PRML（Pattern Recognition And Machine Learning）读书会

第三章 Linear Models for Regression

主讲人 planktonli

QQ 群 177217565

读书会微信公众平台请扫描下面的二维码

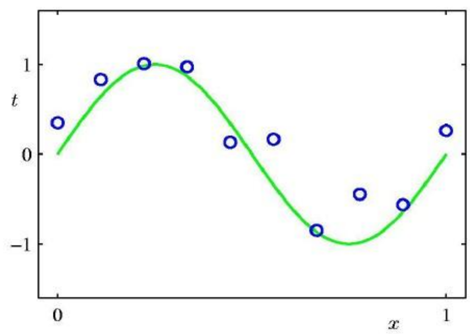
planktonli(1027753147) 18:58:12

大家好，我负责给大家讲讲 PRML 的第 3 讲 linear regression 的内容，请大家多多指教，群主让我们每个主讲人介绍下自己，赫赫,我也说两句，我是 applied mathematics + computer science 的，有问题大家可以直接指出，互相学习。大家有兴趣的话可以看看我的博客: http://t.qq.com/keepuphero/mine，当然我给大家推荐一个好朋友的，他对计算机发展还是很有心得的,他的网页 http://www.zhizhihu.com/ 对 machine learning 的东西有深刻的了解。

好,下面言归正传，开讲第 3 章，第 3 章的名字是 linear regression，首先需要考虑的是: 为什么在讲完 introduction、probability distributions 之后就直讲 linear regression? machine learning 的 essence 是什么?

机器学习的本质问题: 我个人理解,就是通过数据集学习未知的最佳逼近函数，学习的 收敛性\界 等等都是描述这个学习到的 function 到底它的性能如何。但是,从数学角度出发,函数是多样的，线性\非线性\跳跃\连续\非光滑，你可以组合出无数的函数,那么这些函数就组成了函数空间，在这些函数中寻找到一个满足你要求的最佳逼近函数,无疑大海捞针。我们再来回顾下第一章的 曲线拟和问题：

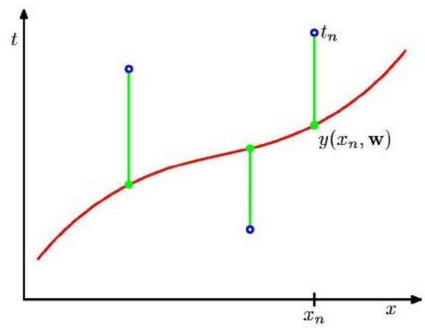
Polynomial Curve Fitting



$$y(x,\mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

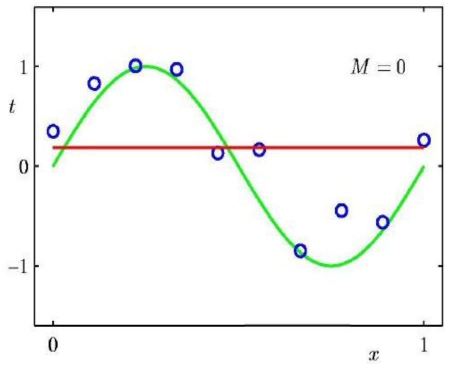
需要逼近的函数是: $\sin(2\pi x)$ ，M 阶的曲线函数可以逼近么？这是我们值得思考的问题。

Sum-of-Squares Error Function

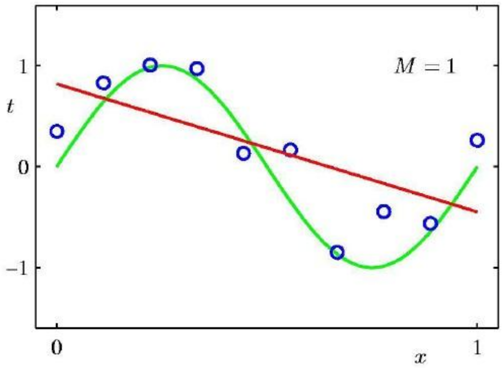


$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N} \{y(x_n,\mathbf{w}) - t_n\}^2$$

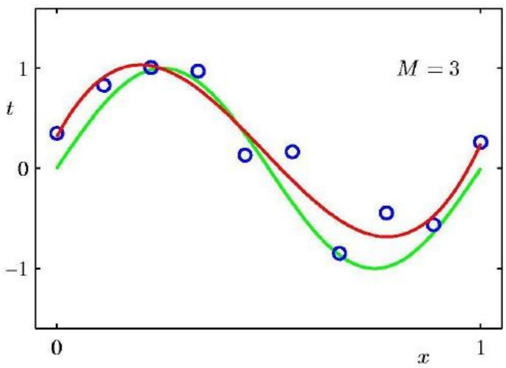
0th Order Polynomial



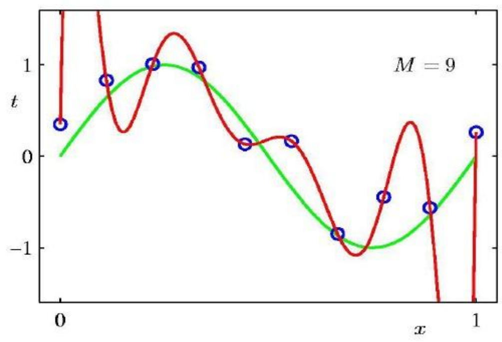
1st Order Polynomial



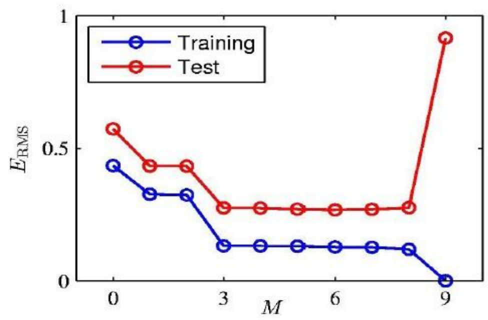
3rd Order Polynomial



9th Order Polynomial



Over-fitting



Root-Mean-Square (RMS) Error: $E_{\mathrm{RMS}} = \sqrt{2E(\mathbf{w}^\star)/N}$

要曲线拟和, 那么拟和的标准是什么?这里用了 2 范数定义,也就是误差的欧式距离，当然,你可以用 L1,L 无穷，等等了 ，只是 objective 不同罢了。现在的疑问是: 为什么要用 Polynomial Fitting?有数学依据么，这里牵扯到 范函的问题，就是函数所张成的空间，举一个简单的例子，大家还都记得 talyor 展式吧：

$$f(x) = f(x_0) + \frac{f'(x)}{1!}(x - x_0) + + \frac{f''(x)}{2!}(x - x_0)^2 + \ldots.$$

这表明 任意一个函数可以表示成 x 的次方之和，也就是 任意一个函数 可以放到 $(1, x, x^2, x^3, \ldots..)$ 所张成的函数空间，如果是有限个基的话就称为欧式空间，无穷的话 就是 Hilbert 空间，其实 傅里叶变换 也是这样的一个例子，既然已经明白了 任意函数可以用 Polynomial Fitting，那么下面就是什么样的 Polynomial 是最好的。

Wilbur_中博(1954123) 19:28:26

泰勒展开是局部的、x0 周围的，而函数拟合是全局的，似乎不太一样吧？

planktonli(1027753147) 19:29:21

恩,泰勒展开是局部的，他是在 x0 点周围的一个 表达，函数拟合是全局的,我这里只是用一个简单的例子说明 函数表达的问题。

Wilbur_中博(1954123) 19:30:41

planktonli(1027753147) 19:31:03

其实,要真正解释这个问题是需要范函的东西的。

Wilbur_中博(1954123) 19:31:45
抱歉，打断了一下，因为我觉得这个问题留到讨论就不太好了，呵呵。了解了，请继续吧。
planktonli(1027753147) 19:31:51
由于大多数群友未学过这个课程,我只是想说下这个思想 ,呵呵,没事 ,讨论才能深刻理解问题 ,其实,wavelet 这些,包括 kernel construcion 这些东西都牵扯到 范函。
Bishop 用上面这个例子说明 ：
1) 可以用 Polynomial Fitting 拟和 sin 类的函数　2) 存在过拟和问题
而且这里的 Polynomial Fitting 是一个线性 model， 这里 Model 是 w 的函数,w 是线性的：

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

$\sin(2\pi x)$ 是线性的么，肯定不是，那么 让我们再来分析下 研究的问题

$\sin(2\pi x)$ 中的 x 是 1 维的

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

上面的 X 变成了 $[1, x, x^2, x^3, ...]^T$

$y(x, w) = W^T X$，非常有意思的是: 维数升高了，同时这个 model 具有了表达非线性东西的能力。这里的思想,可以说贯穿在 NN,SVM 这些东西里，也就是说,线性的 model 如果应用得当的话,可以表达非线性的东西。与其在所有函数空间盲目的寻找,还不如从一个可行的简单 model 开始，这就是为什么 Bishop 在讲完基础后直接切入 Linear regression 的原因 ,当然这个线性 model 怎么构造,是单层的 linear model,还是多层的 linear model 一直争论不休，BP 否定了 perceptron 的 model，SVM 否定了 BP model 现在 deep learning 又质疑 SVM 的 shallow model，或许这就是 machine learning 还能前进的动力。

让咱们再回来看看 linear regression 的模型，这里从标准形式到扩展形式，也就是引入基函数后,Linear regression 的模型可以表达非线性的东西了，因为基函数可能是非线性的：

Linear Regression

The basic form of linear regression

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

where $\mathbf{x} = (x_1, \cdots, x_D)^T$ is the input variable, and $\mathbf{w} = (w_1, \cdots, w_D)^T$ is the parameters.

More general form of linear regression

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

where $\phi_j(\mathbf{x})$'s are known as *basis functions*.

Parameter w_0 is called a *bias* parameter. By adding $\phi_0(\mathbf{x}) = 1$, we have

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x})$$

基函数的形式，这些基函数都是非线性的：

Basis functions

Polynomial functions

$$\phi_j(x) = x^j$$

Gaussian functions

$$\phi_j(\mathbf{x}) = \exp\left\{-\frac{(\mathbf{x}-\boldsymbol{\mu}_j)^T(\mathbf{x}-\boldsymbol{\mu}_j)}{2s^2}\right\}$$

Sigmoid basis functions

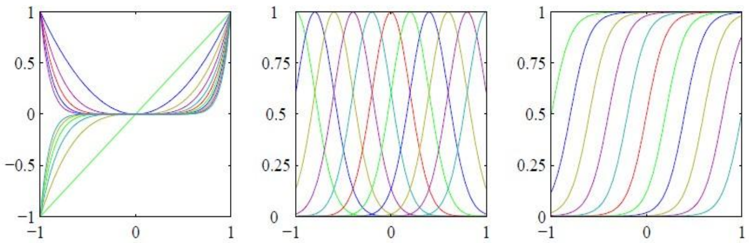
$$\phi_j(\mathbf{x}) = \sigma\left(\frac{\mathbf{b}^T\mathbf{x}-\mu_j}{s}\right)$$

where $\sigma(a)$ is the logistic sigmoid function

$$\sigma(a) = \frac{1}{1+\exp\{-a\}}$$

Other basis functions: wavelets

Basis functions



Polynomial Gaussian Sigmoid

Probabilistic Formulation

Assume the target variable t is given by a deterministic function $y(\mathbf{x},\mathbf{w})$ with additive Gaussian noise

$$t = y(\mathbf{x},\mathbf{w}) + \epsilon$$

where ϵ is a zero mean Gaussian random variable with precision (inverse variance) β. Therefore

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

Conditional mean

$$\mathbb{E}[t|\mathbf{x}] = \int tp(t|\mathbf{x})\mathrm{d}t = y(\mathbf{x},\mathbf{w}) + \mathbb{E}[\epsilon] = y(\mathbf{x},\mathbf{w})$$

在 Gaussian 零均值情况下,Linear model 从频率主义出发的 MLE 就是 Least square：

Maximum Likelihood Estimator

Consider a data set of inputs $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ with corresponding target values $\mathbf{t} = (t_1, \cdots, t_N)^T$. The likelihood function (a function of $\mathbf{w}$ and β) is

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \prod \mathcal{N}(t_n|\mathbf{w}^T\phi(\mathbf{x}_n), \beta^{-1})$$

Log likelihood

$$\begin{aligned}\log p(\mathbf{t}|\mathbf{w}, \beta) &= \sum_{n=1}^{N} \log \mathcal{N}(t_n|\mathbf{w}^T\phi(\mathbf{x}_n), \beta^{-1}) \\ &= \frac{N}{2}\log\beta - \frac{N}{2}\log 2\pi - \beta E_D(\mathbf{w})\end{aligned}$$

where the sum-of-square error function is

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

最小 2 乘的解就是广义逆矩阵乘输出值：

Maximum Likelihood Estimator: $\mathbf{w}$

With β fixed, we obtain the gradient of the log likelihood

$$\nabla \log p(\mathbf{t}|\mathbf{w}, \beta) = \sum_{n=1}^{N}\left(t_n - \mathbf{w}^T\phi(\mathbf{x}_n)\right)\phi(\mathbf{x}_n)^T$$

Setting the gradient to be zero gives

$$\begin{aligned}\mathbf{w}^T\left(\sum_{n=1}^{N}\phi(\mathbf{x}_n)\phi(\mathbf{x}_n)^T\right) &= \sum_{n=1}^{N} t_n\phi(\mathbf{x}_n)^T \\ \mathbf{w}^T\mathbf{\Phi}^T\mathbf{\Phi} &= \mathbf{t}^T\mathbf{\Phi} \\ \mathbf{\Phi}^T\mathbf{\Phi}\mathbf{w} &= \mathbf{\Phi}^T\mathbf{t} \\ \mathbf{w}_{\mathrm{ML}} &= (\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{t}\end{aligned}$$

where the $N \times M$ *design matrix* $\mathbf{\Phi}$ is $\mathbf{\Phi} = [\phi(\mathbf{x}_1) \ \cdots \ \phi(\mathbf{x}_N)]^T$. The quantity $\mathbf{\Phi}^\dagger \equiv (\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T$ is called *pseudo-inverse* of the matrix $\mathbf{\Phi}$.

Gaussian 的 precision 也可以计算出来：

Maximum Likelihood Estimator: β

Recall log likelihood

$$\log p(\mathbf{t}|\mathbf{w}, \beta) = \frac{N}{2}\log\beta - \frac{N}{2}\log 2\pi - \beta E_D(\mathbf{w})$$

Setting the partial derivative w.r.t. β to zero

$$\frac{\partial}{\partial\beta}\log p(\mathbf{t}|\mathbf{w}, \beta) = \frac{N}{2}\frac{1}{\beta} - E_D(\mathbf{w})$$

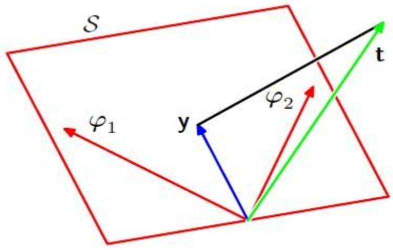
We obtain

$$\beta_{\mathrm{ML}} = \frac{N}{(\mathbf{t} - \mathbf{\Phi}\mathbf{w}_{\mathrm{ML}})^T(\mathbf{t} - \mathbf{\Phi}\mathbf{w}_{\mathrm{ML}})}$$

最小 2 乘的解可以看成到基张成空间的投影：

Geometrical Interpretation of Least Squares

The least-squares regression function is obtained by finding the orthogonal projection of the data vector $\mathbf{t}$ onto the subspace spanned by the basis functions $\phi_j(\mathbf{x})$ in which each basis function is viewed as a vector φ_j of length N with elements $\phi_j(\mathbf{x}_n)$.



频率主义会导致 过拟和，加入正则,得到的最小 2 乘解：

Regularized Least Squares

The over-fitting issue can be addressed through adding a *regularization* term

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

where λ is the regularization coefficient. For example

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

and the error function becomes

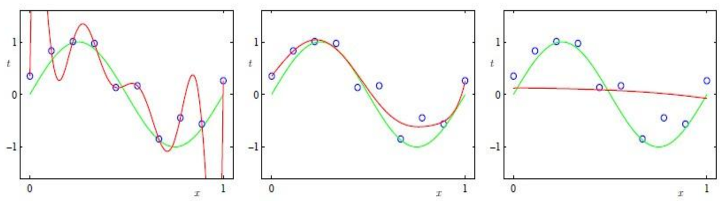
$$\frac{1}{2}(\mathbf{t} - \mathbf{\Phi}\mathbf{w})^T(\mathbf{t} - \mathbf{\Phi}\mathbf{w}) + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}$$

which leads to solution

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

正则参数对 model 结果的影响：

Effects of λ

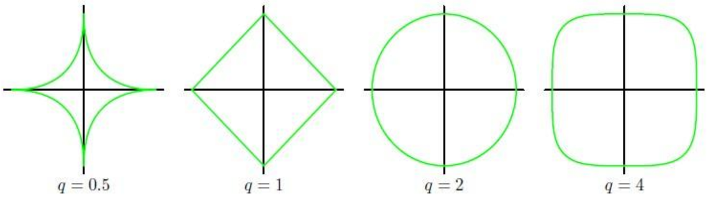


$\log\lambda = -\infty$ $\log\lambda = -18$ $\log\lambda = 0$

消除过拟和，正则的几何解释：

Other Regularizers

$$E_W(\mathbf{w}) = \frac{1}{2}\sum_{j=1}^{M}|w_j|^q$$



When $q = 1$, the regularizer is called *lasso* in statistics as sparse models are preferred.

正则方法不同,就会出现很多 model,例如 lasso, ridge regression。LASSO 的解是稀疏的，例如:sparse coding,Compressed sensing 是从 L0--> L1sparse 的问题，现在也很热的。

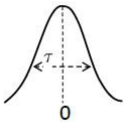
Regularized Least Squares and MAP

What if $(\mathbf{A}^T\mathbf{A})$ is not invertible ?

$$\widehat{\beta}_{\text{MAP}} = \arg\max_{\beta} \underbrace{\log p(\{(X_i, Y_i)\}_{i=1}^{n}|\beta,\sigma^2)}_{\text{log likelihood}} + \underbrace{\log p(\beta)}_{\text{log prior}}$$

I) Gaussian Prior

$$\beta \sim \mathcal{N}(0, \tau^2\mathbf{I}) \qquad p(\beta) \propto e^{-\beta^T\beta/2\tau^2}$$



$$\widehat{\beta}_{\text{MAP}} = \arg\min_{\beta} \sum_{i=1}^{n}(Y_i - X_i\beta)^2 + \underset{\substack{\downarrow \\ \text{constant}(\sigma^2,\tau^2)}}{\lambda}\|\beta\|_2^2$$

Ridge Regression

Closed form: HW

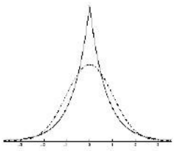
Regularized Least Squares and MAP

What if $(\mathbf{A}^T\mathbf{A})$ is not invertible ?

$$\widehat{\beta}_{\text{MAP}} = \arg\max_{\beta} \underbrace{\log p(\{(X_i, Y_i)\}_{i=1}^{n}|\beta,\sigma^2)}_{\text{log likelihood}} + \underbrace{\log p(\beta)}_{\text{log prior}}$$

II) Laplace Prior

$$\beta_i \overset{iid}{\sim} \text{Laplace}(0, t) \qquad p(\beta_i) \propto e^{-|\beta_i|/t}$$



$$\widehat{\beta}_{\text{MAP}} = \arg\min_{\beta} \sum_{i=1}^{n}(Y_i - X_i\beta)^2 + \underset{\substack{\downarrow \\ \text{constant}(\sigma^2,t)}}{\lambda}\|\beta\|_1$$

Lasso

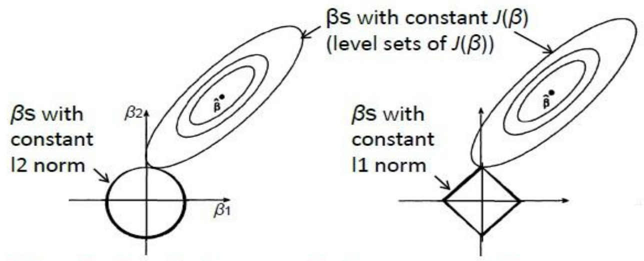
Closed form: HW

Ridge Regression vs Lasso

$$\min_{\beta}(\mathbf{A}\beta - \mathbf{Y})^T(\mathbf{A}\beta - \mathbf{Y}) + \lambda \text{pen}(\beta) = \min_{\beta} J(\beta) + \lambda \text{pen}(\beta)$$

Ridge Regression: $\text{pen}(\beta) = \|\beta\|_2^2$

Lasso: $\text{pen}(\beta) = \|\beta\|_1$ HOT !



Lasso (l1 penalty) results in sparse solutions – vector with more zero coordinates
Good for high-dimensional problems – don't have to store all coordinates!

下面看 Bias-Variance Decoposition，正则就是在 训练数据的模型上加一个惩罚项，shrink 模型的参数,让它不要学习的太过，这里 $E_D(\mathbf{w})$ 是对训练数据学习到的模型,$E_W(\mathbf{w})$是学习到的参数的惩罚模型

Expected Squared Loss

Recall that t is the value for each input $\mathbf{x}$ and $y(\mathbf{x})$ is our estimate. We incur a loss $L(t, y(\mathbf{x}))$, and the expected loss is

$$\mathbb{E}[L] = \int\int L(t, y(\mathbf{x}))p(\mathbf{x}, t)\mathrm{d}\mathbf{x}\mathrm{d}t.$$

A common choice of loss function is the squared loss $L(t, y(\mathbf{x})) = \{y(\mathbf{x}) - t\}^2$, and the expected squared loss is

$$\mathbb{E}[L] = \int\int \{y(\mathbf{x}) - t\}^2 p(\mathbf{x}, t)\mathrm{d}\mathbf{x}\mathrm{d}t$$

Our goal is to choose $y(\mathbf{x})$ to minimize $\mathbb{E}[L]$. From Euler-Lagrange

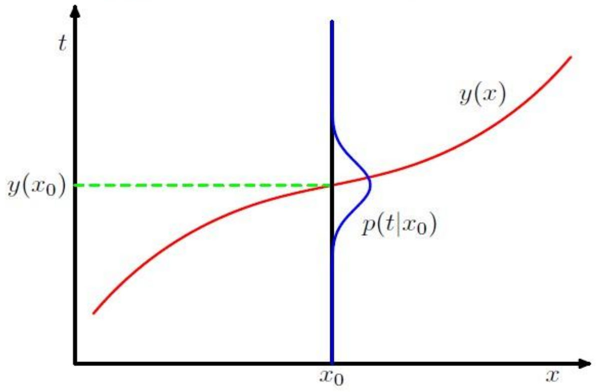
$$\frac{\delta \mathbb{E}[L]}{\delta y(\mathbf{x})} = 2\int \{y(\mathbf{x}) - t\}p(\mathbf{x}, t)\mathrm{d}t = 0$$

So

$$y(\mathbf{x}) = \frac{\int tp(\mathbf{x}, t)\mathrm{d}t}{p(\mathbf{x})} = \int tp(t|\mathbf{x})\mathrm{d}t = \mathbb{E}_t[t|\mathbf{x}]$$

Regression Function

The function $y(\mathbf{x}) = \mathbb{E}[t|\mathbf{x}]$ is known as the *regression function*.



Decomposition of the Loss Function

Define

$$h(\mathbf{x}) = \mathbb{E}[t|\mathbf{x}] = \int tp(t|\mathbf{x})d\mathbf{t}$$

Since the optimal solution is the conditional expectation, we can expand the square term

$$\begin{aligned}\{y(\mathbf{x}) - t\}^2 &= \{y(\mathbf{x}) - \mathbb{E}[t|\mathbf{x}] + \mathbb{E}[t|\mathbf{x}] - t\}^2 \\ &= \{y(\mathbf{x}) - h(\mathbf{x})\}^2 + 2\{y(\mathbf{x}) - h(\mathbf{x})\}\{h(\mathbf{x}) - t\} + \{h(\mathbf{x}) - t\}^2\end{aligned}$$

Therefore

$$\begin{aligned}\mathbb{E}[L] &= \int\int \{y(\mathbf{x}) - t\}^2 p(\mathbf{x}, t)\mathrm{d}\mathbf{x}\mathrm{d}t \\ &= \int \{y(\mathbf{x}) - h(\mathbf{x})\}^2 p(\mathbf{x})\mathrm{d}\mathbf{x} + \int \{h(\mathbf{x}) - t\}^2 p(\mathbf{x}, t)\mathrm{d}\mathbf{x}\mathrm{d}t\end{aligned}$$

When $y(\mathbf{x}) = h(\mathbf{x})$, the first term will vanish. The second term is the variance of the distribution of t averaged over $\mathbf{x}$, representing the intrinsic variability of the target data and can be regarded as noise and the irreducible minimum value of the loss function.

Decomposition of the Loss Function

In practice, $h(\mathbf{x})$ is unknown. We have a data set $\mathcal{D}$ containing only a finite number N of data points. The loss function becomes

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

We can rewrite it as

$$\begin{aligned}\{y(\mathbf{x}; \mathcal{D}) - h(\mathbf{x})\}^2 &= \{y(\mathbf{x}; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] + \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] - h(\mathbf{x})\}^2 \\ &= \{y(\mathbf{x}; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})]\}^2 + \{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] - h(\mathbf{x})\}^2 \\ &\quad + 2\{y(\mathbf{x}; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})]\}\{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] - h(\mathbf{x})\}\end{aligned}$$

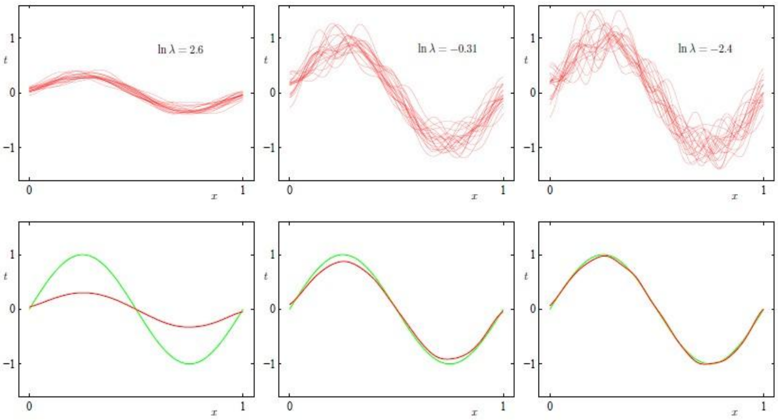
Take the expectation w.r.t. $\mathcal{D}$

$$\begin{aligned}&\mathbb{E}_{\mathcal{D}}[\{y(\mathbf{x}; \mathcal{D}) - h(\mathbf{x})\}^2] \\ &= \underbrace{\{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] - h(\mathbf{x})\}^2}_{\text{bias}^2} + \underbrace{\mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x}; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})]\}^2\right]}_{\text{variance}}\end{aligned}$$

上面这么多 PPT 无非就是说，学习到的模型和真实的模型的期望由 2 部分组成：

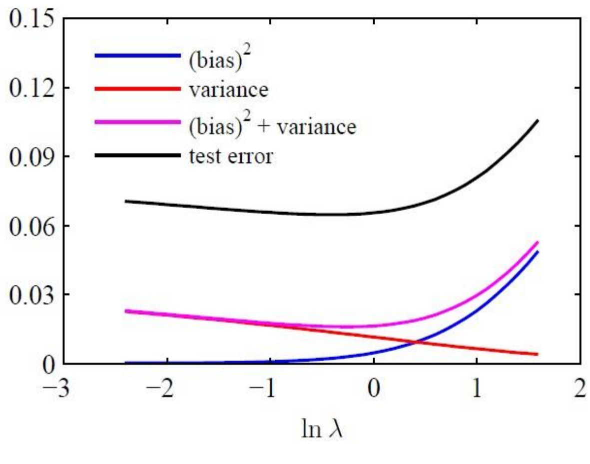
1--> Bias 2--> Variance。Bias 表示的是学习到的模型和真实模型的偏离程度,Variance 表示的是学习到的模型和它自己的期望的偏离程度。从这里可以看到正则项在控制 Bias 和 Variance：

Bias-Variance as a Function of Complexity



Wilbur_中博(1954123) 20:33:07
这个是关键，呵呵
planktonli(1027753147) 20:33:25
Variance 小的情况下,Bias 就大，Variance 大的情况下,Bias 就小，我们就要 tradeoff 它们。
从这张图可以看到 Bias 和 Variance 的关系：

Bias-Variance as a Function of Complexity



这个 Bias-Variance Decoposition 其实没有太大的实用价值，它只能起一个指导作用。
下面看看 Bayesian Linear Regression：

Conjugate Prior

Recall the likelihood function

$$\begin{aligned} p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) &= \prod_{n=1}^{N} \mathcal{N}(t_n|\mathbf{w}^T \phi(\mathbf{x}_n), \beta^{-1}) \\ &= \mathcal{N}(\mathbf{t}|\mathbf{\Phi w}, \beta^{-1}\mathbf{I}) \end{aligned}$$

is the exponential of a quadratic function of $\mathbf{w}$. So the conjugate prior is given by a Gaussian distribution of the form

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0)$$

Bayesian Theorem for Gaussian

Theorem (Bayesian Theorem for Linear Gaussian)

Given prior and likelihood

$$\begin{aligned} p(\mathbf{x}) &= \mathcal{N}(\mathbf{x}|\mu, \mathbf{\Lambda}^{-1}) \\ p(\mathbf{y}|\mathbf{x}) &= \mathcal{N}(\mathbf{y}|\mathbf{Ax} + \mathbf{b}, \mathbf{L}^{-1}) \end{aligned}$$

the marginal $p(\mathbf{y})$ *and posterior* $p(\mathbf{x}|\mathbf{y})$ *are*

$$\begin{aligned} p(\mathbf{y}) &= \mathcal{N}(\mathbf{y}|\mathbf{A}\mu + \mathbf{b}, \mathbf{A}\mathbf{\Lambda}^{-1}\mathbf{A}^T + \mathbf{L}^{-1}) \\ p(\mathbf{x}|\mathbf{y}) &= \mathcal{N}(\mathbf{x}|\mathbf{\Sigma}[\mathbf{A}^T\mathbf{L}(\mathbf{y} - \mathbf{b}) + \mathbf{\Lambda}\mu], \mathbf{\Sigma}) \end{aligned}$$

where $$\mathbf{\Sigma} = (\mathbf{\Lambda} + \mathbf{A}^T\mathbf{L}\mathbf{A})^{-1}.$$

Posterior

Apply the theorem and we obtain the posterior

$$p(\mathbf{w}|\mathbf{t}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N)$$

where

$$\begin{aligned} \mathbf{m}_N &= \mathbf{S}_N(\mathbf{S}_0^{-1}\mathbf{m}_0 + \beta\mathbf{\Phi}^T\mathbf{t}) \\ \mathbf{S}_N^{-1} &= \mathbf{S}_0^{-1} + \beta\mathbf{\Phi}^T\mathbf{\Phi} \end{aligned}$$

The prior can be a zero-mean isotropic Gaussian governed by a single precision parameter α

$$p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I})$$

and the corresponding posterior distribution is

$$\begin{aligned} \mathbf{m}_N &= \beta\mathbf{S}_N\mathbf{\Phi}^T\mathbf{t} \\ \mathbf{S}_N^{-1} &= \alpha\mathbf{I} + \beta\mathbf{\Phi}^T\mathbf{\Phi} \end{aligned}$$

Bayesian MAP vs. Regularization

The log posterior takes the following form

$$\log p(\mathbf{w}|\mathbf{t}) = -\frac{\beta}{2}(\mathbf{t}-\Phi\mathbf{w})^T(\mathbf{t}-\Phi\mathbf{w}) - \frac{\alpha}{2}\mathbf{w}^T\mathbf{w} + \text{const}$$

The regularized least square takes the form

$$\frac{1}{2}(\mathbf{t}-\Phi\mathbf{w})^T(\mathbf{t}-\Phi\mathbf{w}) + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}$$

Therefore, $\lambda = \frac{\alpha}{\beta}$.

从 Bayesian 出发,关注的不是参数的获取,而更多的是 新预测的值，通过后验均值可以得到 linear model 和核函数的联系，当然也可以建立 gaussian process 这些东西。

Wilbur_中博(1954123) 20:51:25

这里可以讲细一点么，如何建立联系？

planktonli(1027753147) 20:54:44

Bayesian Linear Regression (2)

A common choice for the prior is

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I})$$

for which

$$\begin{aligned} \mathbf{m}_N &= \beta \mathbf{S}_N \mathbf{\Phi}^{\mathrm{T}}\mathbf{t} \\ \mathbf{S}_N^{-1} &= \alpha\mathbf{I} + \beta\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}. \end{aligned}$$

Equivalent Kernel (1)

The predictive mean can be written

$$\begin{aligned} y(\mathbf{x}, \mathbf{m}_N) &= \mathbf{m}_N^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}) = \beta\boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}}\mathbf{S}_N\mathbf{\Phi}^{\mathrm{T}}\mathbf{t} \\ &= \sum_{n=1}^{N} \underbrace{\beta\boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}}\mathbf{S}_N\boldsymbol{\phi}(\mathbf{x}_n)}t_n \\ &= \sum_{n=1}^{N} \underbrace{k(\mathbf{x}, \mathbf{x}_n)}t_n. \end{aligned}$$

Equivalent kernel or *smoother matrix.*

This is a weighted sum of the training data target values, t_n.

这里就可以看到了啊，看到了么，Wilbur?

Wilbur_中博(1954123) 20:57:24

在看

planktonli(1027753147) 20:58:08

如果共扼先验是 0 均值情况下,linear model 就可以变成 kernel 了：

Equivalent Kernel (4)

The kernel as a covariance function: consider

$$\begin{aligned}\mathrm{cov}[y(\mathbf{x}), y(\mathbf{x}')] &= \mathrm{cov}[\boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}}\mathbf{w}, \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}')] \\ &= \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}}\mathbf{S}_N\boldsymbol{\phi}(\mathbf{x}') = \beta^{-1}k(\mathbf{x}, \mathbf{x}').\end{aligned}$$

We can avoid the use of basis functions and define the kernel function directly, leading to *Gaussian Processes* (Chapter 6).

Equivalent Kernel (5)

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

for all values of $\mathbf{x}$; however, the equivalent kernel may be negative for some values of $\mathbf{x}$.

Like all kernel functions, the equivalent kernel can be expressed as an inner product:

$$k(\mathbf{x}, \mathbf{z}) = \boldsymbol{\psi}(\mathbf{x})^{\mathrm{T}}\boldsymbol{\psi}(\mathbf{z})$$

where $\boldsymbol{\psi}(\mathbf{x}) = \beta^{1/2}\mathbf{S}_N^{1/2}\boldsymbol{\phi}(\mathbf{x})$.

最后讲了 bayesain model 比较：

Bayesian Model Comparison (1)

How do we choose the 'right' model?

Assume we want to compare models $\mathcal{M}_i$, $i=1, \ldots, L$, using data $\mathcal{D}$; this requires computing

$$p(\mathcal{M}_i|\mathcal{D}) \propto p(\mathcal{M}_i)p(\mathcal{D}|\mathcal{M}_i).$$

Posterior Prior *Model evidence or marginal likelihood*

Bayes Factor: ratio of evidence for two models

$$\frac{p(\mathcal{D}|\mathcal{M}_i)}{p(\mathcal{D}|\mathcal{M}_j)}$$

Bayesian Model Comparison (2)

Having computed $p(\mathcal{M}_i|\mathcal{D})$, we can compute the predictive (mixture) distribution

$$p(t|\mathbf{x},\mathcal{D}) = \sum_{i=1}^{L} p(t|\mathbf{x},\mathcal{M}_i,\mathcal{D})p(\mathcal{M}_i|\mathcal{D}).$$

A simpler approximation, known as *model selection*, is to use the model with the highest evidence.

Bayesian Model Comparison (3)

For a model with parameters $\mathbf{w}$, we get the model evidence by marginalizing over $\mathbf{w}$

$$p(\mathcal{D}|\mathcal{M}_i) = \int p(\mathcal{D}|\mathbf{w},\mathcal{M}_i)p(\mathbf{w}|\mathcal{M}_i)\,\mathrm{d}\mathbf{w}.$$

Note that

$$p(\mathbf{w}|\mathcal{D},\mathcal{M}_i) = \frac{p(\mathcal{D}|\mathbf{w},\mathcal{M}_i)p(\mathbf{w}|\mathcal{M}_i)}{p(\mathcal{D}|\mathcal{M}_i)}$$

选择最大信任的 model 来作为模型选择，而非用交叉验证，信任近似：

The Evidence Approximation (1)

The fully Bayesian predictive distribution is given by

$$p(t|\mathbf{t}) = \iiint p(t|\mathbf{w},\beta)p(\mathbf{w}|\mathbf{t},\alpha,\beta)p(\alpha,\beta|\mathbf{t})\,\mathrm{d}\mathbf{w}\,\mathrm{d}\alpha\,\mathrm{d}\beta$$

but this integral is intractable. Approximate with

$$p(t|\mathbf{t}) \simeq p\left(t|\mathbf{t},\widehat{\alpha},\widehat{\beta}\right) = \int p\left(t|\mathbf{w},\widehat{\beta}\right)p\left(\mathbf{w}|\mathbf{t},\widehat{\alpha},\widehat{\beta}\right)\,\mathrm{d}\mathbf{w}$$

where $\left(\widehat{\alpha},\widehat{\beta}\right)$ is the mode of $p(\alpha,\beta|\mathbf{t})$, which is assumed to be sharply peaked; a.k.a. *empirical Bayes, type II* or *generalized maximum likelihood,* or *evidence approximation*.

The Evidence Approximation (2)

From Bayes' theorem we have

$$p(\alpha, \beta|\mathbf{t}) \propto p(\mathbf{t}|\alpha, \beta)p(\alpha, \beta)$$

and if we assume $p(\alpha,\beta)$ to be flat we see that

$$\begin{aligned} p(\alpha, \beta|\mathbf{t}) &\propto p(\mathbf{t}|\alpha, \beta) \\ &= \int p(\mathbf{t}|\mathbf{w}, \beta)p(\mathbf{w}|\alpha)\,\mathrm{d}\mathbf{w}. \end{aligned}$$

General results for Gaussian integrals give

$$\ln p(\mathbf{t}|\alpha, \beta) = \frac{M}{2}\ln\alpha + \frac{N}{2}\ln\beta - E(\mathbf{m}_N) + \frac{1}{2}\ln|\mathbf{S}_N| - \frac{N}{2}\ln(2\pi).$$

Limitations of Fixed Basis Functions

- M basis function along each dimension of a D-dimensional input space requires M^D basis functions: the curse of dimensionality.
- In later chapters, we shall see how we can get away with fewer basis functions, by choosing these using the training data.

固定基存在缺陷为 NN,SVM 做铺垫，NN,SVM 都是变化基，BP 是梯度下降 error,固定基，RBF 是聚类寻找基，SVM 是 2 次凸优化寻找基。好了,就讲到这里吧，肯定还有讲的不对,或者不足的地方，请大家一起讨论和补充，谢谢。

============================讨论==================================

Wilbur_中博(1954123) 21:08:29

RBF 不是固定径向基找系数的么，SVM 也是固定基的吧，这里寻找基是什么意思？

planktonli(1027753147) 21:09:01

SVM 是寻找那些 系数不为 0 的作为基，RBF,我说的是 RBF 神经网络，不是 RBF 基函数，呵呵

Wilbur_中博(1954123) 21:11:07

嗯，但咱们现在这一章，比如多项式基，也可以说是寻找系数不为 0 的 x^k 吧，SVM 也仍然是固定了某一种核，比如多项式核或者高斯核。嗯，我知道是说 RBF 网络。

planktonli(1027753147) 21:11:40

恩,可以这么说

Wilbur_中博(1954123) 21:12:35

还有就是，固定一组基的话，也有很多选择，有多项式、也有高斯、logisitic 等等，那我们应该怎么选择用什么基去做回归呢？这一章讲得大多都是有了基以后怎么选择 w ,但怎么选择基这一点有没有什么说法。

planktonli(1027753147) 21:13:37

我说的固定指的是,SVM 不知道基是谁，而是通过优化获取的。

Wilbur_中博(1954123) 21:13:41

或者小波傅里叶什么的。。好多基

planktonli(1027753147) 21:14:03

3.6. Limitations of Fixed Basis Functions 这里提出了固定基的问题，基的选择要看样本的几何形状，一般都是 选择 gaussian，当然也可以一个个测试着弄。
Wilbur_中博(1954123) 21:15:55
SVM 里有个叫 multiple kernel learning 的，感觉像是更广泛的变化基的解决方案。嗯，就是说大多是经验性的是吧，选基这个还是蛮有趣的，我觉得。
planktonli(1027753147) 21:16:45
恩,MK 是多个 kernel 的组合，尝试用多个几何形状的 kernl 去寻找一个更 power 的。
Wilbur_中博(1954123) 21:17:05
嗯，呵呵
planktonli(1027753147) 21:17:16
恩,kernel construction 是 ML 的主要研究内容之一
Wilbur_中博(1954123) 21:18:14
好的，我没什么问题了，谢谢，以后多交流。看其他朋友还有什么问题。
planktonli(1027753147) 21:50:29
本次的讲义有些内容是群共享里的 Linear1.pdf
下次的 linear classification 主要讲的内容在群共享中为 Linear2.pdf