Alin Tomescu
6.867 Machine learning | Week 5, Tuesday, October 1st, 2013| Lecture 8

Lecture 8:

Gaussian processes

Multivariate Gaussian distributions

$$y(x), x \in \mathcal{X} = \mathbb{R}^d$$

For any subset of points $x^{(1)}, \dots, x^{(n)}$, the corresponding random variables evaluated at those points is a multivariate Gaussian: $y^{(1)} = x^{(1)}, \dots, y^{(n)} = x^{(n)}$

If it is a Gaussian, we only need to specify their mean and their covariance, to fully specify their definition. We must do that for any point x. So we need a mean function:

$$m(x) = E\{y(x)\}, \forall x$$

In our case, we always assume the mean is zero.

What's left is to specify a covariance function, which tells us for any two points x, x':

$$C(x, x') = E\{\big(y(x) - m(x)\big)\big(y(x') - m(x')\big)\}$$

$$\vec{y} = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(n)} \end{bmatrix} \sim N(\vec{0}, \mathcal{C})$$

($\vec{y}$ and $\vec{0}$ are $n \times 1$ vectors)

$$\mathcal{C} = C\big(x^{(i)}, x^{(j)}\big)$$

Dave's explanation of Gaussian processes

You have a set $Y = \{y_x\}_{x \in \chi}$ indexed by elements from a fixed set χ, and $Y \in \mathbb{R}^d$.

- χ is usually infinite, but it can also be finite. For our machine learning purposes, χ is the set of x_i values in the training set $\{(x_i, y_i)\}_{i=1}^n$

How can you sample a bivariate Gaussian? If v_1, v_2 are the eigenvectors, you can draw $z_1 \sim N(0, \sigma_1^2), z_2 \sim N(0, \sigma_2^2)$, (where σ_1 and σ_2 are the eigenvalues of the covariance matrix) and build $z_1 v_1 + z_2 v_2$.

Last time we went over Bayesian regression:

$$\theta = N(0, I)$$

$$y^{(i)} = \theta \phi\big(x^{(i)}\big) + \varepsilon_i, \varepsilon_i \sim N(0, \sigma^2)$$

Alin Tomescu
6.867 Machine learning | Week 5, Tuesday, October 1st, 2013| Lecture 8
In the problem set, we will get:

$$C\left(x^{(i)}, x^{(j)}\right) = (bayesian\ regression) = K\left(x^{(i)}, x^{(j)}\right) + \delta_{ij}\sigma^2$$

$$\delta_{ij} = \begin{cases} 1, i = j \\ 0, i \neq j \end{cases}$$

How to use a Gaussian process for prediction?

$$\begin{bmatrix} y \\ y_x \end{bmatrix} \sim N\left(\vec{0}_{(n+1)\times 1}, \begin{bmatrix} C & K_x \\ K_x^T & C(x,x) = K(x,x) + \sigma^2 \end{bmatrix}_{(n+1)\times(n+1)}\right)$$

$$K_x = \begin{bmatrix} K\left(x^{(1)}, x\right) \\ \vdots \end{bmatrix}$$

What is $P\left(y_x \mid y^{(1)}, \dots, y^{(n)}\right) = ?$

All I need is mean $\hat{\mu}(x) = E\left\{y_x \mid y^{(1)}, \dots, y^{(n)}\right\}$ and variance $v^2(x) = E\left\{\left(y_x - \mu(x)\right)^2 \mid y^{(1)}, \dots, y^{(n)}\right\}$

$$\hat{\mu}(x) = E\left\{y_x \mid y^{(1)}, \dots, y^{(n)}\right\} = K_x^T C_{n\times n}^{-1} \vec{y}_{n\times 1} = K_x^T (K + \sigma^2 I)^{-1} \vec{y}_{n\times 1}$$

$$v^2(x) = E\left\{\left(y_x - \mu(x)\right)^2 \mid y^{(1)}, \dots, y^{(n)}\right\} = K(x,x) + \sigma^2 - K_x^T (K + \sigma^2 I)^{-1} K_x$$

Let's work with $K(x, x') = xx'$

(see figure 3 & 4)

What is σ^2?

$$\sigma = \underset{\sigma}{\operatorname{argmax}}\ P\left(y^{(1)}, \dots y^{(n)} \mid x^{(1)}, \dots, x^{(n)}, \sigma^2\right)$$

Kernel matrix and covariance matrix are "equivalent", you can use one in place of the other.

The decision boundary is: $K_x^T C^{-1} y = 0$

Regression trees

$$\bigcup_{l \in L(T)} R_l = \mathbb{R}^d, L(T) = \text{leaves of regression tree}$$

$$\hat{y}(x) = \sum_{l \in L(T)} f_l[[x \in R_l]], \text{where } f_l[[x \in R_l]] = f_l, if\ x \in R_l$$

$$J(T) = \sum_{l \in L(T)} \sum_{i: x^{(i)} \in R_l} \left(y^{(i)} - f_l\right)^2$$

$\min J(T)$ over the entire tree is computationally hard. So to partition, we can do it greedily.

Bias is low. Variance is high.

Alin Tomescu

6.867 Machine learning | Week 5, Tuesday, October 1st, 2013| Lecture 8

Over-fitting is a problem.