CS-567 Machine Learning

Nazar Khan

PUCIT

Lectures 18-19
Dec 22, 2015 and Jan 5 2016

The Gaussian Distribution

- The Gaussian distribution for a continuous, multivariate D-dimensional vector $\mathbf{x}$ is given by

$$\mathcal{N}(\mathbf{x}|\mu, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^D|\boldsymbol{\Sigma}|}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

  where the $D \times D$ matrix $\boldsymbol{\Sigma}$ is called the **covariance matrix** and $|\boldsymbol{\Sigma}|$ is its determinant.
- Gaussian distribution is intrinsically uni-modal. Its mode is the same as its mean $\boldsymbol{\mu}$.
- Cannot represent multi-modal data. For that a *mixture of Gaussians* can be used.

Mahalanobis Distance

- The term within the exponent is the so-called *Mahalanobis distance*

$$d(\mathbf{x}) = (\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})$$

- All $\mathbf{x}$ satisfying $d(\mathbf{x}) = k$ constitute the k-th *iso-surface* of function $d(\cdot)$.
- Iso-surfaces of Mahalanobis distance are iso-surfaces of the Gaussian density also.

Σ – The Covariance Matrix

- Covariance matrix $\boldsymbol{\Sigma}$ is
  - Real-valued
  - Symmetric
  - Positive Definite (all eigenvalues are positive)
- Its eigen-decomposition can be written as

$$\boldsymbol{\Sigma} = \sum_{i=1}^{D} \lambda_i \mathbf{u}_i \mathbf{u}_i^T$$

- Using this eigen-decomposition, its inverse can be written as

$$\boldsymbol{\Sigma}^{-1} = \sum_{i=1}^{D} \frac{1}{\lambda_i} \mathbf{u}_i \mathbf{u}_i^T$$

Σ – The Covariance Matrix

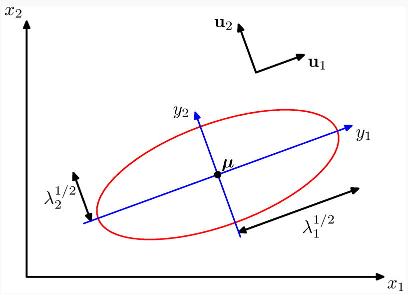
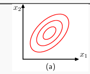
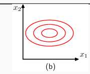
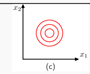


Figure: Elliptical iso-contour of a 2D Gaussian. Center of ellipse is determined by $\boldsymbol{\mu}$, axes are determined by the eigenvectors of $\boldsymbol{\Sigma}$ and axes lengths are determined via the eigenvalues of $\boldsymbol{\Sigma}$.

Σ – The Covariance Matrix

- Covariance matrix $\boldsymbol{\Sigma}$ can be categorised as

Category	$\boldsymbol{\Sigma}$ $(D = 2)$	DoF	Iso-contours $(D = 2)$
General	$\begin{pmatrix} \sigma_1^2 & \sigma_1\sigma_2 \\ \sigma_2\sigma_1 & \sigma_2^2 \end{pmatrix}$	$\frac{D(D+1)}{2}$	
Diagonal	$\begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix}$	D	
Isotropic	$\sigma^2\mathbf{I}$	1	

- Diagonal and isotropic cases are easy to work with but cannot represent data with interesting correlations.

Central Limit Theorem

- For random variables $\mathbf{x}_1, \dots, \mathbf{x}_N$ that belong to any distribution (non-Gaussian), the sum $\mathbf{s} = \mathbf{x}_1 + \cdots + \mathbf{x}_N$ approaches a Gaussian random variable as N approaches ∞.
- This is known as the *Central Limit Theorem*.
- This is one reason for the popularity of the Gaussian distribution – lots of natural phenomena correspond to sums or averages of many (non-Gaussian) random variables. For large enough N, these phenomena can be modelled by Gaussian distributions.

Fitting Gaussian density to data

- We have already covered how ML and MAP estimates for Gaussian density can be obtained.
- For computing log-likelihood of Gaussian, it is sufficient to pre-compute the following 2 statistics from the data:
  - the $D \times 1$ vector $\sum_{n=1}^{N} \mathbf{x}_n$
  - the $D \times D$ matrix $\sum_{n=1}^{N} \mathbf{x}_n \mathbf{x}_n^T$
- These statistics are called *sufficient statistics* for log-likelihood of Gaussian. The individual data items can be discarded once these are computed.

Parametric Density Estimation
Disadvantage

- So far, we have considered fitting a parametric density function to data.
- The density function is governed by some parameters $\boldsymbol{\theta}$ and the goal has been to find the optimal parameters $\boldsymbol{\theta}^*$.
- A major weakness of parametric methods is that if the chosen density function cannot represent the given data then no optimal parameters will exist.
  - For example, fitting Gaussian density to multi-modal data.
- Now we will study non-parametric density estimation methods.

Non-Parametric Density Estimation
Histogram based

- We have already covered a very basic non-parametric density estimation method – via histograms.
- The basic idea is simple.
  - Divide input space into bins.
  - Count number of observations/data points in each bin.
  - Normalise bin values to obtain probabilities.
- A more specific algorithm.
  - Divide input space into bins.
  - Count number of observations/data points n_i in bin i with width/volume Δ_i.
  - Normalise each bin value by dividing by its volume Δ_i. This makes small and large bins comparable.
  - Normalise again by dividing by total number of observations N to obtain probabilities.
- In short, probability of bin i can be obtained as

$$p_i = \frac{n_i}{N\Delta_i}$$

Non-Parametric Density Estimation
Histogram based

- Advantages
  - Once the histogram is computed, the data can be discarded. This is beneficial for
    - large datasets
    - sequential learning
- Disadvantages
  - $p(\mathbf{x})$ is discontinuous *only due to* having bin edges. The underlying distribution that generated the data might not be discontinuos.
  - Curse of dimensionality.
    - If we divide each variable in a D-dimensional space into M bins, then total number of bins will be M^D which scales exponentially with D.
    - To ensure that each bin gets enough data to estimate probability reliably, we will need *lots of data*.

Non-Parametric Density Estimation
Alternative methods

- Better scaling with dimensionality is acheived by two other density estimation techniques
  - Kernel estimators
  - Nearest neighbours
- Based on the same idea as the histogram based method – in order to estimate $p(\mathbf{x})$, consider data *around* $\mathbf{x}$.

Non-Parametric Density Estimation
Alternative methods

- Probability of data points in region $\mathcal{R}$ is given by $P = \int_{\mathcal{R}} p(\mathbf{x}) d\mathbf{x}$.
- P can also be viewed as the probability of a new data point falling in region $\mathcal{R}$.
- For N observation, probability of K observations falling in region $\mathcal{R}$ is given by the Binomial distribution.

$$\text{Bin}(K|N,P) = \frac{N!}{K!(N-K)!} P^K (1-P)^{N-K}$$

- Since $K \sim \text{Bin}(N,P)$, $\mathbb{E}[K] = NP$ and $\text{var}(K) = NP(1-P)$.
- Therefore, $\mathbb{E}[\frac{K}{N}] = P$ and $\text{var}(\frac{K}{N}) = \frac{P(1-P)}{N}$.
- Since $\lim_{N\to\infty} \text{var}(\frac{K}{N}) = 0$, $\frac{K}{N}$ stays close to its expected value P and we can write $\frac{K}{N} \approx P$.

Non-Parametric Density Estimation
Alternative methods

- In a small region $\mathcal{R}$ with volume V around location $\mathbf{x}$, we can assume that probability density of points remains constant. We denote that constant density value by $p(\mathbf{x})$.
- Probability mass P of region $\mathcal{R}$ is the product of density and volume. That is, $P = p(\mathbf{x})V$.
- From the previous slide, we can now write $\frac{K}{N} \approx p(\mathbf{x})V$.
- This yields the following formula for non-parametric density estimation

$$p(\mathbf{x}) = \frac{K}{NV} \tag{1}$$

- Notice that histogram based density estimation also used the same formula with $K = n_i$ and $V = \Delta_i$.

Non-Parametric Density Estimation
Alternative methods

- Now we have 2 options to compute $p(\mathbf{x})$
  1. Fix a volume V around location $\mathbf{x}$, count number of data points K lying within that volume and compute $p(\mathbf{x})$ using Equation (1). This method is known as density estimation through *Kernel Estimators*.
  2. Fix a number K and find the K closest data points around location $\mathbf{x}$, compute volume V of the region encompassing these nearest neighbours and compute $p(\mathbf{x})$ using Equation (1). This method is known as density estimation through *Nearest Neighbours*.

Non-Parametric Density Estimation
Kernel Estimators

- Consider a unit hyper-cube around the origin and a point $\mathbf{u}$.
- We want a function that returns 1 if $\mathbf{u}$ lies inside the hyper-cube and 0 if it lies outside.
- This function/kernel can be written as

$$k(\mathbf{u}) = \left\{ \begin{array}{ll} 1, & \text{if } |u_i| \leq \frac{1}{2} \text{ for } i = 1,\dots,D \\ 0, & \text{otherwise} \end{array} \right\}$$

- To perform the same operation for a unit hyper-cube centered on a location $\mathbf{x}$, we can use the modified kernel

$$k(\mathbf{u}-\mathbf{x}) = \left\{ \begin{array}{ll} 1, & \text{if } |u_i - x_i| \leq \frac{1}{2} \text{ for } i = 1,\dots,D \\ 0, & \text{otherwise} \end{array} \right\}$$

- Similarly, to perform the same operation for a hyper-cube with dimension length h centered on a location $\mathbf{x}$, we can use the modified kernel $k(\frac{\mathbf{u}-\mathbf{x}}{h})$.

Non-Parametric Density Estimation
Kernel Estimators

- This gives us a way of counting number of data points in a hyper-cube of volume h^D around location $\mathbf{x}$ as $K = \sum_{n=1}^{N} k(\frac{\mathbf{u_n}-\mathbf{x}}{h})$.
- Finally, $p(\mathbf{x})$ can be computed using Equation (1) as $p(\mathbf{x}) = \frac{K}{Nh^D}$.
- This method is also known as the *Parzen window* approach.

Non-Parametric Density Estimation
Kernel Estimators

- Use of the hyper-cube with a binary in/out decision leads to artificial, discontinuous estimates for $p(\mathbf{x})$.
- One alternative is to use a smoother (e.g., Gaussian) kernel function instead.

$$p(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{\sqrt{(2\pi h^2)}}\exp\left\{-\frac{||\mathbf{u}_n - \mathbf{x}||^2}{2h^2}\right\}$$

  where h plays the role of a smoothing parameter.
- Any kernel function satisfying $k(\mathbf{u}) \geq 0$ and $\int k(\mathbf{u})d\mathbf{u} = 1$ can be used. This will ensure that the resulting density function also satisfies $p(\mathbf{x}) \geq 0$ and $\int p(\mathbf{x})d\mathbf{x} = 1$.

Non-Parametric Density Estimation
Nearest Neighbours

- Here the idea is to fix K and determine volume V from the data.
- We consider a small hyper-sphere around location $\mathbf{x}$ and allow its radius to grow until it contains exactly K data points.
- $p(\mathbf{x})$ can then be computed using Equation (1) where V is the volume of the resulting hyper-sphere.

Non-Parametric Density Estimation
Disdvantage of KDE and KNN

- For both kernel estimators and nearest neighbours, $p(\mathbf{x})$ is computed using all N points of the training data.
- Therefore, training data cannot be discarded.
- Evaluation cost of $p(\mathbf{x})$ grows linearly with N.