Ch. 3: Linear Models of Regression

王星

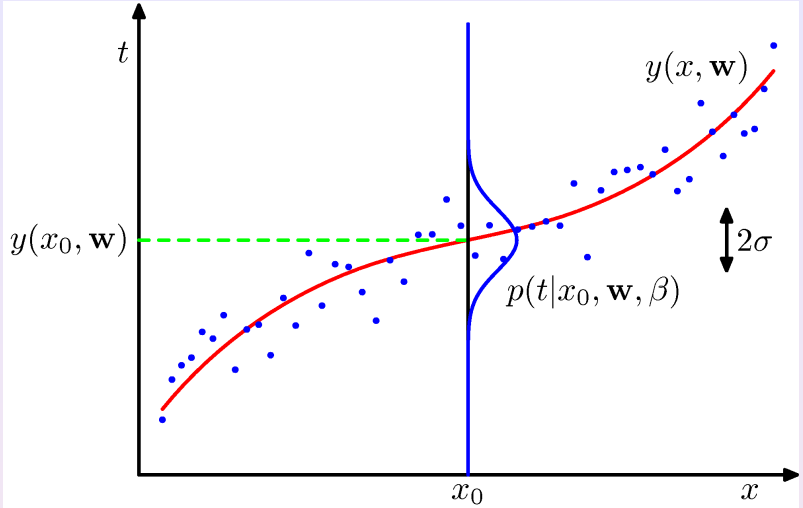
中国人民大学统计学院
E-mail:wangxingwisdom@126.com

November 4, 2020

Chapter content

- 第一章考虑的例子：多项式曲线拟合；
- 线性基函数的组合形式 - 回归 - of a fixed set of nonlinear functions - basis functions
- 有指导的学习: N observations $\{x_n\}$ with corresponding target values $\{t_n\}$ are provided. **The goal is to predict t of a new value x.**
- 任务：构造函数$y(x)$ 用于预测t.
- 概率的视角: 对预测分布建模$p(t|x)$.

Figure 1.16, page 29



- 假设给定x,对应的t值服从高斯分布，均值为$y(x,\omega)$,

$$p(t|x,\omega,\beta) \sim \mathcal{N}(t|y(x,\omega),\beta^{-1}) \quad \beta^{-1} = \sigma^2;$$

- 训练数据$\{x_n,t_n\}_{n=1,\ldots,N}$,对应的似然函数

$$p(t|x,\omega,\beta) = \prod_{n=1}^{N} \sqrt{\frac{\beta}{2\pi}} \exp\{-\frac{\beta(t_n - y(x_n,\omega))^2}{2}\};$$

- 给定ω_{ML}，极大似然估计$\frac{1}{\beta_{ML}} = \frac{1}{N}\sum_{n=1}^{N}\{y(x_n,\omega_{ML}) - t_n)^2\}$

- 对应的预测分布：

$$p(t|x_n, \omega_{ML}, \beta_{ML}) = N(t|y(x_n, \omega_{ML}, \beta_{ML}^{-1});$$

- $$p(\omega|\alpha) = N(\omega|0, \alpha^{-1}I) = (\frac{\alpha}{2\pi})^{\frac{M+1}{2}} \exp(-\frac{\alpha}{2}\omega^T\omega);$$

  M表示多项式的阶，也可以理解为拟合函数中的多项式的项数，

- $$p(\omega|t, x, \alpha, \beta) \propto p(t|x, \omega, \beta)p(\omega|\alpha);$$

- 最大后验分布等价于求

$$\omega_{MAP} = \text{argmax}\frac{\beta}{2}\sum_{n=1}^{N}\{y(x_n, \omega) - t_n\}^2 + \frac{\alpha}{2}\omega^T\omega.$$

- 正则化参数$\lambda = \alpha/\beta$,对应于岭估计，后验分布的众数就是岭估计。

The chapter section by section

Linear Basis Function Models

$$y(x,\omega) = \sum_{j=0}^{M-1} \omega_j \phi_j(x) = \omega^\top \phi(x)$$

where:

- $\omega = (\omega_0, \cdots, \omega_{M-1})^\top$ and $\phi = (\phi_0, \cdots, \phi_{M-1})^\top$ with $\phi_0(x) == 1$ and $\omega_0 = biasparameter$.
- In general $x \in \mathcal{R}^D$ but it will be convenient to treat the case $x \in \mathcal{R}$
- We observe the set $X = x_1, \cdots, x_n, \cdots, x_N$ with corresponding target variables $t = \{t_n\}$.

可以选择基函数Basis function choices

多项式拟合的不足是全局性，即：一个块的输入改变会影响到其他块的拟合，解决方法是将输入切分成若干个不同的块，每个区域使用不同的多项式拟合。

- Polynomial

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

- Gaussian

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

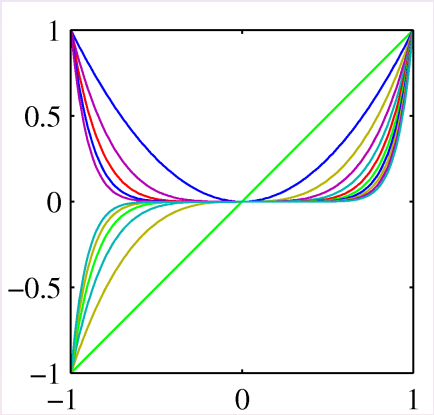
控制基函数在输入空间的位置，参数s控制基函数的空间大小。

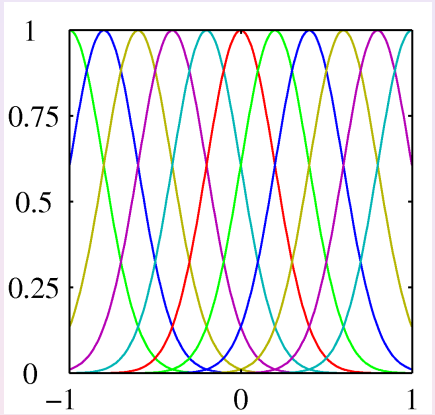
- Sigmoidal

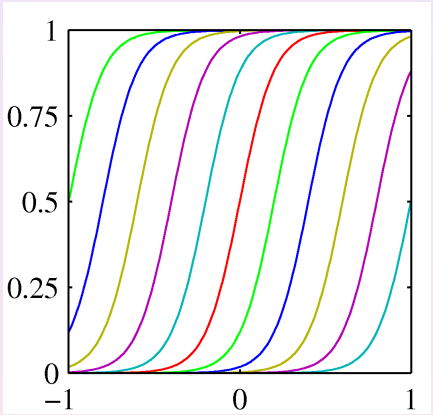
$$\phi_j(x) = \sigma\left(\frac{x-\mu_j}{s}\right) \quad with \quad \sigma(a) = \frac{1}{1+e^{-a}}$$

- splines, Fourier, wavelets, etc.

Examples of basis functions







Maximum likelihood and least squares

$$t = \underbrace{y(x,\omega)}_{\text{deterministic}} + \underbrace{\epsilon}_{\text{Gaussian noise}}$$

For a i.i.d. data set we have the likelihood function:

$$p(\mathbf{t}|x,\omega,\beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|\underbrace{\omega^\top \phi(x_n)}_{\text{mean}}, \underbrace{\beta^{-1}}_{\text{var}})$$

We can use the machinery of MLE to estimate the parameters ω and the precision β:

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

$$\Rightarrow \omega_{ML} = (\Phi^\top\Phi)^{-1}\Phi^\top\mathbf{t} \; with \; \Phi_{M\times N} = [\phi_{mn}(x_n)] (3.15)$$

$$\omega_0 = \bar{t} - \sum_{j=1}^{M-1}\omega_j\bar{\phi}_j.$$

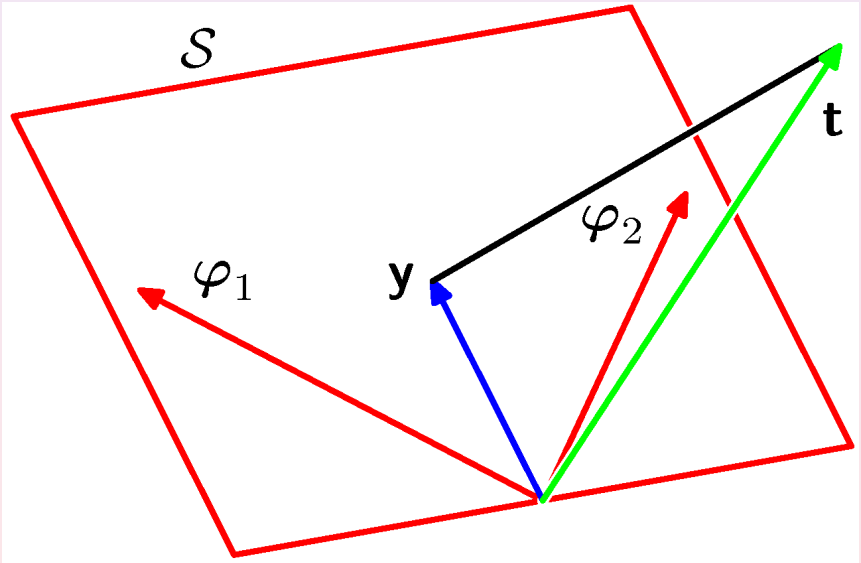
and:

$$\beta_{ML}^{-1} = \frac{1}{N}\sum_{n=1}^{N}(t_n - \omega_{ML}^\top\phi(x_n))^2$$

噪声精度参数的倒数由目标值在回归函数周围的残差方差给出。

最小二乘的几何意义

最小二乘回归解的得到方式：考虑一个N维空间，数据向量$t = \{t_1, ..., t_N\}$, 每个在数据点处的估计也可以看成是一个向量，φ_j 对应于Φ的第j 列，而$\phi(x_n)$ 对应于Φ的第n 行，当基函数的数量M 小于数据点的数量N, 那么第n个数据点的最小二乘估计$y(x_n, \omega)$可以看成是数据向量$t = \{t_1, ..., t_N\}$ 在由M 个向量基函数$\phi_j(x)$ 张成的子空间上的正交投影



Sequential learning,LMS算法

极大似然解的求解过程涉及到批处理整个训练数据，这种批处理技术对于大规模数据的计算而言并不可取，于是需要在线算法。Apply a technique known as 随机或序列梯度下降(stochastic gradient descent or sequential gradient descent), i.e.,
replace:

$$E_D(\omega) = \frac{1}{2}\sum_{n=1}^{N}(t_n - \omega^\top \phi(x_n))^2$$

with (η is a learning rate parameter):

$$\omega^{(\tau+1)} = \omega^\tau + \eta \underbrace{(t_n - \omega^{(\tau)^\top}\phi(x_n))\phi(x_n)}_{\nabla E_n} \qquad (3.23)$$

LMS中比较著名的是自适应滤波算法，它是在维纳滤波理论上运用速下降法后的优化延伸，该算法不需要已知输入信号和期望信号的统计特征，”当前时刻”的权系数是通过”上一时刻”权系数再加上一个负均方误差梯度的比例项求得。有时也称为均方误差的陡坡下降，可以在权向量面上的投影方向更新权重实现模型优化。

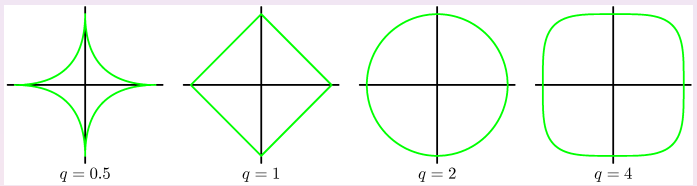
Regularized least squares

The total error function:

$$\frac{1}{2}\sum_{n=1}^{N}(t_n - \omega^\top \phi(x_n))^2 + \frac{\lambda}{2}\omega^\top\omega$$

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

这类正则方法常被称为权值衰减（weight decay）准则，这表明它倾向于让权值向零的方向衰减，除非有更多数据对其非零的支持。



更一般的正则回归表示为:

$$\frac{1}{2}\sum_{n=1}^{N}(t_n - \omega^\top \Phi(x_n))^2 + \frac{\lambda}{2}\sum_{j=1}^{M}|\omega_j|^q$$

Regularization has the advantage of limiting the model complexity (the appropriate number of basis functions). This is replaced with the problem of finding a suitable value of the regularization coefficient λ.

Regularization at large

- Most other modern methods for regression can be expressed as

$$\hat{\beta} = \text{argmin}_{\beta \in \mathbb{R}^p} \|y - X\beta\|_2^2 \text{ subject to} R(\beta) \leq t;$$

  or equivalently,

$$\hat{\beta} = \text{argmin}_{\beta \in \mathbb{R}^p} \|y - X\beta\|_2^2 + \lambda R(\beta).$$

- The term R is called a penalty or regularizer, and modifying the regression problem in this way is called applying regularization.
  - Regularization can be applied beyond regression: e.g., it can be applied to classification, clustering, principal component analysis;
  - Regularization goes beyond sparsity: e.g., design R to induce smoothness or structure, instead of pure sparsity.

偏差方差分解The Bias-Variance Decomposition

正确理解λ的正则作用。

- 在有限的数据集上想建立复杂的模型，无论将极大似然还是最小平方作为目标函数都会发生过度拟合Over-fitting occurs whenever the number of basis functions is large and with training data sets of limited size.
- 限定基函数数量虽然可以解决一定的过拟合问题，但也抑制了模型的灵活性，特别是限制了对数据中潜藏的特色模式的提取Limiting the number of basis functions limits the flexibility of the model.
- 正则是解决过拟合的一种方法，但引入了新的问题：如何确定正则系数Regularization can control over-fitting but raises the question of how to determine λ.
- The bias-variance tradeoff is a frequentist viewpoint of model complexity. 用频率的观点处理模型复杂性是不够的。

Back to section 1.5.5回归问题的决策问题

- 平方损失The regression loss-function: $L(t, y(x)) = (y(x) - t)^2$;
- 最优预测Solution: $y(x) = \int tp(t|x)\mathrm{d}t = E_t[t|x]$;
- The decision problem = minimize the expected loss:

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

  - this is known as the regression function
  - conditional average of t conditioned on x, e.g., figure 1.28, page 47

- Another expression for the expectation of the loss function:

$$E[L] = \int (y(x) - E[t|x])^2 p(x)\mathrm{d}x + \int (E[t|x] - t)^2 p(x)\mathrm{d}x \qquad (1.90)$$

- The optimal prediction is obtained by minimization of the expected squared loss function:

$$h(x) = E[t|x] = \int tp(t|x)dt \tag{3.36}$$

- The expected squared loss can be decomposed into two terms:

$$E[L] = \int (y(x)-h(x))^2 p(x)dx + \int (h(x)-t)^2 p(x,t)dxdt. \tag{3.37}$$

- The theoretical minimum of the first term is zero for an appropriate choice of the function $y(x)$ (for unlimited data and unlimited computing power).
- The second term arises from noise in the data and it represents the minimum achievable value of the expected squared loss.

An ensemble of data sets

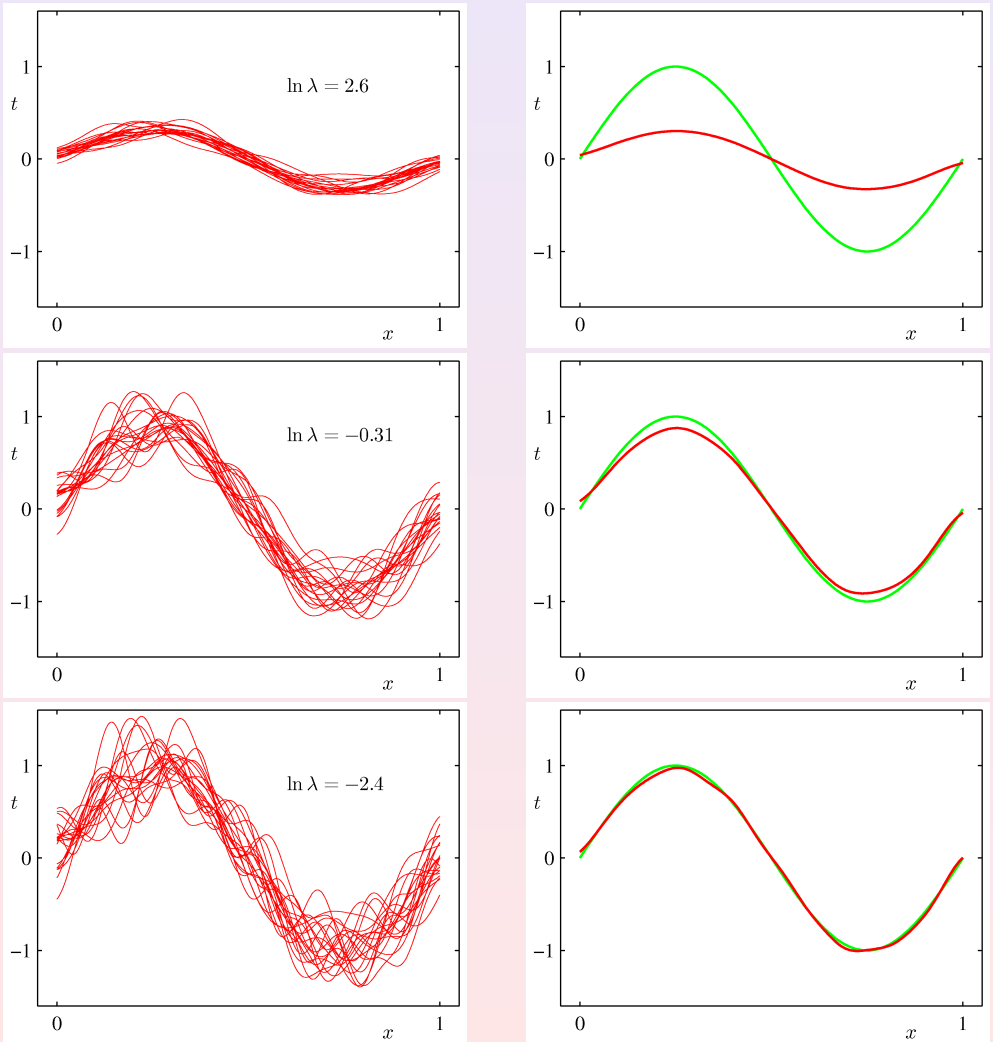
- For any given data set $\mathcal{D}$ we obtain a prediction function $y(x, \mathcal{D})$.
- The performance of a particular algorithm is assessed by taking the average over all these data sets, namely $E_{\mathcal{D}}[L]$. This expands into the following terms:

$$expected\ loss\ =\ (bias)^2\ +\ variance\ +\ noise$$

- There is a tradeoff between bias and variance:
  - flexible models have low bias and high variance
  - rigid models have high bias and low variance
- The bias-variance decomposition provides interesting insights in model complexity, **it is of limited practical value** because several data sets are needed.

Example: L=100, N=25, M=25, Gaussian basis

假设有100组模拟数据，每组有25个数据点，这些数据都是随机地从函数$h(x) = \sin(2\pi)x$ 抽取出来的，数据集的编号记为$L.$，（图中每张图显示的是20 个模型）



Choosing a value of the tuning parameter1/2

- Each regularization method has an associated tuning parameter: e.g., this was λ in the smoothing spline problem, and λ for lasso and ridge regression in the penalized forms (or t in the constrained forms)
- The tuning parameter controls the amount of regularization, so choosing a good value of the tuning parameter is crucial. Because each tuning parameter value corresponds to a fitted model, we also refer to this task as model selection.
- What we might consider a good choice of tuning parameter, however, depends on whether our goal is prediction accuracy or recovering the right model for interpretation purposes. We'll cover choosing the tuning parameter for the purposes of prediction; choosing the tuning parameter for the latter purpose is a harder problem.

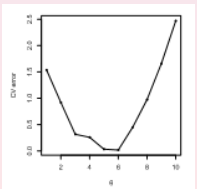
Choosing a value of the tuning parameter2/2

- Suppose that $\hat{f} = \hat{f}_\theta$ depends on a tuning parameter θ, and we want to choose θ to minimize:
- If we actually had training data $y_1, \ldots, y_n$ and test data $y_1, \ldots, y_n$ (meaning that we don't use this to build $\hat{f}_\theta$) in practice, then we could simply use called the test error, as an estimate for $\mathrm{PE}(\hat{f}_\theta)$. The larger that n is, the better this estimate. We usually don't have test data.

Cross-Validation1/10

- Given training data $(x_i, y_i), i = 1, \ldots, n$, An estimator $\hat{f}$ of some unknown function f is constructed. Suppose that $\hat{f} = \hat{\theta}$ depends on a tuning parameter θ.
- How to choose a value of θ to optimize predictive accuracy of $\hat{f}_\theta$. Cross-validation offers one way. Basic idea is simple: divide up training data into K folds (here K is fixed, e.g., K= 5 or K = 10)
- Each fold one at a time, train on the remaining data, and predict the held out observations, for each value of the tuning parameter I.e., for each value of the tuning parameter θ, the cross-validation error is

$$\mathrm{CV}(\theta) = \frac{1}{n} \sum_{k=1}^{K} \sum_{i \in F_k} (y_i - \hat{f}_\theta^{-k}(x_i))^2.$$



- Choose the value of tuning parameter that minimizes the CV error

Choosing λ for the lasso

- Example from last time:$n = 50$, $p = 30$, and the true model is linear with 10 nonzero coefficients. Consider the lasso estimate

$$\hat{\beta}^{lasso} = \text{argmin}_{\beta \in \mathbb{R}^p} \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

- Performing 5-fold crossvalidation, over 60 values of the tuning parameter between 0 and 12, choose $\hat{\lambda} = 3.458$

How the λ for the future prediction

- Recall that standard errors for the CV error curve at each tuning parameter value θ can be computed. First de?ne, for $k = 1, \ldots, K$:

$$\mathrm{CV}(\theta) = \frac{1}{n_k} \sum_{i \in F_k} (y_i - \hat{f}_\theta^{-k}(x_i))^2.$$

  where $n_k$is the number of points in the *kth* fold.
- Compute the sample standard deviation of $\mathrm{CV}_1(\theta), \ldots, \mathrm{CV}_K(\theta)$

$$\mathrm{SD}(\theta) = \sqrt{\mathrm{var}(\mathrm{CV}_1(\theta), \ldots, \mathrm{CV}_K(\theta))};$$

- finally $\mathrm{SE}(\theta) = \mathrm{SD}(\theta)/\sqrt{K}$ for the standard error of $\mathrm{CV}(\theta)$

One standard error rule

- One standard error rule is an alternative way of choosing θ from the CV curve. Start with the usual estimate

$$\hat{\theta} = \text{argmin}_{\theta \in \{\theta_1, \ldots, \theta_m\}} \text{CV}(\theta)$$

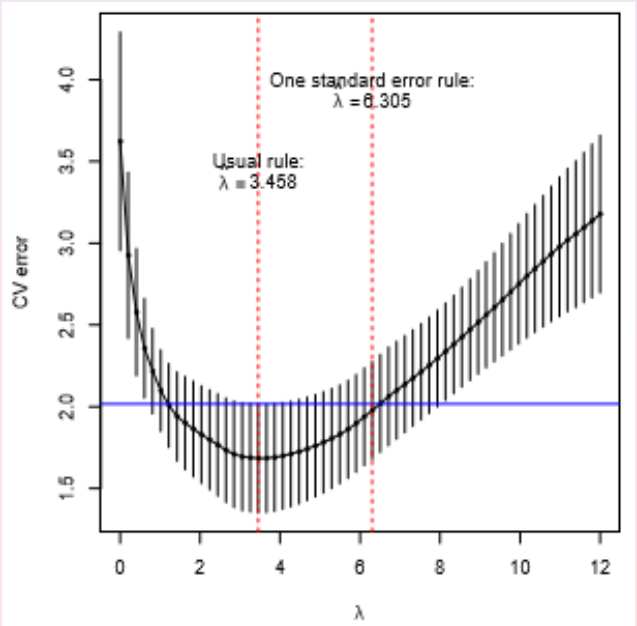
- move θ in the direction of increasing regularization until it ceases to be true that

$$\text{CV}(\theta) \leq \text{CV}(\hat{\theta}) + \text{SE}(\hat{\theta})$$

- In words, we take the simplest (most regularized) model whose error is within one standard error of the minimal error.

Example: choosing λ for the lasso

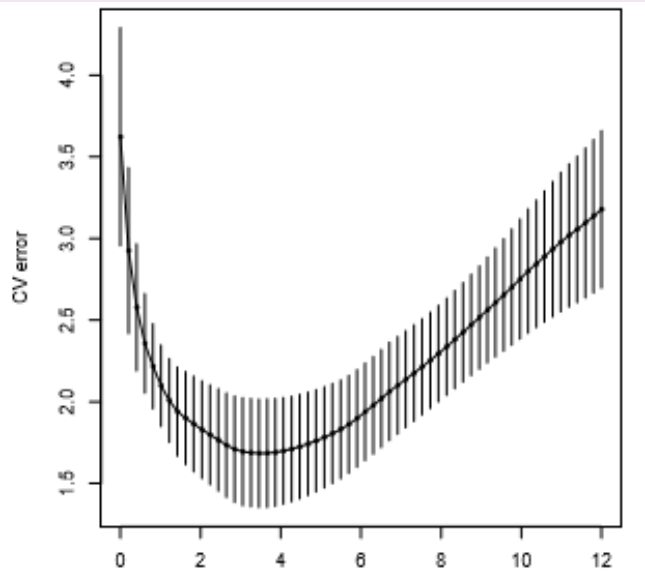
- In the lasso criterion, larger λ means more regularization;
- For last example, applying the one standard error rule has us increase the tuning parameter choice from $\hat{\lambda}$= 3.458 all the way up until $\hat{\lambda}$ = 6.305
- When fitting on the whole training data, this is a difference between a model with 19 and 16 nonzero coefficients(the true model is 10 nonzero coefficients.)



Model assessment

Not only using the CV error curves to pick a value of the tuning parameter. but to use cross-validation to actually estimate the prediction error values themselves,this is called model assessment.

- Report estimate for prediction error to collaborator;
- Compare estimated prediction error to known prediction errors of other estimators;
- Answering the question: is my estimator $\hat{f}$ working at all?
- Choosing between di?erent estimators (each possibly having their own tuning parameters)



Prediction versus interpretation

- cross-validation error estimates prediction error at any fixed value of the tuning parameter, and so by using it, we are implicitly assuming that achieving minimal prediction error is our goal
- Choosing λ for the goal of recovering the true model, for the sake of interpretation, is somewhat of a different task. The value of the parameter that achieves the smallest cross-validation error often corresponds to not enough regularization for these purposes. But the one standard error rule is a step in the right direction
- There are other (often more complicated) schemes for selecting a value of the tuning parameter for the purposes of true model recovery

Model Assessment

Instead of just using the CV error curves to pick a value of the tuning parameter, suppose to use cross-validation to actually estimate the prediction error values themselves

Prediction error and test error

- For model
$$y_i = f(x_i) + \epsilon_i, i = 1, \ldots, n;$$
- where $x_i \in \mathbb{R}^p$ are fixed (nonrandom) predictor measurements, f is the true function we are trying to predict (think $f(x_i) = x_i^T \beta^*$ for a linear model), and ϵ_i are random errors
- $(x_i, y_i), i = 1, \ldots, n$ the training data. Given an estimator $\hat{f}$ built on the training data, consider the average prediction error over all inputs
$$\mathrm{PE}(\hat{f}) = \mathrm{E}[\frac{1}{n}\sum_{i=1}^{n}(y_i^{new} - \hat{f}(x_i))^2];$$
where $y_i^{new} = f(x_i) + \epsilon_i, i = 1, \ldots, n$ are another set of observations, independent of $y_1, \ldots, y_n$

Choosing of K

- Essentially, the larger the choice of K, the more iterations needed, and so the more computation. So taking $K = n$ (i.e., leave-one-out cross-validation) is going to be a lot more costly than $K = 5$.
- Aside from computation, the choice of K affects the quality of cross-validation error estimates for model assessment. Consider:
  - $K = 2$: split-sample cross-validation. CV error estimates are going to be biased upwards, because only training on half the data each time.
  - $K = n$: leave-one-out cross-validation. CV estimates

$$\mathrm{CV}(\theta) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}_\theta^{-i}(x_i))^2.$$

    are going to have high variance because the averaging a bunch of (positively) correlated quantities
- with the bias-variance tradeoff, Choosing K = 5 or K = 10 seems to generally be a good tradeoff:

Ridge CV(θ)

- In each iteration train on a fraction of (about) $(K-1)/K$ the total training set, so this reduces the bias;
- There is less overlap between the training sets across iterations, so the terms in

$$\mathrm{CV}(\theta) = \frac{1}{n}\sum_{k=1}^{K}\sum_{i\in F_k}(y_i - \hat{f}_\theta^{-k}(x_i))^2.$$

are not as correlated, and the error estimate has a smaller

Suppose that our tuning parameter is $\theta = \lambda$ and $\hat{f}_\lambda$ is the ridge regression estimator

$$\hat{f}_\lambda(x_i) = x_i^T\hat{\beta}^{\text{ridge}} = x_i^T(X^TX + \lambda I)^{-1}X^Ty$$

Then it turns out that

$$\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}_\lambda^{-i}(x_i)\right)^2 = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{y_i - \hat{f}_\lambda(x_i)}{1 - S_{ii}}\right]^2$$

where $S = X(X^TX + \lambda I)^{-1}X^T$. This realization provides a huge computational savings!

- 正则通过限制模型的复杂度，使复杂的模型能够对大小有限的数据集上进行训练，而不会产生严重的过度拟合，然而这样做就将模型复杂度的问题从确定合适的基函数数量的问题转移到确定正则化系数λ 取合适值的问题上,虽然Cross-Validation 可以对模型的复杂性起到一定的控制，但是它的偏差- 方差依赖于对所有数据取平均，而在实际应用中却只是假设了模型仅仅作用于一个观测数据集。如果现在的条件变成可以对很多数据集进行观测，那么应该加强数据集对模型训练机制形成作用的认知，而这种认知显然会降低模型的过拟合。

Bayesian Linear Regression (1/5)

在我们讨论用极大似然估计加正则的方法改进回归模型估计时，已经注意到由基函数带来了复杂的模型，但需要借助测试数据加以正则来调整他们在模型中的作用。于是就产生了如何确定模型一个比较恰当的复杂度的问题？引入额外的数据能够验证模型是否发生过拟合，但这毕竟需要调用大量的资源，这里考虑贝叶斯调节方法，为简化起见，本章仅讨论单一目标变量的情形(single response)

Assume additive gaussian noise with known precision β.

The likelihood function $p(\mathbf{t}|\boldsymbol{\omega})$ is the exponential of a quadratic function of $\boldsymbol{\omega}$, its conjugate prior is Gaussian:

$$p(\boldsymbol{\omega}) = \mathcal{N}(\boldsymbol{\omega}|\mathbf{m}_0, \mathbf{S}_0) \tag{3.48}$$

Its posterior is also Gaussian (2.116):

$$p(\boldsymbol{\omega}|\mathbf{t}) = \mathcal{N}(\boldsymbol{\omega}|\mathbf{m}_N, \mathbf{S}_N) \propto p(\mathbf{t}|\boldsymbol{\omega})p(\boldsymbol{\omega}) \tag{3.49}$$

$$\textit{where} \begin{vmatrix} \mathbf{m}_N = \mathbf{S}_N(\mathbf{S}_0^{-1}\mathbf{m}_0 + \beta\Phi^\top\mathbf{t}) \\ \mathbf{S}_N^{-1} = \mathbf{S}_0^{-1} + \beta\Phi^\top\Phi \end{vmatrix} \tag{3.50/3.51}$$

- Note how this fits a sequential learning framework
- The max of a Gaussian is at its mean: $\boldsymbol{\omega}_{MAP} = \mathbf{m}_N$

Bayesian Linear Regression (2/5)

Assume $p(\boldsymbol{\omega})$ is governed by a hyperparameter α following a Gaussian law of scalar covariance (*i.e.* $\mathbf{m}_0 = 0$ *and* $\mathbf{S}_0 = \alpha^{-1}\mathbf{I}$):

$$p(\boldsymbol{\omega}|\alpha) = \mathcal{N}(\boldsymbol{\omega}|0, \alpha^{-1}\mathbf{I}) \tag{3.52}$$

$$\textit{then} \begin{vmatrix} \mathbf{m}_N = \beta \mathbf{S}_N \Phi^\top \mathbf{t} \\ \mathbf{S}_N^{-1} = \alpha\mathbf{I} + \beta\Phi^\top\Phi \end{vmatrix} \tag{3.53/3.54}$$

- $$\textit{Note} \alpha \to 0 \textit{ implies } \mathbf{m}_N \to \boldsymbol{\omega}_{ML} = (\Phi^\top\Phi)^{-1}\Phi^\top\mathbf{t} \tag{3.35}$$

Log of posterior is sum of log of likelihood and log of prior:

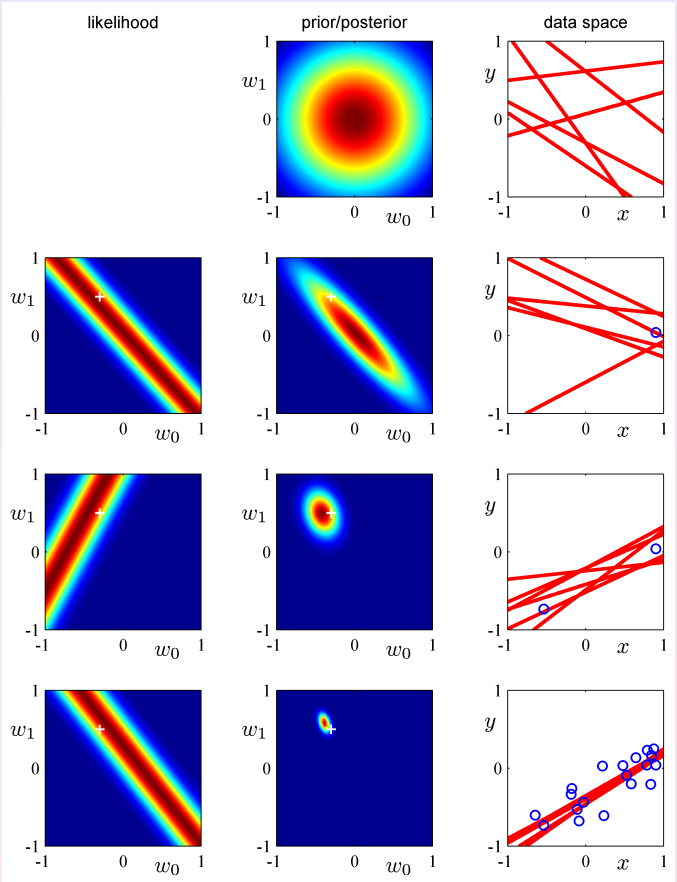
$$ln\, p(\boldsymbol{\omega}|\mathbf{t}) = -\frac{\beta}{2}\sum_{n=1}^{N}(t_n - \boldsymbol{\omega}^T\Phi(\mathbf{x}_n))^2 \; - \; \frac{\alpha}{2}\boldsymbol{\omega}^T\boldsymbol{\omega} + const \tag{3.55}$$

which is equivalent to a quadratic regularizer with coeff. α/β

数据取自$y = a_0 + a_1 x + N(0, 0.2), a_0 = -0.3, a_1 = 0.5$,目标是

1. 恢复a_0和a_1.
2. 理解模型对数据集的依赖关系。

假设：精度参数已知，$\beta = (\frac{1}{0.2})^2 = 25.\alpha == 2.0$



Bayesian Linear Regression-预测分布(3/5)

In practice, 我们真正感兴趣的不是$\boldsymbol{\omega}$而是从x能不能预测出t :we want to make predictions of t for new values of x:

$$p(t|\mathbf{t},\alpha,\beta) = \int p(t|\boldsymbol{\omega},\beta)p(\boldsymbol{\omega}|\mathbf{t},\alpha,\beta)d\boldsymbol{\omega} \tag{3.57}$$

- Conditional distribution:

$$p(t|\boldsymbol{\omega},\beta) = \mathcal{N}(t|y(x,\boldsymbol{\omega}),\beta^{-1}) \tag{3.8}$$

- Posterior:

$$p(\boldsymbol{\omega}|\mathbf{t},\alpha,\beta) = \mathcal{N}(\boldsymbol{\omega}|\mathbf{m}_N.\mathbf{S}_N) \tag{3.49}$$

The convolution is a Gaussian (2.115):

$$p(t|\mathbf{x},\mathbf{t},\alpha,\beta) = \mathcal{N}(t|\mathbf{m}_N^T\Phi(x),\sigma_N^2(x)) \tag{3.58}$$

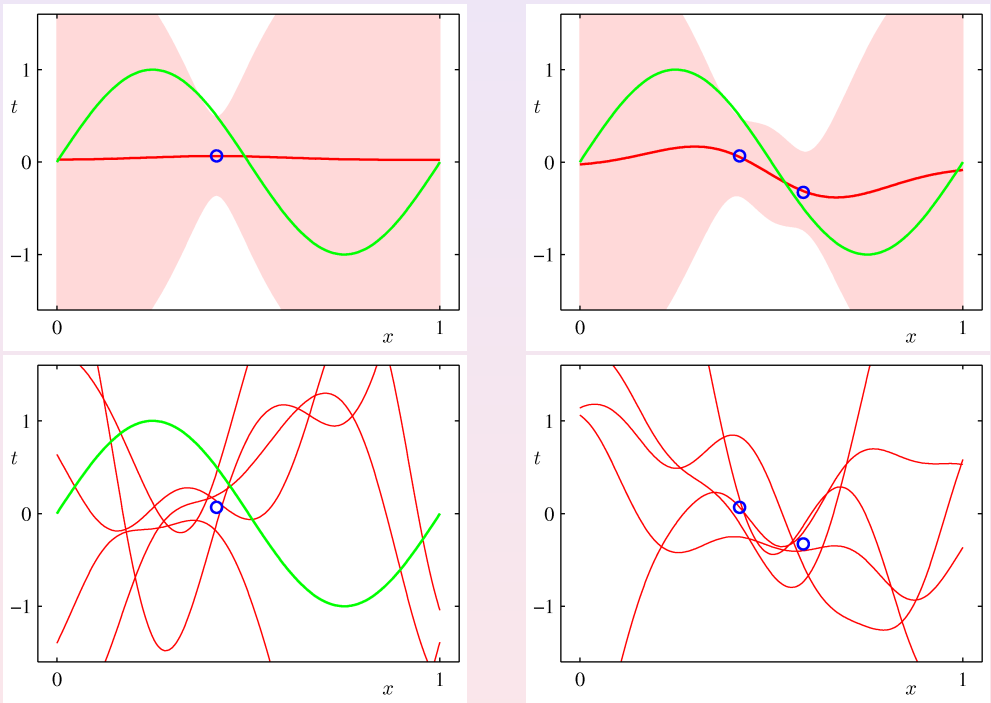
where

$$\sigma_N^2(\mathbf{x}) = \underbrace{\beta^{-1}}_{\text{noise in data}} + \underbrace{\Phi(\mathbf{x})^T\mathbf{S}_N\Phi(\mathbf{x})}_{\text{uncertainty in}\boldsymbol{\omega}} \tag{3.59}$$

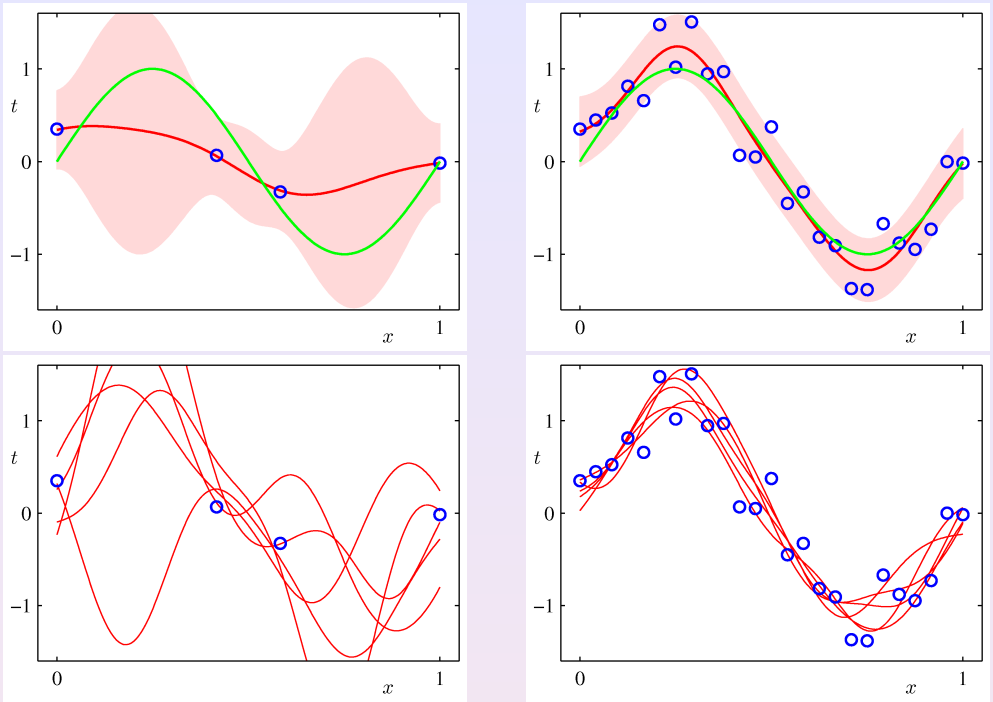
第一项是t的噪声，第二项反映了与参数$\boldsymbol{\omega}$关联的不确定性，两者独立可以相加，当额外数据进来以后，后验概率应该变窄，可以证明$\sigma_{N+1}^2 \leq \sigma_N^2$.所以预测分布的方差只与由参数$\beta$ 控制的具有可加性的噪声有关。

一个贝叶斯回归模型的预测分布建模的过程：

- 绿色曲线对应着真实函数$\sin(2\pi x)$;
- 四个数据量不同的数据用蓝色圆圈表示(N=1,2,4,25);
- 红色曲线对应的是高斯预测曲线的均值;
- 红色阴影表示均值两侧一倍标准差范围的区域;



- 预测的不确定性依赖于x,在数据点的邻域内最小，不确定性的程度随着观测数据点的增多而逐渐减小。



- 如果使用局部基函数（例如高斯基函数），那么在距离基函数中心比较远的区域，公式(3.59) 给出的预测方差的第二项贡献会趋于零，只剩下噪声的贡献β^{-1}; 这表示在距离基函数所在的区域以外进行外插的时候完全不受基函数影响，这不是一个估计问题，这时可能需要另一种方式来避免这种情况的发生。
- 如果ω和β都未知，可以引入高斯+Gamma分布定义的先验分布，在这种情况下，预测分布是学生t 分布。

Bayesian Linear Regression (4/5)-等价核

$y(\mathbf{x}, \mathbf{m}_N)$ rewrites as $\sum_{n=1}^{N} k(\mathbf{x}, \mathbf{x}_n)\mathbf{t}_n$ where

$$k(\mathbf{x}, \mathbf{x}') = \beta\Phi(\mathbf{x})^T\mathbf{S}_N\Phi(\mathbf{x}') \qquad (3.61 - 3.62)$$

- 平滑矩阵Smoother matrix, 等价核equivalent kernel, 线性平滑linear smoother
- 核的作用是距离x较近的点赋予一个比较高的权值，距离x较远的点赋予一个比较低的权值The kernel works as a similarity or closeness measure, giving more weight to evidence that is close to the point where we want to make the prediction
  - Basis functions $\leftrightarrow$ kernel duality
  - With

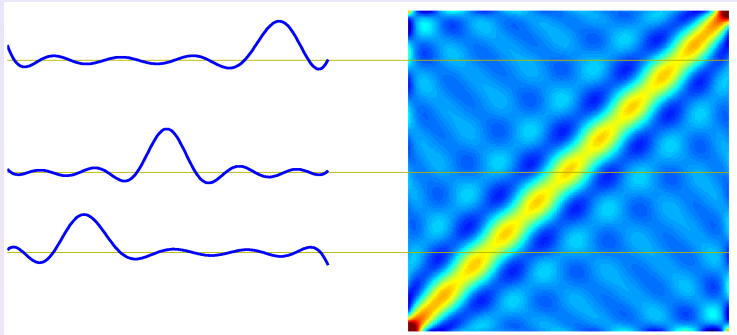
$$\Psi(\mathbf{x}) = \beta^{-1/2}\mathbf{S}_N^{1/2}\Phi(\mathbf{x}), k(\mathbf{x}, \mathbf{x}') = \Psi(\mathbf{x})^T\Psi(\mathbf{x}') \qquad (3.65)$$

  - The kernel sums to one (over the training set)
  -

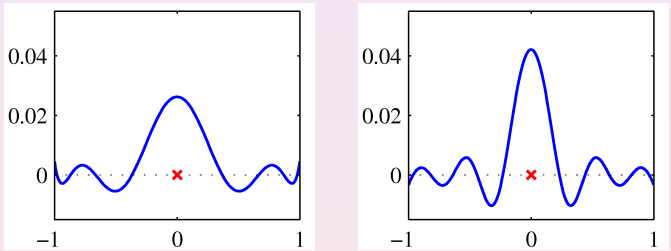
$$\mathrm{cov}(y(\mathbf{x}), y(\mathbf{x}')) = \beta^{-1}k(\mathbf{x}, \mathbf{x}') \qquad (3.63)$$

Bayesian Linear Regression (5/5)

Kernel from Gaussian basis functions



Kernels at $x = 0$ for kernels corresponding (left) to the polynomial basis functions and (right) to the sigmoidal basis functions.



在3.8中我们从预测分布的角度已经看到了不同点附近的预测精度有很大不同，训练数据之间的预测精度比之外的预测精度高，这个原因已经找到了，这是因为这里的精度受到核函数叠加的不通过程度的影响，这样就产生了通过核函数表示线性回归解决回归问题的方法，不引入基函数，而是定义一组局部核函数，在给定观测数据集的条件下，直接用核函数对新的输入进行预测，这个方法叫做回归的高斯过程。

Bayesian Model Comparison (1/3)-Bayes模型比较

The overfitting that appears in ML can be avoided by marginalizing over the model parameters.

- Cross-validation is no more useful
- We can use all the data for better training the model
- We can compare models based on training data alone

- 假设我们想比较的是L个模型$\mathcal{M}_i$,其中$i = 1, \cdots, L$这里，一个模型指的是观测数据$\mathcal{D}$ 上的概率分布。在多项式曲线拟合的问题中，概率分布被定义在目标值t 上，输入值X 被假定是已知的。其他类型的模型是X 和t 的联合分布。我们会假设数据是由这些模型中一个生成的，但是不知道究竟是哪一个，不确定性通过先验概率分布$p(\mathcal{M}_i)$ 表示。给定训练数据集$\mathcal{D}$，估计后验分布如下：

$$p(\mathcal{M}_i|\mathcal{D}) \propto p(\mathcal{M}_i)p(\mathcal{D}|\mathcal{M}_i) \tag{3.66}$$

- $p(\mathcal{D}|\mathcal{M}_i)$:模型证据或边际似然model evidence or marginal likelihood. 反映了数据展现出的不同模型的优先级，也可以被看做在模型空间中的似然函数。
- 两个模型的模型证据的比值$\frac{p(\mathcal{D}|\mathcal{M}_i)}{p(\mathcal{D}|\mathcal{M}_j)}$ 称为贝叶斯因子（Bayes factor）（Kass and Raftery, 1995）。

Bayesian Model Comparison (2/3)-Bayes模型比较

- 一旦知道了模型上的后验概率分布，那么根据概率的加和规则与乘积规则，预测分布为

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

- 对于模型求平均的一个简单的近似是使用最可能的一个模型做预测，这被称为模型选择（model selection）。
- 对于一个由参数w控制的模型，根据概率的加和规则和乘积规则，模型证据为

$$p(\mathcal{D}|\mathcal{M}_i) = \int p(\mathcal{D}|\boldsymbol{\omega},\mathcal{M}_i)p(\boldsymbol{\omega}|\mathcal{M}_i)\mathrm{d}\boldsymbol{\omega}$$

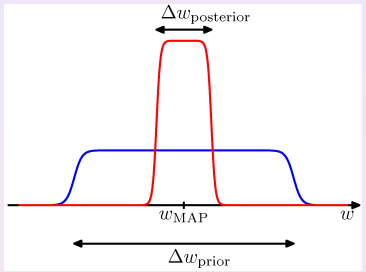
- 首先考虑模型有一个参数ω的情形。这个参数的后验概率正比于$p(\mathcal{D}|\omega)p(\omega)$, 为简化记号，省略了它对于模型$\mathcal{M}_i$ 的依赖。假设后验分布在极大似然值ω_{MAP} 有最大值。Using model selection and assuming the posterior $p(\omega|\mathcal{D},\mathcal{M}_i)$ is sharply peaked at ω_{MAP} (single parameter case):

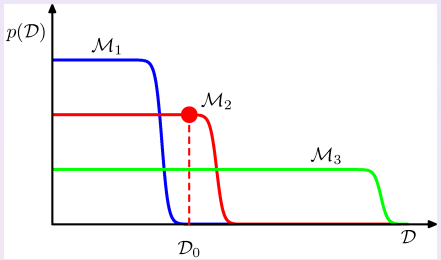
$$p(\mathcal{D}) = \int p(\mathcal{D}|\omega)p(\omega)d\omega \simeq p(\mathcal{D}|\omega_{MAP})\frac{\Delta\omega_{posterior}}{\Delta\omega_{prior}} \tag{3.70}$$

Bayesian Model Comparison (3/3)

Single Parameter Case：假设后验$p(\omega|\mathcal{D},\mathcal{M}_i)$ is sharply peaked at ω_{MAP}.

$$p(\mathcal{D}) = \int p(\mathcal{D}|\omega)p(\omega)\mathrm{d}\omega \simeq p(\mathcal{D}|\omega_{MAP})\frac{\Delta\omega_{posterior}}{\Delta\omega_{prior}} \tag{3.70}$$





Back to multiple parameters, assuming they share the same $\delta\omega$ ratio, the complexity penalty is linear in M:

$$\ln\ p(\mathcal{D}) \simeq \ln\ p(\mathcal{D}|\boldsymbol{\omega}_{MAP}) + M\ln\left(\frac{\Delta\omega_{posterior}}{\Delta\omega_{prior}}\right) \tag{3.72}$$

About $p(\mathcal{D}|M_i)$:

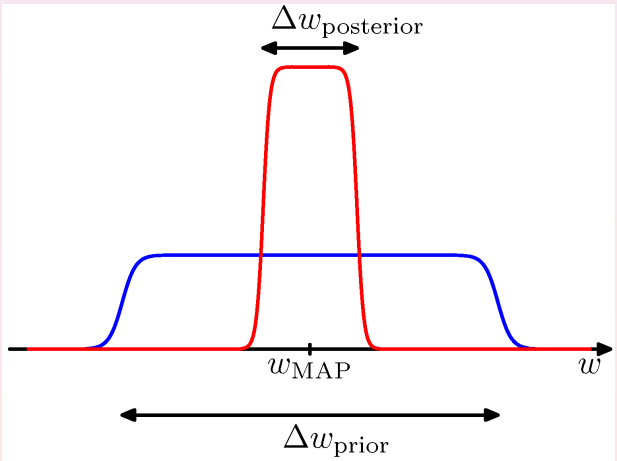
- if $\mathcal{M}_i$ is too simple, bad fitting of the data
- if $\mathcal{M}_i$ is too complex/powerful, the probability of generating the observed data is washed out

贝叶斯框架总结

- 生成数据的真实模型包含在备选的模型中；
- 贝叶斯模型会倾向于选择出更为正确的模型，因为它综合考虑数据和参数的先验信息，因为当我们把贝叶斯因子在数据集上平均，得到期望贝叶斯因子，实际上是一个Kullback-Leibler 散度的形式。

$$\int p(\mathcal{D}|\mathcal{M}_1) \ln \frac{p(\mathcal{D}|\mathcal{M}_1)}{p(\mathcal{D}|\mathcal{M}_2)} \mathrm{d}\mathcal{D} \tag{3.73}$$

  如果两个分布相等，散度为零，也就是说，贝叶斯总会倾向于选择正确的模型。
- 贝叶斯框架能够避免过度拟合，并且允许模型能够基于训练数据进行自我对比，但是与其他模式识别方法类似，贝叶斯需要模型形式的假设，如果先验假设和似然的假设违背常理，那么结果就会出错。在图3.12 中，模型证据对先验的依赖是非常强的，如果先验改变，那么模型证据就会随之改变。所以在实际应用中，保留一个独立的测试集，交叉验证用来对系统的整体性能进行评估也是需要的。



The evidence approximation (1/5)

Fully bayesian treatment would imply marginalizing over hyperparameters and parameters, but this is intractable:
完整地对超参数求积分很多情况下并不具有解析解，一种近似的方法是：先对参数$\mathbf{w}$ 求积分，得到边缘似然，再极大化边缘似然，确定超参数的值，也称为经验贝叶斯（bernardo and Smith 1994），二类极大似然（type 2 maximum likelihood）(Berger 1985)，或被称为广义极大似然。在机器学习中称为证据近似(evidence approximation)

$$p(t|\mathbf{t}) = \int\int\int p(t|\boldsymbol{\omega},\beta)p(\boldsymbol{\omega}|\mathbf{t},\alpha,\beta)p(\alpha,\beta|\mathbf{t})\mathrm{d}\boldsymbol{\omega}\mathrm{d}\alpha\mathrm{d}\beta \qquad (3.74)$$

其中

$$p(t|x,\boldsymbol{\omega},\beta) \sim \mathcal{N}(t|y(x,\boldsymbol{\omega},\beta^{-1}) \;\; \beta^{-1} = \sigma^2;$$

$$p(\boldsymbol{\omega}|\mathbf{t}) = \mathcal{N}(\boldsymbol{\omega}|\mathbf{m}_N,\mathbf{S}_N) \propto p(\mathbf{t}|\boldsymbol{\omega})p(\boldsymbol{\omega}) \qquad (3.49)$$

$$\begin{vmatrix} \mathbf{m}_N = \beta\mathbf{S}_N\Phi^\top\mathbf{t} \\ \mathbf{S}_N^{-1} = \alpha\mathbf{I}+\beta\Phi^\top\Phi \end{vmatrix} \qquad (3.53/3.54)$$

边缘似然函数$p(\mathbf{t}|\alpha,\beta)$通过对$\boldsymbol{\omega}$积分得到：

$$p(\mathbf{t}|\alpha,\beta)=\int p(\mathbf{t}|\boldsymbol{\omega},\beta)p(\boldsymbol{\omega}|\alpha)dw$$

计算这个积分有两种方法：1.使用线性-高斯模型的条件概率分布结果；2. 通过对指数项配平方，使用高斯分布的归一化。得到证据函数：

$$p(t|\alpha,\beta)=\left(\frac{\beta}{2\pi}\right)^{\frac{N}{2}}\left(\frac{\alpha}{2\pi}\right)^{\frac{M}{2}}\int\exp\{-E(\boldsymbol{\omega})\}\mathrm{d}\boldsymbol{\omega}$$

同时还定义了：

$$E(\boldsymbol{\omega})=\beta E_D(\boldsymbol{\omega})+\alpha E_W(\boldsymbol{\omega})=\frac{\beta}{2}\|t-\Phi\boldsymbol{\omega}\|^2+\frac{\alpha}{2}\boldsymbol{\omega}^T\boldsymbol{\omega}$$

然后通过比较多元高斯分布的归一化系数，可以求出$\boldsymbol{\omega}$的积分：

$$\begin{aligned}\int\exp\{-E(\boldsymbol{\omega})\}\mathrm{d}\boldsymbol{\omega}&=\exp\{-E(m_N)\}\int\exp\{-\frac{1}{2}(\boldsymbol{\omega}-m_N)^TA(\boldsymbol{\omega}-m_N)\}\mathrm{d}\boldsymbol{\omega}\\&=\exp\{-E(m_N)\}(2\pi)^{M/2}|A|^{-1/2}\end{aligned}$$

The evidence approximation 3/5

An approximation is found by maximizing the marginal likelihood function $p(\alpha, \beta|\mathbf{t}) \propto p(\mathbf{t}|\alpha, \beta)p(\alpha, \beta)$ to get $(\hat{\alpha}, \hat{\beta})$ (empirical Bayes).

$$\ln\, p(\mathbf{t}|\alpha, \beta) = \frac{M}{2}\ln\alpha + \frac{N}{2}\ln\beta - E(\mathbf{m}_N) - \frac{1}{2}\ln|\mathbf{S}_N^{-1}| - \frac{N}{2}\ln(2\pi) \quad (3.77 \rightarrow 3.86)$$

Assuming $p(\alpha, \beta|\mathbf{t})$ is highly peaked at $(\hat{\alpha}, \hat{\beta})$: 其中

$$\mathbf{S}_N = \alpha I + \beta\Phi^T\Phi$$

$$E(\mathbf{m}_N) = \frac{\beta}{2}||\mathbf{t} - \Phi\mathbf{m}_N||^2 + \frac{\beta}{2}\mathbf{m}_N^T\mathbf{m}_N;$$

$$\mathbf{m}_N = \beta\mathbf{S}_N\phi^T\mathbf{t}$$

M是参数个数，N是数据量

$$p(t|\mathbf{t}) \simeq (t|\mathbf{t}, \hat{\alpha}, \hat{\beta}) = \int p(t|\boldsymbol{\omega}, \hat{\beta})p(\boldsymbol{\omega}, \hat{\alpha}, \hat{\beta})\mathrm{d}\boldsymbol{\omega} \qquad (3.75)$$

最大化证据函数The evidence approximation 4/5

书中首先定义了特征向量方程，为的是在后面最大化的过程中容易进行计算：

$$(\beta\Phi^T\Phi)u_i = \lambda_i u_i$$

Step1:首先对证据函数的对数形式对α 求导：

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\alpha}\ln|A| &= \frac{\mathrm{d}}{\mathrm{d}\alpha}\ln\Pi_i(\lambda_i+\alpha)\\ &= \frac{\mathrm{d}}{\mathrm{d}\alpha}\sum_i \ln(\lambda_i+\alpha)\\ &= \sum_i \frac{1}{\lambda_i+\alpha}\end{aligned}$$

可以得到关于α的驻点

$$0 = \frac{M}{2\alpha} - \frac{1}{2}{m_N}^T m_N - \frac{1}{2}\sum_i \frac{1}{\lambda_i+\alpha}$$

$$\alpha {m_N}^T m_N = M - \alpha\sum_i \frac{1}{\lambda_i+\alpha} = \gamma$$

最大化证据函数The evidence approximation 5/5

因此γ可以写成

$$\gamma = \sum_i \frac{\lambda_i}{\alpha + \lambda_i}. \quad (3.91)$$

得到α的公式：

$$\alpha = \frac{\gamma}{m_N^T m_N}. \quad (3.92)$$

Step2:现在对证据函数的对数形式关于β 求导：

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\beta} \ln |A| &= \frac{\mathrm{d}}{\mathrm{d}\beta} \sum_i \ln(\lambda_i + \alpha) \\ &= \frac{1}{\beta} \sum_i \frac{\lambda_i}{\lambda_i + \alpha} = \frac{\gamma}{\beta} \quad (3.93) \end{aligned}$$

边际似然的β驻点满足下式：

$$0 = \frac{N}{2\beta} - \frac{1}{2} \sum_{n=1}^{N} \{t_n - m_N^T \phi(x_n)\}^2 - \frac{\gamma}{2\beta};$$

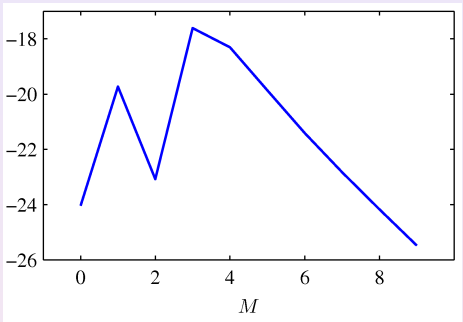
整理之后，我们可以得到β要满足的方程：

$$\frac{1}{\beta} = \frac{1}{N - \gamma} \sum_{n=1}^{N} \{t_n - m_N^T \phi(x_n)\}^2. (3.95)$$

与α一样，这里也是β的一个隐式解，可以通过迭代解出。

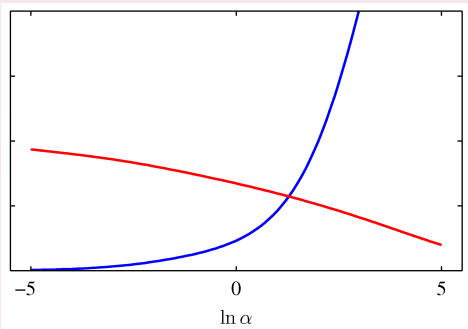
The evidence approximation参数有效量

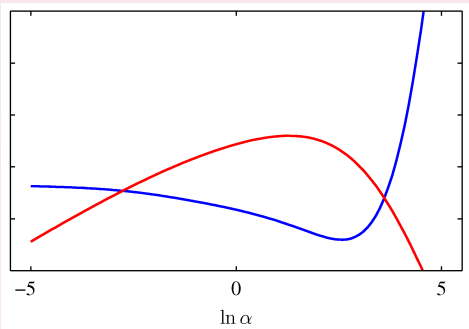
Plot of the model evidence $\ln p(\mathbf{t}|\alpha, \beta)$ versus M, the model complexity, for the polynomial regression of the synthetic sinusoidal example (with fixed α).

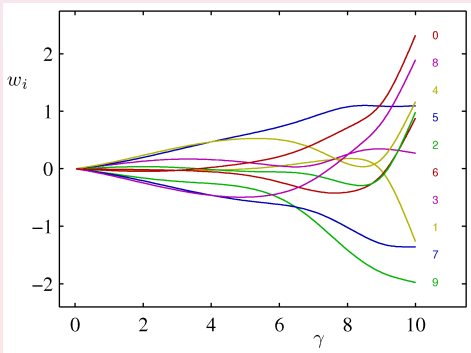


The computation for $(\hat{\alpha}, \hat{\beta})$ give rise to $\gamma = \alpha \mathbf{m}_N^T \mathbf{m}_N$ (3.90)

γ has the nice interpretation of being the effective number of parameters







固定基函数的局限性

- 优点：最小二乘解析解，易于计算的贝叶斯方法；
- 基函数在观测之前就已经固定下来，基函数的数量随输入空间维度D 的增加而迅速增大；
- 真实数据
  - 数据向量$\{x_n\}$通常位于一个非线性流形的内部；
  - 目标变量可能只依赖于流形中的少量可能的方向。