Lecture 14: Unsupervised Learning

Introduction to Machine Learning [25737]

Sajjad Amini

Sharif University of Technology

Contents

References

The material in the slides except cited are inspired from the following reference:

- Murphy, K. P. (2022). *Probabilistic machine learning: an introduction.* MIT press.

Section 1

Approach Definition

Unsupervised Learning

Principle Component Analysis

- Experience E: Set of N samples $\mathcal{D} = \{\boldsymbol{x}_n\}_{n=1}^{N}$
- Task T: Projecting data into low dimensional subspace which captures its main aspects
- Performance measure: Preserving data variations

Clustering

- Experience E: Set of N samples $\mathcal{D} = \{\boldsymbol{x}_n\}_{n=1}^{N}$
- Task T: Partition the input into regions that contains *similar* points.
- Performance measure in *Compression*: Compression loss

Section 2

Principle Component Analysis

Subsection 1

Interpretation Via Maximum Projection Spread

Frame Title

Data Matrix

- Assume $\boldsymbol{x} \in \mathbb{R}^D$ is a random variable and you have observed N copies of it as $\{\boldsymbol{x}_i\}_{i=1}^N$ (Equivalently the dataset).
- As before, we stack these copies into a Matrix $\boldsymbol{X}$ as:

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1^T \\ \boldsymbol{x}_2^T \\ \dots \\ \boldsymbol{x}_N^T \end{bmatrix} = \begin{bmatrix} x_1^1 & \dots & x_1^D \\ x_2^1 & \dots & x_2^D \\ \dots & & \\ x_N^1 & \dots & x_N^D \end{bmatrix} \in \mathbb{R}^{N \times D}$$

  - Each column is a feature (covariate or predictor)
  - Each row is an observation

Characterizing Dataset [1]

Characterizing Dataset

The dataset point create a point cloud in $\mathbb{R}^D$ space.

- The expectation of this point cloud, calculated below, determines the center of point cloud.

$$\mathbb{E}[\boldsymbol{x}] = \begin{bmatrix} \mathbb{E}[x^1] \\ \vdots \\ \mathbb{E}[x^D] \end{bmatrix}$$

- The covariance matrix of this point cloud, calculated below, determines the spread of point cloud.

$$\begin{aligned} \mathrm{Cov}[\boldsymbol{x}] &\triangleq \mathbb{E}\left[(\boldsymbol{x} - \mathbb{E}[\boldsymbol{x}])(\boldsymbol{x} - \mathbb{E}[\boldsymbol{x}])^T\right] = \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^T] - \mathbb{E}[\boldsymbol{x}]\mathbb{E}[\boldsymbol{x}]^T = \boldsymbol{\Sigma} \\ &= \begin{bmatrix} \mathrm{Cov}[X_1, X_1] & \mathrm{Cov}[X_1, X_2] & \cdots & \mathrm{Cov}[X_1, X_D] \\ \mathrm{Cov}[X_2, X_1] & \mathrm{Cov}[X_2, X_2] & \cdots & \mathrm{Cov}[X_2, X_D] \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}[X_D, X_1] & \mathrm{Cov}[X_D, X_2] & \cdots & \mathrm{Cov}[X_D, X_D] \end{bmatrix} \end{aligned}$$

Characterizing Dataset [1]

Utilizing Empirical Distribution

The empirical distribution for dataset $\{\boldsymbol{x}_i\}_{i=1}^N$ is defined as:

$$p_D(\boldsymbol{x}) = \frac{1}{N}\sum_{n=1}^{N}\delta(\boldsymbol{x} - \boldsymbol{x}_n)$$

We can use it to compute the empirical (sample) mean and empirical (sample) covariance matrix as:

$$\mathbb{E}_D[\boldsymbol{x}] = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i = \bar{\boldsymbol{x}}$$

$$\underset{D}{\mathrm{Cov}}[\boldsymbol{x}] = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i\boldsymbol{x}_i^T - \bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^T = \boldsymbol{S}$$

Characterizing Dataset [1]

Eliminating Summation Using Linear Algebra

For sample mean, we have:

$$\bar{\boldsymbol{x}} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i = \frac{1}{N}\begin{bmatrix} | & | & | \\ \boldsymbol{x}_1 & \dots & \boldsymbol{x}_N \\ | & | & | \end{bmatrix}_{D\times N}\begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{N\times 1} = \frac{1}{N}\boldsymbol{X}^T\mathbf{1}$$

For sample covariance matrix, we use forth method for matrix multiplication as:

$$\boldsymbol{S} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i\boldsymbol{x}_i^T - \bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^T = \frac{1}{N}\begin{bmatrix} | & | & | \\ \boldsymbol{x}_1 & \dots & \boldsymbol{x}_N \\ | & | & | \end{bmatrix}\begin{bmatrix} - & \boldsymbol{x}_1^T & - \\ & \vdots & \\ - & \boldsymbol{x}_N^T & - \end{bmatrix} - \bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^T$$

$$= \frac{1}{N}\boldsymbol{X}^T\boldsymbol{X} - \frac{1}{N^2}\boldsymbol{X}^T\mathbf{1}\mathbf{1}^T\boldsymbol{X} = \frac{1}{N}\boldsymbol{X}^T\underbrace{\left(\boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right)}_{\boldsymbol{H}}\boldsymbol{X}$$

Characterizing Dataset [1]

Idempotent Matrix

Matrix $\boldsymbol{P}$ is said to be idempotent matrix if $\boldsymbol{P}^2 = \boldsymbol{P}$

Projection Matrix

Matrix $\boldsymbol{Z}$ is said to be projection matrix if it is symmetric and idempotent.

Working on $\boldsymbol{H}$ Matrix

Matrix $\boldsymbol{H}$ is a projection matrix because:

- $\boldsymbol{H}^T = (\boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T)^T = \boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T$
- Idempotent property:

$$\begin{aligned}\boldsymbol{H}^2 =& \boldsymbol{H}\boldsymbol{H} = (\boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T)(\boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T) = \boldsymbol{I} - \frac{2}{N}\mathbf{1}\mathbf{1}^T + \frac{1}{N^2}\mathbf{1}\overbrace{\mathbf{1}^T\mathbf{1}}^{N}\mathbf{1}^T \\ =& \boldsymbol{I} - \frac{2}{N}\mathbf{1}\mathbf{1}^T + \frac{1}{N}\mathbf{1}\mathbf{1}^T = \boldsymbol{H}\end{aligned}$$

Characterizing Dataset [1]

Characterizing the Projection $\boldsymbol{H}$

Assume $\boldsymbol{v} \in \mathbb{R}^N$, then:

$$\boldsymbol{H}\boldsymbol{v} = \boldsymbol{v} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\boldsymbol{v} = \boldsymbol{v} - \frac{\mathbf{1}^T\boldsymbol{v}}{N}\mathbf{1} = \boldsymbol{v} - \bar{\boldsymbol{v}}$$

Thus $\boldsymbol{H}$ removes the mean of the vector from each coordinate. Equivalently $\overline{\boldsymbol{H}\boldsymbol{v}} = \mathbf{0}$

Thus $\boldsymbol{H}$ is the projection onto the subspace of vectors with zero mean (Projection onto hyperplane which is orthogonal to $\mathbf{1}$ vector).

Re-writing $\boldsymbol{S}$

Based on the projection matrix $\boldsymbol{H}$, we have:

$$\boldsymbol{S} = \frac{1}{N}\boldsymbol{X}^T\boldsymbol{H}\boldsymbol{X} = \frac{1}{N}\boldsymbol{X}^T\boldsymbol{H}^2\boldsymbol{X} = \frac{1}{N}\boldsymbol{X}^T\boldsymbol{H}^T\boldsymbol{H}\boldsymbol{X} = \frac{1}{N}(\boldsymbol{H}\boldsymbol{X})^T(\boldsymbol{H}\boldsymbol{X})$$

where $\boldsymbol{H}\boldsymbol{X}$ result in centered features.

Linear Combination of Features [1]

Original Formulation

Assume an arbitrary direction of $\boldsymbol{u} \in \mathbb{R}^D$, then consider the following value:

$$\begin{aligned}\boldsymbol{u}^T \boldsymbol{\Sigma} \boldsymbol{u} =& \boldsymbol{u}^T \left[\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^T] - \mathbb{E}[\boldsymbol{x}]\mathbb{E}[\boldsymbol{x}^T]\right] \boldsymbol{u} \overset{(a)}{=} \mathbb{E}[(\boldsymbol{u}^T\boldsymbol{x})(\boldsymbol{u}^T\boldsymbol{x})^T] - \mathbb{E}[\boldsymbol{u}^T\boldsymbol{x}]\mathbb{E}[(\boldsymbol{u}^T\boldsymbol{x})^T] \\ =& \mathbb{E}[(\boldsymbol{u}^T\boldsymbol{x})^2] - \mathbb{E}[\boldsymbol{u}^T\boldsymbol{x}]^2 = \mathrm{var}(\boldsymbol{u}^T\boldsymbol{x})\end{aligned}$$

Switching to Empirical Distribution

Using empirical distribution, we have the empirical variance for $\{\boldsymbol{u}^T\boldsymbol{x}_i\}_{i=1}^N$:

$$\begin{aligned}\boldsymbol{u}^T \boldsymbol{S} \boldsymbol{u} =& \boldsymbol{u}^T \left[\frac{1}{N}\sum_{i=1}^N \boldsymbol{x}_i^T\boldsymbol{x}_i - \overline{\boldsymbol{x}}\overline{\boldsymbol{x}}^T\right] \boldsymbol{u} = \frac{1}{N}\sum_{i=1}^N (\boldsymbol{u}^T\boldsymbol{x}_i)(\boldsymbol{x}_i^T\boldsymbol{u}) - (\boldsymbol{u}^T\overline{\boldsymbol{x}})(\overline{\boldsymbol{x}}^T\boldsymbol{u}) \\ =& \frac{1}{N}\sum_{i=1}^N (\boldsymbol{u}^T\boldsymbol{x}_i)^2 - (\boldsymbol{u}^T\overline{\boldsymbol{x}})^2 = \frac{1}{N}\sum_{i=1}^N (\boldsymbol{u}^T\boldsymbol{x}_i)^2 - (\overline{\boldsymbol{u}^T\boldsymbol{x}})^2 = s^2\end{aligned}$$

Intuition Behind Principle Component Analysis [1]

Intuition

Finding the direction $\boldsymbol{u}$ which result in the high projection value spread measured by project value variance

Extreme Cases

- Zero variance: The projection of all points onto $\boldsymbol{u}$ is equal (The points are in the hyper-plane whose normal vector is $\boldsymbol{u}$).
- Large variance: The points are spread along the $\boldsymbol{u}$ direction.

Objective

Fining the direction that maximize the projection variance (or equivalently projection spread)

Formulation

The problem for PCA can be formulated as:

$$\max_{\boldsymbol{u}\in\mathbb{R}^D} \boldsymbol{u}^T\boldsymbol{S}\boldsymbol{u}$$

The maximum value of objective function is infinity, thus we need to constrained $\boldsymbol{u}$ as:

$$\max_{\boldsymbol{u}\in\mathbb{R}^D} \boldsymbol{u}^T\boldsymbol{S}\boldsymbol{u} \text{ subject to } \|\boldsymbol{u}\|_2 = 1$$

Spectal Theorem

Eigenvalues and Eigenvectors of Symmetric Matrices

Based on *Spectral Theorem*, for symmetric matrix $\boldsymbol{S}$ we have:

- All eigenvalues are real
- Eigenvectors are orthonormal ($\boldsymbol{U}$ is orthogonal thus $\boldsymbol{P}^{-1} = \boldsymbol{P}^T$)

Then we have:

$$\begin{aligned}\boldsymbol{S} =& \boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{P}^T = \begin{bmatrix} | & | & & | \\ \boldsymbol{p}_1 & \boldsymbol{p}_2 & & \boldsymbol{p}_n \\ | & | & & | \end{bmatrix} \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix} \begin{bmatrix} - & \boldsymbol{p}_1^T & - \\ - & \boldsymbol{p}_2^T & - \\ & \vdots & \\ - & \boldsymbol{p}_m^T & - \end{bmatrix} \\ =& \sum_{i=1}^{n} \lambda_i \boldsymbol{p}_i \boldsymbol{p}_i^T\end{aligned}$$

Covariance Matrices

Covariance matrices are positive semi-definite, equivalently, all their eigenvalues are non-negative ($\boldsymbol{u}^T\boldsymbol{\Sigma}\boldsymbol{u} \geq 0, \forall \boldsymbol{u}$ and $\boldsymbol{u}^T\boldsymbol{S}\boldsymbol{u} \geq 0, \forall \boldsymbol{u}$).

Characterizing $\boldsymbol{S}$

Using Spectral theorem, we can write $\boldsymbol{S}$ as:

$$\boldsymbol{S} = \boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{P}^T, \quad \begin{cases} \boldsymbol{P} = \begin{bmatrix} | & | & & | \\ \boldsymbol{p}_1 & \boldsymbol{p}_2 & & \boldsymbol{p}_D \\ | & | & & | \end{bmatrix} \\ \boldsymbol{\Lambda} = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_D \geq 0 \end{bmatrix}, \lambda_1 \geq \lambda_2 \ldots \lambda_D \\ \boldsymbol{P}^T\boldsymbol{P} = \boldsymbol{I} \end{cases}$$

Characterizing $\boldsymbol{S}$ [1]

Transforming Using Eigenvectors

Assume we define $\boldsymbol{y} = \boldsymbol{P}^T\boldsymbol{x} \in \mathbb{R}^D$ and $\overline{\boldsymbol{x}} = 0$, then:

$$\overline{\boldsymbol{y}} = \overline{\boldsymbol{P}^T\boldsymbol{x}} = \boldsymbol{P}^T\overline{\boldsymbol{x}} = \boldsymbol{0}$$

Thus the sample covariance matrix for $\boldsymbol{y}$ is:

$$\boldsymbol{S}^y = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{P}^T\boldsymbol{x}_i)(\boldsymbol{P}^T\boldsymbol{x}_i)^T = \boldsymbol{P}^T\left(\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i\boldsymbol{x}_i^T\right)\boldsymbol{P} = \boldsymbol{P}^T\boldsymbol{S}\boldsymbol{P} = \boldsymbol{D}$$

Thus we take one step through whitening:

$$\operatorname{cov}(Y^i, Y^j) = \begin{cases} 0 & i \neq j \\ \lambda_i & i = j \end{cases}$$

Finding Maximum Spread Direction

Assume the maximum spread direction is $\boldsymbol{u}$ and consider the following definition:

$$\boldsymbol{b} = \boldsymbol{P}^T\boldsymbol{u} \Rightarrow \boldsymbol{u} = \boldsymbol{P}\boldsymbol{b}$$

Now we measure the spread as:

$$\boldsymbol{u}^T\boldsymbol{S}\boldsymbol{u} = (\boldsymbol{P}\boldsymbol{b})^T(\boldsymbol{P}\boldsymbol{D}\boldsymbol{P}^T)(\boldsymbol{P}\boldsymbol{u}) = \boldsymbol{b}^T\overbrace{(\boldsymbol{P}\boldsymbol{P}^T)}^{\boldsymbol{I}}\boldsymbol{D}\overbrace{(\boldsymbol{P}^T\boldsymbol{P})}^{\boldsymbol{I}}\boldsymbol{b} = \sum_{j=1}^{D}\lambda_j b_j^2 \leq \lambda_1 \overbrace{\sum_{j=1}^{D} b_j^2}^{\|\boldsymbol{b}\|^2}$$

On the other hand, for $\|\boldsymbol{b}\|^2$, e have:

$$\|\boldsymbol{b}\|^2 = \|\boldsymbol{P}^T\boldsymbol{u}\|^2 = (\boldsymbol{P}^T\boldsymbol{u})^T(\boldsymbol{P}^T\boldsymbol{u}) = \boldsymbol{u}^T(\boldsymbol{P}\boldsymbol{P}^T)\boldsymbol{u} = \|\boldsymbol{u}\|^2 = 1$$

Thus:

$$\forall \boldsymbol{u} \in \mathbb{R}^D : \boldsymbol{u}^T\boldsymbol{S}\boldsymbol{u} \leq \lambda_1$$

Finding Maximum Spread Direction [1]

Finding Maximum Spread Direction

We see:

$$\forall \boldsymbol{u} \in \mathbb{R}^D : \boldsymbol{u}^T \boldsymbol{S} \boldsymbol{u} \leq \lambda_1$$

Now check the variance for $\boldsymbol{u} = \boldsymbol{p}_1$:

$$\boldsymbol{b} = \begin{bmatrix} - & \boldsymbol{p}_1^T & - \\ - & \boldsymbol{p}_2^T & - \\ & \vdots & \\ - & \boldsymbol{p}_m^T & - \end{bmatrix} \boldsymbol{p}_1 = \begin{bmatrix} \boldsymbol{p}_1^T \boldsymbol{p}_1 \\ \boldsymbol{p}_2^T \boldsymbol{p}_1 \\ \vdots \\ \boldsymbol{p}_D^T \boldsymbol{p}_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Thus:

$$\boldsymbol{p}_1^T \boldsymbol{S} \boldsymbol{p}_1 = \boldsymbol{b}^T \boldsymbol{D} \boldsymbol{b} = \sum_{j=1}^{D} \lambda_j b_j = \lambda_1$$

And $\boldsymbol{u} = \boldsymbol{p}_1$is the direction of maximum spread.

Fining Next Maximum Spread Directions

Assume $\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_D$ to be the eigenvectors of $\boldsymbol{S}$ matrix corresponding to eigenvalues sorted in the descending order. Then, we have seen:

$$\boldsymbol{p}_1 \in \underset{\|\boldsymbol{u}\|=1}{\operatorname{argmax}} \, \boldsymbol{u}^T \boldsymbol{S} \boldsymbol{u}$$

We can show the following in an almost similar way:

$$\begin{aligned}
\boldsymbol{p}_2 &\in \underset{\|\boldsymbol{u}\|=1, \boldsymbol{u} \perp \boldsymbol{p}_1}{\operatorname{argmax}} \quad \boldsymbol{u}^T \boldsymbol{S} \boldsymbol{u} \\
\boldsymbol{p}_3 &\in \underset{\|\boldsymbol{u}\|=1, \boldsymbol{u} \perp \boldsymbol{p}_i, i=1,2}{\operatorname{argmax}} \quad \boldsymbol{u}^T \boldsymbol{S} \boldsymbol{u} \\
&\vdots \\
\boldsymbol{p}_j &\in \underset{\|\boldsymbol{u}\|=1, \boldsymbol{u} \perp \boldsymbol{p}_k, k=1,\ldots,(j-1)}{\operatorname{argmax}} \quad \boldsymbol{u}^T \boldsymbol{S} \boldsymbol{u}
\end{aligned}$$

Subsection 2

Interpretation Via Reconstruction

PCA Interpretation Using Reconstruction

PCA Interpretation Using Reconstruction

Assume we have a high-dimensional data $\boldsymbol{x} \in \mathbb{R}^D$ and we want to project it to a low dimensional subspace $\boldsymbol{z} \in \mathbb{R}^L$ such that low dimensional representation is a *good representation*. To approach a mathematical formulation, we need:

- A projection (encoding) operator: $\boldsymbol{z} = \text{Encode}(\boldsymbol{x}; \boldsymbol{\theta})$
- An un-projection (decoding) operator: $\widehat{\boldsymbol{x}} = \text{Decode}(\boldsymbol{z}; \boldsymbol{\theta})$
- A goodness measure: $\|\boldsymbol{x} - \widehat{\boldsymbol{x}}\|^2$

Parameters

Parameters

- Representation in the low dimensional space $\boldsymbol{z} \in \mathbb{R}^L$
- Basis functions for reconstruction $\widehat{\boldsymbol{x}} = \sum_{i=1}^{L} z_i \boldsymbol{w}_i$ such that:

$$\boldsymbol{w}_i^T \boldsymbol{w}_j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

Or equivalently if $\boldsymbol{W} = \begin{bmatrix} \boldsymbol{w}_1 & \boldsymbol{w}_2 & \dots & \boldsymbol{w}_L \end{bmatrix} \in \mathbb{R}^{D \times L}$ then:

$$\boldsymbol{W}^T \boldsymbol{W} = \begin{bmatrix} \boldsymbol{w}_1^T \\ \boldsymbol{w}_2^T \\ \vdots \\ \boldsymbol{w}_L^T \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_1 & \boldsymbol{w}_2 & \dots & \boldsymbol{w}_L \end{bmatrix} = \begin{bmatrix} \boldsymbol{w}_1^T \boldsymbol{w}_1 & \boldsymbol{w}_1^T \boldsymbol{w}_2 & \dots & \boldsymbol{w}_1^T \boldsymbol{w}_L \\ \boldsymbol{w}_2^T \boldsymbol{w}_1 & \boldsymbol{w}_2^T \boldsymbol{w}_2 & \dots & \boldsymbol{w}_2^T \boldsymbol{w}_L \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{w}_L^T \boldsymbol{w}_1 & \boldsymbol{w}_L^T \boldsymbol{w}_2 & \dots & \boldsymbol{w}_L^T \boldsymbol{w}_L \end{bmatrix} = \boldsymbol{I}$$

Altogether

PCA Interpretation Using Reconstruction

You are given a dataset $\{\boldsymbol{x}_i\}_{i=1}^N$ in $\mathbb{R}^D$. You should design $\boldsymbol{W} \in \mathbb{R}^{D\times L}$ and $\{\boldsymbol{z}_i\}_{i=1}^N$ using the following problem:

$$\min_{\boldsymbol{W},\{\boldsymbol{z}_k\}} \frac{1}{N}\sum_{i=1}^N \|\boldsymbol{x}_i - \boldsymbol{W}\boldsymbol{z}_i\|_2^2$$

Basic Problem $L = 1$

Simplifying the Loss

In this case, the loss function is:

$$\begin{aligned}\mathcal{L}(\boldsymbol{w}_1, \{z_k^1\}) =& \frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{x}_i - z_i^1 \boldsymbol{w}_1\|^2 = \frac{1}{N}\sum_{i=1}^{N} (\boldsymbol{x}_i - z_i^1 \boldsymbol{w}_1)^T (\boldsymbol{x}_i - z_i^1 \boldsymbol{w}_1) \\ =& \frac{1}{N}\sum_{i=1}^{N} \left[\boldsymbol{x}_i^T \boldsymbol{x}_i - 2 z_i^1 \boldsymbol{w}_1^T \boldsymbol{x}_i + (z_i^1)^2 \overbrace{\boldsymbol{w}_1^T \boldsymbol{w}_1}^{=1}\right] \\ =& \frac{1}{N}\sum_{i=1}^{N} \left[\boldsymbol{x}_i^T \boldsymbol{x}_i - 2 z_i^1 \boldsymbol{w}_1^T \boldsymbol{x}_i + (z_i^1)^2\right]\end{aligned}$$

Basic Problem $L = 1$

Derivative w.r.t. Representation

$$\frac{\partial \mathcal{L}(\boldsymbol{w}_1, \{z_k^1\}}{\partial z_n^1} = \frac{1}{N}\left[-2\boldsymbol{w}_1^T\boldsymbol{x}_n + 2z_n^1\right] = 0 \Rightarrow z_n^1 = \boldsymbol{w}_1^T\boldsymbol{x}_n$$

Updating Loss Function

$$\mathcal{L}(\boldsymbol{w}_1) = \frac{1}{N}\sum_{i=1}^{N}\left[\boldsymbol{x}_n^T\boldsymbol{x}_n - (z_i^1)^2\right] = \text{const} - \frac{1}{N}\sum_{i=1}^{N}(z_i^1)^2$$

Dropping the constant term, we have:

$$\mathcal{L}(\boldsymbol{w}_1) = -\frac{1}{N}\sum_{i=1}^{N}(z_i^1)^2 = -\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{w}_1^T\boldsymbol{x}_i\boldsymbol{x}_i^T\boldsymbol{w}_1 = -\boldsymbol{w}_1^T\boldsymbol{S}\boldsymbol{w}_1$$

Note that in the above, we assumed the empirical mean vector to be zero ($\overline{\boldsymbol{x}} = \boldsymbol{0}$)

Basic Problem $L = 1$

Solving for $\boldsymbol{w}_1$

We have the following optimization problem:

$$\min_{\boldsymbol{w}_1} \ \boldsymbol{w}_1^T \boldsymbol{S} \boldsymbol{w}_1 \ \text{ subject to } \boldsymbol{w}_1^T \boldsymbol{w}_1 = 1$$

Thus we form the Lagrangian as:

$$\widetilde{\mathcal{L}}(\boldsymbol{w}_1) = \boldsymbol{w}_1^T \boldsymbol{S} \boldsymbol{w}_1 - \lambda_1 (\boldsymbol{w}_1^T \boldsymbol{w}_1 - 1)$$

The partial derivative for the Lagrangian is:

$$\frac{\partial}{\partial \boldsymbol{w}_1} \widetilde{\mathcal{L}}(\boldsymbol{w}_1) = 2\boldsymbol{S}\boldsymbol{w}_1 - 2\lambda_1 \boldsymbol{w}_1 = 0 \Rightarrow \boldsymbol{S}\boldsymbol{w}_1 = \lambda_1 \boldsymbol{w}_1$$

Thus $(\lambda_1, \boldsymbol{w}_1)$ is a pair of (eigenvalue,eigenvector). But which of them?

$$\boldsymbol{w}_1^T \boldsymbol{S} \boldsymbol{w}_1 = \boldsymbol{w}_1^T \boldsymbol{w}_1 = \lambda_1$$

Thus $\boldsymbol{w}_1$ is the direction of eigenvector corresponding to largest eigenvalue.

General Case

Assume we want to find $\boldsymbol{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_L]$ and $\boldsymbol{z} = [z^1, \ldots, z^L]$. Then we have the following problem:

$$\mathcal{L}(\boldsymbol{W}, \{\boldsymbol{z}_k\}) = \frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{x}_i - \sum_{j=1}^{L} z_i^j \boldsymbol{w}_j\|^2$$

And the solution is:

$$\boldsymbol{w}_i = \boldsymbol{p}_i,\ \ i = 1, \ldots, L$$
$$z_i^j = \boldsymbol{p}_j^T \boldsymbol{x}_i, \begin{cases} i = 1, \ldots, N \\ j = 1, \ldots, L \end{cases}$$

where $\{\boldsymbol{p}_i\}$ is the set of eigenvector for $\boldsymbol{S}$ matrix corresponding to eigenvalues sorted in descending order.

Encoding and Decoding

Encoding

$$\mathbb{R}^L \ni \boldsymbol{z} = \text{Encode}(\boldsymbol{x}, \boldsymbol{W}) = \boldsymbol{W}^T\boldsymbol{x} = \begin{bmatrix} \boldsymbol{w}_1^T \\ \boldsymbol{w}_2^T \\ \vdots \\ \boldsymbol{w}_L^T \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} \boldsymbol{w}_1^T\boldsymbol{x} \\ \boldsymbol{w}_2^T\boldsymbol{x} \\ \vdots \\ \boldsymbol{w}_L^T\boldsymbol{x} \end{bmatrix} = \begin{bmatrix} z^1 \\ z^2 \\ \vdots \\ z^L \end{bmatrix}$$

Decoding

$$\mathbb{R}^D \ni \widehat{\boldsymbol{x}} = \text{Decode}(\boldsymbol{x}, \boldsymbol{W}) = \boldsymbol{W}\boldsymbol{z} = \begin{bmatrix} \boldsymbol{w}_1 & \boldsymbol{w}_2 & \dots & \boldsymbol{w}_L \end{bmatrix} \begin{bmatrix} z^1 \\ z^2 \\ \vdots \\ z^L \end{bmatrix} = \sum_i z^i \boldsymbol{w}_i$$

Section 3

Clustering

Clustering

- Experience E: Set of N samples $\mathcal{D} = \{\boldsymbol{x}_n\}_{n=1}^{N}$
- Task T: Partition the input into regions that contains *similar* points.
- Performance measure in *Compression*: Compression loss

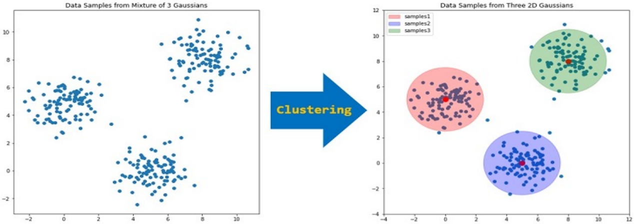

Figure: Sample GMM distribution

Section 4

Mixture Models

Mixture Models

Mixture Models

One way to create more complex probability models is to take a convex combination of simple distributions. This is called a mixture model. This has the form $p(\boldsymbol{y}|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k p_c(\boldsymbol{y}|\boldsymbol{\theta}_k)$ where:

- $p_c(\cdot|\boldsymbol{\theta}_k)$ is the k-th mixture component
- $\{\pi_k\}_{k=1}^{K}$ are mixture weights with the following constraints:
  - $0 \leq \pi_k \leq 1, k = 1, \ldots, K$
  - $\sum_{k=1}^{K} \pi_k = 1$

Mixture Models - Generative Story

Suppose latent variable z to be a categorical RV and distributed as $p(z|\boldsymbol{\theta}) = Cat(z|\boldsymbol{\pi})$ and conditional $p(\boldsymbol{y}|z = k, \boldsymbol{\theta}) = p_c(\boldsymbol{y}|\boldsymbol{\theta}_k)$. We can interpret mixture models as follows:

- We sample a specific component.
- We generate $\boldsymbol{y}$ using sampled value of z.

Using the above procedure, we have:

$$p(\boldsymbol{y}|\boldsymbol{\theta}) = \sum_{k=1}^{K} p(z = k|\boldsymbol{\theta}) p(\boldsymbol{y}|z = k, \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k p(\boldsymbol{y}|\boldsymbol{\theta}_k)$$

Gaussian Mixture Model

Gaussian Mixture Model (GMM) or Mixture of Gaussian (MoG) is defined as:

$$p(\boldsymbol{y}|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

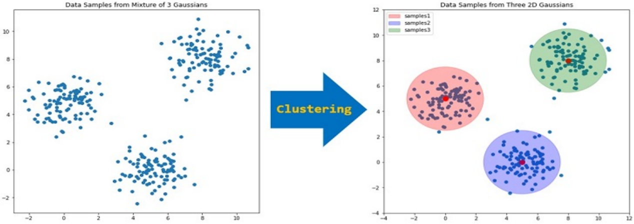

Figure: Sample GMM distribution

Maximum Likelihood Approach to Clustering

Problem Formulation

- Observed data samples $\{\boldsymbol{x}_i\}_{i=1}^{n}$
- Unobserved mixture element corresponding to each data sample $\{z_i\}_{i=1}^{N}$

Using the above two formulation, the complete dataset likelihood is:

$$p(\mathcal{D}|\boldsymbol{\theta}) = p(\{\boldsymbol{x}_i\}, \{z_i\}|\boldsymbol{\theta})$$

The marginal likelihood of dataset is:

$$p(\{\boldsymbol{x}_i\}|\boldsymbol{\theta}) = \sum_{\{z_i\}} p(\{\boldsymbol{x}_i\}, \{z_i\}|\boldsymbol{\theta})$$

and the maximum likelihood estimation for $\theta = \{\boldsymbol{\theta}_1, \ldots, \theta_K, \boldsymbol{\pi}\}$ can be calculated as:

$$\widehat{\boldsymbol{\theta}}_{mle} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, p(\{\boldsymbol{x}_i\}|\boldsymbol{\theta})$$

Challenge and Solution

Challenge

As the scale of the problem increases (dimension of $\boldsymbol{x}$ and number of dataset sample N), it becomes computationally intractable to exactly evaluate (or even optimize) the marginal likelihood.

Solution

One solution is to use expectation maximization algorithm as:

- Initialize $\boldsymbol{\theta}$ randomly (or by using problem-specific heuristics) as $\boldsymbol{\theta}^{(0)}$
- For $t = 1, 2, \ldots, T$, repeat:
  - **E-step:** Compute posterior distribution of $\{z_i\}$ given $\{\boldsymbol{x}_i\}$ and $\boldsymbol{\theta}^{(t-1)}$ as:

$$q^{(t)}(\{z_i\}) = p(\{z_i\}|\{\boldsymbol{x}_i\}, \boldsymbol{\theta}^{(t-1)})$$

  - **M-step:** Find $\boldsymbol{\theta}^{(t)}$ as the maximizer of complete log-likelihood with respect to $q^{(t)}(\{z_i\})$ as:

$$\boldsymbol{\theta}^{(t)} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\ \mathbb{E}_{q^{(t)}}\left[\log p(\{\boldsymbol{x}_i\}, \{z_i\}|\boldsymbol{\theta})\right] = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \sum_{\{z_i\}} q^{(t)}(\{z_i\}) \log p(\{\boldsymbol{x}_i\}, \{z_i\}|\boldsymbol{\theta})$$

General Mixture Model

General Mixture Model

For a general mixture model, the samples are generated using the following distribution:

$$p(x|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k p_c(x|\boldsymbol{\theta}_k)$$

where we have:

$$\boldsymbol{\theta} = \left\{ \boldsymbol{\pi} = \begin{bmatrix} \pi_1 \\ \vdots \\ \pi_K \end{bmatrix}, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K \right\}$$

and $z \sim \mathrm{Cat}(\boldsymbol{\pi})$

General Mixture Model

Complete log-Likelihood Formulation

$$\log p(\{\boldsymbol{x}_i\}, \{z_i\}|\boldsymbol{\theta}) = \log \prod_{i=1}^{N} p(\boldsymbol{x}_i, z_i|\boldsymbol{\theta}) = \log \prod_{i=1}^{N} p(\boldsymbol{x}_i|z_i, \boldsymbol{\theta})p(z_i|\boldsymbol{\theta})$$

On the other hand, we have:

$$p(\boldsymbol{x}_i|z_i, \boldsymbol{\theta}) = p_c(\boldsymbol{x}_i|\boldsymbol{\theta}_{z_i})$$
$$p(z_i|\boldsymbol{\theta}) = \pi_{z_i}$$

Thus we have:

$$\log p(\{\boldsymbol{x}_i\}, \{z_i\}|\boldsymbol{\theta}) = \sum_{i=1}^{N} \left(\log \pi_{z_i} + \log p_c(\boldsymbol{x}_i|\boldsymbol{\theta}_{z_i})\right)$$
$$= \sum_{i=1}^{N} \sum_{k=1}^{K} \delta_{k,z_i} \left(\log \pi_k + \log p_c(\boldsymbol{x}_i|\boldsymbol{\theta}_k)\right)$$

General Mixture Model

E-step

$$p(\{z_i\}|\{\boldsymbol{x}_i\}, \boldsymbol{\theta}) = \prod_{i=1}^{N} p(z_i|\boldsymbol{x}_i, \boldsymbol{\theta})$$

To compute $p(z_i|\boldsymbol{x}_i, \boldsymbol{\theta})$, we use Bayes rule as:

$$p(z_i = k|\boldsymbol{x}_i, \boldsymbol{\theta}) = \frac{p(\boldsymbol{x}_i|z_i = k, \boldsymbol{\theta})p(z_i = k|\boldsymbol{\theta})}{\sum_{l=1}^{K} p(\boldsymbol{x}_i|z_i = l, \boldsymbol{\theta})p(z_i = l|\boldsymbol{\theta})} = \frac{\pi_k p_c(\boldsymbol{x}_i|\boldsymbol{\theta}_k)}{\sum_{l=1}^{K} \pi_l p_c(\boldsymbol{x}_i|\boldsymbol{\theta}_l)}$$

Thus we have:

$$q^{(t)}(\{z_i\}) \prod_{i=1}^{N} q_i^{(t)}(z_i), \quad q_i^{(t)}(z_i) = p(z_i|\boldsymbol{x}_i, \boldsymbol{\theta}^{(t-1)})$$

General Mixture Model

M-step

$$\mathbb{E}_{q^{(t)}}\left(\sum_{i=1}^{N}\sum_{k=1}^{K}\delta_{k,z_i}\left(\log\pi_k+\log p_c(\boldsymbol{x}_i|\boldsymbol{\theta}_k)\right)\right)$$

$$=\sum_{i=1}^{N}\sum_{k=1}^{K}\mathbb{E}_{q^{(t)}}\left[\delta_{k,z_i}\left(\log\pi_k+\log p_c(\boldsymbol{x}_i|\boldsymbol{\theta}_k)\right)\right]$$

$$=\sum_{i=1}^{N}\sum_{k=1}^{K}\mathbb{E}_{q^{(t)}}\left[\delta_{k,z_i}\right]\left(\log\pi_k+\log p_c(\boldsymbol{x}_i|\boldsymbol{\theta}_k)\right)$$

$$=\sum_{i=1}^{N}\sum_{k=1}^{K}q_i^{(t)}(k)\left(\log\pi_k+\log p_c(\boldsymbol{x}_i|\boldsymbol{\theta}_k)\right)$$

Now we should maximize the above over all parameters $\boldsymbol{\theta}$.

General Mixture Model

M-step

The optimization problem for different parameters is:

$$\widehat{\boldsymbol{\theta}}_k^{(t)} = \underset{\boldsymbol{\theta}_k}{\operatorname{argmax}} \sum_{i=1}^{N} q_i^{(t)}(k) \log \; p_c(\boldsymbol{x}_i|\boldsymbol{\theta}_k)$$

$$\widehat{\boldsymbol{\pi}}^{(t)} = \underset{\boldsymbol{\pi}}{\operatorname{argmax}} \sum_{i=1}^{N} q_i^{(t)}(k) \log \pi_k, \text{ subject to } \sum_{k=1}^{K} \pi_k = 1, \pi_k \leq 0$$

The second optimization problem result in the following answer:

$$\widehat{\pi}_k^{(t)} = \frac{1}{N} \sum_{i=1}^{N} q_i^{(t)}(k)$$

Multivariate Gaussian as p_c

Algorithm

The algorithm is as follows:

- Initialize $\{\boldsymbol{\mu}_k^{(0)}, \boldsymbol{\Sigma}_k^{(0)}\}_{k=1}^K$ randomly and $\boldsymbol{\pi}^{(0)} = \frac{1}{K}\mathbf{1}$.
- For $t = 1, 2, \ldots, T$, repeat:
    - **E-step:**

$$q_i^{(t)}(z_i = k) = \frac{\pi_k^{(t-1)} p_c(\boldsymbol{x}_i | \boldsymbol{\mu}_k^{(t-1)}, \boldsymbol{\Sigma}_k^{(t-1)})}{\sum_{l=1}^K \pi_l^{(t-1)} p_c(\boldsymbol{x}_i | \boldsymbol{\mu}_l^{(t-1)}, \boldsymbol{\Sigma}_l^{(t-1)})}, \quad \begin{cases} k = 1, \ldots, K \\ i = 1, \ldots, N \end{cases}$$

    - **M-step:**

$$\pi_k^{(t)} = \frac{1}{N} \sum_{i=1}^N q_i^{(t)}(k)$$

$$\boldsymbol{\mu}_k^{(t)} = \frac{1}{N\pi_k^{(t)}} \sum_{i=1}^N q_i^{(t)}(k) \boldsymbol{x}_i$$

$$\boldsymbol{\Sigma}_k^{(t)} = \frac{1}{N\pi_k^{(t)}} \sum_{i=1}^N q_i^{(t)} \left(\boldsymbol{x}_i - \boldsymbol{\mu}_k^{(t)}\right) \left(\boldsymbol{x}_i - \boldsymbol{\mu}_k^{(t)}\right)^T$$

References I

"Lecture 9: Principal component analysis (pca)," https://ocw.mit.edu/courses/18-650-statistics-for-applications-fall-2016/resources/mit18_650f16_pca/, Accessed: 2022-09-24.