GAUSSIAN PROCESSES

2009年11月26日
上午 11:02

$y(x) = w^T \phi(x)$ prior over w : $p(w) = \mathcal{N}(w | 0, \alpha^{-1} I)$

α precision

probability distribution over w
induces probability distribution over $y(x)$

In practice, we evaluate this function at specific values of x: $x_1, \ldots, x_N$, we get $y(x_1), \ldots, y(x_N)$

What does the joint distribution of $y(x_1), \ldots, y(x_N)$ look like?

$\underline{y = \Phi w}$ $\Phi_{nk} = \phi_k(x_n)$

linear combination of Gaussian distributed variables

So y is Gaussian

$$E[y] = \Phi E[w] = 0$$
$$\mathrm{cov}[y] = E[(y-0)(y-0)^T] = \Phi E[ww^T] \Phi^T = \frac{1}{\alpha} \Phi \Phi^T = K$$

$$K_{nm} = k(x_n, x_m) = \frac{1}{\alpha} \phi(x_n)^T \phi(x_m)$$

---

① A Gaussian process is defined as a probability distribution over functions $y(x)$ such that the set of values of $y(x)$ evaluated at an arbitrary set of point $x_1, \ldots, x_N$ jointly have a Gaussian distribution.

② The joint distribution over N variables $y_1, \ldots, y_N$ is specified completely by the mean and covariance

mean function and covariance function

Gaussian Processes for Regression

2009年11月26日
上午 11:51

$t_n = y_n + \epsilon_n$ $y_n = y(x_n)$ ϵ_n is a random noise variable

$$p(t_n | y_n) = \mathcal{N}(t_n | y_n, \beta^{-1})$$

precision of the noise

$p(\mathbf{t} | \mathbf{y}) = \mathcal{N}(\mathbf{t} | \mathbf{y}, \beta^{-1} I_N)$ isotropic Gaussian

noise is independent for each data point

$p(\mathbf{y}) = \mathcal{N}(\mathbf{y} | 0, K)$ (from the previous pages)

we want to find the marginal
$$p(\mathbf{t}) = \int p(\mathbf{t}|\mathbf{y}) p(\mathbf{y}) \, d\mathbf{y} = \mathcal{N}(\mathbf{t} | 0, C)$$

$$\Rightarrow \quad C = \beta^{-1} I_N + K$$

$$p(a) = \mathcal{N}(a | \mu, \Lambda^{-1})$$
$$p(b|a) = \mathcal{N}(b | Aa + d, L^{-1})$$
$$p(b) = \mathcal{N}(b | A\mu + d, L^{-1} + A\Lambda^{-1}A^T)$$

2009年11月26日
下午 10:02

make predictions

$\mathbf{t}_N = (t_1, \dots, t_N)^T$

predictive distribution $p(t_{N+1} \mid \mathbf{t}_N)$

start by writing down the joint distribution

$p(\mathbf{t}_{N+1}) \longrightarrow p(t_{N+1} \mid \mathbf{t}_N)$

$$p(\mathbf{t}_{N+1}) = \mathcal{N}(\mathbf{t}_{N+1} \mid 0, C_{N+1})$$

$$C_{N+1} = \begin{pmatrix} C_N & \mathbf{k} \\ \mathbf{k}^T & c \end{pmatrix}$$

$c = k(\mathbf{x}_{N+1}, \mathbf{x}_{N+1}) + \beta^{-1}$

$$\mathbf{k} = \begin{pmatrix} \vdots \\ k(\mathbf{x}_n, \mathbf{x}_{N+1}) \\ \vdots \end{pmatrix}$$

$$\mu = \begin{pmatrix} \mu_a \\ \mu_b \end{pmatrix} = \begin{pmatrix} \mathbf{t} \\ t_{N+1} \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{pmatrix}$$

$$\mu_{b|a} = \mu_b + \Sigma_{ba} \Sigma_{aa}^{-1} (x_a - \mu_a)$$

$$\Sigma_{b|a} = \Sigma_{bb} - \Sigma_{ba} \Sigma_{aa}^{-1} \Sigma_{ab}$$

Joint $\rightarrow$ conditional

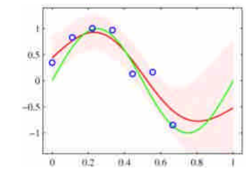
$$m(\mathbf{x}_{N+1}) = \mathbf{k}^T C_N^{-1} \mathbf{t}$$

$$\sigma^2(\mathbf{x}_{N+1}) = c - \mathbf{k}^T C_N^{-1} \mathbf{k}$$

2009年11月26日
下午 10:10

The predictive distribution is a Gaussian whose mean and variance both depend on $\mathbf{x}_{N+1}$

(Fig. 6.8)



λ_i eigenvalue of K

then $\lambda_i + \beta^{-1}$ eigenvalue of C

$$m(\mathbf{x}_{N+1}) = \sum_{n=1}^{N} a_n k(\mathbf{x}_n, \mathbf{x}_{N+1})$$

a_n is the nth component of $C_N^{-1}\mathbf{t}$

e.g. $k(\mathbf{x}, \mathbf{x}') = g(\|\mathbf{x} - \mathbf{x}'\|)$ RBF

---

for kernel methods, C is an $N \times N$ matrix

inversion needs $O(N^3)$

for linear regression S_N is $M \times M$

inversion needs $O(M^3)$ $\quad S_N^{-1} = \alpha I + \beta \Phi^T \Phi$

what if the data set is large?

sparse Gaussian processes

Gaussian Processes for Classification

2009年11月26日
下午 10:20

$t \in \{0, 1\}$ $\quad y = \sigma(a)$ $\quad a(\mathbf{x})$ is linear

$$p(t|a) = \sigma(a)^t (1-\sigma(a))^{1-t}$$

$$\begin{cases} p(a_{N+1}) = \mathcal{N}(a_{N+1} \mid 0, C_{N+1}) \\ C(\mathbf{x}_n, \mathbf{x}_m) = k(\mathbf{x}_n, \mathbf{x}_m) + \nu \delta_{nm} \end{cases} \qquad \delta_{nm} = \begin{cases} 1 & n=m \\ 0 & n \neq m \end{cases}$$

we want to get $p(t_{N+1}=1 \mid \mathbf{t}_N) = \int \underbrace{p(t_{N+1}=1 \mid a_{N+1})}_{= \sigma(a_{N+1})} p(a_{N+1} \mid \mathbf{t}_N)\, d a_{N+1}$

this integration is intractable

we may apply Laplace approximation

first, for the second term

$$p(a_{N+1} \mid \mathbf{t}_N) = \int p(a_{N+1}, \mathbf{a}_N \mid \mathbf{t}_N)\, d\mathbf{a}_N$$

$$= \frac{1}{p(\mathbf{t}_N)} \int p(a_{N+1}, \mathbf{a}_N)\, p(\mathbf{t}_N \mid a_{N+1}, \mathbf{a}_N)\, d\mathbf{a}_N$$

$$= \frac{1}{p(\mathbf{t}_N)} \int p(a_{N+1} \mid \mathbf{a}_N)\, p(\mathbf{a}_N)\, p(\mathbf{t}_N \mid \mathbf{a}_N)\, d\mathbf{a}_N$$ use independence

$$= \int p(a_{N+1} \mid \mathbf{a}_N)\, p(\mathbf{a}_N \mid \mathbf{t}_N)\, d\mathbf{a}_N$$

from GP regression, we know $p(a_{N+1} \mid \mathbf{a}_N) = \mathcal{N}(a_{N+1} \mid \mathbf{k}^T C_N^{-1} \mathbf{a}_N,\ c - \mathbf{k}^T C_N^{-1} \mathbf{k})$

so we need to find a Gaussian to fit $p(\mathbf{a}_N \mid \mathbf{t}_N)$

2009年11月26日
下午 10:46

$p(\mathbf{a}_N)$ is zero-mean Gaussian

$$p(\mathbf{t}_N \mid \mathbf{a}_N) = \prod_{n=1}^{N} \sigma(a_n)^{t_n} (1-\sigma(a_n))^{1-t_n}$$

$$= \prod_{n=1}^{N} \left(\frac{1}{1+e^{-a_n}}\right)^{t_n} \left(\frac{e^{-a_n}}{1+e^{-a_n}}\right)^{1-t_n}$$

$$= \prod_{n=1}^{N} \left(\frac{1}{1+e^{-a_n}}\right)^{t_n} \left(\frac{1}{1+e^{-a_n}}\right)^{1-t_n} \left(e^{-a_n}\right)^{1-t_n}$$

$$= \prod_{n=1}^{N} \frac{1}{1+e^{-a_n}} e^{-a_n} \cdot e^{a_n t_n}$$

$$= \prod_{n=1}^{N} e^{a_n t_n} \frac{1}{1+e^{a_n}} = \prod_{n=1}^{N} e^{a_n t_n} \sigma(-a_n)$$

① find the mode

$$\Psi(\mathbf{a}_N) = \ln p(\mathbf{a}_N \mid \mathbf{t}_N) = \ln p(\mathbf{a}_N) + \ln p(\mathbf{t}_N \mid \mathbf{a}_N)$$

$$= -\frac{1}{2} \mathbf{a}_N^T C_N^{-1} \mathbf{a}_N - \frac{N}{2} \ln(2\pi) - \frac{1}{2} \ln |C_N| + \mathbf{t}_N^T \mathbf{a}_N - \sum_{n=1}^{N} \ln(1+e^{a_n})$$

mode $\quad \nabla \Psi(\mathbf{a}_N) = \mathbf{t}_N - \boldsymbol{\sigma}_N - C_N^{-1} \mathbf{a}_N \qquad \boldsymbol{\sigma}_N = \begin{pmatrix} \vdots \\ \sigma(a_n) \\ \vdots \end{pmatrix}$

optimized by iterative reweighted least squares

IRLS

2009年11月26日
下午 11:01

Compute the Hessian for IRLS

$$\nabla\nabla\Psi(\mathbf{a}_N) = -\mathbf{W}_N - \mathbf{C}_N^{-1}$$

$$\mathbf{W}_N = \begin{bmatrix} \ddots & & \\ & \sigma(a_n)(1-\sigma(a_n)) & \\ & & \ddots \end{bmatrix} \quad \text{positive definite}$$

$\Rightarrow -\nabla\nabla\Psi(\mathbf{a}_N)$ positive definite

$p(\mathbf{a}_N \mid \mathbf{t}_N)$ is log convex $\Rightarrow$ single mode, global maximum

$$\mathbf{a}_N^{new} = \mathbf{C}_N(\mathbf{I} + \mathbf{W}_N\mathbf{C}_N)^{-1}\{\mathbf{t}_N - \boldsymbol{\sigma}_N + \mathbf{W}_N\mathbf{a}_N\}$$

$$\mathbf{a}^* = \mathbf{C}_N(\mathbf{t}_N - \boldsymbol{\sigma}_N) \quad \Big|_{\nabla\Psi(\mathbf{a}_N)=0}$$

Hessian $H = -\nabla\nabla\Psi(\mathbf{a}_N^*) = \mathbf{W}_N + \mathbf{C}_N^{-1}$ evaluated at $\mathbf{a}^*$

so $q(\mathbf{a}_N) = \mathcal{N}(\mathbf{a}_N \mid \mathbf{a}_N^*, H_{\mathbf{a}^*}^{-1})$

is an approximation to $p(\mathbf{a}_N \mid \mathbf{t}_N)$

back to $p(a_{N+1} \mid \mathbf{t}_N) = \int p(a_{N+1} \mid \mathbf{a}_N)\, p(\mathbf{a}_N \mid \mathbf{t}_N)\, d\mathbf{a}_N$

$$\simeq \int \underbrace{p(a_{N+1} \mid \mathbf{a}_N)}_{\text{Gaussian}}\, \underbrace{q(\mathbf{a}_N)}_{\text{Gaussian}}\, d\mathbf{a}_N$$

recall $p(a_{N+1} \mid \mathbf{a}_N) = \mathcal{N}(a_{N+1} \mid \mathbf{k}^T\mathbf{C}_N^{-1}\mathbf{a}_N,\ c - \mathbf{k}^T\mathbf{C}_N^{-1}\mathbf{k})$

combined with $q(\mathbf{a}_N)$

we get $p(a_{N+1} \mid \mathbf{t}_N) \simeq \mathcal{N}(\mu, s^2)$

where

$$\mu = \mathbf{k}^T(\mathbf{t}_N - \boldsymbol{\sigma}_N), \quad s^2 = c - \mathbf{k}^T(\mathbf{W}_N^{-1} + \mathbf{C}_N)^{-1}\mathbf{k}$$

2009年11月26日
下午 11:24

Finally, the predictive distribution

$$p(t_{N+1}=1 \mid \mathbf{t}_N) = \int \underbrace{p(t_{N+1}=1 \mid a_{N+1})}_{\sigma(a_{N+1})}\ \underbrace{p(a_{N+1} \mid \mathbf{t}_N)}_{\mathcal{N}(\mu, s^2)}\, d a_{N+1}$$

$$\simeq \sigma(\kappa(\sigma^2)\mu) \qquad (4.153)$$

We skip the part of deciding parameters for covariance function

SPARSE KERNEL MACHINES

2009年11月26日
下午 11:34

$k(x_n, x_m)$ evaluated for all possible pairs x_n, x_m

Maximum Margin Classifiers $\rightarrow$ sparse

consider a two-class classification problem

$$y(x) = w^T \phi(x) + b \qquad \phi(x): \text{input space} \rightarrow \text{feature space}$$

$x_1, \ldots, x_N$ with labels $t_1, \ldots, t_N$ $\quad t_n \in \{-1, 1\}$

Assume linearly separable in feature space

$$y(x_n) > 0 \text{ for } t_n = +1$$
$$y(x_n) < 0 \text{ for } t_n = -1 \quad \Rightarrow \quad t_n y(x_n) > 0$$

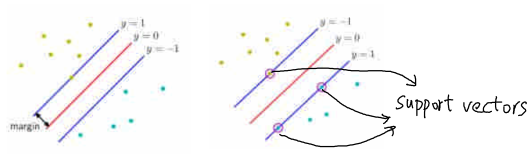
recall the perceptron algorithm, we might get multiple solutions.

we want to find the solution that give the smallest generalization error

SVMs achieve the goal by maximizing the margin

margin : the smallest distance between the decision boundary and any of the samples

$\Rightarrow$ choose the decision boundary of which the margin is maximum



2009年11月26日
下午 11:48

distance of a point x_n to the decision surface :

$$t_n \frac{w^T}{\|w\|} \left(\phi(x_n) - \phi(z) \right) \qquad z \text{ is on the decision surface} \Rightarrow w^T\phi(z) + b = 0$$

$$= \frac{t_n \left(w^T\phi(x_n) - w^T\phi(z) \right)}{\|w\|}$$

$$= \frac{t_n(w^T\phi(x_n) + b)}{\|w\|} = \frac{t_n y(x_n)}{\|w\|}$$

maximize the margin

$$\arg\max_{w,b} \left\{ \frac{1}{\|w\|} \min_n \left[t_n \left(w^T\phi(x_n) + b \right) \right] \right\}$$

$w \rightarrow k w$
$b \rightarrow k b$ $\quad$ scaling does not change $\dfrac{t_n y(x_n)}{\|w\|}$

set $t_n (w^T \phi(x^*) + b) = 1$ as a constraint

x^* is the closest sample to the surface

so we have

$$t_n (w^T\phi(x_n) + b) \geqslant 1, \quad n = 1, \ldots, N$$

equality holds $\longrightarrow$ active constraints

there must be at least two active constraints

we may rewrite the optimization problem as

$$\arg\max_{w,b} \frac{1}{\|w\|} \Rightarrow \arg\min_{w,b} \frac{1}{2}\|w\|^2 \quad \text{subject to} \quad t_n(w^T\phi(x_n) + b) \geqslant 1, \quad n = 1, \ldots, N$$