbol{x}; \boldsymbol{\theta}_k).$$

- Each component f_k is the pdf of a relatively simpler distribution with unknown parameter $\boldsymbol{\theta}_k$.
- The interpretation of the mixture model: classification without class labels. Imagine that the class labels are generated according to $\boldsymbol{\pi} = \{\pi_1, \dots, \pi_K\}$. If the label is $\xi_i = k$, the data $\boldsymbol{X}$ are generated according to the distribution f_k. However, the labels are not actually observed. Instead, only $\boldsymbol{X}_i$ are observed.

Given the finite mixture model, the conditional distribution of ξ given $\boldsymbol{X}$ is multinomial with

$$\mathbb{P}(\xi = k \mid \boldsymbol{X} = \boldsymbol{x}) = \frac{\pi_k f_k(\boldsymbol{x}; \boldsymbol{\theta}_k)}{\sum_{l=1}^{K} \pi_l f_l(\boldsymbol{x}; \boldsymbol{\theta}_k)}.$$

The Bayes rule classifies $\boldsymbol{X} = \boldsymbol{x}$ into

$$\mathcal{C}_{\mathsf{Bayes}}(\boldsymbol{x}) = \arg\max_k \mathbb{P}(\xi = k \mid \boldsymbol{X} = \boldsymbol{x}).$$

Suppose we can estimate $\mathbb{P}(\xi = k \mid \boldsymbol{X} = \boldsymbol{x})$, then a natural clustering algorithm follows from the Bayes rule.

Gaussian mixture model

If each f_k is assumed to be the density of a p-dimensional Gaussian distribution with mean $\boldsymbol{\mu}_k$ and covariance $\boldsymbol{\Sigma}_k$, denoted as $\phi(\cdot; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, then it is called a Gaussian mixture model.

Let $\boldsymbol{\theta} = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, k = 1, \ldots, K\}$, the log-likelihood function is

$$l(\boldsymbol{\theta}; \boldsymbol{X}) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \pi_k \phi(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right).$$

Due to the summation inside the logarithm, $l(\boldsymbol{\theta})$ is non-convex, and the maximum likelihood estimator for $\boldsymbol{\theta}$ cannot be obtained directly.

Expectation-Maximization Algorithm

One popular way to (approximately) compute the MLE is the expectation-maximization (EM) algorithm (Dempster, Laird and Rubin, 1977).

- Introduce n independent multinomial random variable ξ_i such that $\mathbb{P}(\xi_i = k) = \pi_k$ and $(\boldsymbol{X}_i \mid \xi_i = k) \sim N(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$.
- $\{\xi_i\}_{i=1}^n$ are called the missing data or latent variables.
- $\{(\xi_i, \boldsymbol{X}_i)\}_{i=1}^n$ are called the complete data.

The log-likelihood function of the complete data is

$$l_{\text{full}}(\boldsymbol{\theta}; \boldsymbol{X}, \boldsymbol{\xi}) = \sum_{i=1}^{n} \sum_{k=1}^{K} \left[\log \pi_k + \log\left(\phi(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right)\right] \mathbb{1}(\xi_i = k).$$

By considering the complete data, we no longer have summation inside the logarithm, but a new problem arises: we cannot observer ξ_i.

The EM algorithm starts with a given arbitrary initial guess of $\boldsymbol{\theta}^{(0)}$. (The number of clusters K is assumed to be known.)

- At step t, calculate the responsibility $r_{ik}^{(t)}$ of the k-th Gaussian cluster to observation i based on our current guess of the parameter $\boldsymbol{\theta}^{(t)}$

$$r_{ik}^{(t)} = r_{ik}(\boldsymbol{\theta}^{(t)}) = P(\xi_i = k | \boldsymbol{X}_i, \boldsymbol{\theta}^{(t)}) = \frac{\pi_k^{(t)} \phi(\boldsymbol{X}_i; \boldsymbol{\mu}_k^{(t)}, \boldsymbol{\Sigma}_k^{(t)})}{\sum_{l=1}^{K} \pi_l^{(t)} \phi(\boldsymbol{X}_i; \boldsymbol{\mu}_l^{(t)}, \boldsymbol{\Sigma}_l^{(t)})}.$$

- EM uses the responsibility to make a "soft" assignment of the observations to the Gaussian clusters, i.e., $r_{ik}^{(t)}$ indicates how strongly observation i belongs to cluster k, under our current estimation of $\boldsymbol{\theta}^{(t)}$.

- **Expectation step.** Use the conditional probability $\boldsymbol{r}^{(t)} = \{r_{ik}^{(t)}\}$ (which depend on $\boldsymbol{\theta}^{(t)}$) to compute the following conditional expectation:

$$
\begin{aligned}
Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(t)}) &= \mathbb{E}_{\boldsymbol{\xi}\sim\boldsymbol{r}^{(t)}}[l_{\text{full}}(\boldsymbol{\theta};\boldsymbol{X},\boldsymbol{\xi})|\boldsymbol{X}] \\
&= \mathbb{E}_{\boldsymbol{\xi}\sim\boldsymbol{r}^{(t)}}\left[\sum_{i=1}^{n}\sum_{k=1}^{K}[\log\pi_k + \log(\phi(\boldsymbol{X}_i;\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k))]\,\mathbb{1}(\xi_i = k)\right] \\
&= \sum_{i=1}^{n}\sum_{k=1}^{K}[\log\pi_k + \log(\phi(\boldsymbol{X}_i;\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k))]\,r_{ik}^{(t)}.
\end{aligned}
$$

- **Maximization step.** Update $\boldsymbol{\theta}^{(t+1)}$ with the maximizer of $Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(t)})$, i.e.,

$$
\begin{aligned}
\boldsymbol{\theta}^{(t+1)} &= \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(t)}) \\
&= \arg\max_{\{\pi_k,\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\}_{k=1}^{K}} \sum_{i=1}^{n}\sum_{k=1}^{K}[\log\pi_k + \log(\phi(\boldsymbol{X}_i;\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k))]\,r_{ik}^{(t)}
\end{aligned}
$$

- **Parameter update in the M-step.** Note that $r_{ik}^{(t)}$ is known, and the maximization problem have explicit solution

$$\pi_k^{(t+1)} = \frac{1}{n}\sum_{i=1}^{n} r_{ik}^{(t)},$$

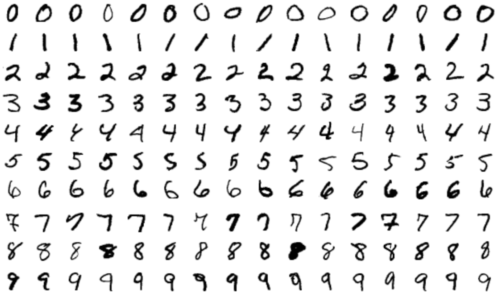
$$\boldsymbol{\mu}_k^{(t+1)} = \frac{\sum_{i=1}^{n} r_{ik}^{(t)} \boldsymbol{X}_i}{\sum_{i=1}^{n} r_{ik}^{(t)}},$$

$$\boldsymbol{\Sigma}_k^{(t+1)} = \frac{\sum_{i=1}^{n} r_{ik}^{(t)} \left(\boldsymbol{X}_i - \boldsymbol{\mu}_k^{(t+1)}\right)\left(\boldsymbol{X}_i - \boldsymbol{\mu}_k^{(t+1)}\right)^T}{\sum_{i=1}^{n} r_{ik}^{(t)}}.$$

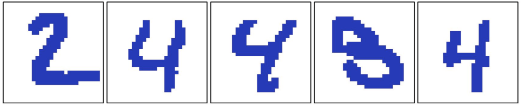
- EM algorithm iterates between E-step and M-step until convergence.
  - One can show that the likelihood is monotonically increasing, hence EM always converges. But it is not guaranteed to converge to the global maximum.

- **Clustering.** Denote $\widehat{\boldsymbol{\theta}}$ as the EM estimate of the parameters. For any observation $\boldsymbol{x}$, the maximum likelihood estimate of the probability that $\boldsymbol{x}$ belongs to cluster k is
$$\widehat{r}_{ik}(\boldsymbol{x}) = \mathbb{P}(\xi_i = k|\boldsymbol{X} = \boldsymbol{x}, \widehat{\boldsymbol{\theta}}) = \frac{\widehat{\pi}_k \phi(\boldsymbol{x}; \widehat{\boldsymbol{\mu}}_k, \widehat{\boldsymbol{\Sigma}}_k)}{\sum_{l=1}^{K} \widehat{\pi}_l \phi(\boldsymbol{x}; \widehat{\boldsymbol{\mu}}_l, \widehat{\boldsymbol{\Sigma}}_l)}.$$
We assign $\boldsymbol{x}$ to cluster $\mathcal{C}_k^*$ for
$$k^* = \arg\max_k \widehat{r}_{ik}(\boldsymbol{x}).$$

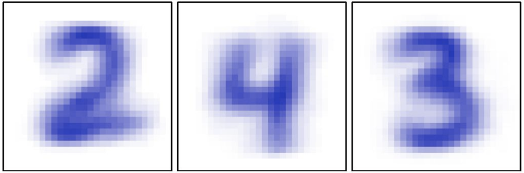
Example: Handwriting recognition using Bernoulli Mixture Model.



Training data



Mean estimations from EM



Remarks – EM Algorithm

- EM algorithm provides only a local maximizer of the likelihood function.
- The outcome of EM algorithm depends on the initial values. Hence it is recommended to try a few initial values and use the one that produced the largest likelihood.
- Sophisticated **initialization** can help EM converge to a better solution, e.g., analogous to k-means$++$.
- Consider a special case with $\boldsymbol{\Sigma}_k = \sigma^2 \boldsymbol{I}$. Then the k-means clustering can be asymptotically recovered when $\sigma^2 \to 0$.
- **Regularization** methods (ridge, lasso, SCAD...) can be applied to EM to achieve variable shrinkage or selection. In that case, the parameter update may not have a closed-form solution, hence additional iterative steps are needed for each EM step.

Reference

- The Element of Statistical Learning, Section 8.5, 14.3, 14.5.
- Statistical Foundations of Data Science, Section 10.1, 13.1.