Module 2: Statistical Modeling

2.1 Maximum Likelihood and Bayesian Inference

2.1.1 Maximum Likelihood Estimation

Maximum Likelihood Estimation

For the problem of **parameter estimation**, given a set of n i.i.d. observations $\{x_i\}$ drawn from a distribution P_{θ^*}, we want to estimate θ^* by maximizing the likelihood of observing the set $\{x_i\}$:

$$\hat{\theta} = \operatorname*{argmax}_{\theta} L(\theta)$$

where $L(\theta)$ is the **likelihood function**:

$$L(\theta) = \prod_{i=1}^{n} P(x_i|\theta)$$

and $\hat{\theta}$ is the **maximum likelihood estimator** of θ^*. In practice it's often easier to consider the **log-likelihood function**:

$$\ell(\theta) = \log L(\theta) = \sum_{i=1}^{n} \log P(x_i|\theta)$$

where

$$\hat{\theta} = \operatorname*{argmax}_{\theta} \ell(\theta) = \operatorname*{argmax}_{\theta} L(\theta)$$

MLE for Discrete Random Variables

In the case that X is a random variable that can take one of k discrete values X_j, the probability that X is a particular value X_j, conditioned on θ, is $P(X = X_j|\theta)$. If we then observe a sequence $D = \{x_1, x_2, \ldots, x_n\}$ of i.i.d. samples x_i, and we count the number of occurrences n_j of each $x_i = X_j$, then the likelihood of observing this sequence is given by

$$L(\theta) = P(D|\theta) = \prod_{i=1}^{n} P(x_i|\theta) = \prod_{j=1}^{k} P(X = X_j|\theta)^{n_j}$$

$$\ell(\theta) = \log L(\theta) = \sum_{j=1}^{k} n_j \log P(X = X_j|\theta)$$

MLE with Density Functions

For distributions of continuous random variables, the maximum likelihood estimator should be defined by their probability density functions $p(x|\theta)$:

$$\begin{aligned}\hat{\theta} &= \underset{\theta}{\operatorname{argmax}} \left\{ L(\theta) = \prod_{i=1}^{n} p(x_i|\theta) \right\} \\ &= \underset{\theta}{\operatorname{argmax}} \left\{ \ell(\theta) = \sum_{i=1}^{n} \log p(x_i|\theta) \right\}\end{aligned}$$

MLE for Regression

In regression we are given a set of pairs $\{\boldsymbol{x}_i, y_i\}$ in order to construct a function that predicts the labels based on the features. First, we assume y follows a distribution of $\mathcal{N}(f(\boldsymbol{x}, \theta), \sigma^2)$ and a density of

$$p(y|\boldsymbol{x}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\Big(- \frac{(y - f(\boldsymbol{x}, \boldsymbol{\theta}))^2}{2\sigma^2} \Big)$$

giving the log-likelihood function:

$$\begin{aligned}\ell(\boldsymbol{\theta}, \sigma) &= \sum_{i=1}^{n} \log p(y_i|\boldsymbol{x}_i, \boldsymbol{\theta}) \\ &= -\frac{1}{2\sigma^2} \sum_{i=1}^{n} \Big(y_i - f(\boldsymbol{x}_i, \boldsymbol{\theta}) \Big)^2 - n \log \sigma - n \log(\sqrt{2\pi})\end{aligned}$$

Optimizing $\boldsymbol{\theta}$ gives

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \sum_{i=1}^{n} \Big(y_i - f(\boldsymbol{x}_i, \boldsymbol{\theta}) \Big)^2$$

which is equivalent to minimizing the mean square error (MSE). Optimizing σ yields

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} \Big(y_i - f(\boldsymbol{x}_i, \hat{\boldsymbol{\theta}}) \Big)^2$$

which is exactly the mean square error obtained by the prediction with $f(\boldsymbol{x}, \hat{\boldsymbol{\theta}})$.

MLE for Classification (Logistic Regression)

Given another set $\{\boldsymbol{x}_i, y_i\}$ with $y_i \in \{0, 1\}$ for binary classification, we assume

$$P(y|\boldsymbol{x}, \boldsymbol{\theta}) = \frac{\exp(yf(\boldsymbol{x}, \boldsymbol{\theta}))}{1 + \exp(f(\boldsymbol{x}, \boldsymbol{\theta}))}$$

The log-likelihood function is

$$\begin{aligned}
\ell(\boldsymbol{\theta}) &= \sum_{i=1}^{n} \log P(y_i|\boldsymbol{x}_i) \\
&= \sum_{i=1}^{n} \log \frac{\exp(y_i f(\boldsymbol{x}_i, \boldsymbol{\theta}))}{1 + \exp(f(\boldsymbol{x}_i, \boldsymbol{\theta}))} \\
&= \sum_{i=1}^{n} \Big(y_i f(\boldsymbol{x}_i, \boldsymbol{\theta}) - \log(1 + \exp(f(\boldsymbol{x}_i, \boldsymbol{\theta}))) \Big)
\end{aligned}$$

and the maximum likelihood estimator is

$$\hat{\boldsymbol{\theta}} = \underset{\theta}{\operatorname{argmax}} \left\{ \ell(\boldsymbol{\theta}) = \sum_{i=1}^{n} \Big(y_i f(\boldsymbol{x}_i, \boldsymbol{\theta}) - \log(1 + \exp(f(\boldsymbol{x}_i, \boldsymbol{\theta}))) \Big) \right\}$$

MLE Theoretical Properties

Given a MLE estimator as a random variable $\hat{\theta} = \hat{\theta}(x_1, \ldots, x_n)$, we can evaluate this estimator against the true distribution parameter θ^* in terms of the bias, variance, and mean square error (MSE).

The **bias** is defined as

$$\text{Bias}(\hat{\theta}) = \mathbb{E}[\hat{\theta}] - \theta^*$$

and we call $\hat{\theta}$ an **unbiased** estimator if $\text{Bias}(\hat{\theta}) = 0$.

The **variance** is defined as usual as

$$\text{Var}(\hat{\theta}) = \mathbb{E}[(\hat{\theta} - \mathbb{E}[\hat{\theta}])^2]$$

The **MSE** is defined as

$$\text{MSE}(\hat{\theta}) = \mathbb{E}[(\hat{\theta} - \theta^*)^2] = \text{Bias}(\hat{\theta})^2 + \text{Var}(\hat{\theta})$$

and we call $\hat{\theta}$ a **consistent** estimator if $\text{MSE}(\hat{\theta}) = 0$.

2.1.2 Bayesian Inference

The **Bayesian Formula**, for the **posterior distribution** of θ is

$$p(\theta|D) = \frac{p(D|\theta)p_0(\theta)}{p(D)}$$

where $p_0(\theta)$ is the **prior distribution** of θ, $p(D|\theta)$ is the likelihood of seeing D given θ, and $p(D)$ is the marginal distribution of D:

$$p(D) = \int p(D|\theta)p_0(\theta)d\theta$$

Since $p(D)$ only serves as a normalization constant and does not depend on θ, it often suffices to write Baye's Rule as

$$p(\theta|D) \propto p(D|\theta)p_0(\theta)$$

When D consists of a set of i.i.d. samples $D = \{x_i\}$,

$$p(D|\theta) = \prod_{i=1}^{n} p(x_i|\theta)$$

and

$$p(\theta|D) \propto \left[\prod_{i=1}^{n} p(x_i|\theta)\right] p_0(\theta)$$

2.2 Clustering, K-means, Mixture, & EM

2.2.1 Clustering and K-means

Clustering, a form of unsupervised learning used mostly during EDA, is the task of grouping a set of objects in a way such that the objects in each cluster are more similar to each other than those in different clusters.

We start with a dataset $\{\boldsymbol{x}^{(i)}\}$ and an integer K, and the goal is to partition the dataset into K clusters. The **K-means** algorithm is the most basic clustering algorithm. It works by optimizing the centroid μ_k for each cluster S_k according to the optimization function:

$$\min_{\boldsymbol{z}} \min_{\boldsymbol{\mu}} \left\{ F(\boldsymbol{z}, \boldsymbol{\mu}) = \sum_{i=1}^{n} \|\boldsymbol{x}^{(i)} - \boldsymbol{\mu}^{(z^{(i)})}\|^2 \right\} \tag{5.1}$$

where $\boldsymbol{z} = \{z^{(i)}\}$ is the set of cluster IDs for each of the n datapoints and $\boldsymbol{\mu} = \{\boldsymbol{\mu}_k\}$ is the set of K cluster centroids.

This optimization is performed by **Coordinate Descent**. We first start with some initialization $(\boldsymbol{z}_0, \boldsymbol{\mu}_0)$ and then alternatively update $\boldsymbol{z}$ and $\boldsymbol{\mu}$ at each iteration t until the algorithm converges.

1. With $\boldsymbol{\mu}$ fixed, update $\boldsymbol{z}$:

$$\boldsymbol{z}_{t+1} = \operatorname*{argmin}_{\boldsymbol{z}} F(\boldsymbol{z}, \boldsymbol{\mu}_t) \tag{5.2}$$

$$\boldsymbol{z}_{t+1}^{(i)} = \operatorname*{argmin}_{k} \|\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_t^{(k)}\|^2 \tag{5.4}$$

   for all $i = 1, \ldots, n$ and $k = 1, \ldots, K$

2. With $\boldsymbol{z}$ fixed, update $\boldsymbol{\mu}$:

$$\boldsymbol{\mu}_{t+1} = \operatorname*{argmin}_{\boldsymbol{\mu}} F(\boldsymbol{z}_{t+1}, \boldsymbol{\mu}) \tag{5.2}$$

$$\boldsymbol{\mu}_{t+1}^{(k)} = \frac{1}{|S_k|} \sum_{i \in S_k} \boldsymbol{x}^{(i)} \tag{5.5}$$

   where $S_k = \{i : z^{(i)} = k\}$ for all $k = 1, \ldots, K$

THEOREM 5.1:

Following the updates in (5.2) and (5.3), for all t,

$$F(\boldsymbol{z}_t, \boldsymbol{\mu}_t) \geq F(\boldsymbol{z}_{t+1}, \boldsymbol{\mu}_{t+1})$$

This guarantees convergence to a local optimum of $F(\boldsymbol{z}, \boldsymbol{\mu})$, but not necessarily to a global optimum, highlighting the importance of a good initialization $(\boldsymbol{z}_0, \boldsymbol{\mu}_0)$.

2.2.2 Gaussian Mixture and EM

The Gaussian mixture model (GMM) is a natural probabilistic model for clustering in which each cluster is represented by a Gaussian distribution. The density function of a GMM is a weighted linear combination of several Gaussian density functions.

$$p(x|\theta) = \sum_{k=1}^{K} w_k \mathcal{N}(x; \mu_k, \sigma_k^2) \tag{5.6}$$

where $\theta = \{w_k, \mu_k, \sigma_k\}$ and $\{w_k\}$ is a set of mixture weights that satisfy $\sum_k w_k = 1$ and $w_k \geq 0$. $\mathcal{N}(x; \mu_k, \sigma_k^2)$ is the density of a Gaussian distribution:

$$\mathcal{N}(x; \mu_k, \sigma_k^2) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{1}{2\sigma_k^2}(x - \mu_k)^2\right)$$

and each $\mathcal{N}(x; \mu_k, \sigma_k^2)$ is called the component of $p(x|\theta)$ that corresponds to the k-th cluster.

If a random variable X is drawn from a GMM, it can be equivalently drawn from a randomly picked Gaussian component (cluster) with probability w_k. We can introduce a latent index variable $Z \in \{1, 2, \ldots, K\}$ and generate (X, Z) with the following procedure:

1. Draw a latent label Z:
$$P(Z = k \mid \theta) = w_k$$

2. Draw observation X:
$$p(X = x \mid Z = k, \theta) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{1}{2\sigma_k^2}(x - \mu_k)^2\right)$$

The joint probability of (X, Z) is obtained by the chain rule:

$$\begin{aligned} p(X = x, Z = k \mid \theta) &= P(Z = k \mid \theta) p(X = x \mid Z = k, \theta) \\ &= w_k \mathcal{N}(x; \mu_k, \sigma_k^2) \end{aligned}$$

from which the marginal distribution of X (5.6) is obtained by summing over all Z

$$p(X = x \mid \theta) = \sum_{k=1}^{K} p(X = x, Z = k \mid \theta) = \sum_{k=1}^{K} w_k \mathcal{N}(x; \mu_k, \sigma_k^2)$$

Given an observation of X, we can infer its cluster ID from the posterior distribution:

$$P(Z = k \mid X = x, \theta) = \frac{P(X = x, Z = k \mid \theta)}{p(X = x \mid \theta)} = \frac{w_k \mathcal{N}(x; \mu_k, \sigma_k^2)}{\sum_{\ell=1}^{K} w_\ell \mathcal{N}(x; \mu_\ell, \sigma_\ell^2)} \quad (5.7)$$

Clustering as Parameter Estimation of GMM

The clustering problem can be formulated as estimating parameters in GMM. We assume our dataset $\{\boldsymbol{x}^{(i)}\}$ is drawn from a GMM with parameters $\theta = \{w_k, \mu_k, \sigma_k\}$ where μ_k and σ_k represent the mean and variance of the k-th cluster, and w_k represents is relative percentage in the dataset. Then by performing Maximum Likelihood Estimation on θ, we can calculate the posterior probability $P(Z = k \mid X = \boldsymbol{x}^{(i)}, \theta)$ for each data point to form a probabilistic clustering:

$$\begin{aligned} \hat{\theta} &= \operatorname*{argmax}_{\theta} \sum_{i=1}^{n} \log p(\boldsymbol{x}^{(i)} \mid \theta) \\ \{\hat{w}_k, \hat{\mu}_k, \hat{\sigma}_k\} &= \operatorname*{argmax}_{\{w_k, \mu_k, \sigma_k\}} \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} w_k \mathcal{N}(x; \mu_k, \sigma_k^2) \right) \end{aligned} \quad (5.8)$$

where the weights are constrained by $\sum_k w_k = 1$ and $w_k \geq 0$ and $\sigma_k \geq 0$.

Expectation-Maximization (EM)

The above optimization of GMM is rather complicated to solve, giving rise to a much more efficient and convenient method for maximizing the log-likelihood of mixture models. Algorithmically, **Expectation-Maximization**

(EM) can be viewed as a "probabilistic variant" of K-means. We denote γ_{ik} to be the posterior probability that the i-th data point is drawn from the k-th component:

$$\gamma_{ik} = P(Z = k \mid X = \boldsymbol{x}^{(i)}, \theta)$$

we first initialize $\theta_0 = \{w_{k;0}, \mu_{k;0}, \sigma_{k;0}\}$ and then perform the following update step until the algorithm converges:

1. **E-Step:** With θ fixed, update $\{\gamma_{ik}\}$:

$$\gamma_{ik;t+1} = P(Z = k \mid X = \boldsymbol{x}^{(i)}, \theta_t) = \frac{w_{k;t}\, \mathcal{N}(x^{(i)}; \mu_{k;t}, \sigma^2_{k;t})}{\sum_{\ell=1}^{K} w_{\ell;t}\, \mathcal{N}(x^{(i)}; \mu_{\ell;t}, \sigma^2_{\ell;t})}$$

2. **M-Step:** With $\{\gamma_{ik}\}$ fixed, update θ:

$$\begin{aligned}
\mu_{k;t+1} &= \frac{\sum_{i=1}^{n} \gamma_{ik;t+1} \boldsymbol{x}^{(i)}}{\sum_{j=1}^{n} \gamma_{jk;t+1}} \\
\sigma_{k;t+1} &= \frac{\sum_{i=1}^{n} \gamma_{ik;t+1} (\boldsymbol{x}^{(i)} - \mu_{k;t+1})^2}{\sum_{j=1}^{n} \gamma_{jk;t+1}} \\
w_{k;t+1} &= \frac{\sum_{i=1}^{n} \gamma_{ik;t+1}}{\sum_{j=1}^{n} \sum_{k=1}^{K} \gamma_{jk;t+1}}
\end{aligned}$$

EM as Coordinate Descent

The above EM procedure can also be viewed as a coordinate descent algorithm that monotonically maximized the function $\ell(\theta)$:

$$\ell(\theta) = \max_{\gamma \in \Gamma} F(\theta, \gamma) \tag{5.9}$$

where Γ is the set of all valid posterior probabilities

$$\Gamma = \{\gamma = \{\gamma_{ik}\}\}$$

such that $\gamma_{ik} \geq 0$ for all i, k and $\sum_{k=1}^{K} \gamma_{ik} = 1$ for all i.

Since

$$\max_{\theta} \ell(\theta) = \max_{\theta} \max_{\gamma \in \Gamma} F(\theta, \gamma)$$

we can optimize both θ and γ alternatively using coordinate descent similar to K-means:

$$\begin{aligned} \gamma_{t+1} &= \operatorname*{argmax}_{\gamma \in \Gamma} F(\theta_t, \gamma) \\ \theta_{t+1} &= \operatorname*{argmax}_{\theta} F(\theta, \gamma_{t+1}) \end{aligned} \tag{5.10}$$

THEOREM 5.2:

With

$$F(\theta, \gamma) = \sum_{i=1}^{n} \sum_{k=1}^{K} \gamma_{ik} \log \left(\frac{P(X = \boldsymbol{x}^{(i)}, Z = k \mid \theta)}{\gamma_{ik}} \right) \tag{5.11}$$

and

$$\ell(\theta) = \max_{\gamma \in \Gamma} F(\theta, \gamma)$$

then for each θ, the maximum γ_{ik}^* is achieved by

$$\gamma_{ik}^* = P(X = \boldsymbol{x}^{(i)} \mid Z = k, \theta)$$

for all i, k.

2.3 Multivariate Normal Distributions

2.3.1 Multivarate Distributions

For a random vector $\mathbf{X} = [X_1, \ldots, X_d]^\top \in \mathbb{R}^d$, its distribution is characterized by its probability density function $p(\boldsymbol{x})$ constrained by $p(x) \geq 0$ and $\int p(\boldsymbol{x})d\boldsymbol{x} = 1$. For any function $h(\boldsymbol{x})$, its expectation under $\mathbf{X}$ is

$$\mathbb{E}[h(\mathbf{X})] = \int h(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}$$

In general, the mean vector of $\mathbf{X}$ is

$$\mathbb{E}[\mathbf{X}] = \begin{bmatrix} \mathbb{E}[X_1] \\ \vdots \\ \mathbb{E}[X_d] \end{bmatrix} = \begin{bmatrix} \int x_1 p(\boldsymbol{x})d\boldsymbol{x} \\ \vdots \\ \int x_d p(\boldsymbol{x})d\boldsymbol{x} \end{bmatrix} \in \mathbb{R}^d$$

The covariance matrix of $\mathbf{X}$ is a $d \times d$ matrix consisting of the pairwise covariances of each of the elements of $\mathbf{X}$

$$\mathrm{Cov}(\mathbf{X}) = [\mathrm{Cov}(X_i, X_j)]_{ij} = \begin{bmatrix} \mathrm{Var}(X_1) & \mathrm{Cov}(X_1, X_2) & \cdots & \mathrm{Cov}(X_1, X_d) \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Var}(X_2) & \cdots & \mathrm{Cov}(X_2, X_d) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}(X_d, X_1) & \mathrm{Cov}(X_d, X_2) & \cdots & \mathrm{Var}(X_d) \end{bmatrix}$$

where each each $\mathrm{Cov}(X_i, X_j)$ represents the covariance between X_i and X_j:

$$\begin{aligned} \mathrm{Cov}(X_i, X_j) &= \mathbb{E}\Big[(X_i - \mathbb{E}[X_i])(X_j - \mathbb{E}[X_j])\Big] \\ &= \mathbb{E}[X_i X_j] - \mathbb{E}[X_i]\mathbb{E}[X_j] \end{aligned}$$

and each diagonal element simplifies to the univariate variance of each element of $\mathbf{X}$

$$\begin{aligned} \mathrm{Var}(X_i) &= \mathrm{Cov}(X_i, X_i) \\ \mathbb{E}[(X_i - \mathbb{E}[X_i])^2] &= \mathbb{E}[X_i^2] - \mathbb{E}[X_i]^2 \end{aligned}$$

In compact matrix form, we can represent the covariance matrix as

$$\begin{aligned}\text{Cov}(\mathbf{X}) &= \mathbb{E}\Big[(\mathbf{X} - \mathbb{E}[\mathbf{X}])(\mathbf{X} - \mathbb{E}[\mathbf{X}])^\top\Big] \\ &= \mathbb{E}[\mathbf{X}\mathbf{X}^\top] - \mathbb{E}[\mathbf{X}]\mathbb{E}[\mathbf{X}]^\top\end{aligned}$$

THEOREM 6.1:

$\boldsymbol{\Sigma} = \text{Cov}(\mathbf{X})$ is always positive semi-definite:

$$\boldsymbol{v}^\top \boldsymbol{\Sigma} \boldsymbol{v} \geq 0$$

Proof: For any $\boldsymbol{v} \in \mathbb{R}^d$,

$$\begin{aligned}\boldsymbol{v}^\top \boldsymbol{\Sigma} \boldsymbol{v} &= \boldsymbol{v}^\top \mathbb{E}\Big[(\mathbf{X} - \mathbb{E}[\mathbf{X}])(\mathbf{X} - \mathbb{E}[\mathbf{X}])^\top\Big] \boldsymbol{v} \\ &= \mathbb{E}\Big[\boldsymbol{v}^\top (\mathbf{X} - \mathbb{E}[\mathbf{X}])(\mathbf{X} - \mathbb{E}[\mathbf{X}])^\top \boldsymbol{v}\Big] \\ &= \mathbb{E}\Big[\mathbf{b}^\top \mathbf{b}\Big] \\ &= \mathbb{E}\Big[\|\mathbf{b}\|^2\Big] \geq 0\end{aligned}$$

where we defined $\mathbf{b} = (\mathbf{X} - \mathbb{E}[\mathbf{X}])^\top \boldsymbol{v}$. Note in practice $\boldsymbol{\Sigma}$ will usually be positive definite, downgrading to semi-definiteness only when some variables are perfectly linearly dependent.

2.3.2 Multivariate Normal Distribution

A random variable $X \in \mathbb{R}$ is univariate normal $\mathcal{N}(\mu, \sigma^2)$ if its density function is

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\Big(-\frac{1}{2\sigma^2}(x-\mu)^2\Big)$$

where $\mathcal{N}(0, 1)$ in particular is called the **standard normal** distribution.

Definition 6.1: A random vector $\mathbf{X} \in \mathbb{R}^d$ is **multivariate normal** if it can be obtained by applying linear transformations on a set of independent standard normal random variables

$$\mathbf{X} = \mathbf{A}\mathbf{Z} + \mathbf{b} \tag{6.1}$$

where $\mathbf{A} \in \mathbb{R}^{d \times d}$ and $\mathbf{b} \in \mathbb{R}^d$ are deterministic parameters and $\mathbf{Z} = [Z_1, \ldots, Z_d]^\top$ is a set of independent standard normal random variables with $Z_i \sim \mathcal{N}(0, 1)$ and $Z_i \perp Z_j$ for all $i \neq j$.

THEOREM 6.2:

For the multivariate normal random variable $\mathbf{X} = \mathbf{A}\mathbf{Z} + \mathbf{b}$,

$$\mathbb{E}[\mathbf{X}] = \boldsymbol{\mu} \qquad \text{Cov}(\mathbf{X}) = \boldsymbol{\Sigma} = \mathbf{A}\mathbf{A}^\top$$

THEOREM 6.3:

By extension of Theorem 6.2, if $\mathbf{Z} \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ and $\mathbf{X} = \mathbf{A}\mathbf{Z} + \mathbf{b}$, then

$$\mathbb{E}[\mathbf{X}] = \mathbf{A}\boldsymbol{\mu}_0 + \mathbf{b} \qquad \text{Cov}(\mathbf{X}) = \mathbf{A}\boldsymbol{\Sigma}_0\mathbf{A}^\top$$

$\mathbf{X}$ is also multivariate normal,

$$\begin{aligned} \mathbf{X} &\sim \mathcal{N}(\mathbf{A}\boldsymbol{\mu}_0 + \mathbf{b},\ \mathbf{A}\boldsymbol{\Sigma}_0\mathbf{A}^\top) \\ &\sim \mathcal{N}(\boldsymbol{\mu},\ \boldsymbol{\Sigma}) \end{aligned}$$

and its probability density function is

$$p(\boldsymbol{x}) = \frac{1}{D}\exp\Big(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\Big) \qquad D = \sqrt{(2\pi)^d \det(\boldsymbol{\Sigma})}$$

where $\boldsymbol{\mu} = \mathbf{A}\boldsymbol{\mu}_0 + \mathbf{b}$ is the mean of $\mathbf{X}$ and $\boldsymbol{\Sigma} = \mathbf{A}\boldsymbol{\Sigma}_0\mathbf{A}^\top$ its covariance matrix. D simply serves as a normalization constant.

Marginal Distributions

If $\mathbf{X}$ is multivariate normal, then each of its sub-vectors $[\mathbf{X}_\alpha, \mathbf{X}_\beta]^\top$ is also multivariate normal, and its covariance matrix is simply the corresponding sub-matrix of the covariance matrix.

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_\alpha \\ \mathbf{X}_\beta \end{bmatrix} \qquad \boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_\alpha \\ \boldsymbol{\mu}_\beta \end{bmatrix} \qquad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{\alpha\alpha} & \boldsymbol{\Sigma}_{\alpha\beta} \\ \boldsymbol{\Sigma}_{\beta\alpha} & \boldsymbol{\Sigma}_{\beta\beta} \end{bmatrix} \tag{6.2}$$

THEOREM 6.4: If $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $\mathbf{X}_\alpha$ is its sub-vector, then $\mathbb{E}[\mathbf{X}_\alpha] = \boldsymbol{\mu}_\alpha$, $\text{Cov}(\mathbf{X}_\alpha) = \boldsymbol{\Sigma}_{\alpha\alpha}$, and $\mathbf{X}_\alpha \sim \mathcal{N}(\boldsymbol{\mu}_\alpha, \boldsymbol{\Sigma}_\alpha)$. Therefore the probability

density function of $\mathbf{X}_\alpha$ is

$$p_{\boldsymbol{x}_\alpha}(\boldsymbol{x}_\alpha) = \frac{1}{D_\alpha} \exp\Big(-\frac{1}{2}(\boldsymbol{x}_\alpha - \boldsymbol{\mu}_\alpha)^\top \boldsymbol{\Sigma}_{\alpha\alpha}^{-1}(\boldsymbol{x}_\alpha - \boldsymbol{\mu}_\alpha)\Big) \qquad D_\alpha = \sqrt{(2\pi)^{d_\alpha} \det(\boldsymbol{\Sigma}_\alpha)}$$

Furthermore, the conditional distribution $p(\mathbf{X}_\alpha|\mathbf{X}_\beta = \mathbf{b})$ is also Gaussian:

$$p(\mathbf{X}_\alpha|\mathbf{X}_\beta = \mathbf{b}) \sim \mathcal{N}(\boldsymbol{\mu}_{\alpha|\beta}, \boldsymbol{\Sigma}_{\alpha|\beta})$$

where

$$\begin{aligned}\boldsymbol{\mu}_{\alpha|\beta} &= \mathbb{E}[\mathbf{X}_\alpha|\mathbf{X}_\beta = \mathbf{b}] = \boldsymbol{\mu}_\alpha + \boldsymbol{\Sigma}_{\alpha\beta}\boldsymbol{\Sigma}_{\beta\beta}^{-1}(\mathbf{b} - \boldsymbol{\mu}_\beta) \\ \boldsymbol{\Sigma}_{\alpha|\beta} &= \mathrm{Cov}(\mathbf{X}_\alpha|\mathbf{X}_\beta = \mathbf{b}) = \boldsymbol{\Sigma}_{\alpha\alpha} - \boldsymbol{\Sigma}_{\alpha\beta}\boldsymbol{\Sigma}_{\beta\beta}^{-1}\boldsymbol{\Sigma}_{\beta\alpha}\end{aligned}$$

here $\boldsymbol{\Sigma}_{\alpha|\beta}$ is called the **Schur complement** of $\boldsymbol{\Sigma}_{\beta\beta}$.

Covariance and Independence

Definition 6.2: Two random variables X_i, X_j are **independent** from each other, $X_i \perp X_j$, if their joint density equals the product of their individual density functions

$$p_{X_i,X_j}(x_i, x_j) = p_{X_i}(x_i)p_{X_j}(x_j)$$

for all x_i, x_j.

THEOREM 6.5:

The following statements regarding two random variables X_i, X_j are equivalent:

1. X_i, X_j are independent from each other, $X_i \perp X_j$
2. $\mathrm{Cov}(h(X_i), h(X_j)) = 0$ for any function h
3. The joint density of X_i, X_j can be written as

$$p_{X_i,X_j}(x_i, x_j) \propto \phi(x_i)\psi(x_j)$$

   where ϕ and ψ are two non-negative functions.

Note:
In general, $\mathrm{Cov}(X_i, X_j) = 0$ does not necessarily imply independence between the two variables, but the statement above requiring zero covariance under

any function h is a much stronger requirement, and does indeed guarantee independence.
Alternatively, we can require $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ to be a multivariate random variable. Then if two elements have zero covariance, $\sigma_{ij} = \text{Cov}(X_i, X_j) = 0$, they are guaranteed to be independent from each other, as shown in Theorem 6.6.

THEOREM 6.6:

Given $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$,

$$\sigma_{ij} = \text{Cov}(X_i, X_j) = 0 \iff X_i \perp X_j$$

2.3.3 Gaussian Graphical Models

Recall that the probability density function of a multivariate normal distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is

$$p(\boldsymbol{x}) = \frac{1}{D} \exp\Big(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu})\Big) \qquad D = \sqrt{(2\pi)^d \det(\boldsymbol{\Sigma})} \quad (6.3)$$

which is called the **standard form** of multivariate normal distributions.

We can also write the PDF in its **natural form** (or information form)

$$p(\boldsymbol{x}) = \frac{1}{C} \exp\Big(-\frac{1}{2}\boldsymbol{x}^\top \mathbf{Q}\boldsymbol{x} + \mathbf{b}^\top \boldsymbol{x}\Big) \quad (6.4)$$

which we denote by $\bar{\mathcal{N}}(\mathbf{b}, \mathbf{Q})$, where $\mathbf{b}, \mathbf{Q}$ are the **natural parameters** of the distribution.

Lemma 6.7: The density functions (6.3) and (6.4) are equivalent if

$$\mathbf{Q} = \boldsymbol{\Sigma}^{-1} \qquad \mathbf{b} = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} \qquad C = D \exp\Big(-\frac{1}{2}\boldsymbol{\mu}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\Big)$$

i.e. $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is equivalent to $\bar{\mathcal{N}}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}, \boldsymbol{\Sigma}^{-1})$. Since $\mathbf{Q}$ is the inverse of the covariance matrix $\boldsymbol{\Sigma}^{-1}$, it is named the inverse covariance matrix, or the **precision matrix**. Note, since $\boldsymbol{\Sigma}$ must be positive definite to invert, $\mathbf{Q}$ must also positive definite.

This natural form proves to be highly convenient for studying the **conditional distributions** and **conditional independence** of multivariate normal distributions, while the standard form is more convenient for studying the **marginal distributions** and **marginal independence (correlation)**.

Definition 6.3: Let α, β, γ be three non-overlapping index subsets of $\{1, \ldots, d\}$. $\mathbf{X}_\alpha$ and $\mathbf{X}_\beta$ are **conditionally independent**, given $\mathbf{X}_\gamma$, $(\mathbf{X}_\alpha \perp \mathbf{X}_\beta | \mathbf{X}_\gamma)$ if

$$p_{\mathbf{X}_{\alpha\cup\beta}|\mathbf{X}_\gamma}(\boldsymbol{x}_\alpha, \boldsymbol{x}_\beta|\boldsymbol{x}_\gamma) = p_{\mathbf{X}_\alpha|\mathbf{X}_\gamma}(\boldsymbol{x}_\alpha|\boldsymbol{x}_\gamma) \times p_{\mathbf{X}_\beta|\mathbf{X}_\gamma}(\boldsymbol{x}_\beta|\boldsymbol{x}_\gamma)$$

for all $\boldsymbol{x}_\alpha, \boldsymbol{x}_\beta, \boldsymbol{x}_\gamma$, where $p_{\mathbf{X}_\alpha|\mathbf{X}_\gamma}(\boldsymbol{x}_\alpha|\boldsymbol{x}_\gamma)$ denotes the probability density of $\mathbf{X}_\alpha$ conditioned on $\mathbf{X}_\gamma = \boldsymbol{x}_\gamma$, and $p_{\mathbf{X}_{\alpha\cup\beta}|\mathbf{X}_\gamma}(\boldsymbol{x}_\alpha, \boldsymbol{x}_\beta|\boldsymbol{x}_\gamma)$ is the joint density function of $[\mathbf{X}_\alpha, \mathbf{X}_\beta]$ conditioned on $\mathbf{X}_\gamma = \boldsymbol{x}_\gamma$. In this case $\mathbf{X}_\gamma$ is called the **Markov Blanket** of $\mathbf{X}_\alpha$.

Lemma 6.8: Let $\mathbf{X} \in \mathbb{R}^d$ be a random variable, α, β, γ be three non-overlapping index subsets of $\{1, \ldots, d\}$, and $p(\boldsymbol{x}_\alpha, \boldsymbol{x}_\beta, \boldsymbol{x}_\gamma)$ be the joint probability density function of $[\mathbf{X}_\alpha, \mathbf{X}_\beta, \mathbf{X}_\gamma]$. Then $\mathbf{X}_\alpha \perp \mathbf{X}_\beta | \mathbf{X}_\gamma$ ($\mathbf{X}_\alpha$ and $\mathbf{X}_\beta$ are conditionally independent) iff

$$p(\boldsymbol{x}_\alpha, \boldsymbol{x}_\beta, \boldsymbol{x}_\gamma) \propto \phi(\boldsymbol{x}_\alpha, \boldsymbol{x}_\gamma)\psi(\boldsymbol{x}_\beta, \boldsymbol{x}_\gamma)$$

where ϕ and ψ are two non-negative functions.

THEOREM 6.9:

Given $\mathbf{X} \sim \bar{\mathcal{N}}(\mathbf{b}, \mathbf{Q})$ a multivariate normal variable with elements X_i and inverse covariance matrix $\mathbf{Q}$ with elements q_{ij},

$$q_{ij} = 0 \iff X_i \perp X_j | \mathbf{X}_{\neg ij}$$

guaranteeing **conditional independence** between X_i and X_j given all other X_k's. Note this is a weaker condition than Theorem 6.6's guarantee of **marginal independence**, which does not require conditioning on other variables.

2.4 Kernel Method

While standard linear regression is performed by taking a linear combination of the d input variables

$$f(\boldsymbol{x};\boldsymbol{\theta}) = \sum_{\ell=1}^{d} \theta_\ell x_\ell = \boldsymbol{\theta}^\top \boldsymbol{x}$$

this method cannot capture any nonlinear relationships. We can slightly alter linear regression to model these nonlinearities by taking a linear combination of m nonlinear basis functions $\phi_\ell(\boldsymbol{x})$:

$$f(\boldsymbol{x};\boldsymbol{\theta}) = \sum_{\ell=1}^{m} \theta_\ell \phi_\ell(\boldsymbol{x}) = \boldsymbol{\theta}^\top \boldsymbol{\phi}(\boldsymbol{x})$$

As an example, in the case of polynomial regression, we assume $\phi_\ell(\boldsymbol{x}) = x^{\ell-1}$, giving

$$f(\boldsymbol{x};\boldsymbol{\theta}) = \sum_{\ell=1}^{m-1} \theta_\ell x^\ell$$

Here, we've manually and explicitly defined our basis functions $\boldsymbol{\phi}(\boldsymbol{x})$, but moving forward, we will discuss two methods for automatically constructing **adaptive basis functions**: the kernel method in this chapter, and neural networks in the next chapter.

2.4.1 Kernel Regression

We define our **kernel function** to be a symmetric function $\mathbf{k}(\boldsymbol{x}, \boldsymbol{x}') = \mathbf{k}(\boldsymbol{x}', \boldsymbol{x})$ that acts as a "similarity measure" between the two points $\boldsymbol{x}$ and $\boldsymbol{x}'$. A typical example might be the **Gaussian radial basis function (RBF)** kernel:

$$\mathbf{k}(\boldsymbol{x}, \boldsymbol{x}') = \exp\Big(-\frac{1}{2h^2}\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2\Big)$$

where h is a positive parameter called the **bandwidth**. Given a dataset $\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1}^n$, we can construct a kernel representation of a point $\boldsymbol{x}$ by comparing

it with each observed point in $\mathcal{D}$:

$$\boldsymbol{\phi}(\boldsymbol{x}) = \begin{bmatrix} \mathbf{k}(\boldsymbol{x}, \boldsymbol{x}_1) \\ \vdots \\ \mathbf{k}(\boldsymbol{x}, \boldsymbol{x}_n) \end{bmatrix}$$

We can then use these features, representing relative similarity to the other points in the dataset, to form a powerful adaptive basis for our linear function class.

$$f(\boldsymbol{x}; \boldsymbol{\theta}) = \sum_i \theta_i \phi_i(\boldsymbol{x}) = \sum_i \theta_i \mathbf{k}(\boldsymbol{x}, \boldsymbol{x}^{(i)})$$

We can estimate $\boldsymbol{\theta}$ by minimizing the empirical loss function:

$$\begin{aligned} L(\boldsymbol{\theta}) &= \sum_{i=1}^{n} \Big(y_i - f(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}) \Big)^2 \\ &= \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{n} \theta_j \mathbf{k}(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) \right)^2 \\ &= \|\mathbf{Y} - \mathbf{K}\boldsymbol{\theta}\|_2^2 \end{aligned}$$

where $\mathbf{Y} = [y_1, \ldots, y_n]^\top \in \mathbb{R}^{n \times 1}$ and $\mathbf{K} = [\mathbf{k}(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})]_{ij=1}^{n} \in \mathbb{R}^{n \times n}$. $\mathbf{K}$ is often called the **gram matrix**, which we assume to be invertible. The above then yields

$$\begin{aligned} L(\boldsymbol{\theta}) &= \min_{\boldsymbol{\theta}} \|\mathbf{Y} - \mathbf{K}\boldsymbol{\theta}\|_2^2 \\ \boldsymbol{\theta} &= \mathbf{K}^{-1}\mathbf{Y} \end{aligned} \tag{7.1}$$

when we fit the curve exactly with n parameters matching the n data points.

Since (7.1) fits all the data exactly and presents a great risk of overfitting, we often introduce a regularization term $\Phi(\boldsymbol{\theta})$ with coefficient α:

$$L(\boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \|\mathbf{Y} - \mathbf{K}\boldsymbol{\theta}\|_2^2 + \alpha\Phi(\boldsymbol{\theta}) \tag{7.2}$$

where typically special regularization term is chosen that incorporates the kernel

$$\Phi(\boldsymbol{\theta}) = \boldsymbol{\theta}^\top \mathbf{K} \boldsymbol{\theta} \tag{7.3}$$

In order to make (7.3) non-negative, we require $\mathbf{K}$ to be **positive semi-definite**, i.e. $\boldsymbol{v}^\top \mathbf{K} \boldsymbol{v} \geq 0$ for any $\boldsymbol{v} \in \mathbb{R}^n$ and $\boldsymbol{v} \neq 0$.

2.5 Neural Networks

Neural networks present another opportunity to construct adaptive basis functions for nonlinear regression. Here each basis function $\phi(\boldsymbol{x})$, called a **neuron**, is assumed to have the form of

$$\phi(\boldsymbol{x}; \boldsymbol{w}) = \sigma\left(\sum_{i=1}^{d} w_i x_i + w_0\right)$$

where $\boldsymbol{w} = \{w_i\}_{i=0}^{d}$ is a set of weight coefficients that will be estimated from the dataset, and $\sigma(\cdot)$ is a 1-d nonlinear **activation function**.

Several common choices of activation function include

- **Rectified Linear Unit (ReLU):** $\sigma(t) = \max(0, t)$
- **Sigmoid:** $\sigma(t) = \dfrac{e^t}{1 + e^t}$
- **Thresholding:** $\sigma(t) = \mathbf{I}(t \geq 0)$

A neural network consists of several neurons, each with its own weight

$$f(\boldsymbol{x}; [\mathbf{a}, \mathbf{W}]) = \sum_{\ell=1}^{m} a_\ell \sigma\left(\sum_{i=1}^{d} w_{\ell,i} x_i + w_{\ell,0}\right) = \sum_{\ell=1}^{m} a_\ell \sigma(\boldsymbol{w}_\ell^\top [1; \boldsymbol{x}])$$

where $\mathbf{W} = \{\boldsymbol{w}_\ell\}$ is a matrix consisting of the weights $\boldsymbol{w}_\ell$ of each ℓ-th neuron. The two parameters $\mathbf{a}$ and $\mathbf{W}$ are estimated from the dataset $\mathcal{D}$ by minimizing the squared loss function by gradient descent.

$$\min_{\mathbf{a},\mathbf{W}} \left\{ L(\mathbf{a}, \mathbf{W}) = \mathbb{E}_{\mathcal{D}}\Big[(y - f(\boldsymbol{x}; [\mathbf{a}, \mathbf{W}]))^2\Big] \right\} \tag{8.1}$$

Appendix

Matrix Multiplication

Given a $n \times m$ matrix $\mathbf{A}$

$$\mathbf{A} = \begin{bmatrix} a_{11} & \dots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nm} \end{bmatrix}$$

and a $k \times l$ matrix $\mathbf{B}$

$$\mathbf{B} = \begin{bmatrix} b_{11} & \dots & b_{1l} \\ \vdots & \ddots & \vdots \\ b_{k1} & \dots & b_{kl} \end{bmatrix}$$

the multiplication of $\mathbf{A} \cdot \mathbf{B}$ is valid only if $m = k$ and will result in a $n \times k$ matrix $\mathbf{C}$ with elements

$$c_{ij} = \sum_{r=1}^{m} a_{ir} \cdot b_{rj}$$

2x2 Example

If $\mathbf{A}$ and $\mathbf{B}$ are both 2×2 matrices

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

then their product $\mathbf{C} = \mathbf{A} \cdot \mathbf{B}$ will be

$$\mathbf{C} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix}$$