PRML学习笔记——第六章

Kernel Methods

6.1 Dual Representations

考虑SSE function:

$$J(\mathbf{w})=\frac{1}{2}\sum_{n=1}^{N}\left\{\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}\left(\mathbf{x}_{n}\right)-t_{n}\right\}^{2}+\frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

求导令为0可得:

$$\mathbf{w}=-\frac{1}{\lambda}\sum_{n=1}^{N}\left\{\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}\left(\mathbf{x}_{n}\right)-t_{n}\right\}\boldsymbol{\phi}\left(\mathbf{x}_{n}\right)=\sum_{n=1}^{N}a_{n}\boldsymbol{\phi}\left(\mathbf{x}_{n}\right)=\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{a}$$

引入*gram matrix* $\mathbf{K}=\boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathrm{T}}$
那么SSE就能写成关于gram matrix的形式:

$$J(\mathbf{a})=\frac{1}{2}\mathbf{a}^{\mathrm{T}}\mathbf{K}\mathbf{K}\mathbf{a}-\mathbf{a}^{\mathrm{T}}\mathbf{K}\mathbf{t}+\frac{1}{2}\mathbf{t}^{\mathrm{T}}\mathbf{t}+\frac{\lambda}{2}\mathbf{a}^{\mathrm{T}}\mathbf{K}\mathbf{a}.$$

model的output为:

$$y(\mathbf{x})=\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x})=\mathbf{a}^{\mathrm{T}}\boldsymbol{\Phi}\boldsymbol{\phi}(\mathbf{x})=\mathbf{k}(\mathbf{x})^{\mathrm{T}}\left(\mathbf{K}+\lambda\mathbf{I}_{N}\right)^{-1}\mathbf{t}$$

可以看到,predict的时候只需要使用kernels的结果,避免使用原始data的feature $\phi(\mathbf{x})$.这在high,even infinity dimension的feature space上非常有用.

6.2 Constructing Kernels

> 一个valid kernel function等价于对应的gram matrix是半正定的.

利用这个等价关系仍然难以构造复杂的kernel function.

$$\begin{aligned}
k\left(\mathbf{x}, \mathbf{x}^{\prime}\right) &= c k_1\left(\mathbf{x}, \mathbf{x}^{\prime}\right) \\
k\left(\mathbf{x}, \mathbf{x}^{\prime}\right) &= f(\mathbf{x}) k_1\left(\mathbf{x}, \mathbf{x}^{\prime}\right) f\left(\mathbf{x}^{\prime}\right) \\
k\left(\mathbf{x}, \mathbf{x}^{\prime}\right) &= q\left(k_1\left(\mathbf{x}, \mathbf{x}^{\prime}\right)\right) \\
k\left(\mathbf{x}, \mathbf{x}^{\prime}\right) &= \exp\left(k_1\left(\mathbf{x}, \mathbf{x}^{\prime}\right)\right) \\
k\left(\mathbf{x}, \mathbf{x}^{\prime}\right) &= k_1\left(\mathbf{x}, \mathbf{x}^{\prime}\right) + k_2\left(\mathbf{x}, \mathbf{x}^{\prime}\right) \\
k\left(\mathbf{x}, \mathbf{x}^{\prime}\right) &= k_1\left(\mathbf{x}, \mathbf{x}^{\prime}\right) k_2\left(\mathbf{x}, \mathbf{x}^{\prime}\right) \\
k\left(\mathbf{x}, \mathbf{x}^{\prime}\right) &= k_3\left(\phi(\mathbf{x}), \phi\left(\mathbf{x}^{\prime}\right)\right) \\
k\left(\mathbf{x}, \mathbf{x}^{\prime}\right) &= \mathbf{x}^{\mathrm{T}} \mathbf{A} \mathbf{x}^{\prime} \\
k\left(\mathbf{x}, \mathbf{x}^{\prime}\right) &= k_a\left(\mathbf{x}_a, \mathbf{x}_a^{\prime}\right) + k_b\left(\mathbf{x}_b, \mathbf{x}_b^{\prime}\right) \\
k\left(\mathbf{x}, \mathbf{x}^{\prime}\right) &= k_a\left(\mathbf{x}_a, \mathbf{x}_a^{\prime}\right) k_b\left(\mathbf{x}_b, \mathbf{x}_b^{\prime}\right)
\end{aligned}$$

已知一些简单的kernel function,并利用这些性质,就可以构造复杂的kernel function.

6.3. Radial Basis Function Networks

将model内的basis function固定为radial basis function:

$$f(\mathbf{x}) = \sum_{n=1}^{N} w_n h\left(\left\|\mathbf{x} - \mathbf{x}_n\right\|\right).$$

6.3.1 Nadaraya-Watson model

总的来说做的就是model $p(\mathbf{t}|\mathbf{x})$.
$\mathbb{E}[t|\mathbf{x}] = \sum_n k(\mathbf{x}, \mathbf{x}_n) t_n$,其中的k满足都大于0且
求和是1.

6.4.Gaussian Processes

Gaussian Processes定义为a probability distribution over functions y(x),such that
the set of values of y(x) evaluated at an arbitrary set of points $x_1, \ldots, x_N$ jointly have a Gaussian distribution。

6.4.1 Gaussian processes for regression

GPR可以看成是Bayes linear regression从weight space转到function space的extension,利用了kernel trick可以计算infinity dimension的basis function的情况.

Gaussian process定义的$\mathbf{y}$:

$$p(\mathbf{y})=\mathcal{N}(\mathbf{y}\mid\mathbf{0},\mathbf{K})$$

其中的Covariance就是用Gram matrix给出的.利用第二章gaussian marginal的结果可以得到:

$$p(\mathbf{t})=\int p(\mathbf{t}\mid\mathbf{y})p(\mathbf{y})\mathrm{d}\mathbf{y}=\mathcal{N}(\mathbf{t}\mid\mathbf{0},\mathbf{C})$$

其中$C\left(\mathbf{x}_n,\mathbf{x}_m\right)=k\left(\mathbf{x}_n,\mathbf{x}_m\right)+\beta^{-1}\delta_{nm}$.

再次利用第二章中Gaussian joint的结果可以得到:

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

这就是我们需要的预测目标.重要的是Covariance可以用kernel trick直接给出结果.

6.4.3 Learning the hyperparameters

Gaussian process的predict部分依赖于covariance function.实际中常通过hyperparameter θ来控制.一种方法确定θ是MLE给出point estimate:

$$\ln p(\mathbf{t}\mid\boldsymbol{\theta})=-\frac{1}{2}\ln\left|\mathbf{C}_N\right|-\frac{1}{2}\mathbf{t}^{\mathrm{T}}\mathbf{C}_N^{-1}\mathbf{t}-\frac{N}{2}\ln(2\pi)$$

一般来说$p(\mathbf{t}|\theta)$是nonconvex function,可以用一些基于gradient的optimize methods:

$$\frac{\partial}{\partial\theta_i}\ln p(\mathbf{t}\mid\boldsymbol{\theta})=-\frac{1}{2}\operatorname{Tr}\left(\mathbf{C}_N^{-1}\frac{\partial\mathbf{C}_N}{\partial\theta_i}\right)+\frac{1}{2}\mathbf{t}^{\mathrm{T}}\mathbf{C}_N^{-1}\frac{\partial\mathbf{C}_N}{\partial\theta_i}\mathbf{C}_N^{-1}\mathbf{t}.$$

另外也可以引入一个θ的prior来maximum posterior.还可以完全使用Bayes treatment,marginal over θ.

现在讨论的GPR model中的covariance内的β是constant,但在一些问题中可能dependent on $\mathbf{x}$.可以使用second gaussian process去model $\beta\mathbf{x}$

6.4.4 Automatic relevance determination

Target关于input variable(多个input的情况)可能具有不同程度的依赖强度.可以使用kernel function:

$$k\left(\mathbf{x}_n, \mathbf{x}_m\right)=\theta_0 \exp \left\{-\frac{1}{2} \sum_{i=1}^D \eta_i\left(x_{n i}-x_{m i}\right)^2\right\}+\theta_2+\theta_3 \sum_{i=1}^D x_{n i} x_{m i}$$

其中的η可以用来控制不同input variable对predict的重要性.这里面有Gaussian kernel,linear kernel和constant,通过θ可以实现soft *kernel selection*的效果.

6.4.5 Gaussian processes for classification

GP model给出的output是整个real axis,但对于classify problem,我们玩玩更需要一个在$(0,1)$区间的probability.

对于binary classify,我们可以在GP的output后接一个logistic sigmoid function实现.记GP的output为a,则target variable服从bernoulli distribution:

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

此时的predict function就变为了:

$$p\left(t_{N+1}=1 \mid \mathbf{t}_N\right)=\int p\left(t_{N+1}=1 \mid a_{N+1}\right) p\left(a_{N+1} \mid \mathbf{t}_N\right) \mathrm{d} a_{N+1}$$

其中$p\left(t_{N+1}=1 \mid a_{N+1}\right)=\sigma\left(a_{N+1}\right)$.

`note:` 这个积分是intractable,所以需要用一些approximate approach.第四章用了sigmoid与Gaussian卷积近似,但由于GP的variable会随着input data增加而增加,并不能用中心极限定理近似成Gaussian.

6.4.6 Laplace approximation

对于上一节中的predict function,重点在确定积分中的$p\left(a_{N+1} \mid \mathbf{t}_N\right)$

$$\begin{aligned}
p\left(a_{N+1} \mid \mathbf{t}_N\right) & =\int p\left(a_{N+1}, \mathbf{a}_N \mid \mathbf{t}_N\right) \mathrm{d} \mathbf{a}_N \\
& =\frac{1}{p\left(\mathbf{t}_N\right)} \int p\left(a_{N+1}, \mathbf{a}_N\right) p\left(\mathbf{t}_N \mid a_{N+1}, \mathbf{a}_N\right) \mathrm{d} \mathbf{a}_N \\
& =\frac{1}{p\left(\mathbf{t}_N\right)} \int p\left(a_{N+1} \mid \mathbf{a}_N\right) p\left(\mathbf{a}_N\right) p\left(\mathbf{t}_N \mid \mathbf{a}_N\right) \mathrm{d} \mathbf{a}_N \\
& =\int p\left(a_{N+1} \mid \mathbf{a}_N\right) p\left(\mathbf{a}_N \mid \mathbf{t}_N\right) \mathrm{d} \mathbf{a}_N
\end{aligned}$$

而这里面的$p\left(a_{N+1} \mid \mathbf{a}_N\right)$在GPR中已经得到:

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

所以现在就需要用laplace approximate去近似$p(\mathbf{t}_n|\mathbf{a}_n)$.

在近似完了之后就是一个对应的linear-Gaussian model.完了还有一部是要确定covariance function里的θ.一个方法是用MLE,likelihood function为:

$$p\left(\mathbf{t}_N \mid \boldsymbol{\theta}\right)=\int p\left(\mathbf{t}_N \mid \mathbf{a}_N\right) p\left(\mathbf{a}_N \mid \boldsymbol{\theta}\right) \mathrm{d} \mathbf{a}_N$$

这个积分仍然intractable,需要再一次laplace approximate.