2009年9月23日
下午 03:44

Linear Models for Regression

Supervised learning problems :
Applications in which the training data comprise examples of the input vectors along with their corresponding target vectors

supervised → regression , classification
unsupervised → density estimation , clustering

The goal of regression is to predict the value of one or more continuous target variables t given the values of a D-dimensional vector $\mathbf{x}$ of input variables

Techniques of basis functions

input: $\{\mathbf{x}_n\}$, $n = 1, \ldots, N$ (observations)
$\{t_n\}$, $n = 1, \ldots, N$ (target values)

goal: predict t for a new observation $\mathbf{x}$

① function $y(\mathbf{x})$
② probabilistic perspective : to model the distribution $p(t|\mathbf{x})$

How to use $p(t|\mathbf{x})$ for predicting t
⇒ to minimize the expected value of a suitably chosen loss function

Basis Functions

2009年9月24日
上午 09:58

Linear Basis Function Models

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + \cdots + w_D x_D$$
$$\mathbf{x} = (x_1, \ldots, x_D)^T$$

linear regression
a linear function of the parameters $w_0, \ldots, w_D$ (model)

not necessarily a linear function of input variables x_i

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x})$$

$\phi_j(\mathbf{x})$: basis functions (could be nonlinear)
w_0 : bias

dummy basis function $\phi_0(\mathbf{x}) = 1$

$$y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x})$$

$$\mathbf{w} = (w_0, \ldots, w_{M-1})^T \quad \boldsymbol{\phi} = (\phi_0, \ldots, \phi_{M-1})^T$$

nonlinear basis functions → $y(\mathbf{x}, \mathbf{w})$ nonlinear
But. linear model : linear in $\mathbf{w}$ (parameters)

examples of basis functions

$$\phi_j(x) = \exp\left\{-\frac{(x - \mu_j)^2}{2s^2}\right\}$$

2009年9月24日
上午 10:10

Sigmoidal basis function

$$\phi_j(x) = \sigma\left(\frac{x-\mu_j}{s}\right) \qquad \sigma(a) = \frac{1}{1+\exp(-a)}$$

$$\left(\tanh(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}} = \frac{1-e^{-2a}}{1+e^{-2a}} = 2\sigma(2a) - 1\right)$$

erratum PRML p.139

others:
Fourier basis, wavelet

Maximum Likelihood and Least Squares

$t = y(x, w) + \varepsilon$ ε is zero mean with precision β ($\beta^{-1} = \sigma^2$)

$$p(t|x, w, \beta) = \mathcal{N}\left(t \mid y(x, w), \beta^{-1}\right)$$

{ assume a squared loss function, then the optimal prediction for a new value x will be given by the conditional mean of the target variable

$$E[t|x] = \int t\, p(t|x)\, dt = y(x, w)$$

a data set of inputs $\mathbf{X} = \{x_1, \ldots, x_N\}$
$\mathbf{t} = [t_1, \ldots, t_N]^T$

$$p(\mathbf{t} \mid \mathbf{X}, w, \beta) = \prod_{n=1}^{N} \mathcal{N}\left(t_n \mid w^T \phi(x_n), \beta^{-1}\right)$$

assumption? data points are drawn independently from $p(t|x, w, \beta)$

drop x for brevity

2009年9月24日
上午 10:29

$$\ln p(\mathbf{t} \mid w, \beta) = \sum_{n=1}^{N} \ln \mathcal{N}\left(t_n \mid w^T \phi(x_n), \beta^{-1}\right)$$
$$= \frac{N}{2} \ln\beta - \frac{N}{2} \ln(2\pi) - \beta E_D(w)$$

$$\mathcal{N}(x \mid \mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp\left\{\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$
$$= \frac{\sqrt{\beta}}{\sqrt{2\pi}} \exp\left\{-0.5\beta(x-\mu)^2\right\}$$

$$E_D(w) = \frac{1}{2} \sum_{n=1}^{N} \left(t_n - w^T \phi(x_n)\right)^2$$

maximize the log likelihood function for w
Gaussian noise likelihood
$\Leftrightarrow$ sum-of-squares error function given by $E_D(w)$

$$\nabla \ln p(\mathbf{t} \mid w, \beta) = \sum_{n=1}^{N} \left\{t_n - w^T \phi(x_n)\right\} \phi(x_n)^T \cdot \beta$$

$$\phi(x_n) = (\phi_0(x_n), \ldots, \phi_{M-1}(x_n))^T$$

appendix C.19
$\frac{\partial}{\partial x}(x^T a) = a^T$

Setting the gradient to zero gives

$$0 = \sum_{n=1}^{N} t_n \phi^T(x_n) - w^T \left(\sum_{n=1}^{N} \phi(x_n) \phi(x_n)^T\right)$$

define the design matrix

$$\Phi_{nj} = \phi_j(x_n) \qquad \Phi = \begin{bmatrix} \phi_0(x_1) & \phi_1(x_1) & \cdots \\ \phi_0(x_2) & & \ddots \\ \vdots & & \\ \phi_0(x_n) & & \end{bmatrix}$$

$$\sum_{n=1}^{N} t_n \phi^T(x_n) = \left(\sum_{n=1}^{N} \phi_0(x_n) t_n, \sum_{n=1}^{N} \phi_1(x_n) t_n, \dots \right)$$

$$= \left(\Phi^T t \right)^T$$

$$\sum_{n=1}^{N} \phi(x_n) \phi(x_n)^T = \sum_{n=1}^{N} \begin{pmatrix} \phi_0(x_n) \\ \phi_1(x_n) \\ \vdots \end{pmatrix} \left(\phi_0(x_n), \phi_1(x_n), \dots \right)$$

$$= \begin{pmatrix} \sum_{n=1}^{N} \phi_0(x_n)\phi_0(x_n) & \sum_{n=1}^{N} \phi_0(x_n)\phi_1(x_n) & \cdots \\ \vdots & \ddots & \\ \end{pmatrix}$$

$$= \Phi^T \Phi$$

$$\Rightarrow \quad (\Phi^T t)^T = w^T (\Phi^T \Phi)$$

$$\Phi^T t = (\Phi^T \Phi)^T w = \Phi^T \Phi w$$

$$\Rightarrow \quad w = (\Phi^T \Phi)^{-1} \Phi^T t$$

$\Phi^\dagger \equiv (\Phi^T \Phi)^{-1} \Phi$: Moore-Penrose pseudo-inverse

why $\Phi^T \Phi$ invertible ?

> Φ has linearly independent columns
>
> ⇕
>
> linearly independent basis functions

Proof: If $\Phi^T \Phi$ is not invertible

$\exists\, v \neq 0$ such that $\Phi^T \Phi v = 0$

$$v^T \Phi^T \Phi v = 0$$

$$\| \Phi v \|^2 = 0$$

$$\Phi v = 0$$

$\Rightarrow \Phi$ columns linearly dependent

contradiction

Bias term

$$E_D(w) = \frac{1}{2} \sum_{n=1}^{N} \left\{ t_n - w_0 - \sum_{j=1}^{M-1} w_j \phi_j(x_n) \right\}^2$$

setting the derivative w.r.t. w_0 equal to zero

$$w_0 = \bar{t} - \sum_{j=1}^{M-1} w_j \bar{\phi}_j$$

$$\bar{t} = \frac{1}{N} \sum_{n=1}^{N} t_n \qquad \bar{\phi}_j = \frac{1}{N} \sum_{n=1}^{N} \phi_j(x_n)$$

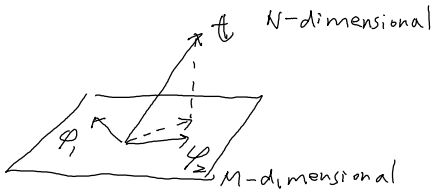
β :

$$\frac{1}{\beta_{ML}} = \frac{1}{N} \sum_{n=1}^{N} \left\{ t_n - w_{ML}^T \phi(x_n) \right\}^2$$

↳ (σ^2)

2009年9月24日
上午 10:57

Geometry of Least Squares



N observations $\Rightarrow$ N-dimensional vector $\mathbf{t}$

M basis functions $\Rightarrow$ M-dimensional subspace

projection onto the M-dimensional subspace spanned by $\{\varphi_j\}$

$$\mathbf{w} = (\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\mathbf{t}$$

$\boldsymbol{\Phi}^T\boldsymbol{\Phi}$ close to singular $\rightarrow$ large parameters

numerical difficulties

regularization

Sequential learning (on-line algorithm)

2009年9月24日
上午 11:04

stochastic gradient descent

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E_n$$

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} + \eta (t_n - \mathbf{w}^{(\tau)T}\phi(\mathbf{x}_n))\phi(\mathbf{x}_n)$$

Regularized Least Squares

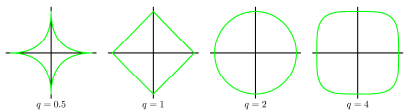
$$E_D(\mathbf{w}) + \lambda E_W(\mathbf{w})$$

simplest regularizer $E_W(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{w}$

$$\frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^T\phi(\mathbf{x}_n)\}^2 + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}$$

$$\mathbf{w} = (\lambda\mathbf{I} + \boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\mathbf{t}$$

general form $\frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^T\phi(\mathbf{x}_n)\}^2 + \frac{\lambda}{2}\sum_{j=1}^{M}|w_j|^q$

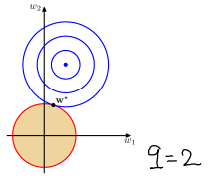


q=0.5 q=1 q=2 q=4

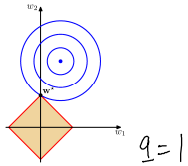
$q \le 1$ sparse

2-dim case

w_1, w_2



q=2



q=1

2009年9月24日
下午 12:19

overfitting : data sets of limit size

model selection :
- number of basis functions
- regularization coefficient

to reduce model complexity

Multiple Outputs

$$y(x, w) = W^T \phi(x)$$

target vector $\mathbb{Z}$ (K-dim)

$$p(\mathbb{Z} \mid x, W, \beta) = \mathcal{N}\left(\mathbb{Z} \mid W^T \phi(x), \beta^{-1} I\right)$$

↖ simple

$$\ln p(\mathbb{Z}, \mathbb{X}, W, \beta) = \frac{NK}{2} \ln\left(\frac{\beta}{2\pi}\right) - \frac{\beta}{2} \sum_{n=1}^{N} \| \mathbb{Z}_n - W^T \phi(x_n) \|^2$$

Assignment:

$$W_{ML} = (\Phi^T \Phi)^{-1} \Phi^T \mathbb{Z}$$

$$W_k = (\Phi^T \Phi)^{-1} \Phi^T \mathbb{Z}_k$$

$\mathbb{Z}$: $N \times K$

$\mathbb{Z}_k$: N-dim column vector

Exercise 3.6

$$\frac{d}{dX}(a^T X^T X b) = X(ab^T + ba^T)$$

↖

$$\frac{d}{dX}(a^T X^T b) = ba^T$$

$$\frac{d}{dX}(a^T X b) = ab^T$$

2009年9月24日
下午 12:32

The Bias-Variance Decomposition

Assume we have the conditional distribution $p(t \mid x)$ (estimated by some approach)

What is the optimal prediction ?

It depends on the loss function you choose.

For the squared loss function, the optimal prediction is given by the conditional expectation

$$h(x) = E[t \mid x] = \int t \, p(t \mid x) \, dt$$

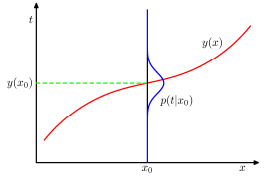
p. 45

The expected squared loss can be written as

$$E[L] = \int \{y(x) - h(x)\}^2 p(x) \, dx + \iint \{h(x) - t\}^2 p(x, t) \, dx \, dt$$

↓ independent of $y(x)$

intrinsic noise



$$E[L] = \iint \{y(x) - t\}^2 p(x, t) \, dx \, dt$$

$$\{y(x) - t\}^2 = \{y(x) - E[t \mid x] + E[t \mid x] - t\}^2$$

$$= \{y(x) - E[t \mid x]\}^2 + 2\{y(x) - E[t \mid x]\}\{E[t \mid x] - t\} + \{E[t \mid x] - t\}^2$$

2009年9月24日
下午 12:45

for a unlimited supply of data we can make $y(x) - h(x) = 0$

In practice, a data set $\mathcal{D}$ contains only a finite number N of data points

$$E[L] = \int \{y(x) - h(x)\}^2 p(x)\, dx + \iint \{h(x) - t\}^2 p(x,t)\, dx\, dt$$

Given a particular data set $\mathcal{D}$, the first term of $E[L]$ takes the form

$$\{y(x;\mathcal{D}) - h(x)\}^2$$

$$\{y(x;\mathcal{D}) - E_{\mathcal{D}}[y(x;\mathcal{D})] + E_{\mathcal{D}}[y(x;\mathcal{D})] - h(x)\}^2$$

$$= \{y(x;\mathcal{D}) - E_{\mathcal{D}}[y(x;\mathcal{D})]\}^2 + \{E_{\mathcal{D}}[y(x;\mathcal{D})] - h(x)\}^2$$

$$+ 2\{y(x;\mathcal{D}) - E_{\mathcal{D}}[y(x;\mathcal{D})]\}\{E_{\mathcal{D}}[y(x;\mathcal{D})] - h(x)\}$$

Take the expectation w.r.t. $\mathcal{D}$

$$E_{\mathcal{D}}[\{y(x;\mathcal{D}) - h(x)\}^2]$$

$$= \{E_{\mathcal{D}}[y(x;\mathcal{D})] - h(x)\}^2 + E_{\mathcal{D}}[\{y(x;\mathcal{D}) - E_{\mathcal{D}}[y(x;\mathcal{D})]\}^2]$$

$(\text{bias})^2$ variance

2009年9月24日
下午 12:58

expected loss = $(\text{bias})^2$ + Variance + noise

$$(\text{bias})^2 = \int \{E_{\mathcal{D}}[y(x;\mathcal{D})] - h(x)\}^2 p(x)\, dx$$

$$\text{variance} = \int E_{\mathcal{D}}[\{y(x;\mathcal{D}) - E_{\mathcal{D}}[y(x;\mathcal{D})]\}^2] p(x)\, dx$$

$$\text{noise} = \iint \{h(x) - t\}^2 p(x,t)\, dx\, dt$$

Run MATLAB fig 3.5