CSE474/574: Introduction to Machine Learning(Fall 2017)

Instructor: Sargur N. Srihari

Probability Concepts

September 11, 2017

1 Overview

Machine Learning methods are based on probability theory and statistics. This project concerns probability distributions of several variables. We will learn to evaluate sufficient statistics: mean and variance of univariate distributions and covariance and correlation coefficient of pairs of variables. We will then use these statistics to construct compact representations of joint probability distributions known as Bayesian networks. Then we will evaluate the goodness of these representations by using the concept of likelihood. Finally we will use the Bayesian networks to answer some queries.

Some useful mathematical definitions and expressions are given here.

1.1 Random Variable

A random variable takes on a set of different values subject to chance. An example of a random variable X is the result of a coin toss which can take values 'Head' and 'Tail' which are not necessarily numeric. However X can be denoted by a random variable x which has a numerical value of 1 and 0. Each value that x can take has an associated probability. The mathematical function describing the possible values of a random variable and associated probabilities is known as a probability distribution.

1.2 Probability Mass Function and Probability Density Function

A random variable can be either discrete or continuous. In the case of a discrete random variable, a probability mass function specifies the exact probability distribution. In the case of a continuous random variable a probability density function specifies the distribution, but it must be integrated over an interval to yield a probability.

1.3 Mean, Variance and Standard Deviation

The sample mean μ of a univariate random variable X with N samples $x(i), i = 1, ..N$ has the form

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x(i) \tag{1}$$

The sample variance σ^2 is computed as

$$\sigma^2 = \frac{1}{N-1}\sum_{i=1}^{N}[x(i)-\mu]^2 \tag{2}$$

where σ is referred to as the standard deviation.

Corresponding Python functions:
numpy.mean() : Average or mean value of array;
numpy.var() : Variance;
numpy.std() : Standard deviation;

1.4 Covariance and Correlation Coefficient

The sample covariance of a pair of random variables X_1, X_2 with samples $x_1(i), x_2(i), i = 1, ..N$ is

$$\sigma_{12} = \frac{1}{N-1}\sum_{i=1}^{N}[x_1(i)-\mu_1][x_2(i)-\mu_2] \tag{3}$$

The correlation coefficient is the normalized covariance given by

$$\rho_{12} = \frac{\sigma_{12}}{\sigma_1\sigma_2} \tag{4}$$

The sample covariance matrix of a set of d variables $\mathbf{X} = \{X_1, ..X_d\}$ is

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \sigma_{13} & \cdots & \sigma_{1d} \\ \sigma_{12} & \sigma_2^2 & \sigma_{23} & \cdots & \sigma_{2d} \\ \sigma_{13} & \sigma_3^2 & \sigma_3^2 & \cdots & \sigma_{3d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sigma_{1d} & \sigma_{2d} & \sigma_{3d} & \cdots & \sigma_d^2 \end{bmatrix}$$

Corresponding Python functions:
numpy.cov() : Covariance;
numpy.corrcoef() : Correlation coefficients;

1.5 Normal Density

The Gaussian (or normal) distribution of a continuous random variable X with mean μ and variance σ^2, denoted as $x \sim \mathcal{N}(\mu, \sigma^2)$, has a probability density function (pdf) of the form

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right] \tag{5}$$

The multivariate form of this distribution for a vector $\mathbf{x}$ of d variables, mean vector μ and covariance matrix Σ, denoted $\mathbf{x} \sim \mathcal{N}(\mu, \Sigma)$ is

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}}\exp\left[-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right] \tag{6}$$

Corresponding Python functions:
numpy.scipy.stats.norm.pdf() : Normal probability density function;

1.6 Normalization

For a univariate population that is normally distributed and known mean and standard deviation, it is useful to convert it to a standard normal distribution $\mathcal{N}(0,1)$ by replacing X by $\frac{X-\mu}{\sigma}$.

1.7 Cumulative Distribution Function (cdf)

A probability can be determined from a cdf, which in turn can be determined from a pdf as follows:

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} p(x)dx \tag{7}$$

Thus a probability of X within a small interval $\pm\delta$ is:

$$P(X - \delta \leq X \leq X + \delta) = F(x + \delta) - F(x - \delta) \tag{8}$$

The multivariate version of cdf is straight-forward. For example, with two variables X_1 and X_2 with pdf $p(x_1, x_2)$ the cdf is

$$F(x_1, x_2) = P(X_1 \leq x_1, X_2 \leq x_2) = \int_{-\infty}^{x_1} \int_{-\infty}^{x_2} p(x_1, x_2)dx \tag{9}$$

Corresponding Python functions:
numpy.scipy.stats.norm.cdf() : Normal cumulative distribution function;
numpy.scipy.stats.multivariate_normal.pdf() : Multivariate normal probability distribution function;

1.8 Log-likelihood function

Given N independent samples $\mathbf{x}_1, ..\mathbf{x}_N$ from a probability distribution $p(\mathbf{x})$, the log-likelihood of observing the samples is given by

$$\mathcal{L}(\mathbf{x}_1, ..\mathbf{x}_N) = \sum_{i=1}^{N} log\ p(\mathbf{x}_i) \tag{10}$$

1.9 Conditional Probabilities

Given two variables X_1 and X_2 the *sum rule* of probability is:

$$p(x_1) = \sum_{Val(x_2)} p(x_1, x_2) \tag{11}$$

where Val is the set of values taken by its argument. The sum rule allows us to obtain the marginal probability $p(x_1)$ from the joint probability $p(x_1, x_2)$.

The *product rule* of probability is:

$$p(x_1, x_2) = p(x_1/x_2)p(x_2) \tag{12}$$

from which we get the *chain rule*

$$p(x_1, x_2, x_3) = p(x_1/x_2, x_3)p(x_2/x_3)p(x_3) \tag{13}$$

1.10 Bayesian Network Factorization

Given a Bayesian network G of d variables $\mathbf{X} = \{X_1, ..X_d\}$, the joint probability distribution is given by

$$p(\mathbf{x}) = \prod_{i=1}^{d} p(x_i/pa(x_i) \tag{14}$$

where $pa(x_i)$ are the parent variables of x_i.

1.11 Bayesian Network Construction

Ideally the Bayesian network for a set of random variables (for a fixed number of links) is one that maximizes the log-likelihood function $\mathcal{L}(\mathbf{x}_1, ..\mathbf{x}_N)$ for the observed data consisting of N samples. The negative of this quantity is referred to as the log-loss of the Bayesian network.

Learning the optimal Bayesian network from data is an NP-hard problem. In this project you are not required to construct the optimal Bayesian network. If you wish to implement a sub-optimal algorithm, you could consider the greedy algorithm. It starts with the most highly correlated pair of variables as being connected and then adds additional variables to the network, at each step maximizing the likelihood function.

1.12 Linear Gaussian Model

Consider a Bayesian network where continuous random variable Y has k continuous parents $\mathbf{X} = \{X_1, .., X_k\}$. Y is said to obey a linear Gaussian model with parameters $\beta_1, .., \beta_k$ and σ^2 if $P(Y|X_1, .., X_k) \sim \ N(\beta_0 + \beta_1 X_1, .. + \beta_k X_k; \sigma^2)$. In vector notation $P(Y|\mathbf{X}) \sim N(\beta_0 + {}^T\mathbf{X}; \sigma^2)$.

1.12.1 Learning the parameters

Task is to learn the parameters $= (\beta_1, .., \beta_k, \sigma^2)$ from a data set D consisting of N samples $\{x_1[1], .., x_k[n], y[n]\}, n = 1, ..N$. In order to determine maximum-likelihood parameters define the log-likelihood function as

$$\mathcal{L}(; \mathbf{x}[1], ..\mathbf{x}[N]) = \sum_{n=1}^{N} -\frac{1}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}(\beta_0 + \beta_1 x_1[n] + ..\beta_k x_k[n] - y[n])^2 \tag{15}$$

It is possible to get a closed-form solution. It involves solving a set of equations which are obtained by taking partial derivatives of the log-likelihood function, as follows.

Gradient of log-likelihood with respect to β_0 is

$$-\frac{1}{\sigma^2}(N\beta_0 + \beta_1 \sum_n x_1[n] + .. + \beta_k \sum_n x_k[n] - y[n]) \tag{16}$$

Equating to zero and rearranging we get

$$\beta_0 + \beta_1 \frac{1}{N}\sum_n x_1[n] + .. + \beta_k \frac{1}{N}\sum_n x_k[n] = \frac{1}{N}\sum_n y[n] \tag{17}$$

All the summations can be obtained from the data, thus giving us a linear equation.

Similarly we get k more linear equations by taking derivatives with respect to β_i. Standard linear algebra techniques are used to solve $k + 1$ simultaneous equations.

To estimate variance, take derivative with respect to σ and setting equal to zero we get

$$\sigma^2 = Cov_D[Y;Y] - \sum_i \sum_j \beta_i \beta_j Cov_D[X_i;X_j] \tag{18}$$