Linear Models for Regression

Regression: predict a continuous variable t given an input vector $x \in \mathbb{R}^D$.

We study linear models on the parameters and not on the input variable.

Given a training data with targets
$$\{(x_1, t_1), (x_2, t_2), \ldots, (x_n, t_n)\}$$
The goal is to predict t for an unobserved input x. From a probabilistic approach we want to construct a model $p(t|x)$.

Basis Function Models

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

called bias

parameters

basis functions

Convenient to define $\phi_0(x) = 1$ thus

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

$w = (w_0, \ldots, w_{M-1})^T$

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

Usually $\{\phi_i(x)\}$ contains some pre-processing or feature extraction.

Examples of Basis:

$\phi_j = x^j$ (Polynomial)

$\phi_j = e^{-\frac{(x-\mu_j)^2}{2s^2}}$ (gaussian)

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

$\phi_j = \sin(w_j x)$ (Fourier Basis)

Basis functions that are localized to finite regions of space and frequency are known as wavelets. Most applicable when x is on a regular lattice.

Maximum likelihood and least squares

Assume $t = y(x) + \varepsilon$ gaussian Noise $\varepsilon \sim \mathcal{N}(0, \beta^{-1})$

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

$\mathbb{E}[t \mid x] = \int t\, p(t \mid x)\, dt = y(x, w)$ (Mean of the gaussian)

Denote $X = \{x_1, \dots, x_N\}$

$t = (t_1, \dots, t_N)^T \in \mathbb{R}^N$

Under the assumption that (x_i, t_i) are drawn independently from the above distribution we have the likelihood function

$$p(t \mid X, w, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n \mid y(x_n, w), \beta^{-1})$$

$$= \prod_{n=1}^{N} \mathcal{N}(t_n \mid w^T \phi(x_n), \beta^{-1})$$

We omit X in the following since we are not trying to model the distribution, so it will always appear as a conditioning variable.

$$\log p(t \mid w, \beta) = \sum_{n=1}^{N} \log \mathcal{N}(t_n \mid w^T \phi(x_n), \beta^{-1})$$

$$= \sum_{n=1}^{N} -\frac{1}{2}\log 2\pi + \frac{1}{2}\log\beta - \frac{\beta}{2}(t_n - w^T \phi_n)^2$$

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

where we defined the sum-of-squares error function

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

Now

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

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

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

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

Recall that $\phi(x_n) = (\phi_0(x_n), \phi_1(x_n), \dots, \phi_{M-1}(x_n))^T$.

Thus the LHS is:

$$\sum_{n=1}^{N} t_n\phi(x_n) = \begin{pmatrix} \sum_n t_n\phi_0(x_n) \\ \sum_n t_n\phi_1(x_n) \\ \vdots \\ \sum_n t_n\phi_{M-1}(x_n) \end{pmatrix} = \Phi\begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{pmatrix} = \Phi t$$

$$\Phi_{mn} = \phi_m(x_n)$$

$$= \begin{pmatrix} \phi_0(x_1) & \phi_0(x_2) & \cdots & \phi_0(x_N) \\ \phi_1(x_1) & \phi_1(x_2) & \cdots & \phi_1(x_N) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{M-1}(x_1) & \phi_{M-1}(x_2) & \cdots & \phi_{M-1}(x_N) \end{pmatrix}_{M\times N}$$

"Fat Matrix"

The RHS is

$$\left(\sum_{n=1}^{N} \phi(x_n)\,\phi(x_n)^T\right) w = \left(\sum_{n=1}^{N} (\phi_0(x_n), \dots, \phi_{M-1}(x_n))^T \cdot \right.$$
$$\left. \cdot\, (\phi_0(x_n), \dots, \phi_{M-1}(x_n))\right) w$$

$$= \sum_{n=1}^{N} \begin{pmatrix} \phi_0(x_n)\phi_0(x_n) & \phi_0(x_n)\phi_1(x_n) & \cdots & \phi_0(x_n)\phi_{M-1}(x_n) \\ \phi_1(x_n)\phi_0(x_n) & \phi_1(x_n)\phi_1(x_n) & \cdots & \phi_1(x_n)\phi_{M-1}(x_n) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{M-1}(x_n)\phi_0(x_n) & \phi_{M-1}(x_n)\phi_1(x_n) & \cdots & \phi_{M-1}(x_n)\phi_{M-1}(x_n) \end{pmatrix}_{M\times M} w$$

The sum can be carried inside the matrix and the element (i,j) has the form

$$\sum_{n=1}^{N} \phi_i(x_n)\,\phi_j(x_n) = (\Phi\,\Phi^T)_{ij}$$

"these are changing"

Therefore we have

$$(\Phi\,\Phi^T)\,w = \Phi t\,, \quad \boxed{w_{ML} = (\Phi\Phi^T)^{-1}\Phi t}$$

Notice that $\Phi\Phi^T \in \mathbb{R}^{M\times M}$ which will be invertible if $N > M$. The matrix

$$\boxed{\Phi^+ = (\Phi\Phi^T)^{-1}\Phi}$$

- is the pseudo-inverse of Φ^T. $\Phi^+\Phi^T = I$ ∴

$$\Phi(\Phi^+)^T = I = \Phi\left(\Phi^T(\Phi\Phi^T)^{-1}\right)$$

So $\boxed{\tilde{\Phi}^+ = \Phi^T(\Phi\Phi^T)^{-1}}$ is the (right) pseudo-inverse of Φ.

In the text the convention is

$$\Phi \to \Phi^T$$

Thus we have

$$\Phi = \begin{pmatrix} \phi_0(x_1) & \cdots & \phi_{M-1}(x_1) \\ \phi_0(x_2) & & \phi_{M-1}(x_2) \\ \vdots & & \vdots \\ \phi_0(x_N) & & \phi_{M-1}(x_N) \end{pmatrix}_{N \times M}$$

"shiny Matrix"

$$\boxed{w_{ML} = (\Phi^T \Phi)^{-1} \Phi^T t}$$

Now $\Phi^+ = (\Phi^T \Phi)^{-1} \Phi^T$ is a left pseudo-inverse.

Let's analyze the bias:

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

$$\frac{\partial E_D}{\partial w_0} = \sum_n \left(t_n - w_0 - \sum_{j=1}^{M-1} w_j \phi_j(x_n)\right) = 0$$

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

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

The bias compensates for the averages in the training set.

Now maximizing over β:

$$0 = \frac{N}{2}\frac{1}{\beta} - E_D(w) \quad \therefore \quad \boxed{\frac{1}{\beta_{ML}} = \frac{2}{N} E_D(w_{ML}) = \frac{1}{N}\sum_{n=1}^{N}\left(t_n - w_{ML}^T \phi(x_n)\right)^2}$$

residual variance of the target values around the regression function.

geometric Interpretation

$t = (t_1, \dots, t_N) \in \mathbb{R}^N$

the jth column of Φ is also a vector in $\mathbb{R}^N$: $\varphi_j = (\phi_j(x_1), \phi_j(x_2), \dots, \phi_j(x_N))^T$.

Now $\hat{y}_n = y(x_n, w_{ML}) = w_{ML}^T \phi(x_n)$ where $\phi(x_n) = (\phi_0(x_n), \phi_1(x_n), \dots, \phi_{M-1}(x_n))^T \in \mathbb{R}^M$.

This vector is the nth row of Φ. The error function becomes

$$\boxed{E_D(w) = \frac{1}{2}\sum_{n=1}^{N}(t_n - \hat{y}_n)^2 = \frac{1}{2}\|t - \hat{y}\|^2}$$

Now form $\hat{y} = (\hat{y}_1, \dots, \hat{y}_N)^T$ and notice that

$$(w_0, w_1, \dots, w_{M-1})\begin{pmatrix} \phi_0(x_1) & \phi_0(x_2) & \\ \phi_1(x_1) & \phi_1(x_2) & \dots \\ \vdots & \vdots & \\ \phi_{M-1}(x_1) & \phi_{M-1}(x_2) & \end{pmatrix}$$

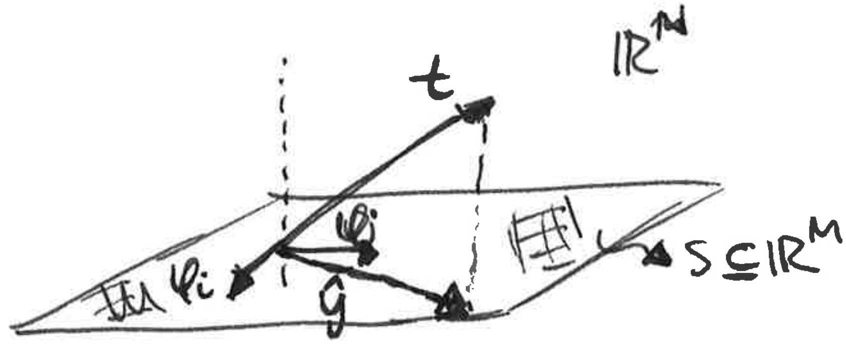
$$= (\hat{y}_1, \hat{y}_2, \dots, \hat{y}_N) \quad \therefore \quad \boxed{\hat{y}^T = w_{ML}^T \Phi^T}$$

or in other words $\boxed{\hat{y} = \Phi w_{ML}}$.

Explicitly: $\hat{y} = \left(\Phi\left(\Phi^T\Phi\right)^{-1}\Phi^T\right)t = Pt$

where $P^2 = P$ and $P = P^T$, so P is an orthogonal projector operator!

If $M < N$ there are at most M linear indep. $\{\varphi_j\}$ which live in a subspace S of dim $\leq M$.
Thus we have the following picture



So the ML solution picks the closest vector to t that lives in S.

In practice if $\Phi^T\Phi$ is singular (when $\varphi_i \parallel \varphi_j$ for some i,j) then $(\Phi^T\Phi)^{-1}$ is not defined.
When this matrix is close to singular the estimated parameters can be very large.
This can be addressed through SVD or regularization techniques.

Sequential Learning

Data points are considered one at a time (on-line algorithms). Usefull when data are streaming and predictions have to be made before all data as seen.

Stochastic gradient Descent: $E = \sum_n E_n$ (error).

$w^{(t+1)} = w^{(t)} - \eta \nabla E_n$, with some starting value $w^{(0)}$. In our case

$$\boxed{w^{(t+1)} = w^{(t)} + \eta \left(t_n - w^{(t)T} \phi(x_n)\right) \phi(x_n)}$$

Least-~~mean~~-square-error algorithm. η must be chosen with care.

Regularized Least Squares

Error function: $E_D(w) \to E_D(w) + \lambda E_W(w)$

This is with the goal of avoiding overfitting.

Sum-of-squares of weight vector:

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

The weight decay

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

encourages the weights $w_j \to 0$ unless supported by the data. In stats. this is known as parameter shrinkage.

Repeating the MLE the only difference is

$$-\left(\Phi^T t - (\Phi^T \Phi) w\right) + \lambda w = 0$$

$$\Phi^T t = (\Phi^T \Phi + \lambda I) w \quad \therefore$$

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

A more general regularized error func. is

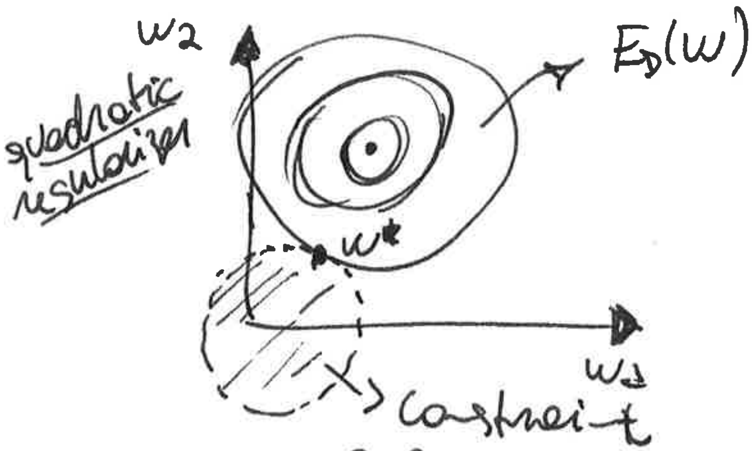
$$E(w) = \frac{1}{2} \sum_{n=1}^{N} (t_n - w^T \phi(x_n))^2 + \frac{\lambda}{2} \sum_{j=1}^{M} |w_j|^q$$

$q=1 \rightarrow$ Lasso. if $\lambda \gg 1$ the some $w_j \rightarrow 0$, leading to a sparse model in which the basis functions play no role.

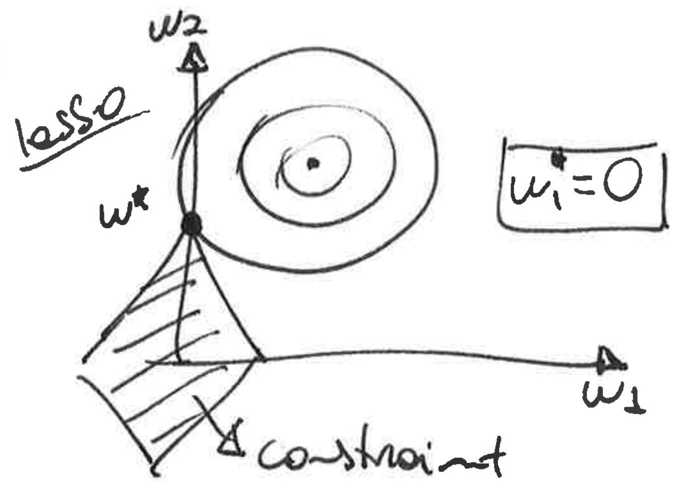
$$\min_w E(w) = \min \frac{1}{2} \sum_{n=1}^{N} (t_n - w^T \phi(x_n))^2$$

$$\text{s.t } \sum_{j=1}^{M} |w_j|^q \leq \eta$$

for some appropriate η.



$q=2$



$q=1$

Regularization allows complex models to be trained on small data sets without severe overfitting, since it limits the effective model complexity. The problem is shifted from finding $\{\phi_i(x)\}$ to finding optimal λ.

Multiple outputs

Before $t \in \mathbb{R}$. We may want $t = (t_1, \dots, t_K) \in \mathbb{R}^K$. We can introduce $\neq \{\phi_j(x)\}$ for each component of t, yielding several independent regression problems. However, it is common to use the same basis functions:

$$\boxed{y(x, W) = W^T \phi(x)}$$

Now a Matrix

$\begin{cases} t \in \mathbb{R}^K \\ y \in \mathbb{R}^K \\ W \in \mathbb{R}^{M \times K} \\ \phi \in \mathbb{R}^M \text{ as before} \end{cases}$

Take $p(t | x, W, \beta) = \mathcal{N}(t | W^T \phi(x), \beta^{-1} I)$

If we have $t_1, t_2, \dots, t_N$ observations we can form the Matrix

$$T = \begin{pmatrix} - t_1^T - \\ - t_2^T - \\ \vdots \\ - t_N^T - \end{pmatrix}_{N \times K}$$

$$\log p = \sum_{n=1}^{N} \log \mathcal{N}(t_n | W^T \phi(x_n), \beta^{-1} I)$$
$$= \frac{NK}{2} \log \frac{\beta}{2\pi} - \frac{\beta}{2} \sum_{n=1}^{N} \| t_n - W^T \phi(x_n) \|^2$$

log likelihood

$$X = \begin{pmatrix} - x_1^T - \\ - x_2^T - \\ \vdots \\ - x_N^T - \end{pmatrix}_{N \times \dim(x)}$$

As before we get

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

For each target ~~vector~~ variable t_k we have

$$\boxed{\vec{w}_k = (\Phi^T\Phi)^{-1}\Phi^T\vec{t}_k}$$

$\vec{t}_k \in \mathbb{R}^N$

$t_{nk} \to$ components.

only need to compute Φ^+ once!

Notice that using another covariance does not change this result which only depends on the mean.

Bias-Variance Decomposition

Maximum likelihood on small data can lead to severe over-fitting if models are complex. Limiting the complexity has the side effect of limiting the flexibility. This is the context of Bias-Variance tradeoff, and we need to find the optimally balance this two effects.

This issue does not arise when we marginalize over the parameters in a Bayesian Approach.

We want to estimate $t = y(x)$. Choose a loss function $L(y(x),t)$ and minimize its expectation

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

If $L = (y-t)^2$ we have

$$\delta L = L[y+\delta y] - L[y] = \frac{\partial L}{\partial y}\delta y = 2(y-t)\,\delta y$$

So $\frac{\delta \mathbb{E}[L]}{\delta y} = 0 = \int dt\, p(x,t)\,(y-t)$

$$\int dt\, p(x,t)\, y(x) = p(x)\, y(x)$$
$$= \int p(x,t)\, t\, dt = \int p(t|x)\, p(x)\, t\, dt$$
$$= p(x) \int p(t|x)\, dt$$

$$\therefore \boxed{y(x) = \mathbb{E}[t|x] \equiv h(x)}$$

using this back

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

$$\int dx\, dt\, p(x,t)\{(y(x)-h(x))(h(x)-t)\} = \int dx\,(y(x)-h(x)) \underbrace{\int dt\, p(x,t)(h(x)-t)}_{=0}$$

since $\int dt\, p(x,t)\, h(x) = h(x)\, p(x)$

$\int dt\, t\, p(x,t) = p(x)\, \mathbb{E}[t|x] = p(x)\, h(x)$

so for the square loss function

$$\mathbb{E}[L] = \underbrace{\int dx\, p(x)\{y(x)-h(x)\}^2}_{\text{this depends on the choice for } y(x).\ \text{This term is } \geq 0.} + \underbrace{\int dx\, dt\, p(x,t)\{h(x)-t\}^2}_{\text{does not depend on } y(x).\ \text{Intrinsic Noise in the data}}$$

Frequentist view: suppose we have data D drawn iid from $p(t,x)$. We model this obtaining $y(x;D)$. If we have an ensemble of data sets $\{D_j\}$ we obtain a different model for each data set $\{y(x;D_j)\}$. The performance is evaluated by averaging over the ensembles.

Expect. over Data

Consider
$$\{y(x;D) - h(x)\}^2 = \{y(x;D) - \mathbb{E}_D[y(x;D)] + \mathbb{E}_D[y(x;D)] - h(x)\}^2$$

The cross term

$$\mathbb{E}_D \; 2(y - \mathbb{E}_D y)(\mathbb{E}_D - h) = 0$$

Therefore

variance over ensemble

$$\mathbb{E}_D[\mathbb{E}[L]] = \int dx\, p(x)\, \mathbb{E}_D\big(y(x;D) - \mathbb{E}_D[y(x;D)]\big)^2$$
$$+ \int dx\, p(x)\, \mathbb{E}_D\big(y(x;D) - h(x)\big)^2 \quad \text{bias}^2$$
$$+ \int dx\, dt\, p(x,t)\, (h(x) - t)^2 \quad \text{Noise}$$

$$= \text{bias}^2 + \text{variance} + \text{Noise}$$

bias^2: Avg prediction differs from the desired regression function.

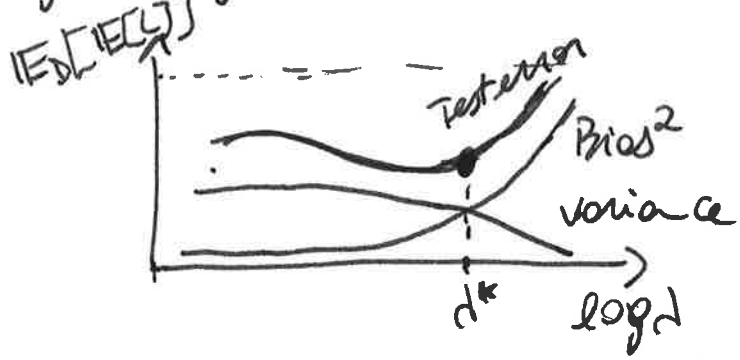
variance: variability of $y(x;D)$ over data sets.

Noise: irreducible error, intrinsic from data.

There is a tradeoff between variance and bias.

Flexible models: low bias, high variance
Rigid models: high bias, low variance

In a regularized regression with parameter λ one typically has



Suppose we have $\{D_1, \dots, D_L\}$ data sets.
Then we estimate $\{y^{(1)}, \dots, y^{(L)}\}$ models.
We then have

$$\bar{y}(x) = \frac{1}{L} \sum_{l=1}^{L} y^{(l)}(x)$$

$$(\text{bias})^2 = \frac{1}{N} \sum_{n=1}^{N} \left(\bar{y}(x_n) - h(x_n) \right)^2$$

$$\text{variance} = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{L} \sum_{l=1}^{L} \left(y^{(l)}(x_n) - \bar{y}(x_n) \right)^2$$

This is how we compute numerically.

- Limited Practical Value: because its based on averages of ensemble data sets. We could better just use the whole dataset as a training data to reduce overfitting.
In practice we usually don't have that much data.

Bayesian Linear Regression

Recall that we have the likelihood function

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

where $X = \{x_1, \dots, x_N\}$ is the data set, and $t = \{t_1, \dots, t_N\}$

$w = (w_0, \dots, w_{M-1})^T$ are the parameters, and $\phi(x) = (\phi_0(x), \dots, \phi_{M-1}(x))^T$ the basis functions.

Therefore,

$$p(t|X,w,\beta) \propto e^{-\frac{\beta}{2}\sum_{n=1}^{N}(t_n - w^T\phi_n)^2}$$

$$= e^{-\frac{\beta}{2}\sum_{n=1}^{N}(t_n^2 - 2w^T\phi_n t_n + (w^T\phi_n)^2)}$$

Now $\sum_{n=1}^{N} w^T\phi_n t_n = w^T \sum_{n=1}^{N} \phi_n t_n = w^T(\Phi^T t)$

$$\sum_{n=1}^{N}(w^T\phi_n)^2 = \sum_{n=1}^{N} w^T\phi_n \phi_n^T w = w^T\left(\sum_{n=1}^{N}\phi_n\phi_n^T\right)w$$

$$= w^T \Phi^T \Phi w$$

where $\Phi = \begin{pmatrix} \phi_0(x_1) & \cdots & \phi_{M-1}(x_N) \\ \vdots & \ddots & \vdots \\ \phi_{M-1}(x_1) & \cdots & \phi_{M-1}(x_N) \end{pmatrix}$

Thus $p(t|X,w,\beta) \propto e^{-\frac{\beta}{2}(t^Tt - 2w^T\Phi^T t + w^T\Phi^T\Phi w)}$

Since this is quadratic in w we choose a prior in the form

$$p(w) = \mathcal{N}(w \mid m_0, S_0) \propto e^{-\frac{1}{2}(w-m_0)^T S_0^{-1}(w-m_0)}$$ (94)

$$= e^{-\frac{1}{2}\{w^T S_0^{-1} w - w^T S_0^{-1} m_0 - m_0^T S_0^{-1} w + m_0^T S_0^{-1} m_0\}}$$

$$= e^{-\frac{1}{2}\{w^T S_0^{-1} w - 2 w^T S_0^{-1} m_0 + cte\}}$$

Therefore

$$p(t \mid w) . p(w) \propto p(w \mid t)$$

$$= \exp\left\{ -\frac{\beta}{2}\left(t^T t - 2 w^T \Phi^T t + w^T \Phi^T \Phi w\right) - \frac{1}{2}\left(w^T S_0^{-1} w\right) - 2 w^T S_0^{-1} m_0 + cte. \right\}$$

Considering the exponent:

$$-\frac{1}{2}\left\{ \beta t^T t + w^T\left(\beta \Phi^T \Phi + S_0^{-1}\right) w - 2 w^T\left(\beta \Phi^T t + S_0^{-1} m_0\right)\right\}$$

Now recall that for a general gaussian

$$\mathcal{N}(w \mid m_N, S_N) \propto e^{-\frac{1}{2}(w - m_N)^T S_N^{-1}(w - m_N)}$$

$$= e^{-\frac{1}{2}\{w^T S_N^{-1} w - 2 w^T S_N^{-1} m_N + m_N^T S_N^{-1} m_N\}}$$

Thus $$\boxed{S_N^{-1} = S_0^{-1} + \beta \Phi^T \Phi}$$

$$S_N^{-1} m_N = \beta \Phi^T t + S_0^{-1} m_0$$

$$\boxed{m_N = S_N\left(S_0^{-1} m_0 + \beta \Phi^T t\right)}$$

So the posterior has the form

$$\boxed{p(w|t) = \mathcal{N}(m_N, S_N)}$$

with m_N and S_N given as stated before. Since it is a gaussian the value w_{MAP} such that the posterior is maximum is $\boxed{w_{MAP} = m_N}$. If $S_0 = \alpha^{-1} I$ with $\alpha \to 0$ we have

$$S_N^{-1} \to \beta \Phi^T \Phi, \quad \boxed{m_N} \to S_N(\beta \Phi^T t)$$

$$= \cancel{\beta^{-1}} (\Phi^T \Phi)^{-1} \cancel{\beta} \Phi^T t$$

$$= (\Phi^T \Phi)^{-1} \Phi^T t = \boxed{w_{ML}}$$

So an infinitely broad prior implies that the mean of the posterior coincides with with the MLE for parameter w_{ML}.

From now on we consider a simplified prior:

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

Thus

$$p(w|t) = \mathcal{N}(m_N, S_N) \quad \text{(posterior)}$$

$$\begin{cases} m_N = \beta S_N \Phi^T t \\ S_N^{-1} = \alpha I + \beta \Phi^T \Phi \end{cases}$$

The log of the posterior as a function of w is

$$\log p(w|t) = -\frac{\beta}{2} \sum_{n=1}^{N} (t_n - w^T \Phi(x_n))^2 - \frac{\alpha}{2} w^T w$$

maximizing this is the same as minimizing the sum-of-squares with a regularization term with parameter $d = \frac{\alpha}{\beta}$.

Example. 1D. (x, t)

$$y(x, w) = w_0 + w_1 x \quad \text{linear model.}$$

generate data: $f(x, a) = a_0 + a_1 x$ $\begin{cases} a_0 = -0.13 \\ a_1 = 0.5 \end{cases}$

$x \sim \text{Unif}(-1, 1)$. Then add noise $\varepsilon = N(0, 0.2)$ to obtain t_n. The goal is to recover a_0, a_1, and study how things behave with increasing N.

As more data is seeing, the posterior becomes sharper and sharper. See the figure in the text.

We can consider other priors

$$p(w|\alpha) = \left(\frac{q}{2}\left(\frac{\alpha}{2}\right)\frac{1}{\Gamma(1/q)}\right)^M e^{-\frac{\alpha}{2}\sum_{j=1}^{M}|w_j|^q}$$

This would correspond to minimize

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

where $\lambda = \frac{\alpha}{\beta}$.

Predictive Distribution

We want to make predictions about t given a new data point x. This requires

$$p(t|\vec{t}, \alpha, \beta) = \int p(t|w, \beta)\, p(w|t, \alpha, \beta)\, dw$$

This is just the convolution of two Gaussians.

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

$p(w|t, \alpha, \beta) = p(w|\vec{t}) = N(w|m_N, S_N)$

Recall that given

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

we have the following marginal and posterior:

$$p(y) = \mathcal{N}(y|A\mu+b, L^{-1}+A\Lambda^{-1}A^T)$$
$$p(x|y) = \mathcal{N}(x|\Sigma\{A^TL(y-b)+\Lambda\mu\}, \Sigma)$$

where $\Sigma = (\Lambda + A^TLA)^{-1}$. Comparing to our case:

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

$$Ax+b \rightarrow y(x,w) = w^T\phi(x) = \underbrace{\phi(x)^T}_{A} w$$

$$L^{-1} \rightarrow \beta^{-1}$$

$$p(x) \rightarrow p(w|\vec{t}) = \mathcal{N}(w|m_N, S_N)$$

$$\mu \rightarrow m_N$$
$$\Lambda^{-1} \rightarrow S_N$$

$$A\mu+b \rightarrow \phi(x)^T m_N$$
$$L^{-1}+A\Lambda^{-1}A^T \rightarrow \beta^{-1} + \phi(x)^T S_N \phi(x)$$

Therefore $p(t|\vec{t},\alpha,\beta) = \mathcal{N}(t|\phi(x)^T m_N, \beta^{-1}+\phi(x)^T S_N \phi(x))$

The variance $\sigma_N^2 = \underbrace{\frac{1}{\beta}}_{\text{noise}} + \underbrace{\phi(x)^T S_N \phi(x)}_{\text{uncertainty about } w}$.

It can be shown that

$$\sigma_{N+1}^2 \leq \sigma_N^2$$

and $\phi(x)^T S_N \phi(x) \rightarrow 0$ as $N \rightarrow \infty$.

Kernel Interpretation

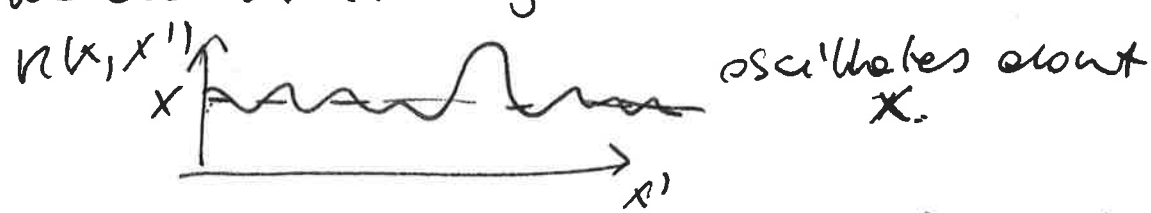
$y(x,w) = w^T\phi(x)$

posterior: $p(w|\vec{t}) = \mathcal{N}(w|m_N, S_N)$ where $m_N = \beta S_N \Phi^T \vec{t}$ and $S_N^{-1} = \alpha I + \beta\Phi^T\Phi$. Thus

$$y(x,m_N) = m_N^T\phi(x) = \phi(x)^T(\beta S_N \Phi^T t)$$
$$= \sum_{n=1}^{N} \beta\phi(x)^T S_N \phi(x_n)\, t_n$$
$$\equiv \sum_{n=1}^{N} k(x,x_n)\, t_n$$

where $k(x,x') = \beta\phi(x)^T S_N \phi(x')$ is the smoother matrix or equivalent kernel. Regression functions that takes linear combinations over the training targets are known as <u>linear</u> smoothers.

If one plots $k(x,x')$ as a function of x', for fixed x, we see something like



Thus the predictive mean $y(x,m_N)$ is obtained by a weighted linear combination of the target values, where weights are higher when close to x.

Consider $\text{cov}[y(x), y(x')] = \text{cov}[\phi(x)^T w, w^T\phi(x')]$
$$= \mathbb{E}[\phi(x)^T w w^T \phi(x')] - \mathbb{E}[\phi(x)^T w]\,\mathbb{E}[w^T\phi(x')]$$
$$= \phi(x)^T\mathbb{E}[ww^T]\phi(x') - \phi(x)^T\mathbb{E}[w]\,\mathbb{E}[w^T]\phi(x')$$

$P(w, \vec{t}) = \mathcal{N}(w \mid m_N, S_N)$. Thus

$$\mathbb{E}[w] = m_N$$

$$\mathbb{E}[ww^T] = \cancel{}\ m_N m_N^T + S_N$$

therefore $\operatorname{cov}[y(x), y(x')] = \phi(x)^T S_N \phi(x')$

$$= \beta^{-1} k(x, x')$$

The predictive mean at nearby points are highly correlated, and distant points not so much.

Thus, instead of introducing basis functions, we can introduce a kernel directly.

It can be