Linear Regression

Maximum Likelihood Estimation

- Given data(x) predict value (t)predict for new data -> model predictive distribution of p(t/x) so every distribution has a predictive distribution and we choose the appropriate distribution for the pbm.
- Since both weights and i/p are linear there are limitations. So extend by linear combinations of fixed nonlinear functions of i/p y(x,w) = w0+summation of w(j)Phi(x) Phi(x) is basis fn w0 is a bias parameter

Generalization of linear regression- replace each input with a function of that input.

- feature extraction- features are expressed in terms of basis functions Phi(x)
- (ex: robot body - arms egs body head - > basis functions / features)

- polynomial fns are global to remove this limitation in modeling use
- spline fns - different polynomial for each region of i/p space
- gaussian, μ_j location, 's'-spatial scale

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

- sigmoid,- space and
- fourier, sinusoidal fnc - specific frequency and infinite space
- wavelets - localized in both space and frequency

- likelihood fn- p(a/b)= p(b/a)p(a) / p(b)
- p(b/a) = likelihood function
- max likelihood fn = 'a' is set to value that maximizes p(b/a)

- Relation between Least Squares of Error and Maximum Likelihood:
- T=y(x,w)+*Epsilon (Epsilon is gaussian random variable with precision* β.
- p(t|x,w,β)=N(t|y(x,w),β−1)=√(β/2π)exp(−β/2(t−y *square of(x,w))

- Use N independent identically distributed observations x1,...xn with corresponding target functions t1...tn
- Jt conditional probability of t/X is
- $p(t|X,w,\beta)=\prod N(ti|w\phi(xi),inv(\beta))$ <- Likelihood Function
- $\log p(t|w,\beta)=(N/2)\log\beta-(N/2)\log 2\pi-\beta Ed(w)$
- Ed(w) is the sum of squares error function
- $Ed(w)=(½)\sum$square of$(ti-w\phi(xi))$ =(½)square of$\|t-\Phi w\|$
- $\phi(xi)$ is Design Matrix (N x M matrix)

- Maximizing log likelihood (= minimizing the sum-of-squares error function) w.r.t. W

-> maximum likelihood estimate of parameters w.

Maximum likelihood estimation ->over-fitting if complex models (e.g. polynomial regression models of high order) are fit to datasets of limited size.

Prevent over-fitting - add a regularization term to error function. ->

by a Bayesian approach (or a Gaussian approach)

- Sum of squares error function is -
- Ed(w) =½ Summ (square of {tn − wTϕ(xn)})

- Maximizing likelihood fn under Gaussian noise distribution for a linear model is equivalent to minimizing sum of squares error fn. given Ed(w)

- Gradient of log likelihood fn
- $\nabla \ln p(t|w, \beta) = \text{Summation}(t_n - w^T \phi(x_n) \phi(x_n)^T$.
- Set gradient to zero
- $0 = \text{Summation}(t_n \phi(x_n)^T - w^T(\text{Summation}(\phi(x_n)\phi(x_n)^T)$
- Solve for W
- $W_{ml} = inv(\Phi^T\Phi)\ \Phi^T t$ <-- Normal equation for least squares pbm (Wml is Weight for Maximum Likelihood)
- Φ is design matrix
- $\Phi^\dagger \equiv inv(\Phi^T\Phi)\ \Phi^T$ <-- Moore Penrose pseudo inverse of matrix Φ

- Bias compensates for difference between averages of target values and weighted sum of averages of basis function values

- Sequential Learning (Online Learning)
- Model updates after each data I/p
- Use Stochastic gradient descent
- $w(\tau+1) = w(\tau) - \eta \nabla En$
- For sum of squares error fn:
- $w(\tau+1) = w(\tau) + \eta(tn - w(\tau)T\phi n)\phi n$
- Above is Least Means Square (LMS) algorithm

Regularization

- Error function : Ed(w) + λEw(w) (Ed - data dependent error)
- Ew is regularization term and λ - control impact of this term
- Simple form of regularization – sum of squares of weight vector terms = Ew(w) =½*transpose(w)*w
- Add E(w) =½ Summation square of{tn − wTϕ(Xn)} to get
- ½ Summation square of {tn − wTϕ(xn)}2 + λ /2 * transpose(w)*w
- Above is the Weight decay / Parameter shrinkage

Regularization -2

- More generalized regularizer:
- }½*Summation square of {tn − wTϕ(xn)} + λ/2 *Summation|Wj| power of q
- If q=1 the function is called Lasso - > regularly used in Deep Learning
- In Lasso when λ is large enough, some of the coefficients of Wj become Zero – > Sparse model -> Basis fn plays no role
- => Avoid overfitting with correct value for regularization
- Generalized to multiple outputs and - solution decouples between different target variables
- Generalize to general Gaussian noise distributions

Bias Variance Decomposition

- Pbms of overfitting and limiting number of basis functions
- Need to determine correct value for λ - (regularization coefficient)

1. Bias Variance trade off
2. Handle overfitting by Bayesian approach (avoid maximum likelihood)

Assuming enough data sets, obtain different prediction fn for each data set. Take average of these functions (squared loss)

Expected squared loss -> E[L] =Integral(square of{y(x) − h(x)} p(x) dx + Integral (square of {h(x) − t}p(x, t) dx dt.

Term 2 represents Noise

Bias Variance decomposition -2

- (Bayesian perspective: posterior distribution over w)
- Average of squared loss over ensemble of data sets , ….
- $E_D\{y(x;D) - h(x)\}^2 = \{E_D[y(x;D)] - h(x)\}^2 + E_D\{y(x;D) - E_D[y(x;D)]\}^2$
- Square(Bias) Variance
- Expected squared difference between y and regression fn h = **squared Bias** (Error of the average prediction over all data sets + **Variance** (sol. varies around average –> sensitivity of y to choice of data)
- Small values of λ -> model become finely tuned to noise on each individual data set -> large variance and low bias
- Large value of λ -> weight parameters to zero -> low variance and large bias.

- Very flexible models <-> low Bias High Variance
- Relatively rigid models <-> High Bias Low Variance
- Need a balance

- Sol: weighted averaging of multiple solutions
- Used in Bayesian approach (Posterior distribution)
- Note: small values of λ -> model is finely tuned to noise on each individual data set -> to large variance
- While large value of λ -> weight parameters to zero -> large bias.

- Bias variance decomposition – uses average over large data sets - which are not available
- Bayesian Linear Regression:
- **Avoid overfitting & determine model complexity**
- Model complexity (# of Basis functions) ∞ size of data set
- Parameter distribution:

- Predictive distribution:

- Equivalent Kernel:

Parameter distribution:

- Prior probability distribution over model parameters 'w'
- (noise precision parameter is considered as a constant)
- P(t/w) is the exponential of a quadratic function of 'w'
- Conjugate prior of likelihood function (posterior and prior in same distribution) -> Gaussian distribution ->
- Normal distribution of (w/m0, S0) m0 – mean, S0 – covariance
- Calculate posterior distribution - ∞ (likelihood fn *prior)
- Evaluate this by completing square in exponential and find normalization co-efficient (using standard result for normalized Gaussian)

- P(w/t) = Normalized fn(w/m_n, S_n)
- Where $M_n = S_n(S_0^{-1}m_0+\beta(Phi)^T t$
- $S_n^{-1}= S_0^{-1}+\beta(Phi)^T Phi$
- Max posterior Weight vector = $w_{map}=m_N$
- Infinitely broad prior then mean of posterior distribution = w_{ml}
- If N=0 then posterior = prior
- If data points arrive sequentially then posterior = prior for subsequent data points

Consider Zero Mean Isotropic Gaussian with a single precision parameter α

=>

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

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

where

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

Gaussian => mode =mean => Maximum Posterior weight vector $\mathbf{w}_{\mathrm{MAP}} = \mathbf{m}_N$

Log of posterior distribution = Sum of log likelihood and log of prior

$$\ln p(\mathbf{w}|\mathbf{t}) = -\frac{\beta}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n)\}^2 - \frac{\alpha}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + \mathrm{const.}$$

Maximization of posterior distribution wrt w <=>

Linear Models for Regression

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

Predictive Distribution

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

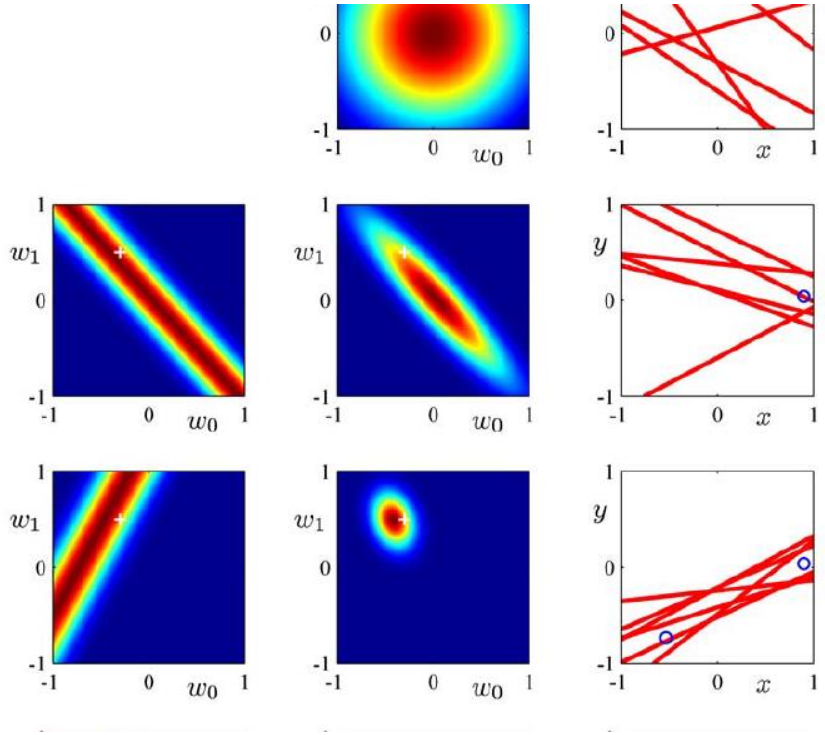
$$p(t|\mathbf{x},\mathbf{t},\alpha,\beta) = \mathcal{N}(t|\mathbf{m}_N^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}),\sigma_N^2(\mathbf{x}))$$

where the variance $\sigma_N^2(\mathbf{x})$ of the predictive distribution is given by

$$\sigma_N^2(\mathbf{x}) = \frac{1}{\beta} + \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}}\mathbf{S}_N\boldsymbol{\phi}(\mathbf{x}).$$

Illustration: Bayesian Linear regression

- Input variable X and target T . Linear model-> $y(x, \mathbf{w}) = w_0 + w_1 x.$
- Only 2 adaptive parameters =>
- Generate synthetic data using $f(x,a) = a_0 + a_1 x$
- Values –> $a_0 = -0.3$ $a_1 = 0.5$ by choosing x_n values from Uniform distribution $U(x|-1,1)$, evaluate $f(x_n,a)$ Add Gaussian Noise with Std. deviation of 0.2 to get t_n
- Objective: Recover values of a_0 and a_1 and study effect of size of data set
- Set Noise $\beta = 25$ and α to 2.0



- **Likelihood Prior/ Posterior Data Space**
- 1st row – initial – Prior distribution (6 samples)
- 2nd row – single data point – rt column – data. Lt column – plot of p(t|x,w)
- Sequential nature of Bayesian learning - Current posterior distribution becomes prior when new data is added

Predictive Distribution

- This is the real interest - predict t for new values of x

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

- Conditional distribution of t & posterior weight distribution
- 2 Gaussian distributions => Predictive distribution is

$$p(t|\mathbf{x}, \mathbf{t}, \alpha, \beta) = \mathcal{N}(t|\mathbf{m}_N^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}), \sigma_N^2(\mathbf{x}))$$

- Variance of predictive distribution is

$$\sigma_N^2(\mathbf{x}) = \frac{1}{\beta} + \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}} \mathbf{S}_N \boldsymbol{\phi}(\mathbf{x}).$$

- 1/β / in Parameter 'w'
- With more sample data posterior distribution gets narrower => Variance becomes lesser. Uncertainty in 'w' become zero and only Noise results in a Variance
- Pbm: Areas away from basis function centers also only noise as the predicted variance value
- Sol: Adopt a Gaussian approach instead of Regression (pl refer book)

Equivalent Kernel

- Kernel methods use

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

- => Predictive mean:

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

- becomes:

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

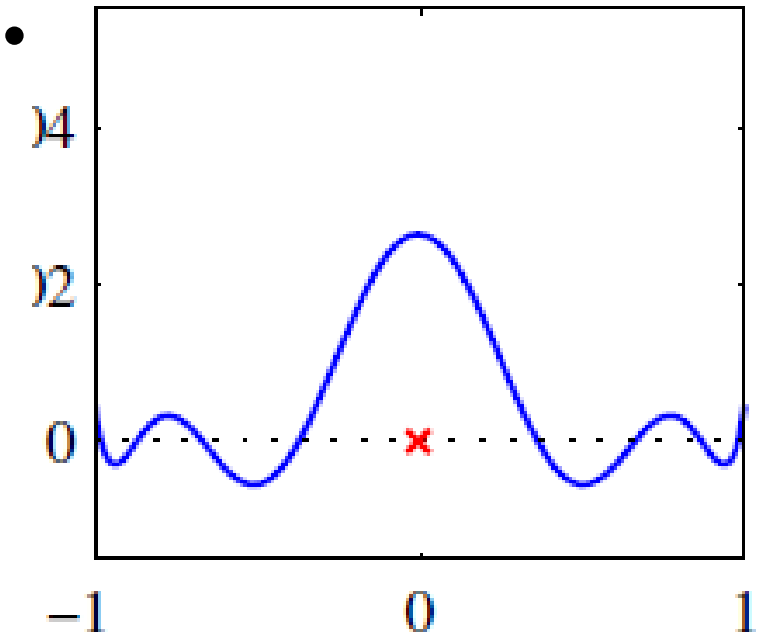
- Predictive distribution at point x is a linear combination of target variables

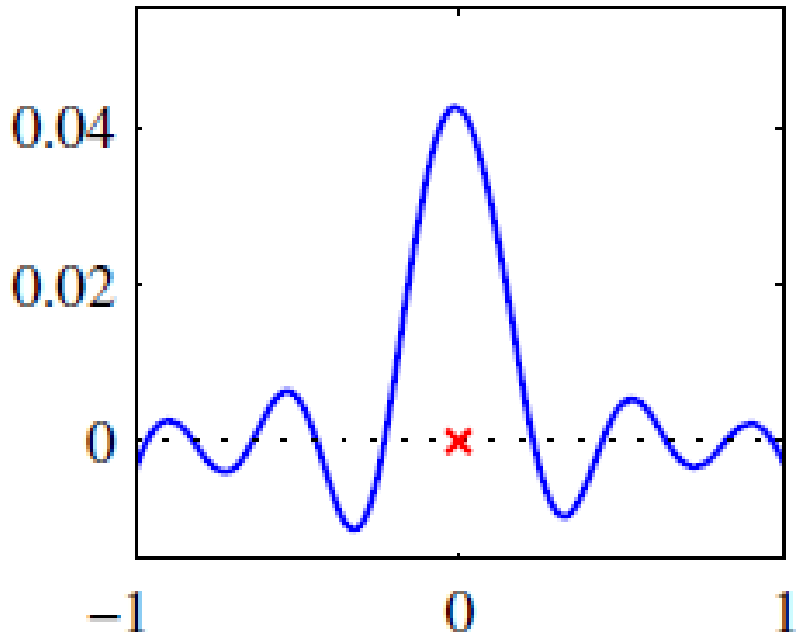
$$y(\mathbf{x}, \mathbf{m}_N) = \sum_{n=1}^{N} k(\mathbf{x}, \mathbf{x}_n) t_n$$

$$k(\mathbf{x}, \mathbf{x}') = \beta \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}} \mathbf{S}_N \boldsymbol{\phi}(\mathbf{x}')$$

- Is called "smoother matrix" or "equivalent kernel"
- Linear Smoothers: Linear combinations of training sets

- Visualization of equivalent kernel for Gaussian Basis functions:
- K(x,x') plotted as a function of x for three values => localized around x
- => Mean of predictive distribution function -> gives more weight to data points close to 'x'
-





- Similar inference from covariance between y(x) and y(x')

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

- Predictive means for nearby points is highly correlated
- **Localized Kernel:**
- Instead of using a set of basis functions *(alternative to model non-linear components)* , we can define a Localized Kernel and use this to make predictions <= Gaussian Process
- Kernel defines weights (weights sum to 'One')

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

- The summation is = predictive means for target data where t_n=1
- Requirement :
- Basis functions are linearly independent,
- More data points than basis functions
- One Basis point is constant (Bias)
- Then => fit training data exactly

- Also $k(\mathbf{x}, \mathbf{x}') = \beta\boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}}\mathbf{S}_N\boldsymbol{\phi}(\mathbf{x}')$
- Equivalent to inner product w.r.t vector $\boldsymbol{\psi}(\mathbf{x})$ of non linear fn

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

- Where

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

Bayesian Model Comparison

- Overfitting due to Maximum Likelihood, approach was to make point estimates of their values.
- Avoid this by marginalizing over model parameters (choosing between alternative models)
- Models can be compared on training data without validation data
- => Avoids cross -validation runs
- (Also allows learning of multiple complexity parameters – Example: Relevance vector machine which has a complexity parameter for every training point)
- Bayesian Model Comparison: Use probabilities to represent uncertainty between models.
- Each model is a probability distribution (that generates the data)

- Uncertainty -> Prior probability distribution $p(M_i)$
- Need to find $p(\mathcal{M}_i|\mathcal{D}) \propto p(\mathcal{M}_i)p(\mathcal{D}|\mathcal{M}_i).$
- Model evidence -> $p(\mathcal{D}|\mathcal{M}_i)$ Marginal Likelihood - > Likelihood function over models -> preference of data for a model
- Bayes factor = Ratio of Model Evidence - $p(\mathcal{D}|\mathcal{M}_i)/p(\mathcal{D}|\mathcal{M}_j)$

Summary

- regression: predict value of continuous target variables t given value of D dimensional vector x of input variables
- Simplest example is a polynomial curve fitting - example of linear regression models
- simplest form - linear functions of input variables
- more useful form - linear combination of a fixed set of non linear functions of input variables (basis functions)
- -> nonlinear wrt to input variables but linear functions of the parameters

- Given N observations Xn, with corresponding target values Tn , goal is to predict T for a new value of X => Model the predictive distribution p(t/x) as the uncertainty is also modeled
- Uncertainty => error or loss function. Minimize Loss function of the model is the goal
- (Squared loss - Optimal solution is the conditional expectation of t)
- Linear models is foundations for more sophisticated models . They cannot handle problems involving higher dimensions)

- Linear Basis function models:

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

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x})$$ <= Dummy basis fn $\phi_0(\mathbf{x}) = 1$

- Basis function $\phi_j(\mathbf{x})$
- Bias parameter (fixed offset) w_0
- When vector 'x' are the original variables then features are in terms of basis function $\phi_j(\mathbf{x})$
- Maximum Likelihood
- With noise modeled by Gaussian random variable with precision β.
- $p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}), \beta^{-1})$. Where $t = y(\mathbf{x}, \mathbf{w}) + \epsilon$

- For a squared loss function the optimal prediction is conditional mean of the target variable.
- For Gaussian conditional distribution conditional mean (optimal prediction) =

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

- Assuming independence of data points the likelihood function =

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

- Logarithm of likelihood fn

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

- Sum of squares error function is

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

- From this likelihood function we use Maximum Likelihood to calculate w and β

1) Maximization wrt 'w'

Maximizing likelihood function under Gaussian distribution <=> Minimize sum of errors function

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

- Set gradient to zero will give

$$0 = \sum_{n=1}^{N} t_n \phi(\mathbf{x}_n)^{\mathrm{T}} - \mathbf{w}^{\mathrm{T}} \left(\sum_{n=1}^{N} \phi(\mathbf{x}_n) \phi(\mathbf{x}_n)^{\mathrm{T}} \right)$$

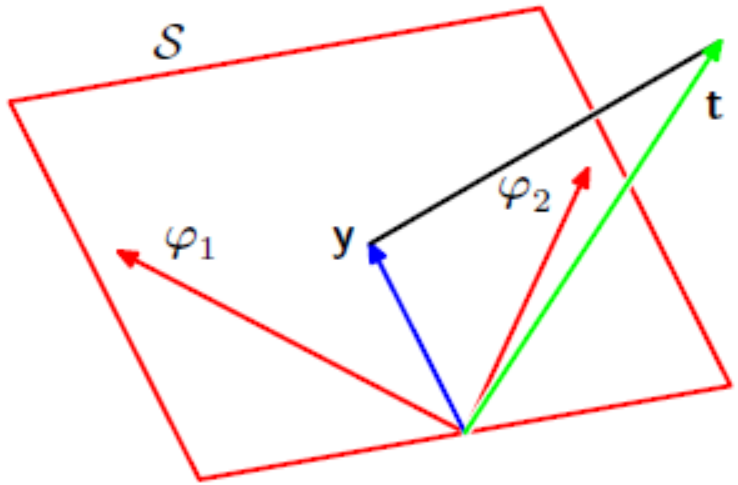
- Solve for 'w' to get $\mathbf{w}_{\mathrm{ML}} = \left(\mathbf{\Phi}^{\mathrm{T}} \mathbf{\Phi} \right)^{-1} \mathbf{\Phi}^{\mathrm{T}} \mathbf{t}$

- Called normal equations for least squares pbm
- NxM matrix called Design matrix

$\mathbf{\Phi}$

$$\mathbf{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}.$$

- The solution to regression problem decouples between the different target variables, => only compute a single pseudo-inverse matrix $\Phi^{\dagger}$, which is shared by all of the vectors w_k
- ---
- The least-squares regression function: find orthogonal projection of the data vector t onto the subspace spanned by the basis functions $\phi_j(x)$ in which each basis function is viewed as a vector
- ϕ_j of length N with elements $\phi_j(x_n)$.
- Φ_j corresponds to the jth column of Φ
- $\phi(x_n)$ corresponds to the nth row of Φ



- If number M of basis functions < N of data points then M vectors $\phi_j(x_n)$ will span a linear subspace S of dimensionality M.
- Take a N dimensional vector 'Y' whose n^{th} element is $y(x_n,w)$
- Sum of squares error = Squared euclidean distance between 'y' and 't'
- Y is to get as close to t as possible
- Solution : orthogonal projection of t on subspace S

- <u>Online learning / sequential learning:</u>
- Stochastic gradient descent: $w^{(\tau+1)} = w^{(\tau)} - \eta \nabla En$
- Update parameter w by $w^{(\tau+1)} = w^{(\tau)} + \eta(t_n - w^{(\tau)T}\phi n)\phi n$ <-- LMS algorithm

- Regularized least squares:
- Error function - $E_D(w) + \lambda E_W(w)$ (E_D – data dependent error E_W – regularization term)
- Simple regularizer: $E_W(w) = \frac{1}{2} (w^T w)$ (sum of squares of weigh vector elements)
- Using sum of errors function $E(w) = \frac{1}{2} \Sigma \{t_n - w^T \phi(x_n)\}^2$
- Total error function = $\frac{1}{2} \Sigma \{t_n - w^T \phi(x_n)\}^2 + \lambda/2\, w^T w$
- Above is Weight Decay (parameter shrinkage)

- Generalized form : $\frac{1}{2} \Sigma \{t_n - w^T \phi(x_n)\}^2 + \lambda/2 |wj|^q$
- q=2 gives above . q=1 – lasso . In this when λ is large enough, coefficients of w_j become zero-> sparse model where basis functions have no relevance

Bias – Variance decomposition

- Bias –Variance trade off

- Pbm. of Overfitting when complex models are trained with limited data
- Sol: Introduce regularization parameter. But this only pushes problem to finding correct co-efficient for regularization term
- Given optimal prediction of squared loss h(x), = $E[t|x] = \int t\, p(t|x)\, dt$)
- Expected squared loss
- $E[L] = \int \{y(x) - h(x)\}^2\, p(x)\, dx + \int \{h(x) - t\}^2 p(x, t)\, dx\, dt$
- *Function y(x)* *Noise*

- Frequentist approach (normal probability approach) -> Making point estimate based on data set D.
- Use Multiple data sets. Learn prediction for each data set -> y(x:D)
- Different data sets give different functions and different values of "Squared Loss"
- Take average over this ensemble of data sets

- For a particular data set the Integral over the Function term in Expected squared loss equation becomes -> $\{y(x;D) - h(x)\}^2$
- Take average over ensemble .
- Add and subtract ED[y(x;D)] gives
- $\{y(x;D) - ED[y(x;D)] + ED[y(x;D)] - h(x)\}^2 =$
- $\{y(x;D) - ED[y(x;D)]\}^2 + \{ED[y(x;D)] - h(x)\}^2 + 2\{y(x;D) - ED[y(x;D)]\}\{ED[y(x;D)] - h(x)\}$.

- Take expectation wrt D gives
- Expected squared difference between y(x;D) and regression fn h(x) is
- $E_D \{y(x;D) - h(x)\}^2 = \{E_D[y(x;D)] - h(x)\}^2 + E_D[\{y(x;D) - E_D[y(x;D)]\}]^2$
- $(\text{bias})^2$ variance
- Squared bias -> represents how average prediction differs from correct regression function
- Variance -> measures variance of individual data sets across their average --> tells us how sensitive y(x;D) is to the particular data set
- Generalizing for the expected squared loss gives
- Expected loss = $(\text{bias})^2$ + variance + noise
- $(\text{bias})^2 = \{E_D[y(x;D)] - h(x)\}^2 p(x)\, dx$
- variance = $E_D \{y(x;D) - E_D[y(x;D)]\}^2 \; p(x)\, dx$
- noise = $\{h(x) - t\}^2 p(x, t)\, dx\, dt$

- Bias and variance are together creating a complexity. One needs large number of data sets to average over. If this were available then overfitting would not happen
- Different approach: As Overfitting happens due to Maximum Likelihood, try approach of Bayesian setting

Bayesian Linear Regression

- Avoids over-fitting problem
- Parametric distribution (w value)
- Likelihood function P(t/w) is exponential of a quadratic function over w
- $p(t|X,w,\beta) = \pi$ Normal distribution $(t_n|w^T\phi(xn), \beta^{-1})$
- Conjugate prior: $p(w) = N(w|m_0, S_0)$ m_0- mean S_0- Covariance
- Pc $p(\mathbf{w}|\mathbf{t}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N)$ prior * likelihood function)

$$\mathbf{m}_N = \mathbf{S}_N \left(\mathbf{S}_0^{-1}\mathbf{m}_0 + \beta\mathbf{\Phi}^{\mathrm{T}}\mathbf{t}\right) \qquad \mathbf{S}_N^{-1} = \mathbf{S}_0^{-1} + \beta\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}$$

- Considering a zero-mean isotropic Gaussian with a single precision parameter α :

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

- Posterior distribution over w is $p(\mathbf{w}|\mathbf{t}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N)$
- With

$$\mathbf{m}_N = \beta \mathbf{S}_N \mathbf{\Phi}^{\mathrm{T}} \mathbf{t} \qquad \mathbf{S}_N^{-1} = \alpha \mathbf{I} + \beta \mathbf{\Phi}^{\mathrm{T}} \mathbf{\Phi}$$

- Taking log: $\ln p(\mathbf{w}|\mathbf{t}) = -\frac{\beta}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 - \frac{\alpha}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + \text{const.}$

- Maximizing above is similar to minimizing sum of squares error function

- Predictive distribution (make predictions of t for new values of x)
- Evaluate predictive distribution $$p(t|\mathbf{t},\alpha,\beta) = \int p(t|\mathbf{w},\beta)p(\mathbf{w}|\mathbf{t},\alpha,\beta)\,\mathrm{d}\mathbf{w}$$

$$p(t|\mathbf{x},\mathbf{t},\alpha,\beta) = \mathcal{N}(t|\mathbf{m}_N^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}),\sigma_N^2(\mathbf{x}))$$

- Where variance is $$\sigma_N^2(\mathbf{x}) = \frac{1}{\beta} + \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}}\mathbf{S}_N\boldsymbol{\phi}(\mathbf{x})$$

- 1/β represents noise. 2nd term represents uncertainty
- With additional data points posterior distribution becomes narrower, so variance in uncertainty become less
- So variance in predictive distribution only depends on noise (β)
- Pbm: model becomes incorrectly confident outside the basis function also.

- Posterior distribution

$$\mathbf{m}_N = \beta \mathbf{S}_N \mathbf{\Phi}^{\mathrm{T}} \mathbf{t} \qquad \mathbf{S}_N^{-1} = \alpha \mathbf{I} + \beta \mathbf{\Phi}^{\mathrm{T}} \mathbf{\Phi}$$

- Substitute this in

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

- Then predictive mean =

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

- Mean of predictive distribution: linear combination of training set variables

- Represent $$k(\mathbf{x}, \mathbf{x}') = \beta\boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}}\mathbf{S}_N\boldsymbol{\phi}(\mathbf{x}')$$
- Then predictive mean $$y(\mathbf{x}, \mathbf{m}_N) = \sum_{n=1}^{N} k(\mathbf{x}, \mathbf{x}_n)t_n$$
- $k(\mathbf{x}, \mathbf{x}')$ Is called "smoother matrix" or "equivalent kernel"
- *Linear smoothers:* Regression functions that make predictions using a linear combinations of training set target values
- Equivalent kernel: similarity measure between new data point and observed evidence, weighted with model parameters
- Equivalent kernel gives more weightage to data points close to 'x'

- Similarly predictive means for nearby points will be more correlated than for distant points

- Basis functions implicitly determine an equivalent kernel.
- Now we define the a localized kernel and use this to make predictions. <-- Gaussian processes
- Kernel defines weights used to combine training set target values when we make a prediction

Bayesian Model comparison

- Context: Validating results ("cross-validation")
- Another approach: Model selection using Bayesian view
- Avoid overfitting due to Maximum Likelihood (making point estimates of values) by Marginalizing (sum or integrate) over model parameters
- Model compare on all data (no need for validation set)
- Model M_i is a probability distribution over observed data D

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

- Model evidence: $p(\mathcal{D}|\mathcal{M}_i)$
- Preference shown by data for different models. Also called "Marginal Likelihood"

- Bayes Factor= Ratio of model evidence $p(\mathcal{D}|\mathcal{M}_i)/p(\mathcal{D}|\mathcal{M}_j)$

- Predictive distribution over models is $p(t|\mathbf{x},\mathcal{D}) = \sum_{i=1}^{L} p(t|\mathbf{x},\mathcal{M}_i,\mathcal{D})p(\mathcal{M}_i|\mathcal{D}).$

- Mixture distribution: average over predictive distribution of individual models weighted by posterior probabilities of these models

- Approximation to model averaging is use single most probable model alone <-- Model selection

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