Lecture 7:
Linear supervised learning

Nando de Freitas

www.cs.ubc.ca/~nando/340-2009/

September 2009

Outline

Linear regression is a supervised learning task. It is of great interest because:

- Many real processes can be approximated with linear models.
- Linear regression appears as part of larger problems.
- It can be solved analytically.
- It illustrates many of the approaches to machine learning.

Least squares

Given the data $\{x_{1:n}, y_{1:n}\}$, with $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$, we want to fit a hyper-plane that maps x to y.

Least squares

Learning and prediction with least squares

Least squares

Mathematically, the linear model is expressed as follows:

$$\widehat{y}_i = \theta_0 + \sum_{j=1}^{d} x_{ij}\theta_j$$

We let $x_{i,0} = 1$ to obtain $\widehat{y}_i = \sum_{j=0}^{d} x_{ij}\theta_j$

In matrix form, this expression is $\widehat{Y} = X\theta$

$$\begin{bmatrix} \widehat{y}_1 \\ \vdots \\ \widehat{y}_n \end{bmatrix} = \begin{bmatrix} x_{10} & \cdots & x_{1d} \\ \vdots & \vdots & \vdots \\ x_{n0} & \cdots & x_{nd} \end{bmatrix} \begin{bmatrix} \theta_0 \\ \vdots \\ \theta_d \end{bmatrix}$$

Least squares with multiple outputs

If we have several outputs $y_i \in \mathbb{R}^c$, our linear regression expression becomes:

Linear classification

Optimization approach

Our aim is to mininimise the quadratic cost between the output labels and the model predictions

$$C(\theta) = (Y - X\theta)^T (Y - X\theta)$$

Optimization approach

We will need the following results from matrix differentiation: $\frac{\partial A\theta}{\partial\theta} = A^T$ and $\frac{\partial\theta^T A\theta}{\partial\theta} = 2A^T\theta$

$$\frac{\partial C}{\partial\theta} =$$

Optimization approach

These are the **normal equations**. The solution (estimate) is:

$$\widehat{\theta} =$$

The corresponding predictions are

$$\widehat{Y} = HY =$$

where H is the “hat” matrix.

Geometric approach

$X^T(Y - \widehat{Y}) =$

Probability approach: Univariate Gaussian distribution

The probability density function of a Gaussian distribution is given by

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

where μ is the mean or center of mass and σ^2 is the variance.

$$X \sim \mathcal{N}(\mu, \sigma^2)$$

Multivariate Gaussian distribution

Let $x \in \mathbb{R}^n$. The pdf of an n-dimensional Gaussian is given by

$$p(x) = \frac{1}{2\pi^{n/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}$$

$$\mu = \begin{pmatrix} \mu_1 \\ : \\ \mu_n \end{pmatrix} = \begin{pmatrix} \mathbb{E}(x_1) \\ : \\ \mathbb{E}(x_n) \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} \sigma_{11} \cdots \sigma_{1n} \\ \cdots \\ \sigma_{n1} \cdots \sigma_{nn} \end{pmatrix} = \mathbb{E}[(X-\mu)(X-\mu)^T]$$

$$\sigma_{ij} = \mathbb{E}[X_i - \mu_i)(X_j - \mu_j)^T]$$

Multivariate Gaussian distribution

We can interpret each component of x, for example, as a feature of an image such as colour or texture. The term $\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)$ is called the **Mahalanobis distance**. Conceptually, it measures the distance between x and μ.

Maximum likelihood approach

If our errors are Gaussian distributed, we can use the model

$$Y = X\theta + \mathcal{N}(0, \sigma^2 I)$$

Note that the mean of Y is $X\theta$ and that its variance is $\sigma^2 I$. So we can equivalently write this expression using the probability density of Y **given** X, θ and σ:

$$p(Y|X,\theta,\sigma) = \left(2\pi\sigma^2\right)^{-n/2} e^{-\frac{1}{2\sigma^2}(Y-X\theta)^T(Y-X\theta)}$$

Maximum likelihood

Maximum likelihood

The maximum likelihood (ML) estimate of θ is obtained by taking the derivative of the log-likelihood, $\log p(Y|X, \theta, \sigma)$. The idea of maximum likelihood learning is to maximise the likelihood of seeing some data Y by modifying the parameters (θ, σ).

Maximum likelihood

The ML estimate of θ is:

Maximum likelihood

Proceeding in the same way, the ML estimate of σ is:

Lecture 8:
Regularization and ridge regression

Nando de Freitas
www.cs.ubc.ca/~nando/340-2009/

September 2009

All the answers so far are of the form

$$\widehat{\theta} = (XX^T)^{-1}X^TY$$

They require the inversion of XX^T. This can lead to problems if the system of equations is poorly conditioned. A solution is to add a small element to the diagonal:

$$\widehat{\theta} = (XX^T + \delta^2 I_d)^{-1}X^TY$$

This is the ridge regression estimate. It is the solution to the following **regularised quadratic cost function**

$$C(\theta) = (Y - X\theta)^T(Y - X\theta) + \delta^2\theta^T\theta$$

Proof

Ridge as constrained optimization

$$\min_{\theta\,:\,\theta^T\theta \le t} \left\{(Y - X\theta)^T(Y - X\theta)\right\}$$

Large values of θ are penalised. We are **shrinking** θ towards zero. This can be used to carry out **feature weighting**. **An input $x_{i,d}$ weighted by a small θ_d will have less influence on the ouptut y_i.**

The Lasso

Spectral view of ridge regression

Again, let $X \in \mathbb{R}^{n \times d}$ be factored as $X = U\Sigma V^T = \sum_{i=1}^{d} u_i \sigma_i v_i^T$,

The least squares prediction is: $\widehat{Y}_{LS} = \sum_{i=1}^{d} u_i u_i^T Y$

Likewise, for ridge regression we have:

$$\widehat{Y}_{ridge} = \sum_{i=1}^{d} \frac{\sigma_i^2}{\sigma_i^2 + \delta^2} u_i u_i^T Y$$

Regularization and noise filtering

The filter factor

$$f_i = \frac{\sigma_i^2}{\sigma_i^2 + \delta^2}$$

penalises small values of σ^2 (they go to zero at a faster rate).

Regularization and noise filtering

Small eigenvectors tend to be wobbly. The Ridge filter factor f_i gets rid of the wobbly eigenvectors. Therefore, the predictions tend to be more stable (smooth, regularised).

The smoothness parameter δ^2 is often estimated by cross-validation or Bayesian hierarchical methods.

Minimax and cross-validation

Lecture 9:
Bayesian learning for linear models

Nando de Freitas

www.cs.ubc.ca/~nando/340-2009/

September 2009

Bayesian linear-Gaussian supervised learning

In the Bayesian linear prediction setting, we focus on computing the posterior:

$$\begin{aligned} p(\theta|X,Y) &\propto p(Y|X,\theta)p(\theta) \\ &= \left(2\pi\sigma^2\right)^{-\frac{n}{2}} e^{\ \frac{1}{2\sigma^2}(Y\ \ X\theta)^T(Y\ \ X\theta)} p(\theta) \end{aligned}$$

We often want to maximise the posterior — that is, we look for the *maximum a poteriori* (MAP) estimate. In this case, the choice of prior determines a type of constraint! For example, consider a Gaussian prior $\theta \sim \mathcal{N}(0, \delta^2\sigma^2 I_d)$. Then

$$p(\theta|X,Y) \propto \left(2\pi\sigma^2\right)^{-\frac{n}{2}} e^{-\frac{1}{2\sigma^2}(Y-X\theta)^T(Y-X\theta)} \left(2\pi\sigma^2\delta^2\right)^{-\frac{d}{2}} e^{-\frac{1}{2\delta^2\sigma^2}\theta^T\theta}$$

$$
\begin{aligned}
p(\theta|X,Y) &= \left|2\pi\sigma^2 M\right|^{-\frac{1}{2}} e^{-\frac{1}{2\sigma^2}(\theta-\mu)^T M^{-1}(\theta-\mu)} \\
&\propto \left(2\pi\sigma^2\right)^{-\frac{n}{2}} e^{-\frac{1}{2\sigma^2}(Y-X\theta)^T(Y-X\theta)} \left(2\pi\sigma^2\delta^2\right)^{-\frac{d}{2}} e^{-\frac{1}{2\delta^2\sigma^2}\theta^T\theta}
\end{aligned}
$$

Bayesian posterior

So the posterior for θ is Gaussian:

$$p(\theta|X,Y) \; = \; \left|2\pi\sigma^2 M\right|^{-\frac{1}{2}} e^{-\frac{1}{2\sigma^2}(\theta-\mu)^T M^{-1}(\theta-\mu)}$$

with **sufficient statistics**:

$$\mathbb{E}(\theta|X,Y) \; = \; (XX^T + \delta^{-2} I_d)^{-1} X^T Y$$
$$var(\theta|X,Y) \; = \; (XX^T + \delta^{-2} I_d)^{-1} \sigma^2$$

Bayesian estimates, ridge and ML

The MAP point estimate is:

$$\widehat{\theta}_{MAP} \; = \; (XX^T + \delta^{-2} I_d)^{-1} X^T Y$$

It is the same as the ridge estimate (except for a trivial negative sign in the exponent of δ), which results from the L_2 constraint. A flat ("vague") prior with large variance (large δ) leads to the ML estimate.

$$\widehat{\theta}_{MAP} = \widehat{\theta}_{ridge} \quad \xrightarrow{\delta^2 \to 0} \quad \widehat{\theta}_{ML} = \widehat{\theta}_{SVD} = \widehat{\theta}_{LS}$$