Machine Learning

Lecture 4,5,6,7

Indian Institute of Information Technology Dharwad

Decision Theory

- Using probability theory to make optimal decisions
- Input vector x, target vector t
  - *Regression:* t is continuous
  - *Classification:* t will consist of class labels
- Summary of uncertainty associated is given by $p(\mathrm{x},\mathrm{t})$
- *Inference problem* is to obtain $p(\mathrm{x},\mathrm{t})$ from data
- *Decision:* make specific prediction for value of t and take specific actions based on t

Medical Diagnosis Problem

- X-ray image of patient
- Whether patient has cancer or not
- Input vector $\mathbf{x}$ is set of pixel intensities
- Output variable t represents whether cancer or not C_1 is cancer and C_2 is absence of cancer
- General inference problem is to determine $p(x,C_k)$ which gives most complete description of situation
- In the end we need to decide whether to give treatment or not. Decision theory helps do this

Bayes Decision

- How do probabilities play a role in making a decision?
- Given input x and classes C_k using Bayes theorem

$$p(C_k \mid \mathrm{x}) = \frac{p(\mathrm{x} \mid C_k)\, p(C_k)}{p(\mathrm{x})}$$

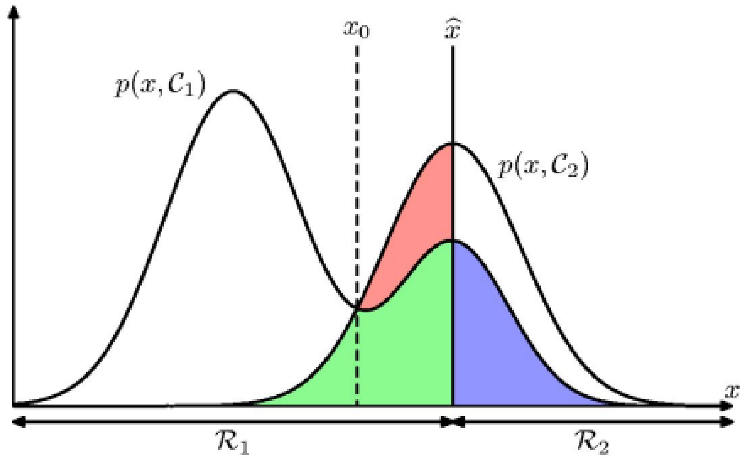
- Quantities in Bayes theorem can be obtained from $p(x,C_k)$ either by marginalizing or conditioning wrt appropriate variable

Minimizing Expected Error

- Probability of mistake (2-class)

$$P(error) = p(x\ \varepsilon\ R_1, C_2) + p(x\ \varepsilon\ R_2, C_1)$$
$$= \int_{R_1} p(\mathrm{x}, C_2)\,d\mathrm{x} + \int_{R_2} p(\mathrm{x}, C_1)\,d\mathrm{x}$$

- Minimum error decision rule
  - For a given x choose class for which integrand is smaller
  - Since $p(\mathrm{x},C_k)=p(C_k|\mathrm{x})p(\mathrm{x})$, choose class for which *a posteriori* probability is highest
  - Called Bayes Classifier



Single input variable x

If priors are equal, decision is based on class-conditional densities $p(x|C_k)$

Minimizing Expected Loss

- Unequal importance of mistakes
- Medical Diagnosis
- Loss or Cost Function given by Loss Matrix
- Utility is negative of Loss
- Minimize Average Loss

$$E[L] = \sum_k \sum_j \int_{R_j} L_{kj}\, p(\mathrm{x}, C_k)\, d\mathrm{x}$$

- Minimum Loss Decision Rule
  - Choose class for which $\sum_k L_{kj}\, p(C_k \mid \mathrm{x})$ is minimum
  - Trivial once we know a posteriori probabilities

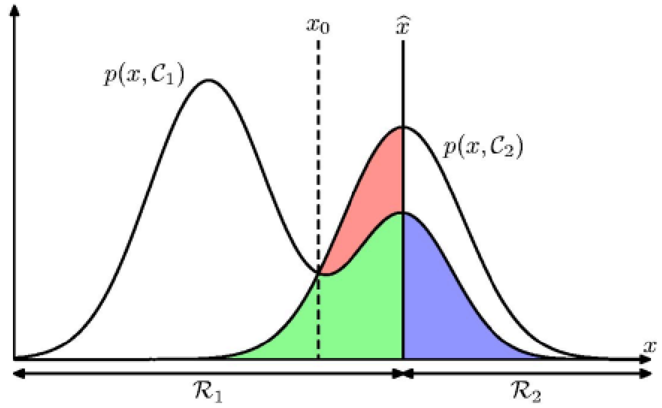
Loss Function for Cancer Decision

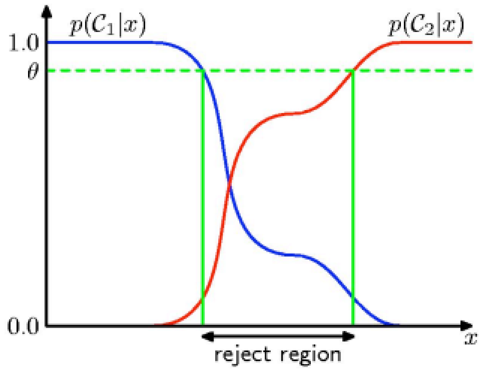
Decision Made

$$\text{True Class}\quad \begin{array}{c c} & \begin{array}{cc} \textit{cancer} & \textit{normal} \end{array} \\ \begin{array}{c} \textit{cancer} \\ \textit{normal} \end{array} & \begin{pmatrix} 0 & 1000 \\ 1 & 0 \end{pmatrix} \end{array}$$

Reject Option

- Decisions can be made when *a posteriori* probabilities are significantly less than unity or joint probabilities have comparable values
- Avoid making decisions on difficult cases





Inference and Decision

- Classification problem broken into two separate stages
  - Inference, where training data is used to learn a model for $p(C_k,x)$
  - Decision, use posterior probabilities to make optimal class assignments
- Alternatively can learn a function that maps inputs directly into labels
- Three distinct approaches to Decision Problems
  1. Generative
  2. Discriminative
  3. Discriminant Function

1. Generative Models

- First solve inference problem of determining class-conditional densities $p(\mathrm{x}|C_k)$ for each class separately
- Then use Bayes theorem to determine posterior probabilities

$$p(C_k \mid \mathrm{x}) = \frac{p(\mathrm{x} \mid C_k)\, p(C_k)}{p(\mathrm{x})}$$

- Then use decision theory to determine class membership

2. Discriminative Models

- First solve inference problem to determine posterior class probabilities $p(C_k|x)$
- Use decision theory to determine class membership

3. Discriminant Functions

- Find a function $f(\mathrm{x})$ that maps each input $\mathbf{x}$ directly to class label
  - In two-class problem, $f(.)$ is binary valued
    - $f=0$ represents class C_1 and $f=1$ represents class C_2
- Probabilities play no role
  - No access to posterior probabilities $p(C_k|\mathrm{x})$

Need for Posterior Probabilities

- Minimizing risk
  - Loss matrix may be revised periodically as in a financial application
- Reject option
  - Minimize misclassification rate, or expected loss for a given fraction of rejected points
- Compensating for class priors
  - When far more samples from one class compared to another, we use a balanced data set (otherwise we may have 99.9% accuracy always classifying into one class)
  - Take posterior probabilities from balanced data set, divide by class fractions in the data set and multiply by class fractions in population to which the model is applied
  - Cannot be done if posterior probabilities are unavailable
- Combining models
  - X-ray images (x_I) and Blood tests (x_B)
  - When posterior probabilities are available they can be combined using rules of probability
  - Assume feature independence $p(\mathrm{x}_I, \mathrm{x}_B|C_k) = p(\mathrm{x}_I|C_k)\, p(\mathrm{x}_B|C_k)$ [Naïve Bayes Assumption]
  - Then

$$
\begin{aligned}
p(C_k|\mathrm{x}_I, \mathrm{x}_B) &\propto p(\mathrm{x}_I, \mathrm{x}_B|C_k)p(C_k) \\
&\propto p(\mathrm{x}_I|C_k)\, p(\mathrm{x}_B|C_k)\, p(C_k) \\
&\propto p(C_k|\mathrm{x}_I)\, p(C_k|\mathrm{x}_B)/p(C_k)
\end{aligned}
$$

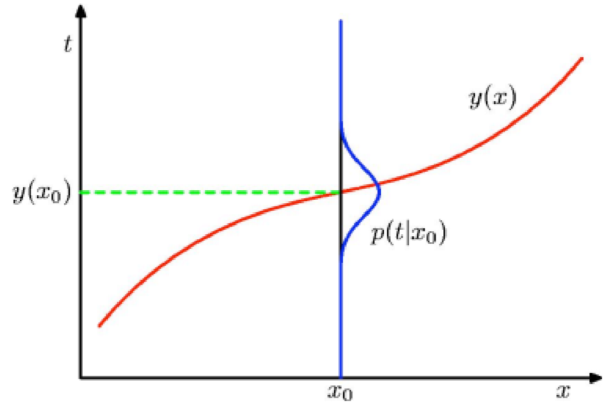
  - Need $p(C_k)$ which can be determined from fraction of data points in each class. Then need to normalize resulting probabilities to sum to one

Loss Functions for Regression

- Curve fitting can also use a loss function
- Regression decision is to choose a specific estimate $y(\mathrm{x})$ of t for a given x
- Incur loss $L(t,y(\mathrm{x}))$
- Squared loss function $L(t,y(x))=\{y(x)-t\}^2$
- Minimize expected loss

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

Taking derivative and setting equal to zero yields a solution $y(x)=E_t[t|x]$



Regression function $y(x)$, which minimizes the expected squared loss, is given by the mean of the conditional distribution $p(t|x)$

Loss function for Regression

$$EPE(f) = E\left[\left(Y - f(X)\right)^2\right]$$

$$= \int \int [y - f(x)]^2 \Pr(y, x) dydx$$

$$= \int_x \int_y [y - f(x)]^2 \Pr(y|x) \Pr(x) \, dydx$$

$$\Pr(X, Y) = \Pr(Y|X) \Pr(X)$$

$$EPE(f) = E_X E_{Y|X}([Y - f(X)]^2 | X).$$

Loss function for Regression

Notice that by conditioning on X, we have freed the dependency of the function f on X and since the quantity $[Y - f]^2$ is convex, there is a unique solution. We can now minimize to solve for f

$$f(x) = \arg\min_f E_{Y|X}([Y - f]^2 | X = x)$$

$$\Rightarrow \frac{\partial}{\partial f} \int [Y - f]^2 \,\mathrm{Pr}(y|x)\, dy = 0$$

$$= \int \frac{\partial}{\partial f} [y - f]^2 \,\mathrm{Pr}(y|X)\, dy = 0$$

$$= 2\int yPr(y|x)dy = 2f \int \quad Pr(y|x)\, dy = 0$$

$$\Rightarrow 2E[Y|X] = 2f$$

$$\Rightarrow f = E[Y|X = x].$$

Inference and Decision for Regression

- Three distinct approaches (decreasing complexity)
- Analogous to those for classifiction

1. Determine joint density $p(\mathrm{x},t)$

   Then normalize to find conditional density $p(t|\mathrm{x})$

   Finally marginalize to find conditional mean $E_t[t|\mathrm{x}]$
2. Solve inference problem of determining conditional density $p(t|\mathrm{x})$

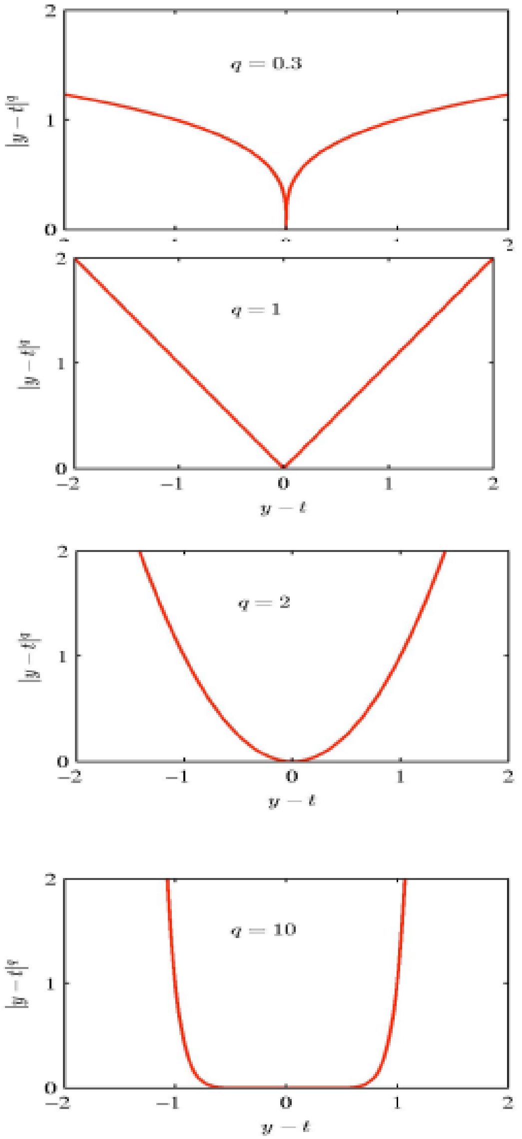
   Marginalize to find conditional mean
3. Find regression function $y(\mathrm{x})$ directly from training data

Minkowski Loss Function

- Squared Loss is not only possible choice for regression
- Important example concerns multimodal $p(t|\mathbf{x})$
- Minkowski Loss

$$\mathbb{E}[L_q] = \iint |y(\mathbf{x}) - t|^q p(\mathbf{x}, t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t$$

- Minimum of $\mathbb{E}[L_q]$ is given by
  - conditional mean for $q=2$,
  - conditional median for $q=1$ and
  - conditional mode for $q \rightarrow 0$



Linear Regression with Basis Function

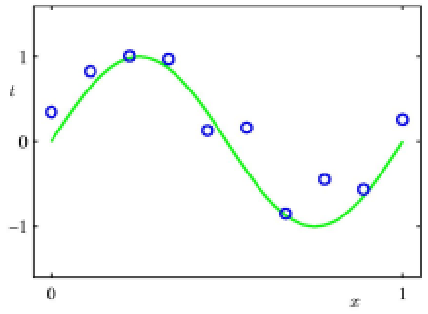
The regression task

- It is a supervised learning task
- Goal of regression:
  - predict value of one or more target variables t
  - given d-dimensional vector x of input variables
  - With dataset of known inputs and outputs
    - $(\mathrm{x}_1, \mathrm{t}_1), ..(\mathrm{x}_N, \mathrm{t}_N)$
    - Where x_i is an input (possibly a vector) known as the predictor
    - t_i is the target output (or response) for case i which is real-valued
  - Goal is to predict t from x for some future test case
    - We are not trying to model the distribution of x
  - We dont expect predictor to be a linear function of x
    - So ordinary linear regression of inputs will not work
    - We need to allow for a nonlinear function of x
    - We don't have a theory of what form this function to take

ML Terminology

- Regression
  - Predict a numerical value t given some input
    - Learning algorithm has to output function $f : \mathrm{R}^n \rightarrow \mathrm{R}$
      - where n =no of input variables
- Classification
  - If t value is a label (categories): $f : \mathrm{R}^n \rightarrow \{1,..,k\}$
- Ordinal Regression
  - Discrete values, ordered categories

Polynomial Curve Fitting with a Scalar



Training data set
$N=10$, Input x, target t

– With a single input variable x

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

M is the order of the polynomial,

x^j denotes x raised to the power j,

Coefficients $w_0,\ldots,w_M$ are collectively denoted by vector $\mathbf{w}$

– Task: Learn $\mathbf{w}$ from training data $D = \{(x_i, t_i)\}$, $i = 1,..,N$

- Can be done by minimizing an error function that minimizes the misfit between $y(x,\mathbf{w})$ for any given $\mathbf{w}$ and training data
- One simple choice of error function is sum of squares of error between predictions $y(x_n,\mathrm{w})$ for each data point x_n and corresponding target values t_n so that we minimize

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

- It is zero when function $y(x,\mathrm{w})$ passes exactly through each training data point

Regression with multiple inputs

- Generalization
  - Predict value of continuous target variable t given value of D input variables $\mathrm{x}=[x_1,..x_D]$
  - t can also be a set of variables (multiple regression)
  - Linear functions of adjustable parameters
    - Specifically linear combinations of nonlinear functions of input variable
- Polynomial curve fitting is good only for:
  - Single input variable scalar x
  - It cannot be easily generalized to several variables, as we will see

Simplest Linear Model with D inputs

- Regression with D input variables

$$y(\mathrm{x},\mathrm{w}) = w_0 + w_1 x_1 + .. + w_d\, x_D = \mathrm{w}^T \mathbf{x}$$

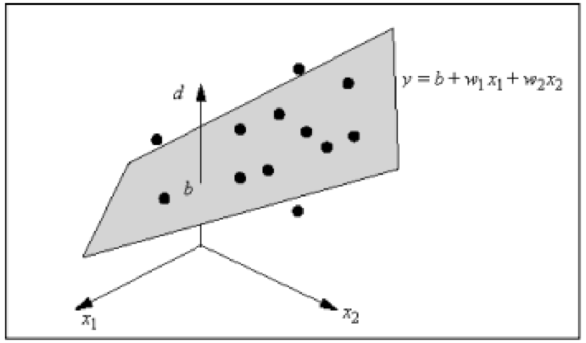
This differs from
Linear Regression with <u>one</u> variable
and Polynomial Reg with <u>one</u> variable

where $\mathrm{x}=(x_1,..,x_D)^{\mathrm{T}}$ are the input variables

- Called Linear Regression since it is a linear function of
  - parameters $w_0,..,w_D$
  - input variables $x_1,..,x_D$
- Significant limitation since it is a linear function of input variables
  - In the one-dimensional case this amounts a straight-line fit (degree-one polynomial)
  - $y(x,\mathrm{w}) = w_0 + w_1 x$

Fitting a Regression Plane

- Assume t is a function of inputs $x_1,\ x_2, \ldots x_D$
  Goal: find best linear regressor of t on all inputs
  - Fitting a hyperplane through N input samples
  - For $D = 2$:



x_1	x_2	t
1	2	2
2	5	1
2	3	2
2	2	2
3	4	1
3	5	3
4	6	2
5	5	3
5	6	4
5	7	3
6	8	4
7	6	2
8	4	4
8	9	3
9	8	4

- Being a linear function of input variables imposes limitations on the model
  - Can extend class of models by considering fixed nonlinear functions of input variables

Basis Functions

- In many applications, we apply some form of fixed-preprocessing, or feature extraction, to the original data variables
- If the original variables comprise the vector $\mathbf{x}$, then the features can be expressed in terms of basis functions $\{\phi_j(\mathbf{x})\}$
  - By using nonlinear basis functions we allow the function $y(\mathbf{x},\mathbf{w})$ to be a nonlinear function of the input vector $\mathbf{x}$
    - They are linear functions of parameters (gives them simple analytical properties), yet are nonlinear wrt input variables

Linear Regression with M Basis Functions

- Extended by considering nonlinear functions of input variables

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

  - where $\phi_j(\mathrm{x})$ are called Basis functions
  - We now need M weights for basis functions instead of D weights for features
  - With a dummy basis function $\phi_0(\mathrm{x})=1$ corresponding to the bias parameter w_0, we can write

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

  - where $\mathrm{w}=(w_0,w_1,..,w_{M-1})$ and $\Phi=(\phi_0,\phi_1,..,\phi_{M-1})^{\mathrm{T}}$
- Basis functions allow non-linearity with D input variables

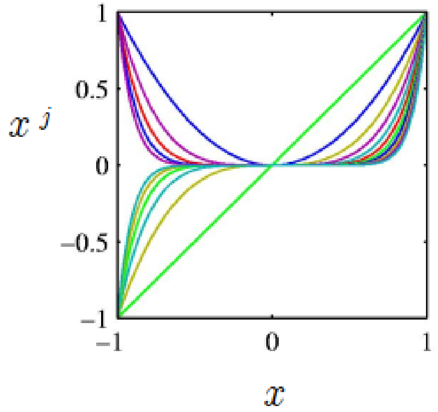
Choice of Basis Functions

- Many possible choices for basis function:
  1. Polynomial regression
     - Good only if there is only one input variable
  2. Gaussian basis functions
  3. Sigmoidal basis functions
  4. Fourier basis functions
  5. Wavelets

1. Polynomial Basis for one variable

- Linear Basis Function Model

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



- Polynomial Basis (for single variable x)

  $\phi_j(x) = x^j$ with degree M-1 polynomial

- Disadvantage
  - Global:
    - changes in one region of input space affects others
  - Difficult to formulate
    - Number of polynomials increases exponentially with M
  - Can divide input space into regions
    - use different polynomials in each region:
    - equivalent to spline functions

Can we use Polynomial with D variables? (Not practical!)

- Consider (for a vector x) the basis: $\phi_j(\mathrm{x}) = \|\mathrm{x}\|^j = \left[\sqrt{x_1^2 + x_2^2 + .. + x_d^2}\right]^j$
  - $\mathrm{x}=(2,1)$ and $\mathrm{x}=(1,2)$ have the same squared sum, so it is unsatisfactory
  - Vector is being converted into a scalar value thereby losing information
- Better polynomial approach:
  - Polynomial of degree M-1 has terms with variables taken none, one, two… M-1 at a time.
  - Use multi-index $j=(j_1,j_2,..j_D)$ such that $j_1+j_2+..j_D \leq M$-1
  - For a quadratic $(M=3)$ with three variables $(D=3)$

$$y(\mathrm{x},\mathrm{w}) = \sum_{(j_1,j_2,j_3)} w_j\phi_j(\mathrm{x}) = w_0 + w_{1,0,0}x_1 + w_{0,1,0}x_2 + w_{0,0,1}x_3 + w_{1,1,0}x_1x_2 + w_{1,0,1}x_1x_3 + w_{0,1,1}x_2x_3 + w_{2,0,0}x_1^2 + w_{0,2,0}x_2^2 + w_{0,0,2}x_3^2$$

  - Number of quadratic terms is $1+D+D(D\text{-}1)/2+D$
  - For $D=46$, it is 1128
- Better to use Gaussian kernel, discussed next

Disadvantage of Polynomial

- Polynomials are *global* basis functions
  - Each affecting the prediction over the whole input space
- Often local basis functions are more appropriate

Review and Derivations

Linear Regression

- The simplest linear model for regression is one that involves a linear combination of the input variables

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + \ldots + w_D x_D \quad = \sum_{j=0}^{D} w_j x^j$$

where $\mathbf{x} = (x_1, \ldots, x_D)^{\mathrm{T}}$

- Extension of linear regression models

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

Linear Regression

- Error Function:

$$E(w) = \frac{1}{2}\sum_{n=1}^{N}\left(t_n - \sum_{i=0}^{D} w_i x_i\right)^2$$

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

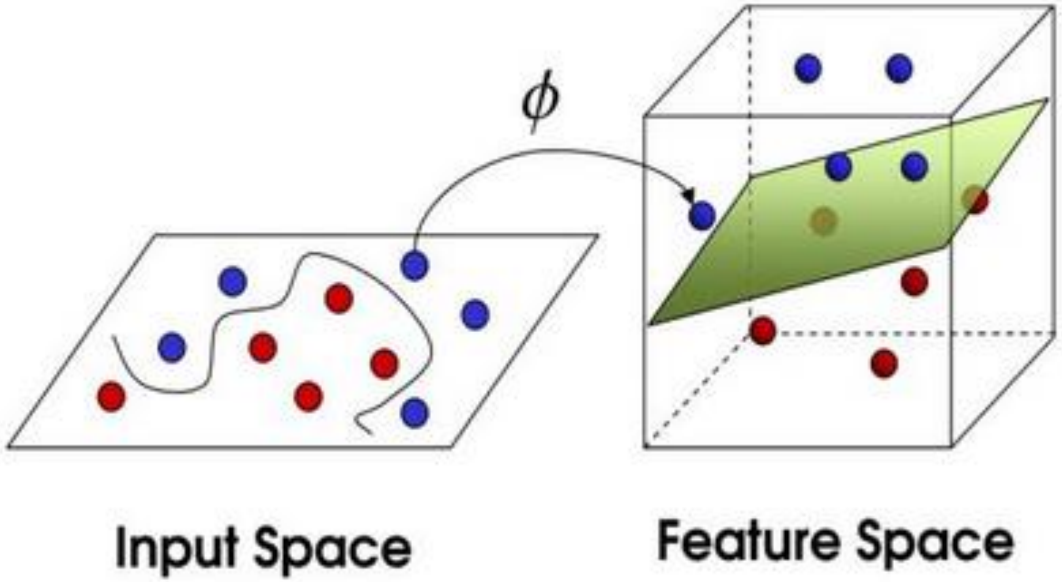
$$E(w) = (t - Xw)^T (t - Xw)$$

- To find *w* which maximizes this function, we differentiate it w.r.t. to *w*; we get

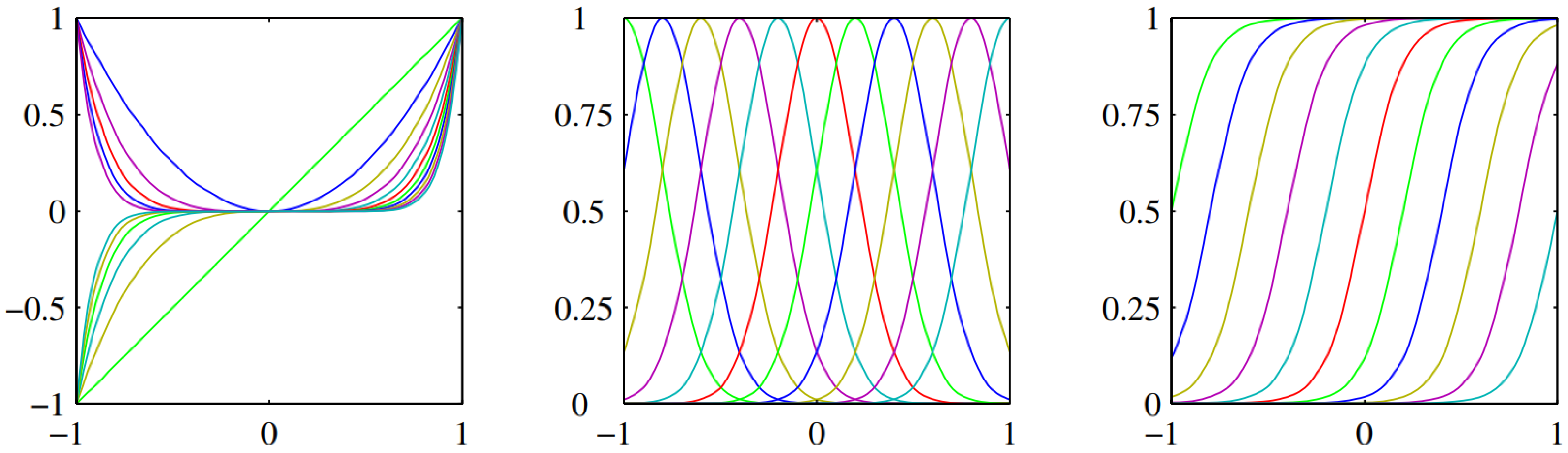
$$X^T(t - Xw) = 0$$

$$w = (X^T X)^{-1} X^T t$$

Basis Function Motivation



Basis Functions



Basis Functions

- Gaussian Basis Function

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

- Sigmoid Function

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

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

Example 1: Linear Regression

- We specify the set of functions $\phi_1, \phi_2, \ldots., \phi_M$ from X to $\mathbb{R}$ and look for function in the form of linear combination

$$y(x) = \sum_{i=0}^{M} w_i \emptyset_i(x)$$

- Performing the regression then reduces to finding the real parameters $w_0, w_1, \ldots, w_M$
- When *x* is only one dimension the **simplest basis function** would be:

$$\phi_0(x) = 1 \text{ and } \phi_1(x) = x$$

- This gives:

$$y(x) = \sum_{i=0}^{1} w_i \emptyset_i(x) = w_0 + w_1 x$$

- In the multidimensional case, where $\boldsymbol{x} = < x_1, \ldots, x_D >$ we would take:

$$\phi_0(x) = 1, \phi_1(x) = x_1, \phi_2(x) = x_2, \ldots\ldots, \phi_D(x) = x_D$$

$$y(x) = \sum_{i=0}^{D} w_i \emptyset_i(x) = w_0 + w_1 x_1 + w_2 x_2 + \ldots\ldots + w_D x_D$$

Example 2: Polynomial Regression

- **Polynomial Regression (when *x* is 1-dimensional):**
- Another possible choice to set $\phi_i(x) = x^i$ for $i = 1,2,\ldots,M$ where M is the degree of the polynomial

$$y(x) = \sum_{i=0}^{M} w_i \emptyset_i(x) = w_0 + w_1 x^1 + w_2 x^2 + \ldots\ldots + w_M x^M$$

- **Polynomial Regression (when *x* is D-dimensional):** For example D = 3 and M = 2
- $X = <x_1,x_2,x_3>$ and $x_1 + x_2 + x_3 <= 2$

- $y(x) = \sum_{(x1,x2,x3)} w_i \emptyset_i(x) = w_0 + w_{(1,0,0)} x_1 + w_{(0,1,0)} x_2 + w_{(0,0,1)} x_3 + w_{(1,1,0)} x_1x_2$
  $+ w_{(1,1,0)} x_1x_2 + w_{(0,1,1)} x_2x_2 + w_{(1,0,1)} x_1x_3 + w_{(2,0,0)} x_1^2 + w_{(0,2,0)} x_2^2 + w_{(0,0,2)} x_3^2$

Maximum likelihood and least squares

Deterministic function with Gaussian noise

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

Alternatively, we can write

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

If we assume squares loss function, then optimal prediction for **x ….???**

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

Likelihood Function for regression

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

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

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

Linear Regression using basis function

- Error Function:

$$E(w) = \frac{1}{2}\sum_{n=1}^{N}\left(t_n - \sum_{i=0}^{M} w_i \emptyset(x_i)\right)^2$$

$$E(w) = \frac{1}{2}\sum_{n=1}^{N}\left(t_n - w^T \emptyset(x)\right)^2$$

- Where, $w = (w_0, w_1, \ldots, w_M)^T$
- To represent the error function in vector form we get:

$$E(w) = \frac{1}{2}(\boldsymbol{t} - \boldsymbol{Qw})^T(\boldsymbol{t} - \boldsymbol{Qw})$$

Linear Regression using basis function

- To represent the error function in vector form we get:

$$E(w) = \frac{1}{2}(\boldsymbol{t} - \boldsymbol{Qw})^T(\boldsymbol{t} - \boldsymbol{Qw})]$$

Where **t =** $(t_1, t_2, \ldots, t_N)^T$ **,** **w** = $(w_0, w_1, \ldots, w_M)^T$ and **Q** is as follows:

$$Q = \begin{pmatrix} \phi_0(x_1) & \phi_1(x_1) & \cdots & \phi_M(x_1) \\ \phi_0(x_2) & \phi_1(x_2) & & \phi_M(x_2) \\ \vdots & & \ddots & \vdots \\ \phi_0(x_N) & \phi_1(x_N) & \cdots & \phi_M(x_N) \end{pmatrix}$$

- To find $\mathbf{w}^*$ (which minimize the error function), set the derivatives of the error function with respect to each w_i equal to zero

$$\frac{\mathrm{d}}{\mathrm{d}(\boldsymbol{w_i})}\left[\frac{1}{2}(\boldsymbol{t} - \boldsymbol{Qw})^T(\boldsymbol{t} - \boldsymbol{Qw})]\right] = \mathbf{0}$$

Linear Regression using basis function

- In gradient vector form it can be written as:

$$\mathbf{0} = \nabla_{\boldsymbol{w}}[\frac{1}{2}(\boldsymbol{t} - \boldsymbol{Q}\boldsymbol{w})^T(\boldsymbol{t} - \boldsymbol{Q}\boldsymbol{w})]$$

$$\mathbf{0} = \nabla_{\boldsymbol{w}}[\frac{1}{2}(\boldsymbol{t}^T\boldsymbol{t} - \mathbf{2}\boldsymbol{t}^T\boldsymbol{Q}\boldsymbol{w} + \boldsymbol{w}^T\boldsymbol{Q}^T\boldsymbol{Q}\boldsymbol{w})]$$

$$\mathbf{0} = [\frac{1}{2}(-\mathbf{2}\boldsymbol{Q}^T\boldsymbol{t} + \mathbf{2}\boldsymbol{Q}^T\boldsymbol{Q}\boldsymbol{w})]$$

- After simplification

$$\boldsymbol{Q}^T\boldsymbol{Q}\boldsymbol{w} = \boldsymbol{Q}^T\boldsymbol{t}$$

$$\boldsymbol{w} = (\boldsymbol{Q}^T\boldsymbol{Q})^{-1}\boldsymbol{Q}^T\boldsymbol{t}$$

- No matter whether we do linear, polynomial or Gaussian, the only thing that changes is the definition of the matrix **Q**

Regularized least squares

- $E(w) + \lambda E_w(w)$ where $E_w(w) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}$

$$E(w) = \frac{1}{2}(\boldsymbol{t} - \boldsymbol{Q}\boldsymbol{w})^T(\boldsymbol{t} - \boldsymbol{Q}\boldsymbol{w}) + \frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w}$$

$$\boldsymbol{0} = \nabla_{\boldsymbol{w}}[\frac{1}{2}(\boldsymbol{t} - \boldsymbol{Q}\boldsymbol{w})^T(\boldsymbol{t} - \boldsymbol{Q}\boldsymbol{w}) + \frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w}]$$

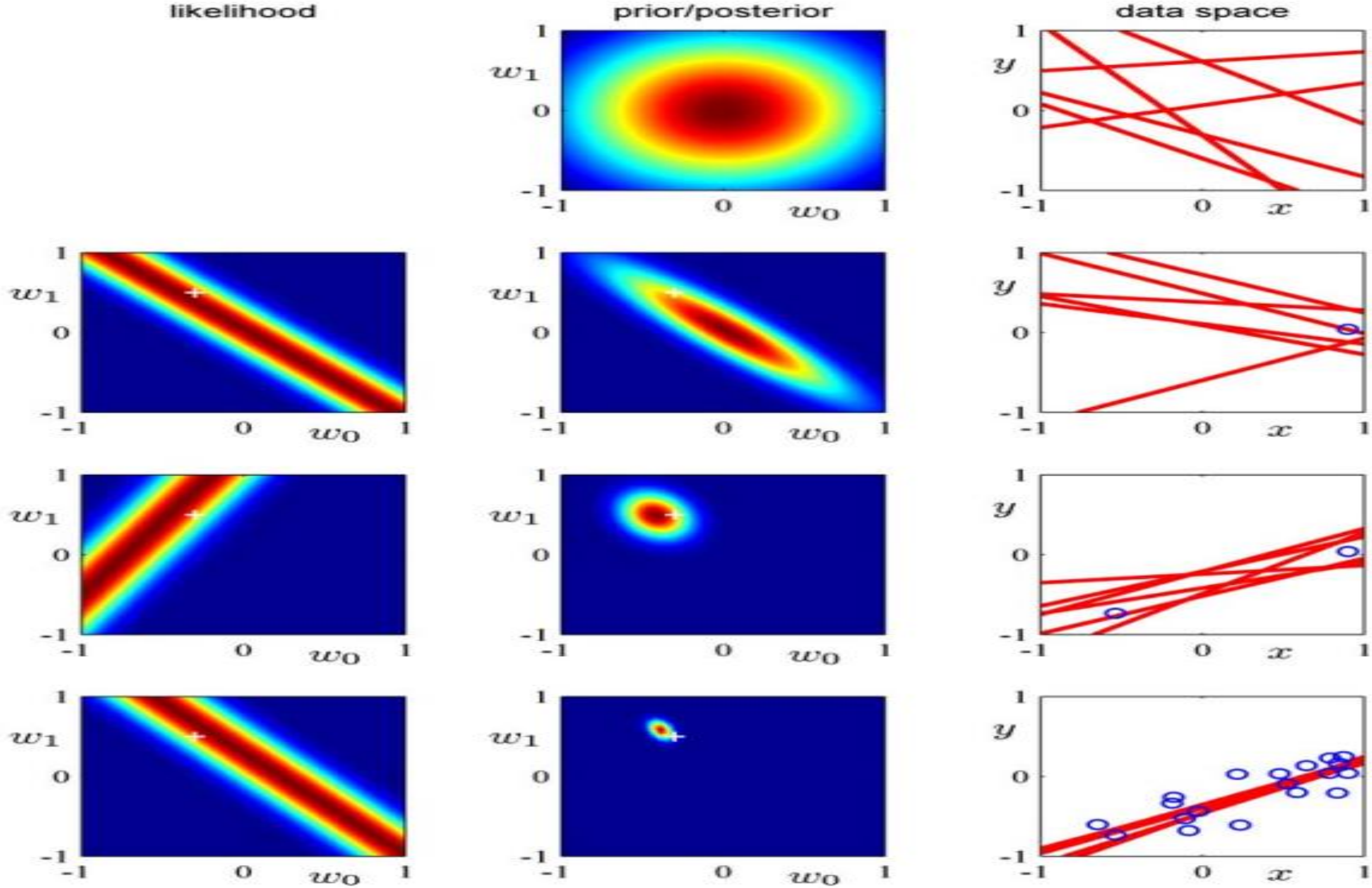
$$\boldsymbol{0} = \nabla_{\boldsymbol{w}}[\frac{1}{2}(\boldsymbol{t}^T\boldsymbol{t} - 2\boldsymbol{t}^T\boldsymbol{Q}\boldsymbol{w} + \boldsymbol{w}^T\boldsymbol{Q}^T\boldsymbol{Q}\boldsymbol{w}) + \frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w}]$$

$$\boldsymbol{0} = [-\boldsymbol{Q}^T\boldsymbol{t} + \boldsymbol{Q}^T\boldsymbol{Q}\boldsymbol{w} + \lambda\boldsymbol{w}]$$

$$\boldsymbol{Q}^T\boldsymbol{Q}\boldsymbol{w} + \lambda\boldsymbol{w} = \boldsymbol{Q}^T\boldsymbol{t}$$

$$(\boldsymbol{Q}^T\boldsymbol{Q} + \lambda\boldsymbol{I})\boldsymbol{w} = \boldsymbol{Q}^T\boldsymbol{t}$$

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



likelihood
prior/posterior
data space
w_1
w_0
x
y

Bayesian Linear Regression

- $D = \{(x_1,y_1), (x_2,y_2),\ldots,(x_N,y_N)\}$ where $x_i \in \mathbb{R}^D$ and $y \in \mathbb{R}$
- Model : $Y_1, Y_2, \ldots, Y_N$ independent given **w**, $Y \sim \mathcal{N}(w^T x_i, \beta)$ [β is precision; $\beta = 1/\sigma^2$]
- $\mathbf{w} \sim \mathcal{N}(0, \alpha^{-1} I)$ where $\mathbf{w} = (w_0, w_1, \ldots, w_D)^T$
- Assume β and α are known
  - ✓ therefore only unknown parameter is **w**
- **Likelihood:**

$$p(\boldsymbol{D}|\boldsymbol{w}) \alpha\, exp(-\frac{\beta}{2}(\boldsymbol{y}-\boldsymbol{Q}\boldsymbol{w})^{\boldsymbol{T}}(\boldsymbol{y}-\boldsymbol{Q}\boldsymbol{w}))$$

- **Posterior:**

$$p(\boldsymbol{w}|\boldsymbol{D})\ \alpha\ p(\boldsymbol{D}|\boldsymbol{w})\ \boldsymbol{p}(\boldsymbol{w})$$

Posterior of *w*

$$p(\boldsymbol{w}|\boldsymbol{D})\ \alpha\ p(\boldsymbol{D}|\boldsymbol{w})\ \boldsymbol{p}(\boldsymbol{w})$$

$$p(\boldsymbol{w}|\boldsymbol{D})\ \alpha\ exp\left(-\frac{\beta}{2}(\boldsymbol{t}-\boldsymbol{Q}\boldsymbol{w})^{\boldsymbol{T}}(\boldsymbol{t}-\boldsymbol{Q}\boldsymbol{w})\right)\ \boldsymbol{exp}(-\frac{\alpha}{2}\boldsymbol{w}^{T}\boldsymbol{w})$$

$$p(\boldsymbol{w}|\boldsymbol{D})\ \alpha\ exp\left(-\frac{\beta}{2}(\boldsymbol{t}-\boldsymbol{Q}\boldsymbol{w})^{\boldsymbol{T}}(\boldsymbol{t}-\boldsymbol{Q}\boldsymbol{w})-\frac{\alpha}{2}\boldsymbol{w}^{T}\boldsymbol{w}\right)$$

$$p(\boldsymbol{w}|\boldsymbol{D})\ \alpha\ exp\left(-\frac{1}{2}(\beta(\boldsymbol{t}-\boldsymbol{Q}\boldsymbol{w})^{\boldsymbol{T}}(\boldsymbol{t}-\boldsymbol{Q}\boldsymbol{w})+\alpha\boldsymbol{w}^{T}\boldsymbol{w})\right)$$

$$p(\boldsymbol{w}|\boldsymbol{D})\ \alpha\ exp\left(-\frac{1}{2}\left(\beta\left(\boldsymbol{t}^{\boldsymbol{T}}\boldsymbol{t}\ -\mathbf{2}\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{Q}^{\boldsymbol{T}}\boldsymbol{t}\ +\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{Q}^{\boldsymbol{T}}\boldsymbol{Q}\boldsymbol{w}\right)\right)+\alpha\boldsymbol{w}^{T}\boldsymbol{w})\right)\ [-\mathbf{2}\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{Q}^{\boldsymbol{T}}\boldsymbol{t}\ = \text{-}\boldsymbol{t}^{\boldsymbol{T}}\boldsymbol{Q}\boldsymbol{w}\ -\ (\boldsymbol{Q}\boldsymbol{w})^{\boldsymbol{T}}\boldsymbol{t}\ ]$$

$$p(\boldsymbol{w}|\boldsymbol{D})\ \alpha\ exp\left(-\frac{1}{2}\left(\ \left(\beta\boldsymbol{t}^{\boldsymbol{T}}\boldsymbol{t}\ -\mathbf{2}\beta\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{Q}^{\boldsymbol{T}}\boldsymbol{t}\ +\beta\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{Q}^{\boldsymbol{T}}\boldsymbol{Q}\boldsymbol{w}\right)\right)+\alpha\boldsymbol{w}^{T}\boldsymbol{w})\right)$$

$$p(\boldsymbol{w}|\boldsymbol{D})\ \alpha\ exp\left(-\frac{1}{2}(\ \left(\beta\boldsymbol{t}^{\boldsymbol{T}}\boldsymbol{t}\ -\mathbf{2}\beta\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{Q}^{\boldsymbol{T}}\boldsymbol{t}\ +\boldsymbol{w}^{\boldsymbol{T}}(\beta\boldsymbol{Q}^{\boldsymbol{T}}\boldsymbol{Q}+\alpha\boldsymbol{I}\right)\boldsymbol{w}\right)$$

Posterior of *w*

$$p(\boldsymbol{w}|\boldsymbol{D})\ \alpha\ exp\left(-\frac{1}{2}\left(\left(\beta \boldsymbol{t}^{\boldsymbol{T}}\boldsymbol{t}\ -2\beta \boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{Q}^{\boldsymbol{T}}\boldsymbol{t}\ +\boldsymbol{w}^{\boldsymbol{T}}(\beta \boldsymbol{Q}^{\boldsymbol{T}}\boldsymbol{Q}+\alpha \boldsymbol{I})\boldsymbol{w}\right)\right)$$

Completing the square:

$$\mathcal{N}(\boldsymbol{\mu},\Lambda^{-1})\ \alpha\ \boldsymbol{exp}\left(-\frac{1}{2}(\boldsymbol{w}-\boldsymbol{\mu})^{\boldsymbol{T}}\Lambda(\boldsymbol{w}-\boldsymbol{\mu})\right)$$

$$\mathcal{N}(\boldsymbol{\mu},\Lambda^{-1})\ \alpha\ \boldsymbol{exp}\left(-\frac{1}{2}(\boldsymbol{w}^{\boldsymbol{T}}\Lambda\boldsymbol{w}\ -\boldsymbol{w}^{\boldsymbol{T}}\Lambda\boldsymbol{\mu}-\boldsymbol{\mu}^{\boldsymbol{T}}\Lambda\boldsymbol{w}+\boldsymbol{\mu}^{\boldsymbol{T}}\Lambda\boldsymbol{\mu})\right)$$

$$\mathcal{N}(\boldsymbol{\mu},\Lambda^{-1})\ \alpha\ \boldsymbol{exp}\left(-\frac{1}{2}(\boldsymbol{w}^{\boldsymbol{T}}\Lambda\boldsymbol{w}\ -2\boldsymbol{w}^{\boldsymbol{T}}\Lambda\boldsymbol{\mu}+\boldsymbol{\mu}^{\boldsymbol{T}}\Lambda\boldsymbol{\mu})\right)$$

Hence:

$$\Lambda=\beta\boldsymbol{Q}^{\boldsymbol{T}}\boldsymbol{Q}+\alpha\boldsymbol{I}$$

$$\boldsymbol{\mu}=\beta\Lambda^{-1}\boldsymbol{Q}^{\boldsymbol{T}}\boldsymbol{t}\qquad(\cancel{\boldsymbol{w}^{\boldsymbol{T}}}\Lambda\boldsymbol{\mu}=\beta\cancel{\boldsymbol{w}^{\boldsymbol{T}}}\boldsymbol{Q}^{\boldsymbol{T}}\boldsymbol{t})$$

Posterior of *w*

$$\Lambda = \beta \boldsymbol{Q}^T \boldsymbol{Q} + \alpha \boldsymbol{I}$$

$$\boldsymbol{\mu} = \beta \Lambda^{-1} \boldsymbol{Q}^T \boldsymbol{t}$$

$$\boldsymbol{\mu} = \beta(\beta \boldsymbol{Q}^T \boldsymbol{Q} + \alpha \boldsymbol{I})^{-1} \boldsymbol{Q}^T \boldsymbol{t} = (\boldsymbol{Q}^T \boldsymbol{Q} + \frac{\alpha}{\beta} \boldsymbol{I})^{-1} \boldsymbol{Q}^T \boldsymbol{t}$$

- $p(\boldsymbol{w}|\boldsymbol{D}) \sim \mathcal{N}(\boldsymbol{\mu}, \Lambda^{-1})$
- MAP estimate of $\boldsymbol{w}$

$$w_{MAP} = (\boldsymbol{Q}^T \boldsymbol{Q} + \frac{\alpha}{\beta} \boldsymbol{I})^{-1} \boldsymbol{Q}^T \boldsymbol{t}$$

[From regularized least square slide

$$\boldsymbol{w} = (\lambda \boldsymbol{I} + \boldsymbol{Q}^T \boldsymbol{Q})^{-1} \boldsymbol{Q}^T \boldsymbol{t}]$$

Prediction

$$p(y|\boldsymbol{x},\boldsymbol{D}) = \int p(y|\boldsymbol{x},\boldsymbol{D},\boldsymbol{w})p(\boldsymbol{w}|\boldsymbol{x},\boldsymbol{D})d\boldsymbol{w}$$

$$p(y|\boldsymbol{x},\boldsymbol{D}) = \int p(y|\boldsymbol{x},\boldsymbol{w})p(\boldsymbol{w}|\boldsymbol{D})d\boldsymbol{w}$$

[Note: $p(\boldsymbol{w}|\boldsymbol{D})\ \alpha\ exp\left(-\frac{1}{2}(\ (\beta\boldsymbol{y}^{\boldsymbol{T}}\boldsymbol{y}\ -2\beta\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{Q}^{\boldsymbol{T}}\boldsymbol{y}\ +\boldsymbol{w}^{\boldsymbol{T}}(\beta\boldsymbol{Q}^{\boldsymbol{T}}\boldsymbol{Q}+\alpha\boldsymbol{I})\boldsymbol{w}\right)$]

$$p(y|\boldsymbol{x},\boldsymbol{D})\ \alpha \int \exp\left(-\frac{\beta}{2}(y\ -\ \boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x})^{\boldsymbol{2}}\right)\exp\left(-\frac{1}{2}(\boldsymbol{w}-\boldsymbol{\mu})^{\boldsymbol{T}}\Lambda(\boldsymbol{w}-\boldsymbol{\mu})\right)dw$$

$$p(y|\boldsymbol{x},\boldsymbol{D})\ \alpha\ \exp(-\frac{\gamma}{2}(y-\upsilon)^{\boldsymbol{2}})$$

where $\upsilon = \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{x}$

$$\frac{1}{\gamma} = \frac{1}{\beta} + \boldsymbol{x}^{\boldsymbol{T}}\,\Lambda^{-1}\boldsymbol{x}$$

$\boldsymbol{\mu} = \beta\Lambda^{-1}\boldsymbol{Q}^{\boldsymbol{T}}\boldsymbol{y}$
$\Lambda = \beta\boldsymbol{Q}^{\boldsymbol{T}}\boldsymbol{Q} + \alpha\boldsymbol{I}$

References

- Chapter 3, Pattern Recognition and Machine Learning, C. Bishop