Lecture notes on CS725 : Machine learning

Contents

1 Lecture 1 : Introdcution to Machine Learning

This lecture was an introduction to machine learning. ...

2 Lecture 2

2.1 Solving Least Squares in General (for Linear models)

$X_{plot} = [0, 0.1, ..., 2]$ *Curve*

- Why noise ?
  - Since observations are not perfect
  - Owing to quantization / precision / rounding

Curve Fitting

Learn $f : X \rightarrow Y$ such that $E(f, X, Y_1)$ is minimized. Here the error function E and form of the function to learn f is chosen by the modeler.

Consider one such form of f,

$$f(x) = w_0 + w_1 x + w_2 x^2 + ... + w_t x^t$$

The sum of squares error is given by,

$$E = \frac{1}{2}\sum_{i=1}^{m}(f(x_i) - y_i)^2$$

So the expression is,

$$\underset{w=[w_1,w_2,...w_t]}{\text{argmin}} \frac{1}{2}\sum_{i=1}^{K}[(w_0 + w_1 x + w_2 x^2 + ... + w_t x^t) - y_1(i)]^2$$

If there are m data points, then a polynomial of degree $m-1$ can exactly fit the data, since the polynomial has m degrees of freedom (where degrees of freedom=no. of coefficients)

As the degree of the polynomial increases beyond m, the curve becomes more and more wobbly, while still passing through the points. Contrast the degree 10 fit in Figure 2.1 against the degree 5 fit in Figure 2.1. This is due to the problem of overfitting (overspecification)

Now E is a convex function. To optimize it, we need to set $\nabla_w E = 0$. The ∇ operator is also called gradient.

Solution is given by,

$$X = (\phi^t \phi)^{-1} \phi^t Y$$

If $m << t$ then

- ϕ becomes singular and the solution cannot be found OR
- The column vectors in ϕ become nearly linearly dependent

RMS (root mean sqare) error is given by :

$$RMS = \sqrt{\frac{2E}{k}}$$

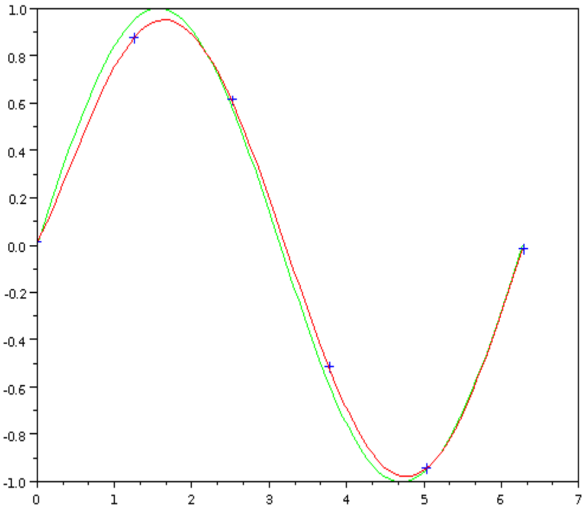


Figure 1: Fit for degree 5 polynomial.

Generally, some test data (which potentially could have been part of the training data) is held out for evaluating the generalized performance of the model. Another held out fraction of the training data, called the validation dataset is typically used to find the most appropriate degree t_{best} for f.

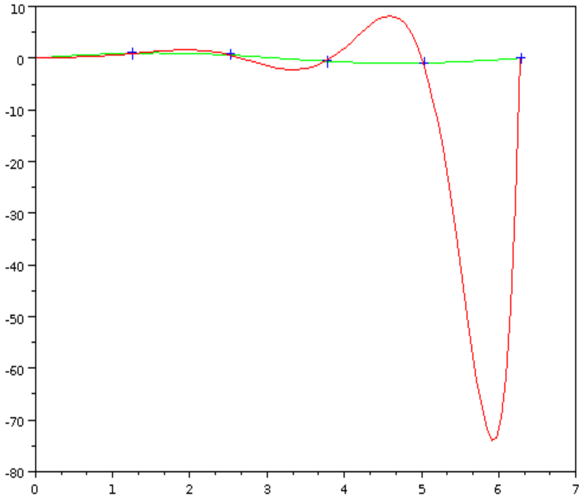


Figure 2: Fit for degree 10 polynomial. Note how wobbly this fit is.

3 Lecture 3 : Regression

This lecture was about regression. It started with formally defining a regression problem. Then a simple regression model called linear regression was discussed. Different methods for learning the parameters in the model were next discussed. It also covered least square solution for the problem and its geometrical interpretation.

3.1 Regression

Suppose there are two sets of variables $\mathbf{x} \in \Re^n$ and $\mathbf{y} \in \Re^k$ such that $\mathbf{x}$ is independent and y is dependant. The regression problem is concerned with determining y in terms of $\mathbf{x}$. Let us assume that we are given m data points $\mathcal{D} = \langle \mathbf{x}_1, \mathbf{y}_1 \rangle, \langle \mathbf{x}_2, \mathbf{y}_2 \rangle, .., \langle \mathbf{x}_m, \mathbf{y}_m \rangle$. Then the problem is to determine a function f^* such that $f^*(\mathbf{x})$ is the best predictor for $\mathbf{y}$, with respect to $\mathcal{D}$. Suppose $\varepsilon(f, \mathcal{D})$ is an error function, designed to reflect the discrepancy between the predicted value $f(\mathbf{x}')$ of $\mathbf{y}'$ and the actual value $\mathbf{y}'$ for any $\langle \mathbf{x}', \mathbf{y}' \rangle \in \mathcal{D}$, then

$$f^* = \underset{f \in \mathcal{F}}{\operatorname{argmin}} \ \varepsilon(f, \mathcal{D}) \tag{1}$$

where, $\mathcal{F}$ denotes the class of functions over which the optimization is performed.

3.2 Linear regression

Depending on the function class we consider, there are many types of regression problems. In Linear regression we consider only linear functions, functions that are linear in the basis function. Here $\mathcal{F}$ is of the form $\{\sum_{i=1}^{p} w_i \phi_i(\mathbf{x})\}$. $\phi_i : \mathbb{R}^n \rightarrow \mathbb{R}^k$ Here, the ϕ_i's are called the **basis functions** (for example, we can consider $\phi_i(x) = x^i$, *i.e.*, polynomial basis functions) .

Any function in $\mathcal{F}$ is characterized by its parameters, the w_i's. Thus, in (1) we have to find $f(\mathbf{w}^*)$ where

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \varepsilon(\mathbf{w}, \mathcal{D})$$

3.3 Least square solution

The error function ε plays a major role in the accuracy and tractability of the optimization problem. The error function is also called the **loss function**. The squared loss is a commonly used loss function. It is the sum of squares of the differences between the actual value and the predicted value.

$$\varepsilon(f, \mathcal{D}) = \sum_{\langle \mathbf{x}_i, y_i \rangle \in \mathcal{D}} (f(\mathbf{x}_i) - y_i)^2$$

So the least square solution for linear regression is given by

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{j=1}^{m} \big(\sum_{i=1}^{p} (w_i \phi_i(x_j) - y_j \big)^2$$

The minimum value of the squared loss is zero. Is it possible to achieve this value ? In other words is $\forall j$, $\displaystyle\sum_{i=1}^{p} w_i \phi_i(x_j) = y_j$ possible ?

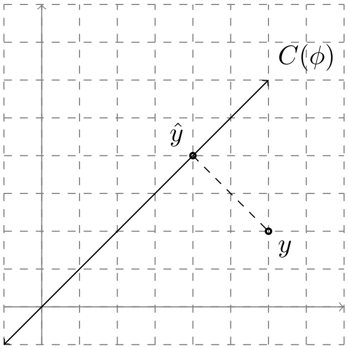


Figure 3: Least square solution $\hat{y}$ is the orthogonal projection of y onto column space of ϕ

The above equality can be written as $\forall u,\ \phi^T(x_u)\mathbf{w} = y_u$
or equivalently $\phi\mathbf{w} = \mathbf{y}$ where

$$\phi = \begin{bmatrix} \phi_1(\mathbf{x}_1) & \cdots & \phi_p(\mathbf{x}_1) \\ \vdots & \vdots & \vdots \\ \phi_1(\mathbf{x}_m) & \cdots & \phi_p(\mathbf{x}_m) \end{bmatrix} \text{ and } \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$$

It has a solution if $\mathbf{y}$ is in the column space (the subspace of $\mathbb{R}^n$ formed by the column vectors) of ϕ. It is possible that there exists no $\mathbf{w}$ which satisfies the conditions? In such situations we can solve the least square problem.

3.4 Geometrical interpretation of least squares

Let $\hat{\mathbf{y}}$ be a solution in the column space of ϕ. The least squares solution is such that the distance between $\hat{\mathbf{y}}$ and $\mathbf{y}$ is minimized. From the diagram it is clear that for the distance to be minimized, the line joining $\hat{\mathbf{y}}$ to $\mathbf{y}$ should be orthogonal to the column space. This can be summarized as

1. $\phi\mathbf{w} = \hat{\mathbf{y}}$
2. $\forall v \in \{1, ..p\},\ (\mathbf{y} - \hat{\mathbf{y}})^T\phi_v = 0$ or $(\hat{\mathbf{y}} - \mathbf{y})^T\phi = 0$

$$\begin{aligned} \hat{\mathbf{y}}^T\phi &= \mathbf{y}^T\phi \\ ie,\ (\phi\mathbf{w})^T\phi &= \mathbf{y}^T\phi \\ ie,\ \mathbf{w}^T\phi^T\phi &= \mathbf{y}^T\phi \\ ie,\ \phi^T\phi\mathbf{w} &= \phi^T\mathbf{y} \\ \therefore\ \mathbf{w} &= (\phi^T\phi)^{-1}\mathbf{y} \end{aligned}$$

In the last step, please note that, $\phi^T\phi$ is invertible only if ϕ has full column rank.

Theorem: If ϕ has full column rank, $\phi^T\phi$ is invertible. A matrix is said to have full column rank if all its column vectors are linearly independent. A set of vectors $\mathbf{v}_i$ is said to be linearly independent if $\sum_i \alpha_i \mathbf{v}_i = 0 \Rightarrow \alpha_i = 0$.

Proof: Given that ϕ has full column rank and hence columns are linearly independent, we have that $\phi\mathbf{x} = 0 \Rightarrow \mathbf{x} = \mathbf{0}$.

Assume on the contrary that $\phi^T\phi$ is non invertible. Then $\exists \mathbf{x} \neq \mathbf{0} \ni \phi^T\phi\mathbf{x} = \mathbf{0}$.

$\Rightarrow \mathbf{x}^T\phi^T\phi\mathbf{x} = 0$

$\Rightarrow (\phi\mathbf{x})^T\phi\mathbf{x} = ||\phi\mathbf{x}||^2 = 0$

$\Rightarrow \phi\mathbf{x} = \mathbf{0}$. This is a contradiction. Hence the theorem is proved.

4 Lecture 4 : Least Squares Linear Regression

In this lecture we discussed how to minimize the error function $\varepsilon(\mathbf{w}, \mathcal{D})$ that we used for the least square linear regression model in the last lecture. To do this, some basic concepts regarding minimization of a function were discussed and we applied these to our error function.

4.1 Least Square Linear Regression Model

In the least squares regression model, we determine the value of $\mathbf{w}$ for which our error function ε attains the minimum value. Here, $\mathcal{D} =< \mathbf{x}_1, y_1 >, < \mathbf{x}_2, y_2 >, .., < \mathbf{x}_m, y_m >$ is the training data set, and ϕ_i's are the basis functions.

$$\begin{aligned}
\mathbf{w}^* &= \underset{\mathbf{w}}{\text{argmin}}\left\{\sum_{j=1}^{m} \left(f\left(\mathbf{x}_j, \mathbf{w}\right) - y_j\right)^2\right\} \\
&= \underset{\mathbf{w}}{\text{argmin}}\left\{\sum_{j=1}^{m} \left(\sum_{i=1}^{p} w_i \phi_i(\mathbf{x}_j) - y_j\right)^2\right\}
\end{aligned}$$

$$\phi = \begin{bmatrix} \phi_1(\mathbf{x}_1) & \phi_2(\mathbf{x}_1) & ...... & \phi_p(\mathbf{x}_1) \\ . & & & \\ . & & & \\ \phi_1(\mathbf{x}_m) & \phi_2(\mathbf{x}_m) & ...... & \phi_p(\mathbf{x}_m) \end{bmatrix}$$

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ y_m \end{bmatrix}$$

$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ . \\ . \\ w_p \end{bmatrix}$$

$$\begin{aligned}
\varepsilon &= \min_{\mathbf{w}} \sum_{j=1}^{m} \left(\phi^T(\mathbf{x}_j)\mathbf{w} - y_j\right)^2 \\
&= \min_{\mathbf{w}} ||\phi\mathbf{w} - \mathbf{y}||^2 \\
&= \min_{\mathbf{w}} (\phi\mathbf{w} - \mathbf{y})^T (\phi\mathbf{w} - \mathbf{y}) \\
&= \min_{\mathbf{w}} \left(\mathbf{w}^T\phi^T\phi\mathbf{w} - 2\mathbf{y}^T\phi\mathbf{w} + \mathbf{y}^T\mathbf{y}\right) \qquad (2)
\end{aligned}$$

How to minimize a function?

Following are some basic concepts which help in minimizing or maximizing a function:

4.2 Level Curves and Surfaces

A *level curve* of a function $f(\mathbf{x})$ is defined as a curve along which the value of the function remains unchanged while we change the value of it's argument $\mathbf{x}$. Note that there can be as many level curves for any function as the number of different values it can attain.

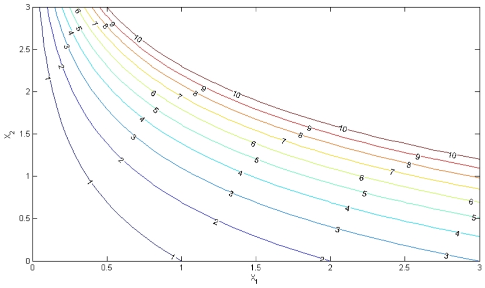


Figure 4: 10 level curves for the function $f(x_1, x_2) = x_1 e_2^x$ (Figure 4.12 from [1])

Level surfaces are similarly defined for any n-dimensional function $f(x_1, x_2, ..., x_n)$ as a collection of points in the argument space on which the value of the function is same at all points while we change the argument values.

Formally we can define a level curve as :

$$L_c(f) = \left\{ \mathbf{x} | f(\mathbf{x}) = c \right\}$$

where c is a constant.

Refer to Fig. 4.15 in class notes [1] for example.

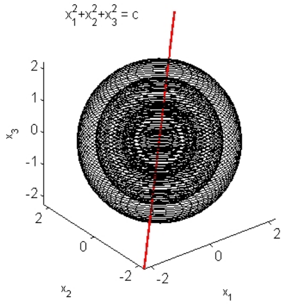


Figure 5: 3 level surfaces for the function $f(x_1, x_2, x_3) = x_1^2 + x_2^2 + x_3^2$ with $c = 1, 3, 5$. The gradient at $(1, 1, 1)$ is orthogonal to the level surface $f(x_1, x_2, x_3) = x_1^2 + x_2^2 + x_3^2 = 3$ at $(1, 1, 1)$ (Fig. 4.14 from [1]).

4.3 Gradient Vector

The *gradient vector* of a function f at a point $\mathbf{x}$ is defined as follows:

$$\nabla f_{\mathbf{x}^*} = \begin{bmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \frac{\partial f(\mathbf{x})}{\partial x_2} \\ . \\ . \\ \frac{\partial f(\mathbf{x})}{\partial x_n} \end{bmatrix} \epsilon \mathbb{R}^n$$

The direction of the gradient vector gives the direction of maximum rate of change of the value of the function at a point. Also the magnitude of the gradient vector gives that maximum value of rate of change.

Refer to Definition 23 in the class notes [1] for more details.

4.4 Directional Derivative

Directional Derivative gives the rate of change of the function value in a given direction at a point. The *directional derivative* of a function f in the direction of a unit vector $\mathbf{v}$ at a point $\mathbf{x}$ can be defined as :

$$D_{\mathbf{v}}(f) = \lim_{h \to 0} \frac{f(\mathbf{x} + h\mathbf{v}) - f(\mathbf{x})}{h}$$

$$||\mathbf{v}|| = 1$$

Note: The maximum value of directional derivative of a function f at any point is always the magnitude of it's gradient vector at that point.

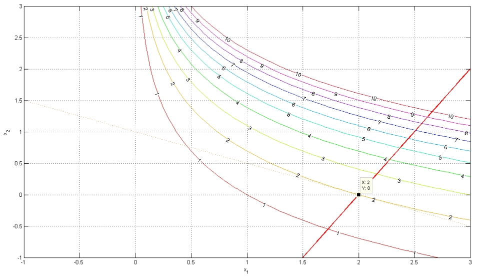


Figure 6: The level curves from Figure 4 along with the gradient vector at (2, 0). Note that the gradient vector is perpenducular to the level curve $x_1 e^{x_2} = 2$ at (2, 0) (Figure 4.13 from [1])

Refer to Definition 22 and Theorem 58 in the class notes [1] for more details.

4.5 Hyperplane

A Hyperplane is a set of points whose direction *w.r.t.* a point $\mathbf{p}$ is orthogonal to a vector $\mathbf{v}$. It can be formally defined as :

$$H_{\mathbf{v},\mathbf{p}} = \left\{ \mathbf{q} \mid (\mathbf{p} - \mathbf{q})^T \mathbf{v} = 0 \right\}$$

4.6 Tangential Hyperplane

There are two definitions of *tangential hyperplane* ($TH_{\mathbf{x}^*}$) to *level surface* ($L_{f(\mathbf{x}^*)}(f)$) of f at $\mathbf{x}^*$:

1. Plane consisting of all tangent lines at $\mathbf{x}^*$ to any parametric curve $c(t)$ on level surface.

2. Plane orthogonal to the gradient vector at $\mathbf{x}^*$.

$$TH_{\mathbf{x}^*} = \left\{ \mathbf{p} \mid (\mathbf{p} - \mathbf{x}^*)^T \nabla f(\mathbf{x}^*) = 0 \right\}$$

Note: By definition, $TH_{\mathbf{x}^*}$ is $n - 1$ dimensional.

Refer to Definition 24 and Theorem 59 in class notes [1] for more details.

4.7 Gradient Descent Algorithm

Gradient Descent Algorithm is used to find minimum value attained by a real valued function $f(\mathbf{x})$. We first start at an intial point $\mathbf{x}^{(0)}$ and make a sequence of steps proportional to negative of gradient of the function at the point. Finally we stop at a point $\mathbf{x}^{(*)}$ where a desired convergence criterion (see notes on Convex Optimization) will be attained.

The idea of gradient descent algorithm is based on the fact that if a real-valued function $f(\mathbf{x})$ is defined and differentiable at a point $\mathbf{x}^k$, then $f(\mathbf{x})$ decreases fastest when you move in the direction of the negative gradient of the function at that point, which is $-\nabla f(\mathbf{x})$.

Here we describe the method of Gradient Descent Algorithm to find the *parameter vector* **w** which minimizes the error function, $\varepsilon(\mathbf{w}, \mathcal{D})$

$$\begin{aligned}
\Delta \mathbf{w}^{(k)} &= -\ \nabla\varepsilon(w^{(k)}) \qquad \textbf{from equation } (2)\\
&= -\nabla(\mathbf{w}^T\phi^T\phi\mathbf{w} - 2\mathbf{y}^T\phi\mathbf{w} + \mathbf{y}^T\mathbf{y})\\
&= -(2\phi^T\phi\mathbf{w} - 2\mathbf{y}^T\phi + 0)\\
&= 2(\phi^T\mathbf{y} - \phi^T\phi\mathbf{w})
\end{aligned}$$

so we got

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + 2t^{(k)}(\phi^T\mathbf{y} - \phi^T\phi\mathbf{w}^{(k)})$$

Gradient Descent Algorithm :

```
Find starting point $\mathbf{w}^{(0)}\epsilon\mathcal{D}$
repeat
  1. $\Delta\mathbf{w}^k = -\nabla\varepsilon(\mathbf{w}^{(k)})$
  2. Choose a step size $t^{(k)} > 0$ using exact or backtracking ray search.
  3. Obtain $\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + t^{(k)}\Delta\mathbf{w}^{(k)}$.
  4. Set $k = k + 1$.
until stopping criterion (such as $\|\nabla\varepsilon(\mathbf{x}^{(k+1)})\ \| \leq \epsilon$) is satisfied
```

Exact Line Search Algorithm :

$$t^{(k)} = \underset{t}{\operatorname{argmin}} \varepsilon\left(\mathbf{w}^{(k)} + 2t\left(\phi^T\mathbf{y} - \phi^T\phi\mathbf{w}^{(k)}\right)\right)$$

In general

$$t^{(k)} = \underset{t}{\operatorname{argmin}} f\left(\mathbf{w}^{(k+1)}\right)$$

Refer to section 4.5.1 in the class notes [1] for more details.

4.8 Local Minimum and Local Maximum

Critical Point : **x** is a called a *critical point w.r.t* to a function f, if $\nabla f(\mathbf{x}) = 0$ i.e. the gradient vanishes at **x** or the gradient fails to exist at **x**.

Local Minimum (or Maximum):

If $\nabla f(\mathbf{x}^*) = \mathbf{0}$ then $\mathbf{x}^*$ can be a point of local minimum (or maximum). [*Neccessary Condition*]

If $\nabla^2 f(\mathbf{x}^*)$ is positive (negative) definite then $\mathbf{x}^*$ is a point of local minimum (maximum). [*Sufficient Condition*]

Note: $\nabla^2 f(\mathbf{x}^*)$ is positive definite means :

$$\forall \mathbf{x} \neq \mathbf{0} \quad \mathbf{x}^T \nabla^2 f(\mathbf{x}^*) \mathbf{x} > \mathbf{0}$$

OR

$$\lambda_i(\nabla^2 f(\mathbf{x}^*)) > 0$$

i.e. matrix eigen values are positive.

Refer to definition 27, theorem 61 and fig. 4.23, 4.24 in the class notes [1] for more details.

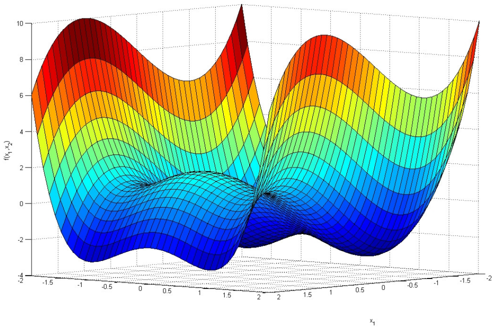


Figure 7: Plot of $f(x_1, x_2) = 3x_1^2 - x_1^3 - 2x_2^2 + x_2^4$, showing the various local maxima and minima of the function (fig. 4.16 from [1])

5 Lecture 5 : Convex functions

In this lecture the concepts of convex sets and functions were introduced.

5.1 Recap

We recall that the problem was to find $\mathbf{w}$ such that

$$\begin{aligned} \mathbf{w}^* &= \underset{\mathbf{w}}{\operatorname{argmin}} \left|\left|\phi\mathbf{w}-\mathbf{y}\right|\right|^2 && (3)\\ &= \operatorname{argmin}_{\mathbf{w}}(\mathbf{w}^T\phi^T\phi\mathbf{w}-2\mathbf{w}^T\phi\mathbf{y}-\mathbf{y}^T\mathbf{y}) && (4) \end{aligned}$$

5.2 Point 1

If $\nabla f(\mathbf{x}^*)$ is defined & $\mathbf{x}^*$ is local minimum/maximum, then $\nabla f(\mathbf{x}^*) = 0$
(A necessary condition) `(Cite : Theorem 60)[2]`

Given that

$$\begin{aligned} f(\mathbf{w}) &= \underset{\mathbf{w}}{\operatorname{argmin}}(\mathbf{w}^T\phi^T\phi\mathbf{w}-2\mathbf{w}^T\phi\mathbf{y}-\mathbf{y}^T\mathbf{y}) && (5)\\ \implies \nabla f(\mathbf{w}) &= 2\phi^T\phi\mathbf{w}-2\phi^T\mathbf{y} && (6) \end{aligned}$$

we would have

$$\begin{aligned} \nabla f(\mathbf{w}^*) &= 0 && (7)\\ \implies 2\phi^T\phi\mathbf{w}^*-2\phi^T\mathbf{y} &= 0 && (8)\\ \implies \mathbf{w}^* &= (\phi^T\phi)^{-1}\phi^T\mathbf{y} && (9) \end{aligned}$$

5.3 Point 2

Is $\nabla^2 f(\mathbf{w}^*)$ positive definite ?
i.e. $\forall \mathbf{x} \neq 0$, *is* $\mathbf{x}^T\nabla f(\mathbf{w}^*)\mathbf{x} > 0$? (A sufficient condition for local minimum)

(Note : Any positive definite matrix is also positive semi-definite)
`(Cite : Section 3.12 & 3.12.1)[3]`

$$\begin{aligned} \nabla^2 f(\mathbf{w}^*) &= 2\phi^T\phi && (10)\\ \implies \mathbf{x}^T\nabla^2 f(\mathbf{w}^*)\mathbf{x} &= 2\mathbf{x}^T\phi^T\phi\mathbf{x} && (11)\\ &= 2(\phi\mathbf{x})^T\phi\mathbf{x} && (12)\\ &= 2\left|\left|\phi\mathbf{x}\right|\right|^2 \geq 0 && (13) \end{aligned}$$

And if ϕ ***has full column rank*** ,

$$\phi\mathbf{x} = 0 \quad iff \quad \mathbf{x} = 0 \qquad (14)$$

$\therefore$ If $\mathbf{x} \neq 0, \quad \mathbf{x}^T \nabla^2 f(\mathbf{w}^*)\mathbf{x} > 0$
Example where ϕ doesn't have a full column rank,

$$\phi = \begin{bmatrix} x_1 & x_1^2 & x_1^2 & x_1^3 \\ x_2 & x_2^2 & x_2^2 & x_2^3 \\ \vdots & \vdots & \vdots & \vdots \\ x_n & x_n^2 & x_n^2 & x_n^3 \end{bmatrix} \tag{15}$$

This is the simplest form of linear correlation of features, and it is not at all desirable.

5.4 Point 3

Definition of convex sets and convex functions (Cite : Definition 32 and 35)[2]

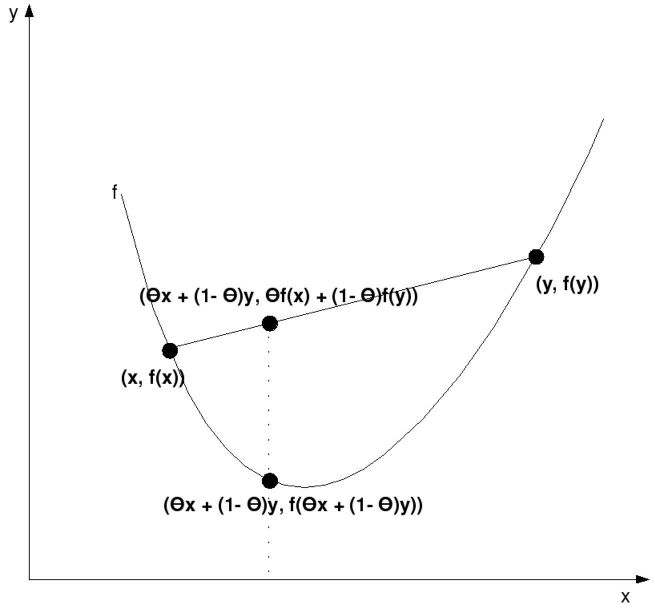

y
f
(Θx + (1- Θ)y, Θf(x) + (1- Θ)f(y))
(y, f(y))
(x, f(x))
(Θx + (1- Θ)y, f(Θx + (1- Θ)y))
x

Figure 8: A sample convex function

$$\therefore f(\theta\mathbf{x} + (1-\theta)\mathbf{y}) \leq \theta f(\mathbf{x}) + (1-\theta)f(\mathbf{y}) \tag{16}$$

Some convex functions : (Cite : Table 4.1, pg-54)[2]

To prove : *Verify that a hyperplane is a convex set.*
Proof :

A Hyperplane $\mathcal{H}$ is defined as $\{\mathbf{x}|\mathbf{a}^T\mathbf{x} = b, \mathbf{a} \neq \mathbf{0}\}$. Let $\mathbf{x}$ and $\mathbf{y}$ be vectors that belong to the hyperplane. Since they belong to the hyperplane, $\mathbf{a}^T\mathbf{x} = b\ and\ \mathbf{a}^T\mathbf{y} = b$. In order to prove the convexity of the set we must show that :

$$\theta\mathbf{x} + (1-\theta)\mathbf{y} \in \mathcal{H},\ where\ \theta \in [0,1] \tag{17}$$

In particular, it will belong to the hyperplane if it's true that :

$$\begin{aligned} \mathbf{a}^T(\theta\mathbf{x} + (1-\theta)\mathbf{y}) &= b && (18)\\ \implies \mathbf{a}^T\theta\mathbf{x} + \mathbf{a}^T(1-\theta)\mathbf{y} &= b && (19)\\ \implies \theta\mathbf{a}^T\mathbf{x} + (1-\theta)\mathbf{a}^T\mathbf{y} &= b && (20)\\ & && (21) \end{aligned}$$

And, we also have $\mathbf{a}^T\mathbf{x} = b\ and\ \mathbf{a}^T\mathbf{y} = b$. Hence $\theta b + (1-\theta)b = b$. [Hence Proved]

So a hyperplane is a convex set.

Q. *Is* $||\phi\mathbf{w} - \mathbf{y}||^2$ *convex?*
A. To check this, we have `(Cite : Theorem 75)[2]` but it is not very practical. We would use `(Cite : Theorem 79)[2]` to check for the convexity of our function. So the condition that has our focus is -

$$\nabla^2 f(\mathbf{w}^*)\ is\ positive\ semi-definite,\ if\ \forall\mathbf{x} \neq 0,\ \mathbf{x}^T\nabla^2 f(\mathbf{w}^*)\mathbf{x} \geq 0 \tag{22}$$

We have,

$$\nabla^2 f(\mathbf{w}) = 2\phi^T\phi \tag{23}$$

So, $||\phi\mathbf{w} - \mathbf{y}||^2$ is convex, since the domain for $\mathbf{w}$ is $\mathbb{R}^n$ and is convex.

Q. *Is* $||\phi\mathbf{w} - \mathbf{y}||^2$ *strictly convex?*
A. Iff ϕ has full column rank.

(Side note : Weka[1])

5.5 Point 4

To prove: *If a function is convex, any point of local minima* $\equiv$ *point of global minima*
Proof - `(Cite : Theorem 69)[2]`

To prove : *If a function is strictly convex, it has a unique point of global minima*
Proof - `(Cite : Theorem 70)[2]`

Since $||\phi\mathbf{w} - \mathbf{y}||^2$ is strictly convex for linearly independent ϕ,

$$\nabla f(\mathbf{w}^*) = 0\ for\ \mathbf{w}^* = (\phi^T\phi)^{-1}\phi^T\mathbf{y} \tag{24}$$

[1] `http://www.cs.waikato.ac.nz/ml/weka/`

Thus, $\mathbf{w}^*$ is a point of global minimum. One can also find a solution to ($\phi^T\phi\mathbf{w} = \phi^T\mathbf{y}$) by Gauss elimination.

5.5.1 Overfitting

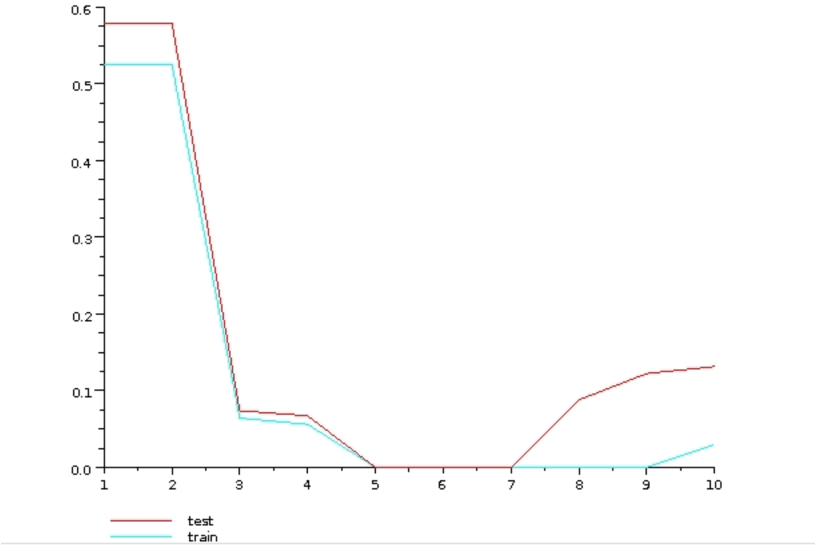


Figure 9: train-RMS and test-RMS values vs t(degree of polynomial) graph

- Too many bends (t=9 onwards) in curve ≡ high values of some $w_i's$
- Train and test errors differ significantly

5.5.2 Next problem

Find

$$\mathbf{w}^* = \text{argmin}_{\mathbf{w}} \|\phi\mathbf{w} - \mathbf{y}\|^2 \;\; s.t. \;\; \|\mathbf{w}\|_p \leq \zeta, \tag{25}$$

where

$$\|\mathbf{w}\|_p = \left(\sum_{i=1}^{n} |w_i|^p\right)^{\frac{1}{p}} \tag{26}$$

5.6 Point 5

Q. *How to solve constrained problems of the above-mentioned type?*
A. General problem format :

$$Minimize\ f(\mathbf{w})\ s.t.\ g(\mathbf{w}) \leq 0 \tag{27}$$

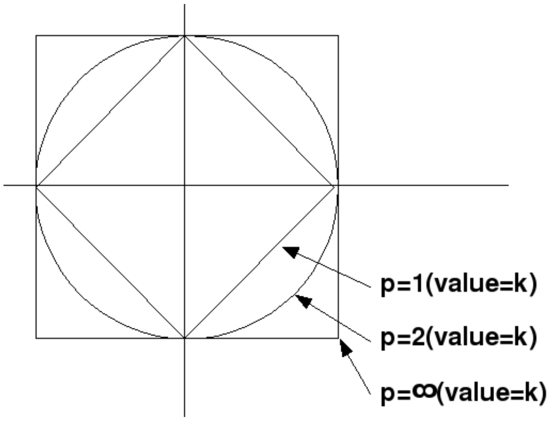


Figure 10: p-Norm curves for constant norm value and different p

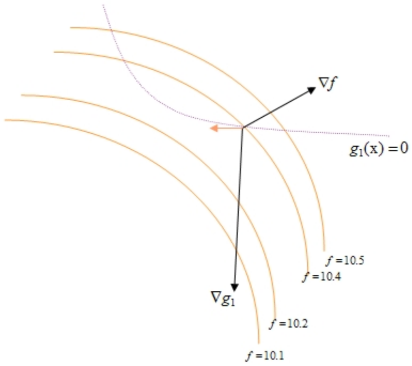


Figure 11: Level curves and constraint regions

At the point of optimality,

$$\textit{Either } g(\mathbf{w}^*) < 0 \quad \& \quad \nabla f(\mathbf{w}^*) = 0 \tag{28}$$

$$\textit{Or } g(\mathbf{w}^*) = 0 \quad \& \quad \nabla f(\mathbf{w}^*) = \alpha \nabla g(\mathbf{w}^*) \tag{29}$$

If $\mathbf{w}^*$ is on the border of g, i.e., $g(\mathbf{w}^*) = 0$,

$$\nabla f(\mathbf{w}^*) = \alpha \nabla g(\mathbf{w}^*) \tag{30}$$

(Duality Theory) `(Cite :  Section 4.4, pg-72)[2]`

Intuition: If the above didn't hold, then we would have $\nabla f(\mathbf{w}^*) = \alpha_1 \nabla g(\mathbf{w}^*) + \alpha_2 \nabla_\perp g(\mathbf{w}^*)$, where by moving in direction $\pm \nabla_\perp g(\mathbf{w}^*)$, we remain on boundary $g(\mathbf{w}^*) = 0$, while decreasing/increasing value of f, which is not possible at the point of optimality.

6 Lecture 6 : Regularized Solution to Regression Problem

In last lecture, we derived solution for the regression problem formulated in least-squares sense which was aimed at minimizing rms error over observed data points. We also analysed conditions under which the obtained solution was guaranteed to be a global minima. However, as we observed, increasing the order of the model yielded larger rms error over test data, which was due to large fluctuations in model learnt and consequently due to very high values of model coefficients (weights). In this lecture, we discuss how the optimization problem can be modified to counter very large magnitudes of coefficients. Subsequently, solution of this problem is provided through lagrange dual formulation followed by discussion over obtained solution and impact over test data. Towards the end of the lecture, a very gentle introduction to axiomatic probability is provided.

6.1 Problem formulation

In order to cease coefficients from becoming too large in magnitude, we may modify the problem to be a constrained optimization problem. Intuitively, for achieving this criterion, we may impose constraint on magnitude of coefficients. Any norm for this purpose might give good working solution. However, for mathematical convenience, we start with the euclidean (L_2) norm. The overall problem with objective function and constraint goes as follows:

$$\begin{aligned} &\underset{\mathbf{w}}{\text{minimize}} && (\Phi\mathbf{w} - Y)^T(\Phi\mathbf{w} - Y) \\ &\text{such that} && ||\mathbf{w}||_2^2 \leq \xi \end{aligned} \tag{31}$$

As observed in last lecture, the objective function, namely $f(\mathbf{w}) = (\Phi\mathbf{w}-Y)^T(\Phi\mathbf{w}-Y)$ is strictly convex. Further to this, the constraint function, $g(\mathbf{w}) = \parallel \mathbf{w} \parallel_2^2 - \xi$, is also a convex function. For convex $g(\mathbf{w})$, the set $S = \{\mathbf{w} | g(\mathbf{w}) \leq 0\}$, can be proved to be a convex set by taking two elements $w_1 \in S$ and $w_2 \in S$ such that $g(w_1) \leq 0$ and $g(w_2) \leq 0$. Since $g(\mathbf{w})$ is a convex function, we have the following inequality:

$$\begin{aligned} g(\theta w_1 + (1-\theta)w_2) &\leq \theta g(w_1) + (1-\theta)g(w_2) \\ &\leq 0; \forall\theta \in [0,1], w_1, w_2 \in S \end{aligned} \tag{32}$$

As $g(\theta w_1 + (1-\theta)w_2) \leq 0$; $\forall\theta \in S$, $\forall w_1, w_2 \in S$, $\theta w_1 + (1-\theta)w_2 \in S$, which is both sufficient and necessary for S to be a convex set. Hence, function $g(\mathbf{w})$ imposes a convex constraint over the solution space.

6.2 Duality and KKT conditions

Given convex objective and constraint functions, minima, $\mathbf{w}^*$, can occur in one of the following two ways:

1. $g(\mathbf{w}^*) = 0$ and $\triangledown f(\mathbf{w}^*) = \alpha \triangledown g(\mathbf{w}^*)$
2. $g(\mathbf{w}^*) < 0$ and $\triangledown f(\mathbf{w}^*) = 0$

This fact might be easily visualized from Figure 1. As we can see, the first condition occurs when minima lies on the boundary of function g. In this case, gradient vectors corresponding to function

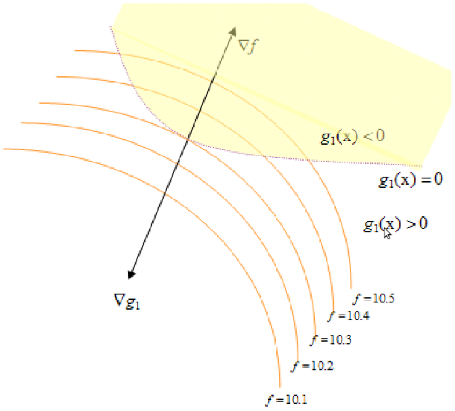


Figure 12: Two plausible scenario for minima to occur: a) When minima is on constraint function boundary, in which case gradients point in the same direction upto constant and b) When minima is inside the constraint space (shown in yellow shade), in which case $\nabla f(\mathbf{w}^*) = 0$.

f and function g, at $\mathbf{w}^*$, point in the same direction barring multiplication by a constant $\alpha \in \mathbb{R}$. Second condition depicts the case when minima lies inside the constraint space (interior of epigraph of function g). This space is shown shaded in Figure 1. Clearly, for this case $\nabla f(\mathbf{w}^*) = 0$ for minima to occur. This primal problem can be converted to dual using lagrange multiplier. According to which, we can convert this problem to objective function augmented by weighted sum of constraint functions in order to get corresponding lagrangian. Such lagrangian might be depicted in the following manner:

$$L(\mathbf{w}, \lambda) = f(\mathbf{w}) + \lambda g(\mathbf{w}); \lambda \in \mathbb{R}^+ \tag{33}$$

Here, we wish to penalize higher magnitude coefficients, hence, we wish $g(\mathbf{w})$ to be negative while minimizing the lagrangian. In order to maintain such direction, we must have $\lambda \geq 0$. Also, for solution $\mathbf{w}^*$ to be feasible, $\nabla g(\mathbf{w}^*) \leq 0$. Due to complementary slackness condition, we further have $\lambda g(\mathbf{w}^*) = 0$, which roughly suggests that lagrange multiplier is zero unless constraint is active at minimum point. As $\mathbf{w}^*$ minimizes lagrangian $L(\mathbf{w}, \lambda)$, gradient must vanish at this point and hence we have $\nabla f(\mathbf{w}^*) + \lambda \nabla g(\mathbf{w}^*) = 0$. In general, optimization problem with inequality and equality constraints might be depicted in the following manner:

$$\begin{aligned} &minimize f(\mathbf{w}) \\ &subject\ to\ g_i(\mathbf{w}) \leq 0; i = 1, 2, \ldots, m \\ &h_j(\mathbf{w}) = 0; j = 1, 2, \ldots, p \end{aligned} \tag{34}$$

Here, $\mathbf{w} \in \mathbb{R}^n$ and domain is intersection of all functions. Lagrangian for such case might be depicted in the following manner:

$$L(\mathbf{w}, \lambda, \mu) = f(\mathbf{w}) + \sum_{i=1}^{m} \lambda_i g_i(\mathbf{w}) + \sum_{j=1}^{p} \mu_j h_j(\mathbf{w}) \tag{35}$$

Lagrange dual function is given as the minimum value of the lagrangian over $\lambda \in \mathbb{R}^m$, $\mu \in \mathbb{R}^p$. Such function might be given in the following manner:

$$z(\lambda, \mu) = \inf_{\mathbf{w}} L(\mathbf{w}, \lambda, \mu) \tag{36}$$

The dual function always yields lower bound for minimizer of the primal formulation. Such dual function is used in characterization of a dual gap, which depicts suboptimality of the solution. Duality gap may be denoted as the gap between primal and dual objectives, $f(\mathbf{w}) - z(\lambda, \mu)$. When functions f and $g_i, \forall i \in [1, m]$ are convex and $h_j, \forall j \in [1, p]$ are affine, Karush-Kuhn-Tucker (KKT) conditions are both necessary and sufficient for points to be both primal and dual optimal with zero duality gap. For above mentioned formulation of the problem, KKT conditions for all differentiable functions (i.e. f, g_i, h_j) with $\hat{\mathbf{w}}$ primal optimal and $(\hat{\lambda}, \hat{\mu})$ dual optimal point may be given in the following manner:

1. $\nabla f(\hat{\mathbf{w}}) + \sum_{i=1}^{m} \hat{\lambda_i} \nabla g_i(\hat{\mathbf{w}}) + \sum_{j=1}^{p} \hat{\mu_j} \nabla h_j(\hat{\mathbf{w}}) = 0$

2. $g_i(\hat{\mathbf{w}}) \leq 0; i = 1, 2, \ldots, m$

3. $\hat{\lambda_i} \geq 0; i = 1, 2, \ldots, m$

4. $\hat{\lambda_i} g_i(\hat{\mathbf{w}}) = 0; i = 1, 2, \ldots, m$

5. $h_j(\hat{\mathbf{w}}) = 0; j = 1, 2, \ldots, p$

6.3 Bound on λ in the regularized least square solution

As discussed earlier, we need to minimize the error function subject to constraint $\|\mathbf{w}\| \leq \xi$. Applying KKT conditions to this problem, if $\mathbf{w}^*$ is a global optimum then from the first KKT condition we get,

$$\nabla_{\mathbf{w}^*}(f(\mathbf{w}) + \lambda g(\mathbf{w})) = 0 \tag{37}$$

$$\tag{38}$$

where, $f(\mathbf{w}) = (\Phi\mathbf{w} - Y)^T(\Phi\mathbf{w} - Y)$ and $g(\mathbf{w}) = \|\mathbf{w}\|^2 - \xi$
Solving we get,

$$2(\Phi^T\Phi)\mathbf{w}^* - 2\Phi^T - 2\lambda\mathbf{w}^* = 0$$

i.e.

$$\mathbf{w}^* = (\Phi^T\Phi + \lambda I)^{-1}\Phi^T\mathbf{y} \tag{39}$$

From the second KKT condition we get,

$$\|\mathbf{w}^*\|^2 \leq \xi \tag{40}$$

From the third KKT condition,

$$\lambda \geq 0 \tag{41}$$

From the fourth condition

$$\lambda\|\mathbf{w}^*\|^2 = \lambda\xi \tag{42}$$

Thus values of $\mathbf{w}^*$ and λ which satisfy all these equations would yield an optimum solution.
Consider equation (39),

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

Premultiplying with $(\Phi^T\Phi + \lambda I)$ on both sides we have,

$$(\Phi^T\Phi + \lambda I)\mathbf{w}^* = \Phi^T\mathbf{y}$$

$$\therefore (\Phi^T\Phi)\mathbf{w}^* + (\lambda I)\mathbf{w}^* = \Phi^T\mathbf{y}$$

$$\therefore \|(\Phi^T\Phi)\mathbf{w}^* + (\lambda I)\mathbf{w}^*\| = \|\Phi^T\mathbf{y}\|$$

By triangle inequality,

$$\|(\Phi^T\Phi)\mathbf{w}^*\| + (\lambda)\|\mathbf{w}^*\| \geq \|(\Phi^T\Phi)\mathbf{w}^* + (\lambda I)\mathbf{w}^*\| = \|\Phi^T\mathbf{y}\| \tag{43}$$

Now , $(\Phi^T\Phi)$ is a nxn matrix which can be determined as Φ is known .
$\|(\Phi^T\Phi)\mathbf{w}^*\| \leq \alpha\|\mathbf{w}^*\|$ for some α for finite $|(\Phi^T\Phi)\mathbf{w}^*\|$. Substituting in previous equation,

$$(\alpha + \lambda)\|\mathbf{w}^*\| \geq \|\Phi^T\mathbf{y}\|$$

i.e.

$$\lambda \geq \frac{\|\Phi^T\mathbf{y}\|}{\|\mathbf{w}^*\|} - \alpha \tag{44}$$

Note that when $\|\mathbf{w}^*\| \to 0, \lambda \to \infty$. This is obvious as higher value of λ would focus more on reducing value of $\|\mathbf{w}^*\|$ than on minimizing the error function.

$$\|\mathbf{w}^*\|^2 \leq \xi$$

Eliminating $\|\mathbf{w}^*\|$ from the equation (14) we get,

$$\therefore \lambda \geq \frac{\|\Phi^T\mathbf{y}\|}{\sqrt{\xi}} - \alpha \tag{45}$$

This is not the exact solution of λ but the bound (15) proves the existance of λ for some ξ and Φ.

6.4 RMS Error variation

Recall the polynomial curve fitting problem we considered in earlier lectures. Figure 2 shows RMS error variation as the degree of polynomial (assumed to fit the points) is increased. We observe that as the degree of polynomial is increased till 5 both train and test errors decrease. For degree > 7, test error shoots up. This is attributed to the overfitting problem (The datasize for train set is 8 points.)
Now see Figure 3 where variation in RMS error and Lagrange multiplier λ has been explored (keeping the polynomial degree constant at 6). Given this analysis, what is the optimum value of λ that must be chosen? We have to choose that value for which the test error is minimum (Identified as optimum in the figure.).

6.5 Alternative objective function

Consider equation (37). If we substitute $g(\mathbf{w}) = \|\mathbf{w}\|^2 - \xi$, we get

$$\nabla_{\mathbf{w}^*}(f(\mathbf{w}) + \lambda \cdot (\|\mathbf{w}\|^2 - \xi)) = 0 \tag{46}$$

This is equivalent to finding

$$\min(\| \Phi\mathbf{w} - \mathbf{y} \|^2 + \lambda \| \mathbf{w} \|^2) \tag{47}$$

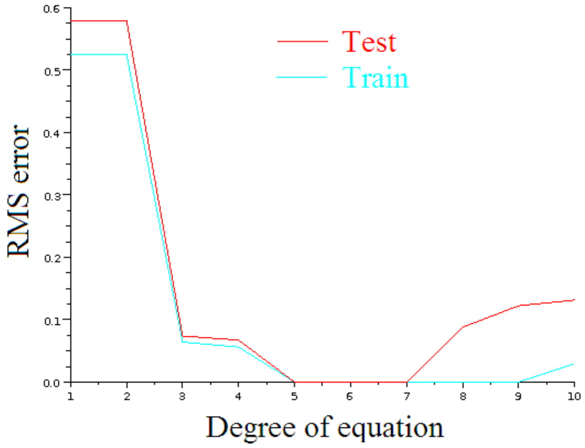


Figure 13: RMS error Vs degree of polynomial for test and train data.

For same λ these two solutions are the same.This form or regression is known as Ridge regression. If we use L_1 norm then it's called as 'Lasso'. Note that $\mathbf{w}^*$ form that we had derived is valid only for L_2 norm.

6.6 A review of probability theory

Let's now review some basics of the probability theory. More details will be covered in the next lecture.

Definition 1. *Sample space (S) : A sample space is defined as a set of all possible outcomes of an experiment. Example of an experiment would be a coin pair toss. In this case $S = \{HH,\ HT,\ TH,\ TT\}$.*

Definition 2. *Event (E) : An event is defined as any subset of the sample space. Total number of distinct events possible is 2^S, where S is the number of elements in the sample space. For a coin pair toss experiment some examples of events could be*

$$\text{for at least one head, E} = \{HH, HT\}$$
$$\text{for all tails, E} = \{TT\}$$
$$\text{for either a head or a tail or both, E} = \{HH, HT, TH, TT\}$$

Definition 3. *Random variable (X) : A random variable is a mapping (or function) from set of events to a set of real numbers. Continuous random variable is defined thus*

$$X : 2^S \to \mathbb{R}$$

On the other hand a discrete random variable maps events to a countable set (e.g. discrete real numbers)

$$X : 2^S \to \textit{Discrete}\ \mathbb{R}$$

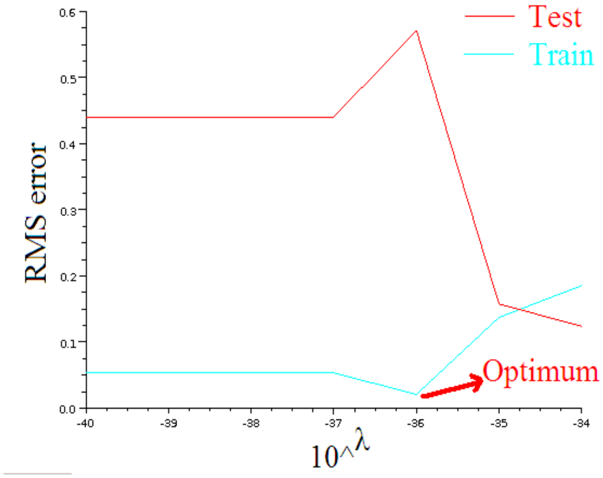


Figure 14: RMS error Vs $10^{\lambda}$for test and train data (at Polynomial degree = 6).

6.6.1 The three axioms of probability

Probability Pr is a number corresponding to events . It satisfies following three axioms,

Axiom 1. *For every event E,* $Pr(E) \in [0, 1]$

Axiom 2. $Pr(S) = 1$ *(Equivalently,* $P(\emptyset) = 0$*)*

Axiom 3. *If* $E_1, E2, \dots, E_n$ *is a set of pairwise disjoint events, then*

$$Pr(\bigcup_{i=1}^{n} E_i) = \sum_{i=1}^{n} Pr(E_i)$$

6.6.2 Bayes' theorem

Let $B_1, B_2, ..., B_n$ be a set of mutually exclusive events that together form the sample space S. Let A be any event from the same sample space, such that $P(A) > 0$. Then,

$$Pr(B_i/A) = \frac{Pr(B_i \cap A)}{Pr(B_1 \cap A) + Pr(B_2 \cap A) + \cdots + Pr(B_n \cap A)} \tag{48}$$

Using the relation $P(B_i \cap A) = P(B_i) \cdot P(A/B_i)$

$$Pr(B_i/A) = \frac{Pr(B_i) \cdot Pr(A/B_i)}{\sum_{j=1}^{n} Pr(B_j) \cdot Pr(A/B_j)} \tag{49}$$

Example 1. *A lab test is* 99% *effective in detecting a disease when in fact it is present. However, the test also yields a false positive for* 0.5% *of the healthy patients tested. If* 1% *of the population has that disease, then what is the probability that a person has the disease given that his/her test is positive?*

Solution 1. *Let, H be the event that a tested person is actually healthy.*

D be the event that a tested person does have the disease.

T be the event that the test comes out positive for a person.

We want to find out $Pr(D/T)$

H and D are disjoint events. Together they form the sample space.

Using Bayes' theorem,

$$P(D/T) = \frac{Pr(D) \cdot Pr(T/D)}{Pr(D) \cdot Pr(T/D) + Pr(H) \cdot Pr(T/H)} \tag{50}$$

Now, Pr(D) = 0.01 (Given)

Since Pr(D)+Pr(H)=1, Pr(H)=0.99

The lab test is 99% effective when the disease is present. Hence, Pr(T/D)=0.99

There is 0.5% chance that the test will give false positive for a healthy person. Hence, Pr(T/H)=0.005 Plugging these values in equation (50) *we get,*

$$\begin{aligned} Pr(D/T) &= \frac{0.01 * 0.99}{0.01 * 0.99 + 0.99 * 0.005} \\ &= \frac{2}{3} \end{aligned}$$

What does this mean? It means that there is 66.66% chance that a person with positive test results is actually having the disease. For a test to be good we would have expected higher certainty. So, despite the fact that the test is 99% effective for a person actually having the disease, the false positives reduce the overall usefulness of the test.

6.6.3 Independent events

Two events E_1 and E_2 are called independent iff their probabilities satisfy

$$P(E_1\, E_2) = P(E_1) \cdot P(E_2) \tag{51}$$

where $P(E_1\, E_2)$means$P(E_1 \cap E_2)$

In general, events belonging to a set are called as mutually independent iff, for every finite subset, $E_1, \cdots, E_n$, of this set

$$Pr(\bigcap_{i=1}^{n} E_i) = \prod_{i=1}^{n} Pr(E_i) \tag{52}$$

7 Lecture 7 : Probability

This lecture gives an overview of the probability theory. It discusses distribution functions; notion of expectation, variance; bernoulli and binomial random variables; and central limit theorem

7.1 Note

- Pr - probability in general of an event
- F - cumulative distribution function
- p - probability distribution function(pdf) or probability mass function(pmf)
- pdf – continuous random variable case
- pmf – discrete random variable case

7.2 Part of speech(pos) example

Problem Statement:-
A set of 'n' words, each of a particular part of speech(noun/verb/etc) is picked. Probability that a word is of part of speech type 'k' is p_k. Assuming the picking of words is done independently, find probability that the set contains a 'noun' given that it contains a 'verb'.

Solution
Let A_k be the probability that the set contains pos type 'k'.
$Pr(A_k) = 1 - (1 - p_k)^n$
where $(1 - p_k)^n$ is that all 'n' words are not of pos of type 'k'.

$Pr(A_{noun}/A_{verb}) = \frac{Pr(A_{noun} \bigcap A_{verb})}{Pr(A_{verb})}$

$Pr(A_{k1} \bigcap A_{k2}) = 1 - (1 - p_{k1})^n - (1 - p_{k2})^n + (1 - p_{k1} - p_{k2})^n$

$Pr(A_{noun}/A_{verb}) = \frac{1-(1-p_{noun})^n-(1-p_{verb})^n+(1-p_{noun}-p_{verb})^n}{1-(1-p_{verb})^n}$

7.3 Probability mass function(pmf) and probability density function(pdf)

pmf :- It is a function that gives the probability that a discrete random variable is exactly equal to some value(Src: wiki).
$p_X(a) = Pr(X = a)$

Cumulative distribution function(Discrete case)
$F(a) = Pr(X <= a)$

pdf :- A probability density function of a continuous random variable is a function that describes the relative likelihood for this random variable to occur at a given point in the observation space(Src:

wiki).

$Pr(X \in D) = \int_D p(x)dx$
where D is set of reals and p(x) is density function.

Cumulative distribution function(Continuous case)
$F(a) = Pr(X <= a) = \int_{-\infty}^{a} p(x)dx$

$f(a) = \frac{dF(x)}{dx}|_{x=a}$

7.3.1 Joint distribution function

If p(x,y) is a joint pdf i.e. for continuous case:
$F(a, b) = Pr(X <= a, Y <= b) = \int_{-\infty}^{b} \int_{-\infty}^{a} p(x, y)dxdy$
$p(a, b) = \frac{\partial^2 F(x,y)}{\partial x \partial y}|_{a,b}$

For discrete case i.e. p(x,y) is a joint pmf:
$F(a, b) = \sum_{x<=a} \sum_{y<=b} p(x, y)$

7.3.2 Marginalization

Marginal probability is then the unconditional probability P(A) of the event A; that is, the probability of A, regardless of whether event B did or did not occur. If B can be thought of as the event of a random variable X having a given outcome, the marginal probability of A can be obtained by summing (or integrating, more generally) the joint probabilities over all outcomes for X. For example, if there are two possible outcomes for X with corresponding events B and B', this means that $P(A) = P(A \bigcap B) + P(A \bigcap B')$. This is called marginalization.

Discrete case:
$P(X = a) = \sum_y p(a, y)$

Continuous case:
$P_x(a) = \int_{-\infty}^{\infty} p(a, y)dy$

7.4 Example

Statement :- X and Y are independent continuous random variables with same density functions.

$$p(x) = \begin{cases} e^{-x} & \text{if } x > 0; \\ 0 & \text{otherwise.} \end{cases}$$

Find density $\frac{X}{Y}$.
Note:- They are indepedent.

Solution

$F_{\frac{X}{Y}}(a) = Pr(\frac{X}{Y} <= a)$
$= \int_0^\infty \int_0^{ya} p(x,y)dxdy$
$= \int_0^\infty \int_0^{ya} e^{-x}e^{-y}dxdy$
$= 1 - \frac{1}{a+1}$
$= \frac{a}{a+1}$

$f_{\frac{X}{Y}}(a) =$ derivative of $F_{\frac{X}{Y}}(a)$ w.r.t a
$= \frac{1}{(a+1)^2} > 0$

7.5 Conditional Density

Discrete case:
$p_X(\frac{x}{Y=y}) = P(\frac{X=x}{Y=y}) = \frac{P(X=x,Y=y)}{P(Y=y)}$

Continuous case:
$p_X(\frac{x}{Y=y}) = \frac{p_{X,Y}(\frac{X}{Y})}{p_Y(y)} = \frac{p_{X,Y}(\frac{X}{Y})}{\int_{-\infty}^{\infty} p(x,y)dx}$

7.6 Expectation

Discrete case: Expectation is equivalent to probability weighted sums of possible values.
$E(X) = \Sigma_i x_i Pr(x_i)$ where X is a random variable

Continuous case: Expectation is equivalent to probability density weighted integral of possible values.
$E(X) = \int_{-\infty}^{\infty} xp(x)dx$

If the random variable is a function of x, then Discrete case:
$E(X) = \Sigma_i f(x_i)Pr(x_i)$ where X is a random variable

Continuous case:
$E(X) = \int_{-\infty}^{\infty} f(x)p(x)dx$

7.6.1 Properties of E(x)

$E[X+Y] = E[X] + E[Y]$

For any constant c and any random variable X
$E[(X-c)^2] \geq E[(X-\mu)^2]$
where $\mu = E[X]$

$E[cX] = cE[X]$

7.7 Variance

For any random variable X, variance is defined as follows:
$Var[X] = E[(X-\mu)^2]$
$\Rightarrow Var[X] = E[X^2] - 2\mu E[X] + \mu^2$
$\Rightarrow Var[X] = E[X^2] - (E[X])^2$

$Var[\alpha X + \beta] = \alpha^2 Var[X]$

7.8 Covariance

For random variables X and Y, covariance is defined as:
$Cov[X,Y] = E[(X-E(X))(Y-E(Y))] = E[XY] - E[X]E[Y]$
If X and Y are independent then their covariance is 0, since in that case
$E[XY] = E[X]E[Y]$
However, covariance being 0 does not necessarily imply that the variables are independent.

7.8.1 Properties of Covariance

$Cov[X+Z,Y] = Cov[X,Y] + Cov[Z,Y]$

$Cov[\Sigma_i X_i, Y] = \Sigma_i Cov[X_i, Y]$

$Cov[X,X] = Var[X]$

7.9 Chebyshev's Inequality

Chebyshev's inequality states that
if X is any random variable with mean μ and variance σ then $\forall k > 0$
$Pr[|X-\mu| \geq k] \leq \frac{\sigma^2}{k^2}$

If n tends to infinity, then the data mean tends to converge to μ, giving rise to the *weak law of large numbers.*

$Pr[|\frac{X_1+X_2+..+X_n}{n} - \mu| \geq k]$ tends to 0 as n tends to ∞

7.10 Bernoulli Random Variable

Bernoulli random variable is a discrete random variable taking values 0,1
Say, $Pr[X_i = 0] = 1 - q$ where $q \epsilon [0, 1]$
Then $Pr[X_i = 1] = q$
$E[X] = (1 - q) * 0 + q * 1 = q$
$Var[X] = q - q^2 = q(1 - q)$

7.11 Binomial Random Variable

Binomial random variable is discrete variable where the distribution is a series of n experiments with 0,1 value. The probability that the outcome of a particular experiment is 1 being q.

$Pr[X = k] = \binom{n}{k} q_k (1 - q)^{n-k}$
$E[X] = \Sigma_i E[Y_i]$ where Y_i is a bernoulli random variable $E[X] = nq$

$Var[X] = \Sigma_i Var[Y_i]$ (since Y_i's are independent)
$Var[X] = nq(1 - q)$

An example of Binomial distribution can be a coin tossed n times and counting the number of times heads shows up.

7.12 Central Limit Theorem

If $X_1, X_2, .., X_m$ is a sequence of i.i.d. random variables each having mean μ and variance σ^2
Then for large m, $X_1 + X_2 + .. + X_m$ is approximately normally distributed with mean mμ and variance mσ^2
If $X\tilde{N}(\mu, \sigma^2)$
Then $P[x] = \frac{1}{\sigma \sqrt[2]{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$

It can be shown by CLT

- $\frac{X_1+X_2+..+X_n-n\mu}{\sigma \sqrt[2]{n}} \tilde{N}(0, 1)$
- Sample Mean: $\hat{\mu}\tilde{N}(\mu, \frac{\sigma^2}{m})$

7.13 Maximum Likelihood and Estimator

Estimator is a function of random space which is meant to approximate a parameter.

- $\hat{\mu}$ is an estimator for μ
- Maximum Likelihood estimator $\hat{q}_{MLE}$ for q
  $\hat{q}_{MLE}$ for q (Parameter of Bern(q) from which we get sample data)

  $L(\hat{X_1}, \hat{X_2}, .., \hat{X_n}|q) = q^{\hat{X_1}}(1-q)^{1-\hat{X_1}}..q^{\hat{X_n}}(1-q)^{1-\hat{X_n}}$

  $\hat{q}_{MLE} = argmax_q L(\hat{X_1}, \hat{X_2}, .., \hat{X_n}|q)$

E.g.: Bernoulli Random Variable

$$p(x) = \mu^x(1-\mu)^{(1-x)}$$
$$D = X_1, X_2, .., X_m \text{is a random sample}$$
$$L(D\|\mu) = \Pi_i \mu_i^x(1-\mu)^{(1-x_i)}$$

GOAL :

$$\hat{\mu}_{MLE} = argmax_\mu L(D|\mu)$$

Equivalently :

$$\hat{\mu}_{MLE} = argmax_\mu logL(D|\mu)$$
$$\frac{dlogL(D|\mu)}{d\mu} = 0 \text{ gives } \hat{\mu}_{MLE} = \frac{\Sigma X_i}{m}$$

Summary :

1. $\hat{\mu}_{MLE}$ is a function of the random sample
2. It is called an estimator in terms of X_i's
3. It is called an estimate in terms of x_i
4. It coincides with sample mean

Recall from CLT

for large m $\Sigma_i X_i$ is similar to $N(m\mu, m\sigma^2)$ if each X_i has
$E(X_i) = \mu$
$V(X_i) = \sigma^2$
Thus :

$\frac{\Sigma_i X_i - m\mu}{\sigma \sqrt[2]{m}} \tilde{N}(0,1)$ and $\hat{\mu}_{MLE}\tilde{N}(\mu, \frac{\sigma^2}{m})$

Question : Given an instantiation of $X_1, X_2, .., X_m$ called Data D $x_1, x_2, .., x_m$
You have MLE estimate
$\frac{\Sigma X_i}{m}$, which is a point estimate
How confident can you be that the actual μ is $\frac{\Sigma X_i}{m}$± z for some z, this is called the interval estimate

7.14 Bayesian estimator

$L(X_1, X_2, .., X_n|\mu) = \Pi_i \mu^{X_i}(1-\mu)^{1-X_i}$

$p(\mu) = \frac{1}{\theta} \forall \mu \epsilon (0,1) \theta \geq 0$

$\int_0^1 p(\mu)\, d\mu = 1$, implies that $\theta = 1$

Posterior $= P(\mu|x_1, x_2, .., x_n)$ (Bayesian Posterior)

$= \frac{L(X_1,X_2,..,X_n\|\mu)p(\mu)}{\int_0^1 L(X_1,X_2,..,X_n|\mu)p(\mu)|,d\mu}$
$= \frac{\mu^{\Sigma_i x_i} 1 - \mu^{\Sigma_i 1 - x_i}}{\int_0^1 \mu^{\Sigma_i x_i} 1 - \mu^{\Sigma_i 1 - x_i}\, d\mu}$

Bayes Estimate $E(\mu|x_1, x_2, .., x_n) = \int \mu P(\mu|x_1, x_2, .., x_n)\, d\mu$
Expected μ under posterior $= \frac{\Sigma_i^m x_i + 1}{m+2}$

Expected value under posterior of parameter μ is called bayes estimate

Beta Distribution is the conjugate prior for Bernoulli distribution
$\beta(\mu|a,b) = \frac{\Gamma(a+b)\mu^(a-1)(1-\mu)^(b-1)}{\Gamma(a)\Gamma(b)}$
Note : Prior and Likelihood should have same form for posterior to have same form as prior. If so ,the chosen prior is called a conjugate prior.

8 Lecture 8

8.1 Bernoulli Distribution

The general formula for probability of a random variable 'x' is

$$p(x) = \mu^x (1-\mu)^{1-x}$$

The likelihood of the data given μ is

$$L(D|\mu) \;=\; \prod_{i=1}^{m} \mu^{x_i} (1-\mu)^{1-x_i}$$

Our goal is to find the maximum likelihood estimate $\hat{\mu}_{MLE} = argmax_\mu \; L(D|\mu)$

Since log is a monotonically increasing function, we can write $\hat{\mu}_{MLE}$ equivalently as :

$$\hat{\mu}_{MLE} = argmax_\mu \; LL(D|\mu)$$

where, LL represents the *log of the likelihood* of the data given μ.

This can also be represented as

$$\hat{\mu}_{MLE} = argmax_\mu \; log(L(D|\mu))$$

$$\Rightarrow \quad \hat{\mu}_{MLE} = argmax_\mu \; \sum_{i=1}^{m} [X_i \ln \mu \; + \; (1-X_i) \ln(1-\mu)]$$

$$\Rightarrow \quad \hat{\mu}_{MLE} = argmax_\mu \; \ln \mu \; (\sum_{i=1}^{m} X_i) \; + \; \ln(1-\mu) \; \sum_{i=1}^{m} (1-X_i)$$

such that, $0 \leq \mu \leq 1$
To find the maxima for $LL(D|\mu)$, we put $\frac{d \; LL(D|\mu)}{d\mu} \;=\; 0$
$\Rightarrow \quad \hat{\mu}_{MLE} = \frac{\sum_{i=1}^{m} X_i}{\sum_{i=1}^{m} X_i \; + \; \sum_{i=1}^{m} (1-X_i)} \;=\; \frac{\sum_{i=1}^{m} X_i}{m}$

Thus, we know that
(1) $\hat{\mu}_{MLE}$ is a function of the random sample.
(2) It is called an *estimator* in terms of $X_i s$.
(3) It is called an *estimate* in terms of $x_i s$.
(4) It coincides with the sample mean.

From central-limit theorem, we know that for large m

$$\sum_{i=1}^{m} X_i \;\sim\; N(m\mu, m\sigma^2)$$

If each X_i has $E[X_i] = \mu$ and $V[X_i] = \sigma^2$

i.e $X_i s$ are normally distributed over m with mean μ and variance σ^2

Thus,

$$\frac{\sum_{i=1}^{m} X_i \ - \ m\mu}{\sigma\sqrt{m}} \ \sim \ N(0,1)$$

and,

$$\hat{\mu}_{MLE} \ \sim \ N(\mu, \frac{\sigma^2}{m})$$

Question. Given an instantiation of ($X_1, X_2, ..., X_m$) called training data D ($x_1, x_2, ..., x_m$), you have Maximum Likelihood Estimation

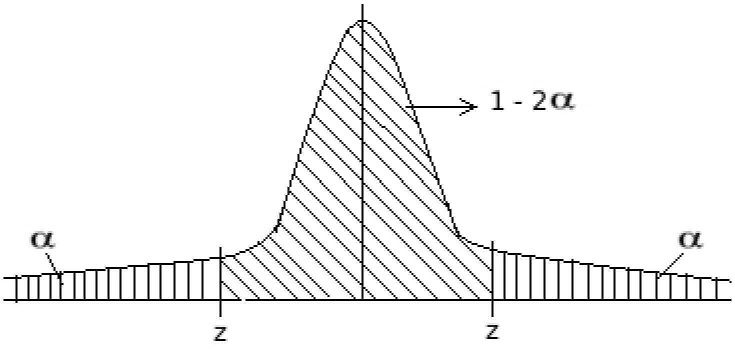
$$\sum_i x_i/m$$

(point estimate). How confident can you be that actual μ is within $(\sum x_i)/m \pm Z$ for some Z.

Answer: Here, we are looking for an interval estimate.

$$\hat{\mu}_{MLE} \pm Z\sqrt{\frac{\hat{\mu} * (1-\hat{\mu})}{m}}$$

where we lookup Z from a table for standard normal distribution value of α for given Z such that P_r(x≥Z) = α



$\mu \in (\hat{\mu}_{MLE} \pm Z\sqrt{\frac{\hat{\mu}*(1-\hat{\mu})}{m}}) is (1-2\alpha)$

8.2 Bayesian Estimation

Likelihood L($X_1, X_2, ..., X_m|\mu) = \prod_{i=1}^{m} \mu^{x_i} * (1-\mu)^{1-x_i}$
p(μ) = $\frac{1}{\theta}$ $\theta \geq 0 \ for all \ \ \mu \in [0,1]$

$$\int_0^1 p(\mu)\,\mathrm{d}\mu \quad = \quad 1 \qquad \Rightarrow \theta = 1$$

Posterior (Bayesian Posterior) is:

$$p_r(\mu|X_1, X_2, ..., X_m) \quad = \quad \frac{L(x_1,x_2,....,x_m|\mu)*p(\mu)}{\int_0^1 L(x_1,x_2,....,x_m|\mu)p(\mu)d\mu}$$

$$= \quad \frac{\mu^{\Sigma x_i} * (1-\mu)^{\Sigma(1-x_i)} * 1}{\int_0^1 \mu^{\Sigma x_i} * (1-\mu)^{\Sigma(1-x_i)} d\mu}$$

$$= \frac{(m+1)!\mu^{\Sigma x_i * (1-\mu)^{\Sigma}(1-x_i)}}{\sum x_i!(m-\sum x_i)!}$$

Expectation $\mathrm{E}(\mu|x_1, x_2, ..., x_m) = \displaystyle\int \mu * p(\mu|x_1, x_2, ..., x_m)d(\mu)$

Expected is under posterior $= \frac{\sum_{i=1}^m x_i + 1}{m+2}$

Thus if we tossed a coin 2 times and $X_i = 1, X_2 = 1$ then

$\mathrm{E}(\mu|1,1) \quad = \quad \frac{2+1}{2+2} \quad = \quad \frac{3}{4}$

$$\hat{\mu}_B = E[\mu|D] = p(\mu|D)$$

$$Beta(\mu|a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\mu^{a-1}\mu^{b-1}$$

Beta is conjugate prior to bernoulli distribution and $\frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}$ is a normalization constant

$$L(x_1...x_n) = \mu^{\Sigma_{i=1}^n x_i}(1-\mu)^{\Sigma_{i=1}^n (1-x_i)}$$

Prior should have the same form as the likelihood with some normalization const

$$p(\mu|D) \propto L(D|\mu)p(\mu)$$

$$If\ \mu_{prior} \approx Beta(\mu|a,b)$$

$$p(\mu|x_1...x_n) = L(x_1...x_n|\mu)P(\mu) = L(x_1...x_n|\mu)Beta(a,b) = \int_o^1 L(x_1...x_n|\mu)Beta(\mu|a,b)du$$

$$= \frac{\Gamma(m+a+b)\mu^{\Sigma_{i=1}^n x_i + a - 1}(1 - \mu^{\Sigma_{i=1}^n (1-x_i)+b-1})}{\Gamma(\Sigma_{i=1}^n x_i + a)\Gamma(\Sigma_{i=1}^n (1-x_i)+b)}$$

$$\approx Beta(\Sigma_{i=1}^n x_i + a, \Sigma_{i=1}^n (1-x_i)+b)$$

$$E_{Beta(a,b)}(\mu|x_1...x_n) = \frac{a}{a+b}$$

$$E_{Beta(a+\Sigma_{i=1}^n x, b+\Sigma_{i=1}^n (1-x_i))}(\mu|x_1...x_n) = \frac{\Sigma_{i=1}^n x_i + a}{\Sigma_{i=1}^n (1-x_i) + \Sigma_{i=1}^n x_i + a + b}$$

for large m, a and $b \ll m$

$$\hat{\mu}_{Bayes} \rightarrow \hat{\mu}_{MLE}$$

Let us say we make k more obsrvations: $y_1 ... y_k$

$$\approx Beta(\Sigma_{i=1}^{n} x_i + \Sigma_{i=1}^{n} y_i + a, \Sigma_{i=1}^{n}(1 - x_i) + \Sigma_{j=1}^{n}(1 - y_j) + b)$$

Multinomial

Say : observation are dice tossing, there are n possible outcomes and each modeled by a vector X_k = $(0_1 0_2 0_3 .....1_k.....)$

$\sum_i x_i^k = 1$ (Note: $X_k^k = 1$ and $X_i^k = 0$ for all i $\neq$ k)

Also, $p(X = x^k) = y_k \approx$ such that $\sum_{k=1}^{n} \mu_k = 1$

Then, p(X = x) = $\prod_{i=1}^{n} \mu_k^{x_k}$

$$E(x) \quad = \quad \sum_{k=1}^{n} X^k * p(X^k) \quad = \quad \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ . \\ . \\ . \\ \mu_n \end{bmatrix}$$

9 Lecture 9 : Multinomial Distribution

This lecture was a continuation of the previous discussion, where we were discussing Multinomial Distribution.Here we discuss about conjugate prior and posterior of Multinomial Distribution. Then we extend the discussion to Gaussian Distribution.

Question: What will be conjugate prior α_i's, which are params of Multinomial?
Answer: Joint Distribution.

$$P\left(\mu_1, \dots \mu_n | \alpha_1, \dots \alpha_n\right) \propto \pi_{i=1}^{n} \mu_i^{\alpha_i - 1} \tag{53}$$

Note: For normalising constant, if P(x) $\propto$ f(x)

$$P(x) = \tfrac{1}{\int f(x)dx} f(x)$$

Since

$$\textstyle\int_{i=1}^{n} P\left(\mu_1, \dots \mu_n | \alpha_1, \dots \alpha_n\right) = 1$$

By Integrating, we get normalisation constant.

$$P\left(\mu_1, \dots \mu_n | \alpha_1, \dots \alpha_n\right) = \frac{\Gamma(\sum_{i=1}^{n} \alpha_i)}{\pi_{i=1}^{n} \Gamma(\alpha_i)} \pi_{i=1}^{n} \mu_i^{\alpha_i - 1} \tag{54}$$

which follows Dir($\alpha_1 \; \dots \; \alpha_n$)

Dirichlet distribution is a generalisation of Beta distribution, just as multinomial is generalisation of Bernouli distribution.

9.0.1 Posterior probability

$$P\left(\mu_1, \dots \mu_n | X_1, \dots X_m\right) = \tfrac{P(X_1, \dots X_m | \mu_1, \dots \mu_n) P(\mu_1, \dots \mu_n)}{P(X_1, \dots X_m)}$$

$$\Rightarrow P\left(\mu_1, \dots \mu_n | X_1, \dots X_m\right) = \frac{\Gamma(\sum_{i=1}^{n} \alpha_i + m)}{\pi_{i=1}^{n} \Gamma(\alpha_i + \sum_{k=1}^{m} X_{k,i})} \pi_{i=1}^{n} \mu_i^{(\alpha_i - 1 + \sum_{k=1}^{m} X_{k,i})} \tag{55}$$

9.0.2 Summary

- For multinomial, the mean at maximum likelihood is given by:

$$\hat{\mu}_{i_{MLE}} = \frac{\sum_{k=1}^{m} X_{k,i}}{m} \tag{56}$$

- Conjugate prior follows Dir($\alpha_1 \dots \alpha_n$)
- Posterior is Dir($\dots \alpha_i + \sum_{k=1}^{m} X_{k,i} \dots$)
- The expectation of μ for Dir($\alpha_1 \dots \alpha_n$) is given by:

$$E\left[\mu\right]_{Dir(\alpha_1 \dots \alpha_n)} = \left[\frac{\alpha_1}{\sum \alpha_1} \cdots \frac{\alpha_1}{\sum \alpha_i}\right] \tag{57}$$

- The expectation of μ for Dir($\ldots \alpha_i + \sum_{k=1}^{m} X_{k,i} \ldots$) is given by:

$$E\left[\mu\right]_{Dir(\ldots\alpha_i+\sum_{k=1}^{m} X_{k,i}\ldots)} = \left[\frac{\alpha_1+\sum_k X_{k,1}}{\sum\alpha_1+m}\ldots\frac{\alpha_j+\sum_k X_{k,j}}{\sum\alpha_i+m}\ldots\right] \tag{58}$$

Observations:

- $\alpha_1 \ldots \alpha_n = 1 \Rightarrow$ Uniform Distribution

- As m$\rightarrow \infty \Rightarrow \hat{\mu}_{Bayes} \rightarrow \hat{\mu}_{MLE}$

- If m=0, $\hat{\mu}_{Bayes} = \mu_{prior}$

9.1 Gaussian Distribution

9.1.1 Information Theory

Let us denote I(X=x) as the measure of information conveyed in knowing value of X=x.

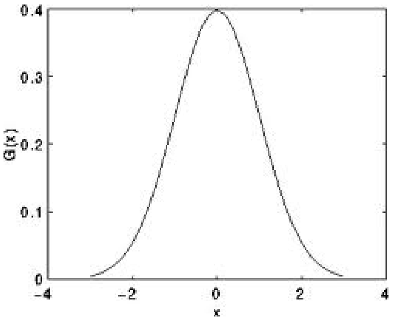


Figure 15: Figure showing curve where Information is not distributed all along.

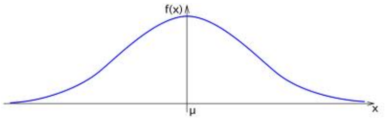


Figure 16: Figure showing curve where Information is distributed.

Question:Consider the following two graphs, say you know probability function p(x), then when is knowing value of X more useful(carries more information).
Ans: It is more useful in the case(2), because more information is conveyed in Figure 15 than in Figure 16.

9.1.2 Expectation for I(X=x):

- If X and Y are independant random variables from same distribution.

$$I(X = x, Y = x) = I(X = x) + I(Y = y) \tag{59}$$

  The above equation can be equivalently stated as follows:

$$I(P(x)P(y)) = I(P(x)) + I(P(y)) \tag{60}$$

  where P(x),P(y) are the probability functions respectively.

- $Ifp(x){>}P(y)$, then

$$I(p(x)){<}I(p(y))$$

  There is only one function which satisfies the above two properties.

$$I(p(x)) = -c\log(p(x)) \tag{61}$$

- The Entropy in the case of discrete random variable can be defined as:

$$E_P\left[I(p(x))\right] = \sum_x -c\log[p(x)] \tag{62}$$

- In the case of continuous random variable it is,

$$E_P\left[I(p(x))\right] = \int_x -c\log[p(x)] \tag{63}$$

  The constant 'C' in the above two equations is traditionally 1.

9.1.3 Observations:

- For Discrete random variable($\sim$ countable domain), the information is maximum for Uniform distribution.
- For Continuous random variable ($\sim$ Finite mean and finite variance), the information for Gaussian Distribution.

Finding $argmax_p E_p \sim$ Infinite domain, subject to

$$\int xp(x)dx = \mu, \int (x-\mu)^2 p(x)dx = \sigma^2$$

The solution would be

$$p(x) = \frac{e^{\frac{-(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma_2}}$$

9.1.4 Properties of gaussian univariate distribution

- If $X \sim N(\mu, \sigma^2)$

$$p(x) = \frac{1}{\sigma\sqrt{2\Pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \ \ where - \infty < x < \infty$$

  then $w_1 X + w_0 \sim N(w_1\mu + w_0, w_1^2\sigma^2)$
  (can prove this using moment generating function)

$$\Phi(N(\mu, \sigma^2)) = E_{N(\mu,\sigma^2)}[e^{tx}] = e^{\mu t + \frac{(\sigma t)^2}{2}}$$

  <u>*Recall*</u>
  $E(X) = \frac{d\phi(p)}{dt}$
  $var(x) = \frac{d^2\phi(p)}{dt^2}$

$$E_{N(\mu,\sigma^2)}[e^{t(w_1 x + w_0)}] = (w_1\mu t + w_0 t + \frac{(\sigma t)^2}{2} \times w_1^2) \sim N(w_1\mu + w_0, w_1^2\sigma^2)$$

- Sum of i.i.d $X_1, X_2, ..., X_n \sim N(\mu, \sigma^2)$ is also normal(gaussian)

  $X_1 + X_2 + ... + X_n \sim N(n\mu, n\sigma^2)$

  In genaral if $X_i \sim N(\mu_i, \sigma_i^2) \implies \sum_{i=1}^n X_i \sim N(\sum \mu_i, \sum \sigma_i^2)$

- Corollary from (1) If $X \sim N(\mu, \sigma^2)$

$$z = \frac{X-\mu}{\sigma} \sim N(0,1) \text{ (Useful in setting interval estimate)}$$

$$\text{(take } w_1 = \frac{1}{\sigma} \ \ and \ \ w_0 = \frac{\mu}{\sigma})$$

- Maximum Likelihood estimate for μ and σ^2

  Given $X_1, X_2, ...X_m .....$ Random Sample.

$$\hat{\mu}_{MLE} = argmax_\mu \prod_{i=1}^m [\frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(X_i-\mu)^2}{2\sigma^2}}]$$

$$= argmax_\mu \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-\sum(X_i-\mu)^2}{2\sigma^2}}$$

  $\hat{\mu}_{MLE} = \frac{\sum_{i=1}^m X_i}{m}$ = sample mean

- With out relaying on central limit theorem Properties (2) and (1) i.e. Sum of i.i.d's $X_1, X_2, ..., X_n \sim N(\mu, \sigma^2)$

$\hat{\mu}_{MLE} = N(\mu, \frac{\sigma^2}{m})$

Similarly

$\hat{\sigma}^2_{MLE} = \frac{\sum_{i=1}^{m}(X_i - \hat{\mu}_{MLE})^2}{m}$ is χ^2 distrbution

$\sim \chi^2_m$

Note:- If $X_1, X_2, ...X_m \sim N(0,1)$

$\sum_i X_i^2 \sim \chi^2_m$ m-degree of freedom

- Coming up with conjugate prior of $N(\mu, \sigma^2)$

  Case (1) σ^2 is fixed and prior on μ

  $\Rightarrow \mu \sim N(\mu_0, \sigma_0^2)$

  Case (2) μ is fixed and σ^2 has prior

  $\Rightarrow \sigma^2 \sim \Gamma$

  case (3) if μ and σ^2 both having the prior

  $\Rightarrow (\mu, \sigma^2) \sim$ Normal gamma distribution $\sim$ Students-t distribution

10 Lecture 10 : Multivariate Gaussian Distribution

We start the lecture by discussing the question given in the previous lecture and then move over to **Multivariate Gaussian Distribution**.

The question was : If $X_1...X_m \sim N(\mu, \sigma^2)$. Then assuming σ^2 is known

$$\hat{\mu}_{ML} = \frac{\displaystyle\sum_{i=1}^{m} X_i}{m}$$
$$\hat{\sigma^2}_{ML} = \frac{\displaystyle\sum_{i=1}^{m} (X_i - \mu)^2}{m}$$

Here $\hat{\sigma^2}_{ML}$ follows the $chi - squared$ distribution

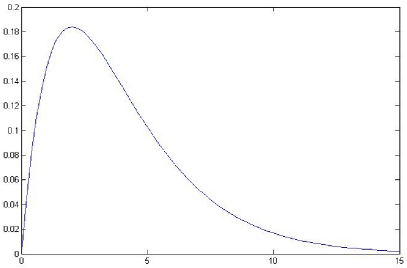


Figure 10 : Figure showing the nature of the $(chi - square)$ distribution of $\hat{\sigma^2}_{ML}$

$$LL(D|\mu, \sigma) = \frac{\displaystyle\sum_{i=1}^{m} (X_i - \mu)^2}{2\sigma^2} - ln(\sigma\sqrt{2\pi})$$
$$\text{ie}, L(D|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \prod_{i=1}^{m} exp(\frac{-(X_i - \mu)^2}{2\sigma^2})$$

Question : What is the conjugate prior p(μ) if σ^2 is known?

Answer : Gaussian distribution.

Question : What is the conjugate prior p(σ^2) if μ is known?

Answer : Gamma distribution.

Question : What is the conjugate prior p(μ, σ^2) if both are unknown?

Answer : If $X_i \sim N(0,1)$ then

$$\sum_{i=1}^{m} X_i \sim \mathcal{X}_m^2 \text{ and}$$

$$y = \frac{z}{\sum X_i^2} \sim t_n \text{ . (where } z \sim \mathcal{N}(0,1))$$

Here y follows *students-t* distribution

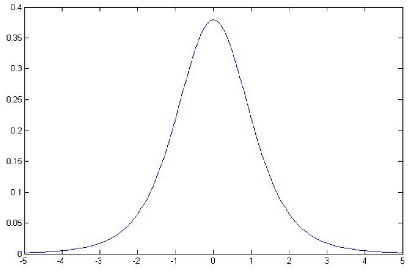


Figure 10: Figure showing students-t distribution of y

10.1 Multivariate Gaussian Variable

Definition : If $\mathbf{X} \sim \mathcal{N}\mu\Sigma)$ (where $\mathbf{x} \in R^n$)) then

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|}\, exp \frac{-(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)}{2}$$

(Note : In pattern recognition, $(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)$ is called the Mahalanobis distance between $\mathbf{x}$ and μ. If Σ = I, it is called Eucledian distance.)

i) Σ can be assumed to be symmetric.
$A = A_{sym} + A_{antisym}$

ii) If Σ is symmetric
$\Sigma = \sum_{i=1}^{n} \lambda_i(q_i q_i^T)$ where q_i's are orthogonal.

Here $\Sigma^{-1} = \sum_{i=1}^{n} \frac{1}{\lambda_i} q_i q_i^T$

$\mathbf{x}' = Q(\mathbf{x}-\mu)$ (Q is a matrix of $q_i^\prime s$ as column vectors)

$$\text{p}(\mathbf{x}') = \prod_{j=1}^{n} (\frac{1}{\sqrt{2\pi\lambda_j}}) exp \frac{-\sum_{i=1}^{n} \frac{(\mathbf{x}'_i)^2}{\lambda_i}}{2\pi\lambda_j}$$

Here the joint distribution has been decomposed as a product of marginals in a shifted and rotated co-ordinate system.

ie, $\mathbf{x}_i's$ are independent.

$$\text{LL}(\mathbf{x}_1...\mathbf{x}_m|\mu,\Sigma) = \frac{-n}{2}ln(2\pi) - \frac{n}{2}ln[|\Sigma|] - \frac{-1}{2}\sum_{i=1}^{m}((\mathbf{x}_i-\mu)^T\Sigma^{-1}(\mathbf{x}_i-\mu))$$

Set $\nabla_\mu LL = 0, \qquad \nabla_\Sigma LL = 0$

$\nabla_\mu LL = [\frac{-1}{2}\sum 2(\mathbf{x}_i-\mu)]\Sigma^{-1} = 0$

Since Σ is invertible,

$\sum(\mathbf{x}_i-\mu) = 0$

ie, $\mu = \frac{\sum \mathbf{x}_i}{m}$

$\hat{\Sigma}_{ML} = \frac{1}{M}\sum_{i=1}^{m}(\mathbf{x}_i-\hat{\mu}_{ML})(\mathbf{x}_i-\hat{\mu}_{ML})^T$

Here $\hat{\Sigma}_{ML}$ is called emperical co-variance matrix in statistics.

$\hat{\mu}_{ML} \sim N(\mu,\Sigma)$

$\text{E}[\hat{\mu}_{ML}] = \mu$

Here $\hat{\mu}_{ML}$ is an unbiased estimator.

10.1.1 Unbiased Estimator

An estimator e(θ) is called unbiased estimator of θ if $E[e(\theta)] = \theta$

If $e_i(\theta), e_2(\theta), ..., e_k(\theta)$ are unbiased estimators and $\sum_{i=1}^{k}\lambda_i = 1$ then $\sum_{i=1}^{k}\lambda_i e_i(\theta)$ is also unbiased estimator.

Since $E(\hat{\Sigma}_{ML}) = \frac{m-1}{m}\Sigma$, $\hat{\Sigma}_{ML}$ is a biased estimator.

An unbiased estimator for Σ is therefore $\frac{1}{m-1}\sum_{i=1}^{m}(\mathbf{x}_i-\hat{\mu}_{ML})(\mathbf{x}_i-\hat{\mu}_{ML})^T$

Question : If $\epsilon \sim N(0,\sigma^2)$

$y = \mathbf{w}^T\phi(x) + \epsilon \qquad$ where $\mathbf{w},\ \phi(x) \in R^n$

then $y \sim N(\mathbf{w}^T\phi(x), \sigma^2)$

$$\text{p}(y|x,w) = \frac{1}{\sqrt{2\pi\sigma^2}} exp(\frac{(y-\phi^T(x)\mathbf{w})^2}{2\sigma^2})$$

$$E[Y(\mathbf{w},x)] = \mathbf{w}^T\phi(x) = \mathbf{w}_0^T + \mathbf{w}_1^T\phi_1(x) + ... + \mathbf{w}_n^T\phi_n(x)$$

$$\phi(x) = [1\ \phi_1(x)\ ...\ \phi_n(x)]$$

Given random sample D

$$D = \begin{bmatrix} y_1 & \phi_1(x) \\ y_2 & \phi_2(x) \\ : & : \\ : & : \\ y_n & \phi_n(x) \end{bmatrix}$$

$$\text{LL}(y_1...y_m|x_1...x_m, \mathbf{w}, \sigma^2) = \frac{-m}{2} ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(\mathbf{w}^T\phi(x)_i - y_i)^2$$

Given σ^2

$$\hat{\mathbf{w}}_{ML} = \text{argmax}\ \ LL(y_1...y_m|x_1...x_m, \mathbf{w}, \sigma^2)$$

$$= \text{argmax}\ \sum_{i=1}^{m}(\mathbf{w}^T\phi(x)_i - y_i)^2$$

10.2 Dealing with Conjugate Priors for Multivariate Gaussian

The congjugate prior for multivariate gaussian distibution if μ, σ^2 are known is given as

$$P(\mu) = \mathcal{N}(\mu_0, \sigma_0^2)$$

$$P(x) \sim \mathcal{N}(\mu, \sigma^2)$$

$$P(\mu|x_1...x_m) = \mathcal{N}(\mu_m, \sigma_m^2)$$

$$\mu_m = (\frac{\sigma^2}{m\sigma_0^2 + \sigma^2}\mu_0) + (\frac{m\sigma_0^2}{m\sigma_0^2 + \sigma^2}\hat{\mu}_{ML})$$

$$\frac{1}{\sigma_m^2} = \frac{1}{\sigma_0^2} + \frac{m}{\sigma^2}$$

If for $y(x, \mathbf{w}) \sim \mathcal{N}(\phi^T(x)\mathbf{w}, \sigma^2)$

$P(\mathbf{w}) \sim \mathcal{N}(\mu_0, \Sigma_0)$

$P(\mathbf{w}|x_1...x_m) = \mathcal{N}(\mu_m, \sigma_m^2)$

$\mu_m = \Sigma_m(\Sigma_0^{-1}\mu_0 + \sigma^2\phi^T y)$

$\Sigma_m^{-1} = \Sigma_0^{-1} + \sigma^2\phi^T\phi$

11 Lecture 11

11.1 Recall

For bayesian estimation in the univariate case with fixed σ where $\mu \sim \mathcal{N}(\mu_0, {\sigma^2}_0)$ and $x \sim \mathcal{N}(\mu, \sigma^2)$

$$\frac{1}{\sigma^2} = \frac{m}{\sigma^2} + \frac{1}{\sigma_0^2}$$
$$\frac{\mu_m}{{\sigma^2}_m} = \frac{m}{\sigma^2}\hat{\mu}_{mle} + \mu_0$$

such that $p(x|D) \sim \mathcal{N}(\mu_m, {\sigma_m}^2)$. m/σ^2 is due to noise in observation while $1/\sigma_0^2$ is due to uncertainity in μ. For the Bayesian setting for the multivariate case with fixed Σ

$$\mathbf{x} \sim \mathcal{N}(\mu, \Sigma),\ \mu \sim N(\mu_0, \Sigma_0)\ \&\ p(\mathbf{x}|D) \sim \mathcal{N}(\mu_m, \Sigma_m)$$
$$\Sigma_m^{-1} = m\Sigma^{-1} + \Sigma_0^{-1}$$
$$\Sigma_m^{-1}\mu_m = m\Sigma^{-1}\hat{\mu}_{mle} + \Sigma_0^{-1}\mu$$

11.2 Bayes Linear Regression

The Bayesian interpretation of probability can be seen as an extension of logic that enables reasoning with uncertain statements. Bayesian linear regression is a Bayesian alternative to ordinary least squares regression.

$$y = w^T\phi(x) + \varepsilon$$
$$\varepsilon \sim N(0, \sigma^2)$$
$$w \sim N(0, \Sigma_0)$$
$$\hat{w}_{MLE} = (\phi^T\phi)^{-1}y$$

Finding μ_m and Σ_m :

$$\Sigma_m^{-1}\mu_m = \Sigma_0^{-1}\mu_0 + \phi^T y/\sigma^2$$
$$\Sigma_m^{-1} = \Sigma_0 + \frac{1}{\sigma^2}\phi^T\phi$$

Setting $\Sigma_0 = \alpha I$ and $\mu_0 = \mathbf{0}$

$$\Sigma_m^{-1}\mu_m = \phi^T y/\sigma^2$$
$$\Sigma_m^{-1}\mu_m = \frac{1}{\alpha}I + \phi^T\phi/\sigma^2$$
$$\mu_m = \frac{(I/\alpha + \phi^T\phi/\sigma^2)^{-1}\phi^T y}{\sigma^2}$$

But since σ^2/α is nothing but the λ in ridge regression, this can be written as

$$\mu_m = (\lambda I + \phi^T\phi/\sigma^2)^{-1}\phi^T y$$
$$\Sigma_m^{-1} = \frac{I}{\alpha} + \frac{\phi^T\phi}{\sigma^2}$$

which are similar to the results obtained in ridge regression.

What is the Bayes Estimator here?

$$\begin{aligned}\hat{w}_{Bayes} &= E_{p(w|X_1...X_m)}[w] \\ &= (\phi^T\phi + \frac{\sigma}{\alpha})^{-1}\phi^T y\end{aligned}$$

which is the least square solution for ridge regression.

$$\hat{w}_{MAP} = \text{argmax}_w\, p(w|X_1...X_n)$$

which is the point w at which posterior distribution peak (mode).
For Gaussian distribution, mode is the same as the mean.

$$mean = \hat{w}_{Bayes}. \tag{64}$$

Find $P(y|X_1...X_m) \sim$ for linear regression
In the context of multivariate gaussian.

$$\begin{aligned}p(x|D) &= p(X|X_1...X_m) \\ &= \int_\mu p(x|\mu)p(\mu|D)d\mu \\ &\sim N(\mu_n, \Sigma + \Sigma_n)\end{aligned}$$

	Point?	$p(x\|D)$
MLE	$\hat{\theta}_{MLE} = \text{argmax}_\theta\, LL(D\|\theta)$	$p(x\|\theta_{MLE})$
Bayes Estimator	$\hat{\theta}_B = E_{p(\theta\|D)}E[\theta]$	$p(x\|\theta_{MLE})$
MAP	$\hat{\theta}_{MAP} = \text{argmax}_\theta\, p(\theta\|D)$	$p(x\|\theta_{MAP})$
Pure Bayesian		$p(\theta\|D) = \frac{p(D\|\theta)p(\theta)}{\int p(D\|\theta)p(\theta)d\theta}$ $p(D\|\theta) = \prod_{i=1}^{m} p(x_i\|\theta)$ $p(x\|D) = \int p(x\|\theta)p(\theta\|D)d\theta$

where θ is the prior vector.

11.3 Pure Bayesian - Regression

$$
\begin{aligned}
p(y|X_1...X_m) &= \int_w p(y|w)p(w|D)dw \\
&where \\
y &= w^T\phi(X) + \varepsilon \\
\varepsilon &\sim N(0,\sigma^2) \\
w &\sim N(0,\alpha I) \\
&\sim N(\mu_m^T\phi(x),\sigma_m^2)
\end{aligned}
$$

Also we have

$$
\begin{aligned}
\sigma_m^2 &= \sigma^2 + \phi^T\Sigma_m\phi \\
\Sigma_m^{-1} &= \frac{I}{\alpha} + \frac{\phi^T\phi}{\sigma^2} \\
\mu_m &= \frac{\Sigma_m\phi^T y}{\sigma^T} \\
\mu_m &= (\phi^T\phi + \frac{\sigma^2}{\alpha})^{-1}\phi^T y
\end{aligned}
$$

Since $x \sim N(\mu,\sigma^2)$, we have $\alpha x + \beta = N(\alpha\mu + \beta, \alpha^2\sigma^2)$. And since we know that $\varepsilon \sim N(0,\sigma^2)$. Using $y = w^T x + \varepsilon$, we get

$$y \sim N(w^T x, \sigma^2)$$

11.4 Sufficient Statistic

A Statistic is called sufficient for θ if $p(D|s,\theta)$ is independent of θ. It can be proved that a statistic s is sufficient for θ iff $p(D|\theta)$ can be written as $p(D|\theta) = g(s,\theta)h(D)$. For case of gaussian we have

$$g(s,\mu) = \exp(-m/2\mu^T\Sigma^{-1}\mu + \mu^T\Sigma^{-1}\sum_{i=1}^{n} x_i)$$

$$h(x_1,x_2...x_m) = 1/2\pi^{nm/2}|\Sigma|^{m/2}\exp(-1/2\sum_{i=1}^{m} x_i^T\Sigma^{-1}x_i)$$

Thus, we see that for the normal distribution, $p(D|\mu) = g(s,\mu)h(D)$.

11.5 Lasso

We have $Y = \mathbf{w}^T\phi(x) + \varepsilon$ where $\varepsilon \sim N(0,\sigma^2)$. Here $w \sim$ Laplace distribution. Then it turns out that $\hat{w}_{MAP}$ with laplace prior is $\hat{w}_{MAP} = \text{argmax}_w \sum_{i=1}^{m} ||w^T\phi(x_i) - y_i||^2 + \lambda||w||_{l1}$

Here $\lambda||w||_{l1}$ is basically the penalty function, also called **lasso**. Recall $\hat{w}_{MAP}$ with guassian prior is $\hat{w}_{MAP} = \text{argmax}_w \sum_{i=1}^{m} ||w^T\phi(x_i) - y_i||^2 + \lambda||w||^2{}_{l1}$

Here $\lambda||w||^2{}_{l2}$ is basically penalty function also called **ridge regression**.
Refer Section 7 from Sir's notes for more details on this topic.

Lasso generally yields sparser solutions. What this means is that if you have $\phi_1(x), \phi_2(x)....\phi_k(x)$ each of them with weights $w_1, w_2...w_k$ We may want to reduce the number of non-zero k. In general, if we use ridge regression those parameter which are irrelevant may have some small weights but lasso tends to set have has zero weights

12 Lecture 12 : Bias-Variance tradeoff

Key terms : Bias, Variance, Expected loss, Noise

When we design a particular machine learning model (function), we wish that it should be able to predict the output accurately and it should be able to do it independent of the sample training data it has been trained with. In the following section we are going to show that there is a trade off between the two main objectives. Let us understand the terms bias and variance. **Variance** of a machine learning model is the variance in the prediction of a value when trained over different training data. **Bias** is related to how much the prediction varies from the actual observed values in the training data.

12.1 Expected Loss

Suppose we are interested in finding the expected loss value of the function. We are given a training data $T_{\mathcal{D}}$, a distribution over x and target variable y say $P(x, y)$ and desired fuction $f(x)$ (here by $f(x)$, we actually mean $f(x, T_{\mathcal{D}})$, i.e. a function trained with respect to the training data $T_{\mathcal{D}}$)to approximate y. We want to find out the expected loss of a model with respect to all the data we have in our distribution. Usually squared error is chosen as a measure of loss the model. First we will derive the loss for a training data and then for the whole distribution.

$$L(y, x, f(x)) = (f(x) - y)^2$$

We want to find a function $f(.)$ whose expected value of the loss over the instances in the training data, is minimum. Find

$$\text{argmin}_f \; E_{f,P(x,y)}[L(y, x, f(x))]$$

where $E_{f,P(x,y)}[L(y, x, f(x))]$ denotes expectation of the loss with a fixed $f()$ and $P(,)$.

$$\begin{aligned}
E_{f,P(x,y)}[L] &= \int_y \int_x L(y, x, f)P(x, y)dxdy \\
&= \int_y \int_x (f(x) - y)^2 P(x, y)dxdy \\
&= \int_y \int_x (f(x) - E[y/x] + E[y/x] - y)^2 P(x, y)dxdy \\
&= \int_y \int_x (f(x) - E[y/x])^2 P(x, y)dxdy + \int_y \int_x (E[y/x] - y)^2 P(x, y)dxdy \\
&+ \int_y \int_x 2(f(x) - E[y/x])(E[y/x] - y)P(x, y)dxdy
\end{aligned}$$

Let us rewrite the third term.

$$\int_y \int_x 2(f(x) - E[y/x])(E[y/x] - y)P(x, y)dxdy \quad = \int_x 2(f(x) - E[y/x]) \int_y (E[y/x] - y)P(y/x)dyP(x)dx \tag{65}$$

Since $\int_y yP(y/x)dy = E[y/x]$, the inner integral in (65) is 0. So

$$E_{f,P(x,y)}[L] \quad = \int_y \int_x (f(x) - E[y/x])^2 P(x, y)dxdy + \int_y \int_x (E[y/x] - y)^2 P(x, y)dxdy \tag{66}$$

Here the second term is independent of $f()$. We have to consult only the first term to find out the $f()$ which will minimize th expecation. From the first term, it is clear that **minimum loss will**

be obtained when $f(x)$ **equals** $E[y/x]$. The minimum expected loss is given by

$$E_{f,P(x,y)}[L] \quad = \int_y \int_x (E[y/x] - y)^2 P(x,y) dx dy \tag{67}$$

This is the minimum loss we can expect for a given training data. Now let us find out the expected loss over different training data. Let

$$E_{T_\mathcal{D}}[f(x, T_\mathcal{D})] = \int_{T_\mathcal{D}} f(x, T_\mathcal{D}) p(T_\mathcal{D}) dT_\mathcal{D}$$

Earlier we have found that the only tweakable component in expected loss is $(f(x) - E[y/x])^2$. Now we will find out the expected loss over all the trainind data by finding expectation of the expression over the distribution of training data.

$$\begin{aligned}\int_{T_\mathcal{D}} (f(x, T_\mathcal{D}) - E[y/x])^2 p(T_\mathcal{D}) dT_\mathcal{D} &= E_{T_\mathcal{D}}[(f(x, T_\mathcal{D}) - E[y/x])^2] \\ &= E_{T_\mathcal{D}}\Big[\big\{f(x, T_\mathcal{D}) - E_{T_\mathcal{D}}[f(x, T_\mathcal{D})] + E_{T_\mathcal{D}}[f(x, T_\mathcal{D})] - E[y/x]\big\}^2\Big] \\ &= E_{T_\mathcal{D}}\Big[\big\{f(x, T_\mathcal{D}) - E_{T_\mathcal{D}}[f(x, T_\mathcal{D})]\big\}^2 + \big\{E_{T_\mathcal{D}}[f(x, T_\mathcal{D})] - E[y/x]\big\}^2 \\ &\quad - 2\big\{f(x, T_\mathcal{D}) - E_{T_\mathcal{D}}[f(x, T_\mathcal{D})]\big\}\big\{E_{T_\mathcal{D}}[f(x, T_\mathcal{D})] - E[y/x]\big\}\Big]\end{aligned}$$

Since

$$E_{T_\mathcal{D}}\Big[\big\{f(x, T_\mathcal{D})\Big] = E_{T_\mathcal{D}}[f(x, T_\mathcal{D})]$$

and the other factors are independent of $T_\mathcal{D}$, the third term vanishes. Finally

$$\begin{aligned}\int_{T_\mathcal{D}} (f(x) - E[y/x])^2 p(T_\mathcal{D}) dT_\mathcal{D} &= E_{T_\mathcal{D}}\Big[\big\{f(x, T_\mathcal{D}) - E_{T_\mathcal{D}}[f(x, T_\mathcal{D})]\big\}^2 + \big\{E_{T_\mathcal{D}}[f(x, T_\mathcal{D})] - E[y/x]\big\}^2\Big] \\ &= E_{T_\mathcal{D}}\Big[\big\{f(x, T_\mathcal{D}) - E_{T_\mathcal{D}}[f(x, T_\mathcal{D})]\big\}^2\Big] + \big\{E_{T_\mathcal{D}}[f(x, T_\mathcal{D})] - E[y/x]\big\}^2 \\ &= Variance + Bias^2\end{aligned}$$

Variance of $f(x, \mathcal{D}) = E_{T_\mathcal{D}}\Big[\big\{f(x, T_\mathcal{D}) - E_{T_\mathcal{D}}[f(x, T_\mathcal{D})]\big\}^2\Big]$ and Bias $= E_{T_\mathcal{D}}[f(x, T_\mathcal{D})] - E[y/x]$ Putting back in (66) the $Expectedloss = Variance + Bias^2 + Noise$

Let us try to understand what this means. Consider the case of regression. The loss of the prediction depends on many factors such as complexity of the model (linear, ..) the parameters and the measurements etc. The noise in the measurement can cause loss of prediction. That is given by third term. Similarly the complexity of the model can contribute to the loss.

If we were to take the linear regression with a low degree polynomial, we are introducing a bias that the dependency of the predicted variable is simple. Similarly when we add a regularizer term, we are implicitly telling that the weights are not big, is also a kind of bias. The prediction we otained may not be accurate. In these cases the predicted values may not have much correlation with the sample points we took. So the predictions remains more or less the same over different samples. That is for different samples the predicted values does not vary much. The prediction is more generalizable over the samples.

Suppose we complicate our regression model by increasing degree of the polynomial used. As we have seen in previous classes, we used to obtain a highly wobbly curve which pass through almost all points in the training data. This is an example for less bias. For a given training data our prediction could be very good. If we were to take another sample data we would have obtained another curve which pass through all the new points but with a drastic difference from the current curve. Our predictions are accurate for the training sample chosen, but at the same time they are highly correlated to the sample we have chosen. For different training data chosen, the variance of the prediction is very high. So the model is not generalizable over the samples.

We saw that when we decrease the bias the variance increases and vice versa. The more complex the model is, the less bias we have and more the variance. Both are contrary to each other. The ideal complexity of the model should be related with the complexity of the actual relation between the dependent and independent variable.

I recommend the reference [4] for a good example.

13 Lecture 13

These are the topics discussed in today's lecture:

1. Conclude Bias-Variance
2. Shrinkage - Best Subset
3. Mixture models
4. Empirical Bayes

13.1 Conclude Bias-Variance

13.1.1 Summary

$$\int_{T_D}\int_y\int_x (f(x) - E[y/x])^2 dx\ dy\ dT_D = \int_{\mathbf{x}} \Bigg\{ \Big\{ E_{T_D}\Big[f(x,T_D) - E[y/x] \Big] \Big\} \tag{1}$$

$$+ E_{T_D}\Bigg[\Big\{ f(x,T_D) - E_{T_D}\big[f(x,T_D) \big] \Big\}^2 \Bigg] \Bigg\} d\mathbf{x} \tag{2}$$

$$- \int_y \int_{\mathbf{x}} \Big\{ E[y/x] - y \Big\}^2 d\mathbf{x}\ dy \tag{3}$$

In the above equation, T_D represents the random sample. $\mathbf{x}$, y represent the *data distribution*. E$[Y/X]$ is the expected value optimal with respect to least squares.

(1) represents the *bias*,
(2) represents the *variance* and
(3) represents the *intrinsic noice*. More the flexbility of the *hypothesis* function, less is the *bias* and more is the *variance*

Now, does this analysis apply to Bayesian Regression?

E$[X/T_D]$ is something like a posterior distribution and expected value is there. But there is no concept of f(x, T_D) which is the posterior estimate.

13.1.2 Bayesian Linear Regression(BLR)

$$P(y/x, \mathbf{y}_T, \alpha, \sigma^2) = N({\mu_m}^T\ \phi(x), \sigma^2 m)$$

where

$$\mu_m = \frac{1}{\sigma^2}\ \sigma_m \phi^T \mathbf{y}_T \quad and$$

$$\Sigma_m{}^{-1} = \frac{1}{\alpha}\ I + \frac{1}{\sigma^2}\ \phi^T\ \phi$$

equation

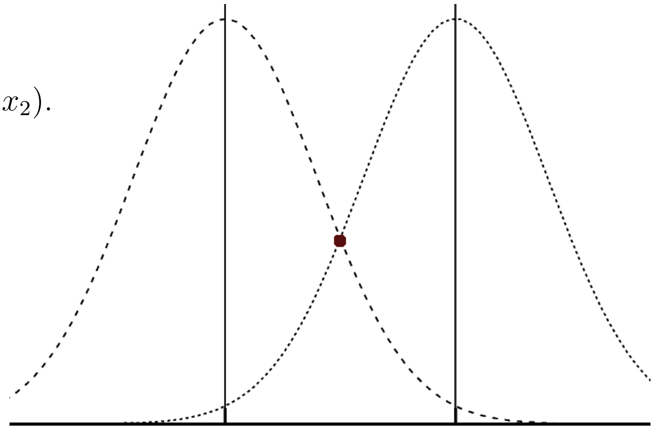
The Basis function:

$$\phi^T = \begin{bmatrix} \phi_1(x_1) & \phi_2(x_1) & .... & .... \\ \phi_1(x_2) & \phi_2(x_2) & .... & .... \\ .... & .... & .... & .... \\ .... & .... & .... & .... \end{bmatrix}$$

If two points are far way from mean, but close to each other then $\phi^T\ \phi$ will increase and $\Sigma_m{}^{-1}$ will also increase. Therefore variance will decrease.

This can also be interpreted as, points which are far apart are positively contributing by giving less variance i.e; less uncertainity.

Also, Uncertainity with the prior is represented by $\frac{1}{\alpha}\ I$, and uncertainity in the estimator that we are explicitly modelling for, is represented by $\frac{1}{\alpha^2}\ \phi^T\ \phi$.



Assume, the peak points of the two *gaussian* distributions are at $\phi(x_1)$ *and* $\phi(x_2)$. The point represented in **red** has equal contributions from both the *gaussians*.

Now, If $\phi(x_1)^T\ \phi(x_2)$ is large, assuming ϕ is *normalized*, standard deviation increases. In *gaussian* distribution, at regions away from the mean, $\phi(x_i)$ will be large.

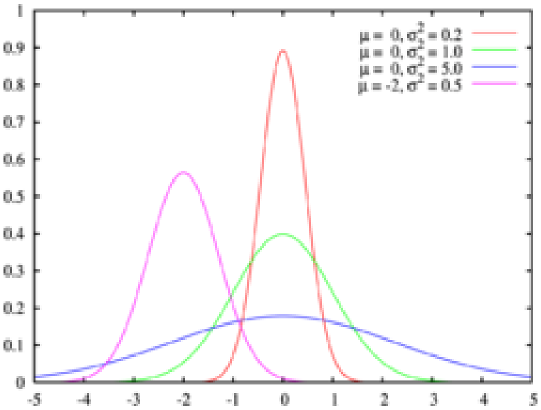
13.1.3 General Problems with Standard Distribution

1. Single Mode - N(μ, Σ)

2. A non-trivial $\Sigma \implies {}^nC_r$ parameters. As dimension of data n grows, n^2 will grow in Σ

3. Search space is sparse.

Mixture of Gaussians



$$p(x) = \sum_{i=1}^{n} \alpha_i p_i(x|z = i) \text{ } (p_i(x) \text{ is a different distribution})$$

$$\sum_{i=1}^{n} \alpha_i = 1$$

p $\implies$ p is a convex combination of individual distribution.

What is form of distribution ?

For the i^{th} distribution taking α_i as the mean where α_i is a multinomial

Ex : *Mixture of Gaussians* :

$p(x) = \Sigma_{i=1}^{n} \alpha_i N(x|\mu_i, \Sigma_i)$

$X \sim N(\mu_i, \Sigma_i)$

$p(x) = p(x|\mu_i, \Sigma_i)$

Issues

1. Number of K's

2. Estimating μ_i's and Σ_i's

3. Estimating α_i's

Classification Perspective

Assume data :

$$\begin{bmatrix} X_1, & 1 \\ X_2, & 1 \\ .... & .. \\ .... & .. \\ X_m, & 3 \end{bmatrix}$$

(z is a class label)

Question : What is MLE ??

$\Longrightarrow$ It is a Classification Problem

If z value is given :

$P(z = t|y) = P(x|z = t)P(z = t)/P(x)$
It is a supervised classfication problem.

Given data : $\begin{bmatrix} X_1 \\ X_2 \\ .... \\ .... \\ X_m \end{bmatrix}$

We have to estimate using hidden variables as z is not explicitly given. (EM algorithm which can be shown to converge).

Target
1. Implicit estimation of z $E[x|x_i]$
2. Estimate μ_i's with $E[z|x_i]$ in place of z_i
i.e

$$\begin{bmatrix} X_1, & E[z|x_1] \\ X_2, & E[z|x_2] \\ .... & .. \\ .... & .. \\ X_m, & E[z|x_m] \end{bmatrix}$$

Estimating number of K's is hard.It also can be classified as a modelling problem and thus

number of K's depend on the model we chose.

EM Algorithm (Source : Wikipedia)

The Expectation-maximization algorithm can be used to compute the parameters of a parametric mixture model distribution (the a_i's and θ_i's). It is an iterative algorithm with two steps: an "expectation step" and a "maximization step"..

The expectation step

With initial guesses for the parameters of our mixture model, "partial membership" of each data point in each constituent distribution is computed by calculating [[expectation value]]s for the membership variables of each data point. That is, for each data point x_j and distribution Y_i, the membership value $y_{i,j}$ is:

$$y_{i,j} = \frac{a_i f_Y(x_j;\theta_i)}{f_X(x_j)}.$$

The maximization step

With expectation values in hand for group membership, “plug-in estimates“ are recomputed for the distribution parameters.

The mixing coefficients a_i are the arithmetic mean’s of the membership values over the N data points.

$$a_i = \frac{1}{N}\sum_{j=1}^{N} y_{i,j}$$

The component model parameters θ_i are also calculated by expectation maximization using data points x_j that have been weighted using the membership values. For example, if θ is a mean μ

$$\mu_i = \frac{\sum_j y_{i,j} x_j}{\sum_j y_{i,j}}.$$

With new estimates for a_i and the θ_i’s, the expectation step is repeated to recompute new membership values. The entire procedure is repeated until model parameters converge.

13.2 Emperical Bayes

There are two approaches to solve the equation

$\Pr(y \mid D) = \Pr(y \mid < y_1, \phi(x_1) >, < y_2, \phi(x_2) >, ... < y_n, \phi(x_n) >)$

$= \iiint \Pr(y \mid w, \sigma^2) \Pr(w \mid \bar{y}, \alpha, \sigma^2) \Pr(\alpha, \sigma^2 \mid \bar{y}), dw\, d\alpha\, d\sigma^2$
where $\bar{y}$ is the data D

13.2.1 First Approach: Approximate the posterior

The first approach involves approximating the posterior i.e, the second term $\Pr(w \mid \bar{y}, \alpha, \sigma^2)$ as w_{MAP}, i.e, as the mode of the posterior distribution of w which is gaus-

sian. Note that as the number of data points keep increasing $\phi^T\phi$ keeps increasing, hence from the relation
$\Sigma_m^{-}1 = \Sigma_0^{-}1 + \frac{1}{\sigma^2}\phi^T\phi$

it is clear that the posterior variance decreases, hence the distribution of w peaks.

13.2.2 Second Approach: Emperical Bayes

The second approach is to emperically assume some value of the hyperparameters α and σ^2 , say $\hat{\alpha}$ and $\hat{\sigma^2}$ for which the posterior will peak. i.e, we have
$\Pr(y \mid D) \approx \int\limits_{\alpha} \sigma^2 \ \Pr(y \mid w_{MAP}, \sigma^2) \Pr(\alpha, \sigma^2 \mid \bar{y}) \Pr(w \mid \bar{y}, \alpha, \sigma^2), dw\, d\alpha\, d\sigma^2$

$\approx \int \Pr(y \mid w, \sigma^2) \Pr(w \mid \bar{y}, \hat{\alpha}, \hat{\sigma^2})\ dw$ for the chosen $\hat{\alpha}$ and $\hat{\sigma^2}$

$\approx N(\phi^T\mu_m\sigma_m^2)N(\mu_m, \Sigma_m)$

Emperical Bayes finds the $\hat{\alpha}$ and $\hat{\sigma^2}$ such that $\prod_i \Pr(y_i \mid T_D)$ i.e, conditional likelihood is maximised.

13.2.3 Sove the eigenvalue equation

$$(\frac{1}{\sigma^2}\phi^T\phi)u_i = \lambda_i u_i$$

define parameter γ as $\gamma = \sum_i \frac{\lambda_i}{\alpha+\lambda_i}$

then emperical $\hat{\alpha}$ is $\hat{\alpha} = \frac{\gamma}{\mu_m^T\mu_m}$
and emperical $\hat{\sigma^2}$ is $\hat{\sigma^2} = \frac{1}{m-\gamma}\sum_{i=1}^{m}(Y_i - \mu_m^T\phi(x_m))^2$

14 Lecture 14 : Introduction to Classification

The goal in classification is to take an input vector $\mathbf{x}$ and assign it to one of D discrete classes

$$f(\mathbf{x}) : \mathbb{R}^n \to D$$

There are many techniques of performing the task of classification. The two main types are

1. **Using a discriminant function**: For e.g.: Consider D = {+, -}.

   $$y = \begin{cases} + & \text{if } f(\mathbf{x}) \geq 0 \\ - & \text{if } f(\mathbf{x}) < 0 \end{cases}$$

   Here $y : \{f(\mathbf{x}) = 0\}$ is called disciminant / decision surface (decision boundary)

   Examples: Least square, Fischer disciminant, Support Vector Machines, Perceptron, Neural Networks

2. **Probabilistic Classification**

   (a) **Disciminative models**: Here we model $Pr(y \,\epsilon\, D \,|\, \mathbf{x})$ directly. For e.g.: we can say that $Pr(D = + \,|\, data)$ comes from a multinomial distribution.

   Examples: Logistic Regression, Maximum Entropy models, Conditional Random Fields

   (b) **Generative models**: Here we model $Pr(y \,\epsilon\, D \,|\, \mathbf{x})$ by modeling $Pr(\mathbf{x} \,|\, y \,\epsilon\, D)$ For Example:

   $$Pr(\mathbf{x} \,|\, y = c_1) \sim \mathbf{N}(\mu_1, \Sigma_1)$$
   $$Pr(\mathbf{x} \,|\, y = c_2) \sim \mathbf{N}(\mu_2, \Sigma_2)$$

   We can find $Pr(y \,|\, \mathbf{x})$ as

   $$Pr(y \,|\, \mathbf{x}) = \frac{Pr(\mathbf{x} \,|\, y) Pr(y)}{\Sigma_y Pr(\mathbf{x} \,|\, y) Pr(y)}$$

   Here, $Pr(y = c_1), Pr(y = c_2)$ are called priors or mixture components.

   Examples: Naive Bayes, Bayes Nets, Hidden Markov Models

15 Lecture 15: Linear Models for Classification

The goal in classification is to take an input vector $\mathbf{x}$ and assign it to one of K discrete classes $\mathcal{C}_k$ where $k = 1, ..., K$. In most cases, the classes are taken to be disjoint, so that each input is assigned to one and only one class. The input space is thus divided into *decision regions* whose boundaries are called *decision boundaries* or *decision surfaces*. We consider only linear models for classification in this lecture, which means that the decision surfaces are linear functions of the input vector $\mathbf{x}$ and hence are defined by $(D-1)$-dimensional hyperplanes within the D-dimensional input space.

The simplest method of classification (for 2 classes) is to design a function f such that

$$f(\mathbf{x}_i) = \begin{cases} v_{c+} & \text{if } \mathbf{x}_i \in \mathcal{C}_+ \\ v_{c-} & \text{if } \mathbf{x}_i \in \mathcal{C}_- \end{cases}$$

15.1 Generalized linear models

In these models we adopt linear regression to model the classification problem. This is done by modeling a function f as follows:

$$f(\mathbf{x}) = g(\mathbf{w}^T\phi(\mathbf{x}))$$

where g is known as *activation function* and ϕ the vector of basis functions. Classification is achieved by:

$$g(\theta) = \begin{cases} v_{c+} & \text{if } \theta > 0 \\ v_{c-} & \text{if } \theta < 0 \end{cases}$$

The decision surface in this case is given by $\mathbf{w}^T\phi(\mathbf{x}) = 0$

15.2 Three broad types of classifiers

1. The first method involves explicit construction of $\mathbf{w}$ for $\mathbf{w}^T\phi(\mathbf{x}) = 0$ as the decision surface.

2. The second method is to model $P(\mathbf{x}|\mathcal{C}_+)$ and $P(\mathbf{x}|\mathcal{C}_-)$ together with the prior probabilities $P(\mathcal{C}_k)$ for the classes, from which we can compute the posterior probabilities using Bayes' theorem

   $$P(\mathcal{C}_k|\mathbf{x}) = \frac{P(\mathbf{x}|\mathcal{C}_k)P(\mathcal{C}_k)}{P(\mathbf{x})}$$

   These types of models are called *generative models.*

3. The third method is to model $P(\mathcal{C}_+|\mathbf{x})$ and $P(\mathcal{C}_-|\mathbf{x})$ directly. These types of models are called *discriminative models*. In this case $P(\mathcal{C}_+|\mathbf{x}) = P(\mathcal{C}_-|\mathbf{x})$ gives the required decision boundary.

15.2.1 Examples

An example of generative model is as follows:

$$P(\mathbf{x}|\mathcal{C}_+) = \mathcal{N}(\mu_+, \Sigma)$$
$$P(\mathbf{x}|\mathcal{C}_-) = \mathcal{N}(\mu_-, \Sigma)$$

With prior probabilities $P(\mathcal{C}_+)$ and $P(\mathcal{C}_-)$ known, we can derive $P(\mathcal{C}_+|\mathbf{x})$ and $P(\mathcal{C}_+|\mathbf{x})$. In this case it can be shown that the decision boundary $P(\mathcal{C}_+|\mathbf{x}) = P(\mathcal{C}_-|\mathbf{x})$ is a hyperplane.

An example of discriminative model is

$$P(\mathcal{C}_+|\mathbf{x}) = \frac{e^{\mathbf{w}^T\phi(\mathbf{x})}}{1 + e^{\mathbf{w}^T\phi(\mathbf{x})}}$$
$$P(\mathcal{C}_-|\mathbf{x}) = \frac{1}{1 + e^{\mathbf{w}^T\phi(\mathbf{x})}}$$

Examples of first model (which directly construct the classifier) include

- Linear Regression
- Perceptron
- Fisher's Discriminant

15.3 Handling Multiclasses

We now consider the extension of linear discriminants to $K > 2$ classes. One solution is to buid a K-class discriminant by combining a number of two-class discriminant functions.

- *one-versus-the-rest*: In this approach, $K - 1$ classifiers are constructed, each of which separtes the points in a particular class $\mathcal{C}_k$ from points not in that classes
- *one-versus-one*: In this method, ${}^KC_2$ binary discriminant functions are introduced, one for every possible pair of classes.

Attempting to construct a K class discriminant from a set of two class discriminants leads to ambiguous regions. The problems with the first two approaches are illustrated in the figures 1 and 2, where there are ambiguous regions marked with '?'.

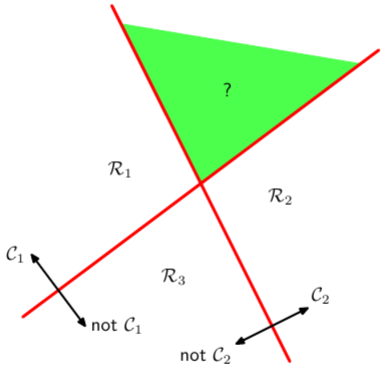


Figure 17: Illustrates the ambiguity in *one-versus-rest* case

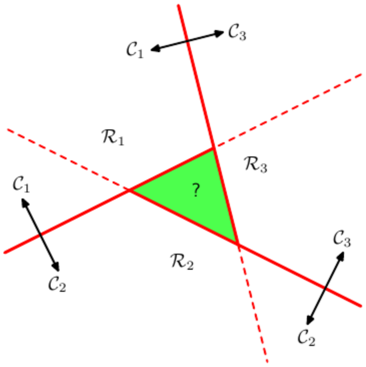


Figure 18: Illustrates the ambiguity in *one-versus-one* case

15.3.1 Avoiding ambiguities

We can avoid above mentioned difficulties by considering a single K-class discriminant comprising K functions $g_{\mathcal{C}_k}(\mathbf{x})$. Then $\mathbf{x}$ is assigned to a class $\mathcal{C}_k$ that has the maximum value for $g_{\mathcal{C}_k}(\mathbf{x})$

If $g_{\mathcal{C}_k}(\mathbf{x}) = \mathbf{w}_{\mathcal{C}_k}^T \phi(\mathbf{x})$ the decision boundary between class $\mathcal{C}_j$ and class $\mathcal{C}_k$ is given by $g_{\mathcal{C}_k}(\mathbf{x}) = g_{\mathcal{C}_j}(\mathbf{x})$ and hence corresponds to

$$(\mathbf{w}_{\mathcal{C}_k}^T - \mathbf{w}_{\mathcal{C}_j}^T)\phi(\mathbf{x}) = 0$$

15.4 Least Squares approach for classification

We now apply the Least squares method to the classification problem. Consider a classification problem with K classes. Then the target values are represented by a K component target vector $\mathbf{t}$. Each class is described by its own model

$$y_k(\mathbf{x}) = \mathbf{w}_k^T \phi(\mathbf{x})$$

where $k \in \{1, ...K\}$. We can conveniently group these together using vector notation so that

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

where $\mathbf{W}$ is a matrix whose k^{th} column comprises the unknown parameters $\mathbf{w}_k$ and $\phi(\mathbf{x})$ is the vector of basis function values evaluated at the input vector $\mathbf{x}$. The procedure for classification is then to assign a new input vector $\mathbf{x}$ to the class for which the output $y_k = \mathbf{w}_k^T \phi(\mathbf{x})$ is largest.

We now determine the parameter matrix $\mathbf{W}$ by minimizing a sum-of-squares error function. Consider a training data set $\{\mathbf{x}_n, \mathbf{t}_n\}$ where $n \in \{1, .., N\}$, where $\mathbf{x}_n$ is input and $\mathbf{t}_n$ is corresponding target vector. We now define a matrix $\mathbf{\Phi}$ whose n^{th} row is given by $\phi(\mathbf{x}_n)$.

$$\mathbf{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \dots & \phi_{K-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \dots & \phi_{K-1}(\mathbf{x}_2) \\ \dots & \dots & \dots & \dots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \dots & \phi_{K-1}(\mathbf{x}_N) \end{pmatrix}$$

We further define a matrix $\mathbf{T}$ whose n^{th} row is given by the vector $\mathbf{t}_n^T$. Now, the sum-of-squares error function can then be written as

$$err(\mathbf{W}) = \frac{1}{2} Tr\{(\mathbf{\Phi W} - \mathbf{T})^T (\mathbf{\Phi W} - \mathbf{T})\}$$

We can now minimize the error by setting the derivative with respect to $\mathbf{W}$ to zero. The solution we obtain for $\mathbf{W}$ is then of the form

$$\mathbf{W} = (\mathbf{\Phi}^T \mathbf{\Phi})^{-1} \mathbf{\Phi}^T \mathbf{T}$$

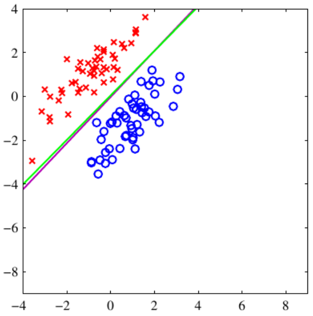


Figure 19: Data from two classes classified by least squares (magenta) and logistic (green)

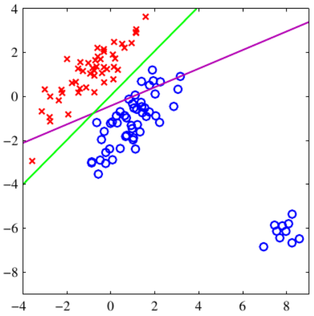


Figure 20: Response of the classifiers to addition of outlier points

15.4.1 Limitations of Least Squares

Even as the least-squares approach gives a closed form solution for the discriminant function parameters, it suffers from problems such as lack of robustness to outliers. This is illustrated in figures 3 and 4 where we see that introduction of additional data points in the figure 4 produce a significant change in the location of the decision boundary, even though these points would be correctly classified by the original boundary in figure 3. For comparison, least squares approach is contrasted with logisitc regression, which remains unaffected due to the additional points.

16 Lecture 16

16.1 Introduction

We will discuss the problems of the Linear regression model for classifications. We will also look at some of the possible solutions of these problems. Our main focus is on two class classification problem.

16.2 Problems of linear regression

The following are the problems with linear regression model for classification:

1. Sensitivity to outliers
2. Masking

16.2.1 Sensitivity to outliers

Outliers : They are points which have noise and adversely affect the classification.

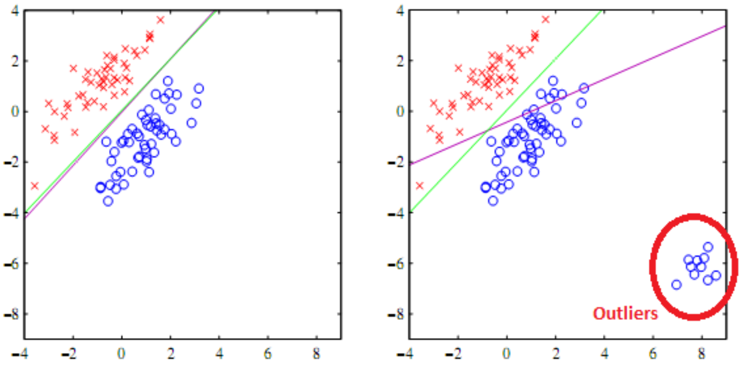


Figure 21: Outliers

In the right hand figure , the separating hyperplane has changed because of the outliers.

16.2.2 Masking

It is seen empirically that linear regression classifier may mask a given class. This is shown in the left hand figure. We had 3 classes one in between the other two. The between class points are not classified.

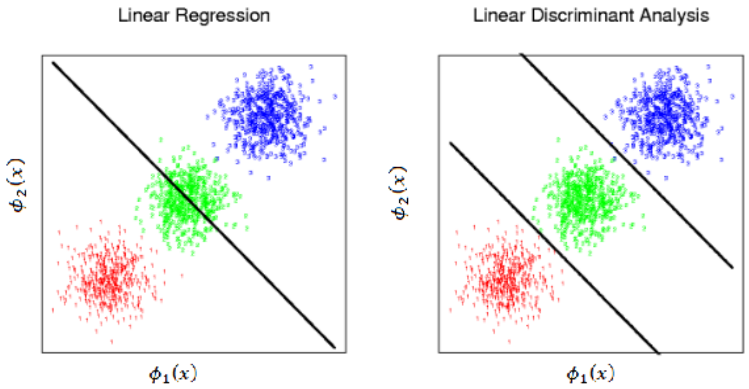


Figure 22: Masking

The right hand figure is the desirable classification.

The equation of the classifier between class C1(red dots) and class C2(green dots) is
$(\omega_1 - \omega_2)^T \phi(x) = 0$
and the equation of the classifier between the classes C2(green dots) and C3(blue dots) is
$(\omega_2 - \omega_3)^T \phi(x) = 0$

16.3 Possible solutions

1. **Mapping to new space**

   We will transform the original dimensions to new dimensions. New dimensions are function of original dimensions. This is a work around solution.

   $\phi_1^{'}(x) = \sigma_1(\phi_1, \phi_2)$
   $\phi_2^{'}(x) = \sigma_2(\phi_1, \phi_2)$

   Here we try to determine the transformations $\phi_1^{'}$ and $\phi_2^{'}$ such that we can get a linear classifier in this new space. When we map back to the original dimensions , the separators may not remain linear.

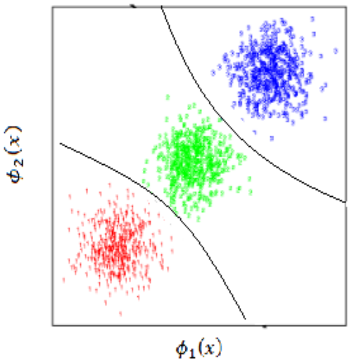


Figure 23: Mapping back to original dimension class separator not linear

Problem : Exponential blowup of number of parameters $(w's)$ in order $O(n^{k-1})$.

2. **Decision surface perpendicular bisector to the mean connector.**

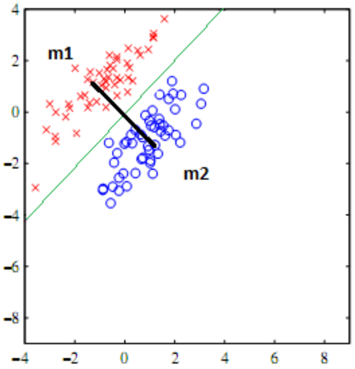


Figure 24: Class separator perpendicular to the line joining mean

Decision surface is the perpendicular bisector of the line joining mean of class $C_1(m_1)$ and mean of class $C_2(m_2)$.

$m_1 = (1/N_1)\sum_{n\in C_1} x_n$ where m_1 is the mean of class C_1 and N_1 is the number of points in class C_1.

$m_2 = (1/N_2)\sum_{n\in C_2} x_n$ where m_2 is the mean of class C_2 and N_2 is the number of points in class C_2.

$||\phi(x) - m_1|| < ||\phi(x) - m_2|| => x \in C_1$
$||\phi(x) - m_2|| < ||\phi(x) - m_1|| => x \in C_2$

Comment : This is solving the masking problem but not the sensitivity problem as this does not capture the orientation(eg: spread of the data points) of the classes.

3. **Fisher Discrimant Analysis.**

Here we consider the mean of the classes , within class covariance and global covariance.
Aim : To increase the separation between the class means and to minimize within class variance. Considering two classes.
S_B is Inter class covariance and S_W is Intra class covariance.

$m_1 = (1/N_1) \sum_{n \in C_1} x_n$ where m_1 is the mean of class C_1 and N_1 is the number of points in class C_1.

$m_2 = (1/N_2) \sum_{n \in C_2} x_n$ where m_2 is the mean of class C_2 and N_2 is the number of points in class C_2.

$N_1 + N_2 = N$ where N is the total number of training points.

$S_B = (m_2 - m_1)(m_2 - m_1)^T$

$S_W = \sum_{n \in C_1}(x_n - m_1)(x_n - m_1)^T + \sum_{n \in C_2}(x_n - m_2)(x_n - m_2)^T$

$J(w) = (w^T S_B w)/(w^T S_w w)$

By maximizing $J(w)$ we get the following:

$w \alpha S_w^{-1}(m_2 - m_1)$

16.4 Summary

	Sensitivity to outliers	**Masking**
Perpendicular Bisector of means connector	Does not solve	Solves
Fischer Discriminant	Does not solve	Solves

We have seen that the fisher discriminant analysis is better compared to the other two possible solutions. Fisher Discriminant analysis for k classes is discussed in the next lecture.

17 Lecture 17

Not submitted

18 Lecture 18:Perceptron

- Was Fisher's discriminant robust to noise?
- Perceptron training

18.1 Fisher's discriminant

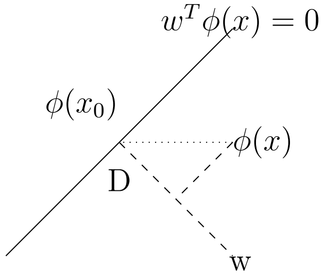
From the figure, it can be seen that Fischer Discriminant method is not robust to noise. The difference in inclination of blue (without noise) and magenta (with noise) lines shows that the outlying points affect the classifier more than desired. This is because Fischer discriminant does not take into account the distance of the data points from the hyperplane. Perceptron, unlike Fischer, considers the distance and is thus robust to noise.

18.2 Perceptron training

- Explicitly account for signed distribution of points(misclassified points) from hyperplane

  $$w^T\phi(x) = 0$$

  Distance from hyperplane can be calculated as follows

  D = $w^T(\phi(x) - \phi(x_0))$
  Since $w^T(\phi(x_0)) = 0$ we get distance = $w^T(\phi(x))$

- Perceptron works for two classes only. We label them as y=1 and y=-1. A point is misclassified if $w^T(\phi(x)) < 0$

Perceptron Algorithm:

- INITIALIZE: w=ones()
- REPEAT:

- If given $< x, y >, w^T\Phi(x).y \leq 0$
- then, $w = w + \Phi(x).y$
- endif

18.2.1 Intuition

$$\begin{aligned} yw_{k+1}^T\phi(x) &= y(w_k + y\phi(x)^T\phi(x) \\ &= yw_k^T\phi(x) + y^2\|\phi(w)\|^2 \\ &> yw_k^T\phi(x) \end{aligned}$$

Note: We applied the update for this point,

$$Since\ yw_k^T\phi(x) \leq 0$$

We have $yw_k^T\phi(x) \ > \ yw_k^T\phi(x)$ So we have more hope that this point is classified correctly now.

More formally, perceptron tries to minimize the error function

$$E = -\sum_{x \in M} y\phi^T(x)\omega$$

where M is the set of misclassified examples.

Perceptron algorithm is **similar** (Its not exactly equivalent) to a gradient descent algorithm, which can be shown as follows:

Gradient Descent (Batch Perceptron) Algorithm

Since ∇E is given by,
$\nabla E = -\sum_{x \in M} y\phi(x)$
So,

$$\begin{aligned} w_{k+1} &= w_k - \eta\nabla E \\ &= w_k + \eta\sum_{x \in M} y\phi(x) \qquad \text{(This takes all misclassified points at a time)} \end{aligned}$$

But what we are doing in standard Perceptron Algorithm, is basically *Stochastic Gradient Descent*:

$\nabla E = -\sum_{x \in M} y\phi(x) = -\sum_{x \in M} \nabla E(x)$, where $E(x) = y\phi(x)$

$$\begin{aligned} w_{k+1} &= w_k - \eta\nabla E(x) \\ &= w_k + \eta y\phi(x) \qquad \text{(for any } x \in M) \end{aligned}$$

Earlier it was intuition, now, **Formally,**:-
If $\exists$ an optimal separating hyperplane with parameters w^* such that,

$$\phi^T(x)w^* = 0$$

then perceptron algorithm converges.

Proof:-

$$\lim_{k\to\infty} \|w_{k+1} - \rho w^*\|^2 = 0 \tag{68}$$

(If this happens for some constant ρ, we are fine.)

$$\|w_{k+1} - \rho w^*\|^2 = \|w_k - \rho w^*\|^2 + \|y\phi(x)\|^2 + 2y(w_k - \rho w^*)^T\phi(x) \tag{69}$$

Now, we want L.H.S. to be less than R.H.S. at every step, although by some small value, so that perceptron will converge overtime.
So, if we can obtain an expression of the form:

$$\|w_{k+1} - \rho w^*\|^2 < \|w_k - \rho w^*\|^2 - \theta^2 \tag{70}$$

Then, $\|w_{k+1} - \rho w^*\|^2$ is reducing by atleast θ^2 at every iteration.
So, from the above expressions (2) and (3), we need to find θ such that,

$$\|\phi(x)\|^2 + 2y(w_k - \rho w^*)^T\phi(x) < -\theta^2$$

(Here, $\|y\phi(x)\|^2 = \|\phi(x)\|^2$ because $\|y\| = 1, y$ is either $+1$ or -1)

So, the no. of iterations would be: $O\left(\frac{\|w_0 - \rho w^*\|^2}{\theta^2}\right)$

Observations:-

1. $yw_k^T\phi(x) \leq 0$ ($\because x$ was misclassified)
2. $\Gamma^2 = \max_{x\in\mathcal{D}} \|\phi(x)\|^2$
3. $\delta = \max_{x\in\mathcal{D}} -2y{w^*}^T\phi(x)$

Here, margin $= {w^*}^T\phi(x) =$ dist. of closest point from hyperplane
and, $\mathcal{D}$ is the set of all points, **not** just misclassifed ones.

$$\begin{aligned}\delta &= \max_{x\in\mathcal{D}} -2y{w^*}^T\phi(x)\\ &= \min_{x\in\mathcal{D}} y{w^*}^T\phi(x)\end{aligned}$$

Since, ${w^*}^T\phi(x) \geq 0$, so, $\delta \leq 0$.
So, what we are interested in, is the 'least negative' value of δ

From the observations, and eq.(2), we have:

$$0 \leq \|w_{k+1} - \rho w^*\|^2 \leq \|w_k - \rho w^*\|^2 + \Gamma^2 + \rho\delta$$

Taking, $\rho = \dfrac{2\Gamma^2}{-\delta}$, then,

$$0 \leq \|w_{k+1} - \rho w^*\|^2 \leq \|w_k - \rho w^*\|^2 - \Gamma^2$$

Hence, we got, $\Gamma^2 = \theta^2$, that we were looking for in eq.(3).
$\therefore \|w_{k+1} - \rho w^*\|^2$ decreases by atleast Γ^2 at every iteration.

Here is the notion of convergence:-
w_k converges to ρw^* by making atleast some decrement at each step.

Thus, for $k \to \infty, \|w_k - \rho w^*\| \to 0,$
Hence, eq.(1) is proved.

19 Lecture 19

19.1 Introduction

In this lecture,we extend the margin-concept towards our goal of classification and introduce ourselves to Support Vector Machines(SVM)

19.2 Margin

Given $w^{\star}$,the unsigned minimum distance of x from the hyperplane:

$$ {w^{\star}}^{T} \phi(x) = 0 \tag{71} $$

is given by:

$$ \min_{\mathbf{x}\epsilon\mathbf{D}} y{w^{\star}}^{T} \phi(x) \tag{72} $$

where $y = \pm 1$.Here,y is the corresponding target classifier value for the case of 2-class classifiers.Note that multiplication with y makes the distance unsigned. This classification is greedy.

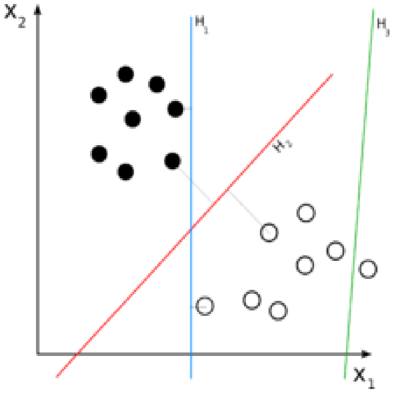


Figure 25: H3(green) doesn't separate the 2 classes. H1(blue) does, with a small margin and H2(red) with the maximum margin. [5]

Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training data points of any class, since in general the larger the margin the lower the generalization error of the classifier. [5]

19.3 Support Vector Machines

The idea in a **Support Vector Machine(SVM)** is to **Maximize** the **Minimum Unsigned Distance** $(yw^T\phi(x))$ of a point x from the hyperplane

$$w^T\phi(x) = 0 \tag{73}$$

The factor y which is also the target classifier ensures that the unsigned distance is positive semi-definite. Posing this as an optimization problem where:

$$\phi = [1, \phi_0, \phi_1, ...\phi_n] \tag{74}$$

$$w = [w_0, w_1, ...w_n] \tag{75}$$

where 1 and w_0 together denote the bias parameters and $[w_1....w_n]$ denote the slope.

Optimization Goal: To adjust the Slope and Bias to Maximize the Minimum unsigned distance from the hyperplane.

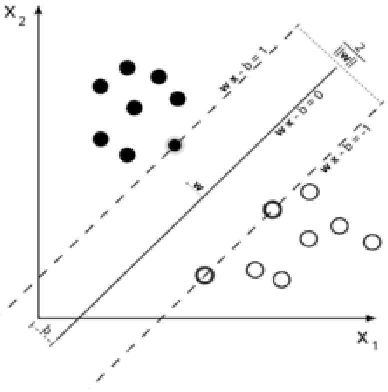


Figure 26: Maximum Margin Hyperplanes and Margins for SVM trained with 2 classes.Samples on the margin are called the support vectors. [5]

19.4 Support Vectors

The seperating hyperplane is dependent only on the support vectors. Those points which are not support vectors are irrelevant.

19.5 Objective Design in SVM

Keeping the Bias(w_0) and Slope(w) separate,

$$y(x) = w^T\phi(x) + w_0 \tag{76}$$

$y(x)$ represents the Separating Hyperplane.

19.5.1 Step 1:Perfect Separability

The condition for existence of a Separating Hyperplane(Perfect Separability) is:

$$\exists w, w_0 \forall y_i, x_i \epsilon D \quad \mathbf{w^T}\phi(\mathbf{x_i}) + \mathbf{w_0} > \mathbf{0} \quad if\ y_i = +1 \tag{77}$$

$$\mathbf{w^T}\phi(\mathbf{x_i}) + \mathbf{w_0} < \mathbf{0} \quad if\ y_i = -1 \tag{78}$$

19.5.2 Step 2:Optimal Separating Hyperplane For Perfectly Separable Data

$$\max_{\mathbf{w},\mathbf{w_0}} \delta \tag{79}$$

$$\mathbf{w^T}\phi(\mathbf{x_i}) + \mathbf{w_0} > \delta \quad if\ y_i = +1 \tag{80}$$

$$\mathbf{w^T}\phi(\mathbf{x_i}) + \mathbf{w_0} < -\delta \quad if\ y_i = -1 \tag{81}$$

$$\delta \geq 0 \tag{82}$$

But for two distinct w_1, w_2 defining the same Hyperplane,signed distance can be different for the same point depending on the value of $||w||$.

Thus fixing $||w||$,the Optimization Objective is:

$$\max_{\mathbf{w},\mathbf{w_0}} \delta \tag{83}$$

$$y_i(\mathbf{w^T}\phi(\mathbf{x_i}) + \mathbf{w_0}) > \delta \tag{84}$$

$$\delta \geq 0 \tag{85}$$

$$||\mathbf{w}|| = \theta \tag{86}$$

$$\tag{87}$$

Without restriction on $||\mathbf{w}||$,δ goes unbounded.

We can get rid of the assumption that $||\mathbf{w}|| = \theta$ by replacing the conditions with the following while redefining $w_0' = \frac{\theta w_0}{||\mathbf{w}||}$ to get:

$$\max_{\mathbf{w},\mathbf{w_0'}} \delta \tag{88}$$

$$\frac{\theta}{||\mathbf{w}||} y_i(\mathbf{w^T}\phi(\mathbf{x_i}) + \mathbf{w_0'}) > \delta \tag{89}$$

$$\delta \geq 0 \tag{90}$$

or Equivalenty,

$$\max_{\mathbf{w},\mathbf{w_0'}} \quad \delta \tag{91}$$

$$y_i(\mathbf{w^T}\phi(\mathbf{x_i}) + \mathbf{w_0'}) > \delta \frac{||\mathbf{w}||}{\theta} \tag{92}$$

$$\delta \geq 0 \tag{93}$$

Since for any $\mathbf{w}$ and w_0' satisfying these inequalities, any positive multiple satisfies them too, we can arbitrarily set $||\mathbf{w}|| = \frac{\theta}{\delta}$. Thus, the equivalent problem is:

$$\max_{\mathbf{w},\mathbf{w_0'}} \frac{1}{||\mathbf{w}||} \tag{94}$$

$$y_i(\mathbf{w^T}\phi(\mathbf{x_i}) + \mathbf{w_0'}) > \mathbf{1} \tag{95}$$

Or Equivalently:

$$\min_{\mathbf{w},\mathbf{w_0'}} ||\mathbf{w}||^{\mathbf{2}} \tag{96}$$

$$y_i(\mathbf{w^T}\phi(\mathbf{x_i}) + \mathbf{w_0'}) > \mathbf{1} \tag{97}$$

The above transformation is fruitful as the determination of model parameters reduces to a Convex Optimization problem and hence,any locally optimum solution is globally optimum.
[6]

19.5.3 Step 2:Separating Hyperplane For Overlapping Data

Unlike the previous case wherein Data was either "Black" or "White",herein we have a Region of "Gray".

Earlier,we implicitly used an error function that gave infinite error if a data point was misclassified and zero error if it was classified correctly. We now modify this approach so that data points are allowed to be on the "wrong side" of the margin boundary, but with a penalty that increases with the distance from that boundary.

Thus, the objective to account for the Noise.Hence, we shall introduce a slack variable ζ_i

Now the Optimization Objective is:

$$\min_{\mathbf{w},\mathbf{w_0'}} \frac{||\mathbf{w}||^{\mathbf{2}}}{2} \tag{98}$$
$$y_i(\mathbf{w^T}\phi(\mathbf{x_i}) + \mathbf{w_0'}) > \mathbf{1} - \zeta_{\mathbf{i}} \tag{99}$$
$$\zeta_i \geq 0 \tag{100}$$

or Equivalently:

$$\min_{\mathbf{w},\mathbf{w_0'}} \frac{||\mathbf{w}||^{\mathbf{2}}}{2} + c\sum_{i=1}^{N} \zeta_i^2 \tag{101}$$
$$y_i(\mathbf{w^T}\phi(\mathbf{x_i}) + \mathbf{w_0'}) > \mathbf{1} - \zeta_{\mathbf{i}} \tag{102}$$
$$\zeta_i \geq 0 \tag{103}$$

Thus,the objective is analogous to the Minimization of Error subject to Regulariser.

Here,the Error is:

$$E = c\sum_{i=1}^{N} \zeta_i^2 \tag{104}$$

And the Regulariser is:

$$R = \frac{||\mathbf{w}||^{\mathbf{2}}}{2} \tag{105}$$

Both $EandR$ can be proved to be convex.

Partially differentiating E w.r.t. ζ_j:

$$\frac{\delta E}{\delta \zeta_j} = 2\zeta_j \tag{106}$$
$$\frac{\delta^2 E}{\delta \zeta_j^2} = 2 > 0 \tag{107}$$

Thus E is convex w.r.t $\zeta_j \forall 1 \leq j \leq n$

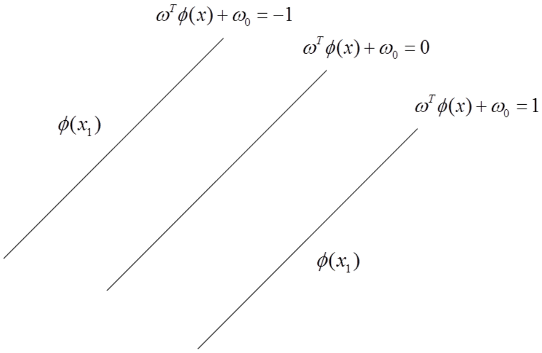
Consider,R:

$$\bigtriangledown(R) = 2w \tag{108}$$
$$\bigtriangledown^2(R) = 2I \tag{109}$$

$2I$ is clearly positive definite with the eigenvalues 2(twice).

Thus R is convex.

Therefore,$E + R$ is convex and the Optimization again reduces to a Convex Optimization Problem.


$\omega^T\phi(x)+\omega_0=-1$
$\omega^T\phi(x)+\omega_0=0$
$\omega^T\phi(x)+\omega_0=1$
$\phi(x_1)$
$\phi(x_1)$

Proof.

Figure 27: Margins in SVM

20 Lecture 20: Support Vector Machines (SVM)

This lecture formulates the primal expression of SVM. Then it applies KKT conditions on top of that. Then we proceed to form the dual problem.

20.1 Recap

The expression from previous day:

$$y_i(\phi^T(x_i)w + w_0') \geq \frac{||w||}{\theta} \tag{110}$$

So, any multiple of w and w_0' would not change the inequality.

20.2 Distance between the points

Important Result 1. *The distance between the points x_1 and x_2 in 20.2 is $\frac{2}{||w||}$*

The distance between the points x_1 and x_2 in 20.2 turns out to be:

$$||\phi(x_1) - \phi(x_2)|| = ||rw|| \tag{111}$$

We have,

$$w^T\phi(x_1) + w_0 = -1 \tag{112}$$

and

$$w^T\phi(x_2) + w_0 = 1 \tag{113}$$

Subtracting Equation 113 from Equation 112 we get,

$$w^T(\phi(x_1) - \phi(x_2)) = -2 \tag{114}$$

$$\implies w^T r w = -2 \tag{115}$$

$$\implies r = -\frac{2}{||w||^2} \tag{116}$$

$$\implies ||rw|| = -\frac{2}{||w||} \tag{117}$$

Hence proved. □

20.3 Formulation of the optimization problem

$$\max \frac{1}{||w||} \tag{118}$$

$$s.t. \forall i \tag{119}$$

$$y_i(\phi^T(x_i)w + w_0') \geq 1 \tag{120}$$

This means that if we maximize the separation between margin planes and at the same time ensure that the respective points are not crossing corresponding margin plane (the constraint), we are done.

It can be proved that,

$$\begin{aligned}
&if \max ||\text{distance of closest point from hyperplane}||_p \\
&then, \max \frac{1}{||w||_q} \\
&s.t. \forall i \\
&y_i(\phi^T(x_i)w + w_0') \geq 1 \\
&where \frac{1}{p} + \frac{1}{q} = 1
\end{aligned}$$

In our case $p = 2$ so $q = 2$. For $p = 1$, $q = \infty$ means maximum.

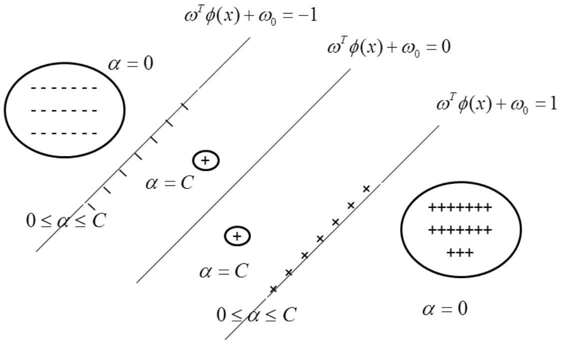


Figure 28: Different types of points

20.4 Soft Margin SVM

$$\min_{w,w_0} ||w||^2 + c\sum_i \xi_i \tag{121}$$

$$s.t.\forall i \tag{122}$$

$$y_i(\phi^T(x_i)w + w_0') \geq 1 - \xi_i \tag{123}$$

$$where, \tag{124}$$

$$\forall i \xi_i \geq 0 \tag{125}$$

In soft margin we account for the the errors. The above formulation is one of the many formulation of soft SVMs. In the above formulation, large value of c means overfitting.

20.4.1 Three types of g points

In subsubsection 20.4.1 we can see three types of points. They are:

1. Correctly classified but $\xi_i > 0$ or violates margin
2. Correctly classified but $\xi_i = 0$ or on the margin
3. Inorrectly classified but $\xi_i > 1$

20.5 Primal and Dual Formulation

20.5.1 Primal formulation

$$p^* = \min f(x) \tag{126}$$
$$x \in D \tag{127}$$
$$s.t. g(x) \leq 0 \tag{128}$$
$$i = 1, \ldots, m \tag{129}$$

20.5.2 Dual Formulation

$$d^* = \max_{\lambda \in \mathbb{R}} \min_{x \in D} \left(f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) \right) \tag{130}$$
$$s.t. \lambda_i \geq 0 \tag{131}$$

Equation 130 is and convex optimization problem. Also, $d^* \leq p^*$ and $(p^* - d^*)$ is called the duality gap.

If for some (x^*, λ^*) where x^* is primal feasible and λ^* is dual feasible and we see the KKT conditions are satisfied and f is and all g_i are convex then x^* is optimal solution to primal and λ^* to dual.

Also, the dual optimization problem becomes,

$$d^* = \max_{\lambda \in \mathbb{R}^m} \mathcal{L}(x^*, \lambda) \tag{132}$$
$$s.t. \tag{133}$$
$$\lambda_i \geq 0 \forall i \tag{134}$$
$$where, \tag{135}$$
$$\mathcal{L}(x, \lambda) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) \tag{136}$$
$$\mathcal{L}^*(\lambda) = \min_{x \in D} \mathcal{L}(x, \lambda) \tag{137}$$
$$= \max_{\lambda \in \mathbb{R}} \mathcal{L}(KKT(x), \lambda) \tag{138}$$
$$\lambda_i \geq 0 \forall i \tag{139}$$

It happens to be,

$$p^* = d^* \tag{140}$$

20.6 Duality theory applied to KKT

$$\mathcal{L}(\bar{w}, \bar{\xi}, w_0, \bar{\alpha}, \bar{\lambda}) = \frac{1}{2}\left|\left|w\right|\right|^2 + c\sum_{i=1}^{m}\xi_i + \sum_{i=1}^{m}\alpha_i\left[1 - \xi_i - y_i\left(\phi^T(x_i)w + w_0\right)\right] - \sum_{i=1}^{m}\lambda_i\xi_i \tag{141}$$

Now we check for KKT conditions at the point of optimality,

KKT 1.a

$$\nabla_w\mathcal{L} = 0 \tag{142}$$

$$\implies w - \sum_{j=1}^{m}\alpha_j y_j \phi^T(x_j) = 0 \tag{143}$$

KKT 1.b

$$\nabla_{xi_i}\mathcal{L} = 0 \tag{144}$$

$$\implies c - \alpha_i - \lambda_i = 0 \tag{145}$$

KKT 1.c

$$\nabla_{w_0}\mathcal{L} = 0 \tag{146}$$

$$\implies w - \sum_{i=1}^{m}\alpha_i y_i = 0 \tag{147}$$

KKT 2

$$\forall i \tag{148}$$

$$y_i\left(\phi^T(x_i)w + w_0\right) \geq 1 - \xi_i \tag{149}$$

$$\xi_i \geq 0 \tag{150}$$

KKT 3

$$\mathcal{L}_j \geq 0 and \lambda_k \geq 0 \tag{151}$$

$$\forall j, k = 1, \ldots, m \tag{152}$$

KKT 4

$$\mathcal{L}_j\left[y_i\left(\phi^T(x_j)w + w_0\right) - 1 + \xi_j\right] = 0 \tag{153}$$

$$\lambda_k\xi_k = 0 \tag{154}$$

(a)

$$w^* = \sum_{j=1}^{m}\alpha_j y_i \phi(x_j) \tag{155}$$

w^* is weighted linear combination of points $\phi(x)$s.

(b)

If $0 < \alpha_j < c$ then, by Equation 145
$0 < \lambda_j < c$ and by Equation 154, $\xi_j = 0$ and $y_i \left(\phi^T(x_j)w + w_0\right) = 1$

If however, $\alpha_j = c$ then $\lambda_j = 0$ and $y_i \left(\phi^T(x_j)w + w_0\right) \leq 1$.

If α_0 then $\lambda_j = c$ and $\xi_j = 0$, we get $y_i \left(\phi^T(x_j)w + w_0\right) \geq 1$. Then $\alpha_j = 0$

21 Lecture 21:The SVM dual

21.1 SVM dual

SVM can be formulated as the following optimization problem,

$$\min_{w}\{\frac{1}{2}\left\|w\right\|^2 + C\sum_{i=0}^{m}\xi_i\}$$

subject to constraint,

$$\forall i : y_i(\phi^T(x_i)w + w_0) \geq 1 - \xi_i$$

The dual of the SVM optimization problem can be stated as,

$$\max\{-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} y_i y_j \alpha_i \alpha_j \phi^T(x_i)\phi(x_j) + \sum_{j=1}^{m}\alpha_j\}$$

subject to constraints,

$$\forall i : \sum_{i}\alpha_i y_i = 0$$

$$\forall i : 0 \leq \alpha_i \leq c$$

The duality gap $= f(x^*) - L^*(\lambda^*) = 0$, as shown in last lecture. Thus, as is evident from the solution of the dual problem,

$$w^* = \sum_{i=1}^{m}\alpha_i^* y_i \phi(x_i)$$

To obtain w_o^*, we can use the fact (as shown in last lecture) that, if $\alpha_i \in (0, C)$, $y_i(\phi^T(x_i)w + w_0) = 1$. Thus, for any point x_i such that, $\alpha_i \in (0, C)$, that is, α_i is a point on the margin,

$$\begin{aligned} w_o^* &= \frac{1 - y_i(\phi^T(x_i)w^*)}{y_i} \\ &= y_i - \phi^T(x_i)w^* \end{aligned}$$

The decision function,

$$\begin{aligned} g(x) &= \phi^T(x)w^* + w_0^* \\ &= \sum_{i=0}^{m}\alpha_i y_i \phi^T(x)\phi(x_i) + w_0^* \end{aligned}$$

21.2 Kernel Matrix

A kernel matrix

$$K = \begin{bmatrix} \phi^T(x_1)\phi(x_1) & \phi^T(x_1)\phi(x_2) & \dots & \dots & \phi^T(x_1)\phi(x_n) \\ \phi^T(x_2)\phi(x_1) & \phi^T(x_2)\phi(x_2) & \dots & \dots & \phi^T(x_2)\phi(x_n) \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \phi^T(x_n)\phi(x_1) & \phi^T(x_n)\phi(x_2) & \dots & \dots & \phi^T(x_n)\phi(x_n) \end{bmatrix}$$

In other words, $K_{ij} = \phi^T(x_i)\phi(x_j)$. The SVM dual can now be re-written as,

$$\max\{-\frac{1}{2}\alpha^T K_y \alpha + \alpha^T ones(m,1)\}$$

subject to constraints,

$$\sum_i \alpha_i y_i = 0$$
$$0 \leq \alpha_i \leq c$$

Thus, for $\alpha_i \in (0, C)$

$$\begin{aligned} w_0^* &= y_i - \phi^T(x_i)w \\ &= y_i - \sum_{j=0}^{m} \alpha_j^* y_j \phi^T(x_i)\phi(x_j) \\ &= y_i - \sum_{j=0}^{m} \alpha_j^* y_j K_{ij} \end{aligned}$$

21.2.1 Generation of ϕ space

For a given $\mathbf{x} = [x_1, x_2, \dots, x_n] \rightarrow \phi(x) = [x_1^d, x_2^d, x_3^d, \dots, x_1^{d-1}x_2, \dots]$.
For $n = 2, d = 2$, $\phi(x) = [x_1^2, x_1x_2, x_2x_1, x_2^2]$, thus,

$$\begin{aligned} \phi^T(x).\phi(\bar{x}) &= \sum_{i=1}^{m}\sum_{j=1}^{m} x_i x_j.\bar{x}_i\bar{x}_j \\ &= (\sum_{i=1}^{m} x_i\bar{x}_i).(\sum_{j=1}^{m} x_j\bar{x}_j) \\ &= (\sum_{i=1}^{m} x_i\bar{x}_i)^2 \\ &= (x^T\bar{x})^2 \end{aligned}$$

In general, for $n \geq 1$ and $d \geq 1$, $\phi^T(x).\phi(\bar{x}) = (x^T\bar{x})^d$.

A polynomial kernel, in general, is defined as $K_{ij} = (x_i^T x_j)^d$.

21.3 Requirements of Kernel

1. Since

   $$\begin{aligned} K_{ij} &= \phi^T(x_i)\phi(x_j) \\ &= \phi^T(x_j)\phi(x_i) \end{aligned}$$

   Hence K should be a Symmetric Matrix.

2. The Cauchy Schwarz Inequality

   $$(\phi^T(x)\phi(\bar{x}))^2 \leq \|\phi^T(x)\|^2 \|\phi(\bar{x})\|^2$$

   $$\Rightarrow {K_{ij}}^2 \leq K_{ii} K_{jj}$$

3. Positivity of Diagonal

   $$K = V \Lambda V^T$$

   Where V is the eigen vector matrix (an orthogonal matrix), and Λ is the Diagonal matrix of eigen values.

Goal is to construct a ϕ. Which can be constructed as

$$\phi(x_i) = \sqrt{\lambda_i} V_i \quad (\lambda_i \geq 0)$$
$$K_{ii} = \lambda_i \|V_i\|^2$$

Hence K must be

1. Symmetric.
2. Positive Semi Definite.
3. Having non-negative Diagonal Entries.

21.3.1 Examples of Kernels

1. $K_{ij} = ({x_i}^T x_j)^d$
2. $K_{ij} = ({x_i}^T x_j + 1)^d$

3. Gaussian or Radial basis Function (RBF)
   $K_{ij} = e^{-\frac{\|x_i - x_j\|}{2\sigma^2}}$ $(\sigma \in R,\, \sigma \neq 0)$
4. The Hyperbolic Tangent function
   $K_{ij} = \tanh(\sigma x_i^T x_j + c)$

21.4 Properties of Kernel Functions

If K' and K'' are Kernels then K is also a Kernel if either of the following holds

1. $K_{ij} = K'_{ij} + K''_{ij}$
2. $K_{ij} = \alpha K'_{ij}$ $(\alpha \geq 0)$
3. $K_{ij} = K'_{ij} K''_{ij}$

Proof : (1) and (2) are left as an exercise.
(3)

$$\begin{aligned} K_{ij} &= K'_{ij} K''_{ij} \\ &= \phi'^T(x'_i)\phi'(x'_j) * \phi''^T(x''_i)\phi''(x''_j) \end{aligned}$$

Define $\phi(x_i) = \phi'^T(x'_i)\phi''^T(x''_i)$. Thus, $K_{ij} = \phi(x_i)\phi(x_j)$.
Hence, K is a valid kernel.

22 Lecture 22: SVR and Optimization Techniques

Topics covered in this lecture

- Variants of SVM (other occurance of kernel)
- Support Vector Regression
- L_1 SVM
- Projection Method (kernel Adatron)

22.1 Other occurance of kernel

- for regression, we have

$$W = (\phi^T \phi)\phi^T y$$
$$f(X) = W^T \phi(x) = \phi(\phi^T \phi)^T \phi(x) y$$

- For perceptron

$$W = \sum_{i=i}^{M} \alpha_i \phi(x_i) y_i$$

α_i = no of times update on w was made for $< x_i, y_i >$

$$g(x_j) = \phi^T(x_j).w = \sum_i \alpha_i \phi^T(x_j)\phi(x_i) y_i = \sum_i \alpha_i k_{ij} y_i$$

22.1.1 Some variants of SVM's

We have considered some variants of SVM,

$$min_{w,\xi} \quad \frac{1}{2}||w||^2 + C\sum_i \xi_i$$
$$st., \quad \forall_i \quad y_i(\phi^T(x_i)w + w_0) \geq 1 - \xi_i \quad and \quad \xi_i > 0$$

This is 1-Norm SVM, i.e norm of the slack (ξ). We can also formulate withou $\xi > 0$ as

$$min_{w,\xi} \quad \frac{1}{2}||w||^2 + C\sum_i \xi_i^2$$

if C = 0 , w=0 is a trivial solution

Dual for this problem:-

$$\begin{aligned}
max \quad & -\frac{1}{2}\sum_i\sum_j \alpha_i\alpha_j(k_{ij}+\frac{1}{C}\delta_{ij})y_iy_j+\sum_i\alpha_i \\
s.t., \quad & \sum_i \alpha_i y_i = 0 \\
& \delta_{ij}=1 \quad if \quad i=j \\
& and = 0 \quad otherwise \\
also \quad & 0 \le \alpha_i \le C
\end{aligned}$$

The constraint $\sum_i \alpha_i y_i = 0$ keep appearing again and again. This appear because of w_0 being explicit. It will dissapear if w_0 were absorbed in w.
In linear regression (error)

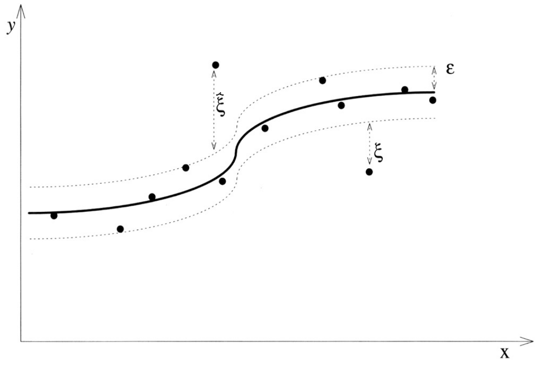
$$and \quad W = (\phi^T\phi)^{-1}\phi^T y$$

In ridge regression $(error + \lambda||w||^2)$

$$and \quad W = (\phi^T\phi - \lambda I)^{-1}\phi^T y$$

22.2 Support Vector Regression

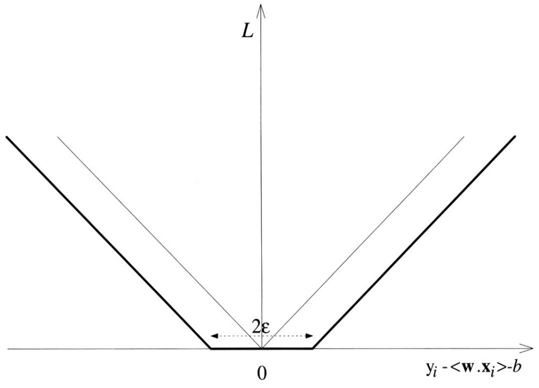
The Support Vector method can also be applied to the case of regression (apart from classification problem), maintaining all the main features that characterise the maximal margin algorithm, preserving the property of sparseness.
A non-linear function is learned by a linear learning machine in a kernel-induced feature space while the capacity of the system is controlled by a parameter that does not depend on the dimensionality of the space. The figure below shows a situation for a non-linear regression function.



The insensitive band (slackness) for a non-linear regression function.

As long as points lie inside the ϵ margin, they do not contribute to the error.
We can define the ϵ-insensitive loss function L^{ϵ} (x, y, f) as:-
For linear :

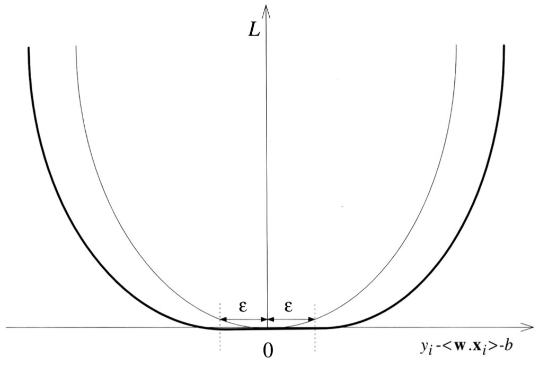
$$L^{\epsilon}(x, y, f) = |y - f(x)|_{\varepsilon} = max(0, |y - f(x)| - \varepsilon)$$



The linear ϵ-insensitive loss for zero and non-zero ϵ.

For Quadratic :

$$L_2^{\epsilon}(x, y, f) = |y - f(x)|_{\varepsilon}^2$$



The quadratic ϵ-insensitive loss for zero and non-zero ϵ.

In adapted ridge regression when slackness is introduced we have:-

$$min \frac{1}{2}||w||^2 + C \sum \xi^2$$
$$st.(\Phi^T(x_i)w + w_0) - y_i = \xi_i$$

we can optimise the generalisation of our regressor by minimising the sum of the quadratic ϵ-insensitive losses. For SVR-2 norm

$$min\frac{1}{2}||w||^2 + C\sum(\xi_i^2 + \xi_i^2)$$

$$st, \quad \forall_i(\Phi^T(x_i)w + w_0) - y_i \leq \varepsilon + \xi_i$$

$$and, \quad y_i - (\Phi^T(x_i)w + w_0) \geq \varepsilon + \xi_i^{`}$$

$$also \quad \xi_i\xi_i^{`} = 0$$

and for 1-norm we have

$$C\sum(\xi_i + \xi_i^{`}) \quad and$$

$$\xi_i \geq 0, \xi_i^{`} \geq 0$$

For 2-norm case, the dual problem can be derived using the standard method and taking into account that $\xi_i\xi_i^{`} = 0$ and therefore that the same relation $\alpha_i\alpha_i^{`}$ holds for the corresponding Lagrange multipliers:

$$maximise\sum_{i=i}^{M} y_i(\alpha_i^{`} - \alpha_i) - \varepsilon\sum_{i=i}^{M} y_i(\alpha_i^{`} + \alpha_i) - \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{M} y_i(\alpha_i^{`} - \alpha_i)(\alpha_j^{`} - \alpha_j)(K_{ij} + \frac{1}{C}\delta_{ij})$$

$$subjectto: \quad \sum_{i=i}^{M}(\alpha_i^{`} - \alpha_i) = 0$$

$$\alpha_i^{`} \geq 0, \quad \alpha_i \geq 0, \quad \alpha_i^{`}\alpha = 0$$

The corresponding Karush - Kuhn - Tucker complementarity conditions are [7]

$$\alpha_i(< w.\phi(x_i) > +b - y_i - \varepsilon - \xi_i) = 0$$

$$\alpha_i^{`}(y_i - < w - \phi(x_i) > -b - -\varepsilon - \xi_i^{`}) = 0$$

$$\xi_i\xi_i^{`} = 0 \quad \alpha_i\alpha_i^{`} = 0$$

By substituting $\beta = \alpha^{`} - \alpha$ and using the relation $\alpha_i\alpha_i^{`} = 0$, it is possible to rewrite the dual problem in a way that more closely resembles the classification case

$$maximise\sum_{i=1}^{M} y_i\alpha_i - \varepsilon\sum_{i=1}^{M}|\alpha_i| - \frac{1}{2}\sum_{i,j=1}^{M}\alpha_i\alpha_j(K(\phi(x_i).\phi(x_j)) + \frac{1}{C}\delta_{ij})$$

$$subjectto\sum_{i=1}^{M}\beta_i = 0$$

Notes:-

- Ridge Regression has 1 parameter - λ,
  SVM 2-norm has 2 parameters - $(C - 1/\lambda)$ and ε

- SVR with 2 norm and $\varepsilon = 0 \equiv$ Ridge Regression
- if $\varepsilon = 0$ and as C $\rightarrow \infty$,
  SVR 2-norm $\rightarrow$ linear regression
- SVR 2-norm and SVM 2-norm have $[\frac{1}{C}\delta_{ij}]$ added to kernel matrix in dual.

22.3 L_1 SVM

Let training datum be $x_i(i = 1, ..., M)$ and its label be $y_i = 1$ if x_i belongs to Class 1, and $y_i = -1$ if Class 2. In SVMs, to enhance linear separability, the input space is mapped intoa high dimensional feature space using the mapping function $g(x)$. To obtain the optimal separating hyperplane of the L1-SVM in the feature space, we consider the following optimization problem:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|W\|^2 + C\sum_{i=0}^{M}\xi_i \\ \text{subject to} \quad & y_i(W^t g(x_i) + b) \geqslant 1 - \xi_i \\ \text{for} \quad & i = 1, \ldots, M, \end{aligned}$$

where W is a weight vector, C is the margin parameter that determines the tradeoff between the maximization of the margin and the minimization of the classification error, ξ_i $(i = 1, ..., M)$ are the nonnegative slack variables and b is a bias term. Introducing the Lagrange multipliers α_i, we obtain the following dual problem:

maximize

$$Q(\alpha) = \sum_{i=0}^{M}\alpha_i - \frac{1}{2}\sum_{i=0}^{M}\alpha_i\alpha_j y_i y_j g(x_i)^t g(x_j),$$

$$\text{subject to} \quad \sum_{i=0}^{M} y_i\alpha_i = 0, \quad 0 \leq \alpha_i \leq C$$

We use the mapping function that satisfies

$$H(x, x') = g(x)^t g(x'),$$

where $H(x, x')$ is a kernel function. By this selection, we need not treat the variables in the feature space explicitly. Solving the above dual problem, we obtain the decision function:

$$D(x) = \sum_{i=0}^{M}\alpha_i^* y_i H(x_i, x) + b^*$$

L_1 SVM is like Lasso and it gives sparse solution.

22.4 Kernel Adatron

The "Perceptron with optimal stability" has been the object of extensive theoretical and exper- imental work, and a number of simple iterative procedures have been proposed, aimed at finding hyperplanes which have "optimal stability" or maximal margin. One of them, the Adatron, comes with theoretical guarantees of convergence to the optimal solution, and of a rate of convergence exponentially fast in the number of iterations, provided that a solution exists. Such models can be adapted, with the introduction of kernels, to operate in a high- dimensional feature space, and hence to learning non- linear decision boundaries. This provides a procedure which emulates SV machines but doesn't need to use the quadratic programming toolboxes.The Adatron is a an on-line algorihm for learning perceptrons which has an attractive xed point cor- responding to the maximal-margin consistent hyper- plane, when this exists.By writing the Adatron in the data-dependent repre- sentation, and by substituting the dot products with kernels, we obtain the following algorithm:

Kernel Adatron Algoritm

1. Initialise $\alpha_i = 1$.

2. Starting from pattern $i = 1$, for labeled points $(x_i, y_i$ calculate $z_i = y_i \sum_{j=0}^{p} \alpha_i y_i K(x, x')$.

3. For all patterns i calculate $\gamma_i = y_i z_i$ and execute steps 4 to 5 below. 4. Let $\delta\alpha^i = \eta(1 - \gamma^i)$ be the proposed change to the multipliers α^i.
5.1. If $(\alpha^i + \delta\alpha^i) \leq 0$ then the proposed change to the multipliers would result in a negative α^i . Consequently to avoid this problem we set $\alpha^i = 0$. 5.2 If $(\alpha^i + \delta\alpha^i) \geq 0$ then the multipliers are updated through the addition of the $\delta\alpha^i$ i.e. $\alpha^i \leftarrow \alpha^i + \delta\alpha^i$.
6. Calculate the bias b from $b = \frac{1}{2}(min(z_i^+) + max(z_i^-))$
where z_i^+ those patterns i with class label $+1$ and z_i^- are those with class label -1.
7. If a maximum number of presentations of the pat- tern set has been exceeded then stop, otherwise return to step 2.

Every stable point for adatron algorithm is a maximal margin point and vice versa. The algorithm converges in a finite number of steps to a stable point if a solution exists. The primal problem for adatron is given below.

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j K_{ij} y_i y_j - \sum \alpha_i, \\ \text{subject to} \quad & \sum \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C \end{aligned}$$

23 Lecture 23

Key terms : SMO, Probabilistic models, Parzen window

In previous classes we wrote the convex formulation for maximum margin classification, Lagrangian of the formulation etc. Then dual of the program was obtained by first minimizing the lagrangian with respect to weights w. The dual will be maximization of it with respect to α which is same as minimizing the negative of the objective function under the same constraints. The dual problem, given in equation (156) is an optimization with respect to α and its solution will correspond to optimal of original objective when the KKT conditions are satisfied. We are interested in solving dual of the objective because we have already seen that most of the dual variable will be zero in the solution and hence it will give a sparse solution (based on the KKT conidtion).

$$\begin{aligned} \text{Dual:} \quad \min_{\alpha} \quad & -\sum \alpha_i + \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j K_{ij} \\ \text{s. t.} \quad & \sum_i \alpha_i y_i = 0 \\ & \alpha_i \in [0, c] \end{aligned} \tag{156}$$

The above program is a quadratic program. Any quadratic solvers can be used for solving (156), but a generic solver will not take consider speciality of the solution and may not be efficient. One way to solve (156) is by using projection methods(also called Kernel adatron). You can solve the above one using two ways - chunking methods and decomposition methods.

The chunking method is as follows

1. Initialize α_is arbitrarily
2. Choose points(I mean the components α_i) that violate KKT condition
3. Consider only K working set and solve the dual for the variables in working set

   $\forall \alpha \in$ working set

$$\begin{aligned} \min_{\alpha} \quad & -\sum_{\alpha_i inWS} \alpha_i + \frac{1}{2}\sum_{i \in WS} \sum_{j \in WS} \alpha_i \alpha_j y_i y_j K_{ij} \\ \text{s. t.} \quad & \sum_{i \in WS} \alpha_i y_i = -\sum_{j \notin WS} \alpha_j y_j \\ & \alpha_i \in [0, c] \end{aligned} \tag{157}$$

4. set $\alpha^{new} = [\alpha^{new}_{WS}, \alpha^{old}_{nonWS}]$

Decompsition methods follow almost the same procedure except that in step 2 we always take a fixed number of points which violate the KKT conditions the most.

23.1 Sequential minimization algorithm - SMO

We can't take just one point at a time in working set and optimize with respect to it, because the second last constraint can't be satisified. So choose 2 points and optimize with respect to that. This is what is done in SMO, that is SMO is a decompostion method with 2 points taken at a time for opimization. The details are given below

Without loss of generality take the points α_1 and α_2 in the working set. Then the program (157) can be rewritten as

$$\begin{aligned}
\min_{\alpha_1,\alpha_2} \quad & -\alpha_1 - \alpha_2 - \sum_{i\neq 1,2} \alpha_i + \frac{1}{2}\alpha_1^2 K_{11} + \alpha_2^2 K_{22} + \alpha_1\alpha_2 K_{12} y_1 y_2 \\
& + \alpha_1 y_1 \sum_{i\neq 1,2} K_{1i}\alpha_i y_i + \alpha_2 y_2 \sum_{i\neq 1,2} K_{2i}\alpha_i y_i \qquad (158) \\
\text{s. t.} \quad & \alpha_1 y_1 + \alpha_2 y_2 = -\sum_{j\neq 1,2} \alpha_j y_j = \alpha_1^{old} + \alpha_2^{old} \\
& \alpha_1, \alpha_2 \in [0, c]
\end{aligned}$$

From the second last constraint, we can write α_1 in terms of α_2.

$$\alpha_1 = -\alpha_2 \frac{y_2}{y_1} + \alpha_1^{old} + \alpha_2^{old} \frac{y_2}{y_1}$$

Then the objective is just a function of α_2, let the objective is $-D(\alpha_2)$. Now the program reduces to

$$\begin{aligned}
\min_{\alpha_2} \quad & -D(\alpha_2) \\
\text{s. t.} \quad & \alpha_2 \in [0, c]
\end{aligned}$$

Find α_2^* such that $\frac{\partial D(\alpha_2)}{\partial \alpha_2} = 0$. We have to ensure that $\alpha_1 \in [0, c]$. So based on that we will have to clipp α_2 , ie, shift it to certain interval. The condition is as follows

$$0 <= -\alpha_2 \frac{y_2}{y_1} + \alpha_1^{old} + \alpha_2^{old} \frac{y_2}{y_1} <= c$$

- case 1: $y_1 = y_2$
$$\alpha_2 \in [max(0, -c + \alpha_1^{old} + \alpha_2^{old}), min(c, \alpha_1^{old} + \alpha_2^{old})]$$
- case 2: $y_1 = -y_2$
$$\alpha_2 \in [max(0, \alpha_2^{old} - \alpha_1^{old}), min(c, c - \alpha_1^{old} + \alpha_2^{old})]$$

If α_2 is already in the inerval then there is no problem. If it is more than the maximum limit then reset it to the maximum limit. This will ensure the optimum value of the objective constrained to this codition. Similarly if α_2 goes below the lower limit then reset it to the lower limit.

23.2 Probablistic models

In one of the previous lectures probablistic models were mentioned. They are of two types *conditional* and *generative* based on the variable over which the distribution is defined. Conditional models define a distribution over class given the input and the Generative models define a joint distribution of both dependent and independent variables.

The classification models can again be divided into two - parametric and non-parametric. The parametric forms assumes a distribution over the class or the input which are controlled by parameters, for example the class output $\sim (N)(w^T\phi(x), \sigma)$ where σ,w are parameters . During the training phase the parameters would be learned.

For a classification task we will have a scoring function $g_k(x)$ based on which we will dot classification. The point x will be classified to $\text{argmax}_k\, g_k(x)$.

For a discriminative model the function $g_k(x) = ln(p(C_k|x))$, ie it models the conitional probability of the class variable with respect to the input.

For a generative models $g_k(x) = ln(p(x|C_k))p(C_k) - ln(p(x))$(can be obtained by Bayes rule). The generative model model a joint distribution of the input and class variables.

24 Lecture 24: Prob. Classifiers

24.1 Non Parametric Density Estimation

The models in which the form of the model is not specified a priori but is instead determined from data are called *Non parametric models*. The term non-parametric is not meant to imply that such models completely lack parameters but that the number and nature of the parameters are flexible and not fixed in advance. Non Parametric models are generally generative methods.

The probability P that a vector $\mathbf{x}$ will fall in a region R is given by

$$P = \int_R p(\mathbf{x})\, d\mathbf{x} \tag{159}$$

Suppose that n iid samples are randomly drawn according to the probability distribution $p(\mathbf{x})$. Probability that k of these n fall in R is

$$P_k = \binom{n}{k} P^k (1-P)^{(n-k)} \tag{160}$$

Expected value of k is

$$E[k] = nP \tag{161}$$

If n is very large then $\frac{k}{n}$ will be a good estimate for the probability P.

$$\int_R p(\mathbf{x}')\, d\mathbf{x}' \simeq p(\mathbf{x})V \tag{162}$$

where $\mathbf{x}$ is a point within R and V is the volume enclosed by R. Combining Eqs (159) & (162) we get:

$$p(\mathbf{x}) \simeq \frac{k}{nV} \tag{163}$$

- **Kernel Density Estimation**
  One technique for nonparametric density estimation is the kernel density estimation where, effectively, V is held fixed while K, the number of sample points

lying withing V is estimated. The density is given by treating K as a kernel function, *e.g.*, a Gaussian function, centered on each data point and then adding the functions together. The quality of the estimate depends crucially on the kernel function. For the Gaussian case, the non-parametric density estimator is known as the Parzen-window density estimator.

In the Eq (163) we could keep V fixed & estimate K. For example, we could consider V to be a hypercube of length d & dimension n:

$$P(\mathbf{x}) = \Sigma_{\mathbf{x}' \epsilon D} \frac{K(\mathbf{x}, \mathbf{x}')}{d^n * |D|} \tag{164}$$

$$K(\mathbf{x}, \mathbf{x}') = \begin{cases} 1 & if\ \|x_i - x'_i\| \leq d \ \forall \ i \in [1:n] \\ 0 & if\ \|x_i - x'_i\| > d\ for\ some\ i \in [1:n] \end{cases} \tag{165}$$

For smooth kernel density estimation, we could use

$$K(\mathbf{x}, \mathbf{x}') = e^{\frac{-\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2}} \tag{166}$$

Since K is smooth we need not specify volume (V); σ implicitly defines V.

$$P(\mathbf{x}) = \frac{1}{|D|} \Sigma_{\mathbf{x}' \epsilon D} K(\mathbf{x}, \mathbf{x}') \tag{167}$$

Parzen Window Classifier:

– Given r classes, for each class build a density model.

$$D = D_1 \cup D_2 ...... \cup D_r$$

$$P(\mathbf{x}|C_i) = P_i(\mathbf{x}) = \frac{1}{|D_i|} \Sigma_{\mathbf{x}' \epsilon D_i} K(\mathbf{x}, \mathbf{x}')$$

– estimate $P(C_i) = \frac{|D_i|}{|D|}$

– The class chosen is the one that maximizes the posterior distribution, *i.e.*, $\underset{i}{\text{argmax}}\ a_i(\mathbf{x}) = log[P(\mathbf{x}|C_i)P(C_i)]$
*(assuming the same σ for all classes)

A potential problem with kernel density classifiers can be that they could be biased toward more populated classes, owing to the class prior.

$$P(C_i) = \frac{|D_i|}{|D|}$$

A more severe issue is that σ is fixed for all classes.

- **K-Nearest Neighbours(K-NN)**
The idea behind this class of kernel desity estimators is to hold K constant and instead determine the volume V of the tighest sphere that encompasses the K samples. The volumne is a non-decreasing function of K. K-NN is non-smooth.

$$P(C_i|\mathbf{x}) = \frac{K_i}{K}$$

where K_i is number of points that fall in class C_i out of K points nearest to a given point $\mathbf{x}$. The steps in K-NN density estimation are as follows:

  - Given a point $\mathbf{x}$, find the set of K nearest neighbours ($\mathcal{D}_k$)
  - For each class C_i compute $K_i = |\{\mathbf{x}\ |\mathbf{x} \in \mathcal{D}_k\ and\ \mathbf{x} \in C_i\}|$, that is K_i is the number of points from class C_i that belong to the nearest neighbour set $\mathcal{D}_k$.
  - Classify $\mathbf{x}$ into $C_j = \underset{C_i}{\text{argmax}} \frac{K_i}{K}$

The decision boundaries of K-NN are highly non-linear.

24.2 Parametric Density Estimation

Parametric methods assume a form for the probability distribution that generates the data and estimate the parameters of the distribution. Generally parametric methods make more assumptions than non-parametric methods.

- *Gaussian Discriminant*:

$$P(\mathbf{x}|C_i) = \mathcal{N}(\phi(\mathbf{x}), \mu_i, \Sigma_i) = \frac{1}{(2\pi)^{n/2}|\Sigma_i|^{1/2}} exp(\frac{-(\phi(\mathbf{x}) - \mu_i)\Sigma_i^{-1}(\phi(\mathbf{x}) - \mu_i)}{2}) \tag{168}$$

Given point $\mathbf{x}$,classify it to class C_i such that,

$$C_i = \text{argmax}_i\ log[P(\mathbf{x}|C_i)P(C_i)]$$

let $a_i = log[P(\mathbf{x}|C_i)] + log[P(C_i)] = w_i^T \phi(\mathbf{x}) + w_{i_0}$
where,

$w_i = \Sigma_i \mu_i$
$w_{i_0} = \frac{-1}{2}\mu_i^T \Sigma_i^{-1}\mu_i + ln[P(C_i)] - \frac{1}{2}ln[(2\pi)^n|\Sigma_i|] - \frac{1}{2}\phi(\mathbf{x})^T\Sigma_i^{-1}\phi(\mathbf{x})$

Maximum Likelihood Estimation:

$$(\mu_i^{MLE}, \Sigma_i^{MLE}) = argmax_{\mu,\Sigma}\ LL(D,\mu,\Sigma) = argmax_{\mu,\Sigma}\ \Sigma_i\Sigma_{\mathbf{x}\epsilon D_i} log[\mathcal{N}(\mathbf{x},\mu_i,\Sigma_i)] \tag{169}$$

1. Σ 's are common across all classes i.e., $\Sigma_i = \Sigma \ \ \forall i$
   Maximum Likelihood estimates using (169) are:
   $$\mu_i^{MLE} = \Sigma_{\mathbf{x}\epsilon D_i}\frac{\phi(\mathbf{x})}{|D_i|}$$
   $$\Sigma^{MLE} = \Sigma_{i=1}^k \frac{1}{|D|}\Sigma_{\mathbf{x}\epsilon D_i}(\phi(\mathbf{x}) - \mu_i)(\phi(\mathbf{x}) - \mu_i)^T$$
2. $\Sigma_i' s$ are also parameters
   Maximum Likelihood estimates are:
   $$\mu_i^{MLE} = \Sigma_{\mathbf{x}\epsilon D_i}\frac{\phi(\mathbf{x})}{|D_i|}$$
   $$\Sigma_i^{MLE} = \Sigma_{\mathbf{x}\epsilon D_i}\frac{1}{|D_i|}(\phi(\mathbf{x}) - \mu_i)(\phi(\mathbf{x}) - \mu_i)^T$$

We could do this for exponential family as well.

- *Exponential Family*:
  For a given vector of functions $\phi(\mathbf{x}) = [\phi_1(\mathbf{x}), \ldots, \phi_k(\mathbf{x})]$ and a parameter vector $\eta\epsilon\Re^k$, the exponential family of distributions is defined as
  $$P(\mathbf{x},\eta) = h(\mathbf{x})exp\{\eta^T\phi(\mathbf{x}) - A(\eta)\} \tag{170}$$
  where the $h(\mathbf{x})$ is a conventional reference function and A(η) is the log normalization constant designed as
  $$A(\eta) = log[\int_{\mathbf{x}\epsilon Range(\mathbf{x})} exp\{\eta^T\phi(\mathbf{x})\}h(\mathbf{x})d\mathbf{x}]$$

  Example:

  * **Gaussian Distribution:** Gaussian Distribution $X \sim \mathcal{N}(\mu,\sigma)$ can be expressed as
    $$\eta = [\tfrac{\mu}{\sigma^2}, \tfrac{-1}{2\sigma^2}]$$
    $$\phi(x) = [x, x^2]$$

- **Multivariate Gaussian**:

$$\eta = \Sigma^{-1}\mu$$

$$p(\mathbf{x}, \Sigma^{-1}, \eta) = exp\{\eta^T\mathbf{x} - \frac{1}{2}\mathbf{x}^T\Sigma^{-1}\mathbf{x} + Z\}$$

where $Z = \frac{-1}{2}(nlog(2\pi) - log(|\Sigma^{-1}|))$

- **Bernoulli**:
Bernoulli distribution is defined on a binary (0 or 1) random variable using parameter μ where $\mu = Pr(X = 1)$. The Bernoulli distribution can be written as

$$p(x|\mu) = exp\{xlog(\tfrac{\mu}{1-\mu}) + log(1-\mu)\}$$

$\Rightarrow \phi(x) = [x]$ and $\eta = [log(\frac{\mu}{1-\mu})]$

Bernoulli is important if $\phi(x)$ contains discrete values.

Say for each class C_k, we have parameter η_k
$a_k(\mathbf{x}) = ln(p(\mathbf{x}|C_k).p(C_k)) = \eta_k^T\phi(\mathbf{x}) - A(\eta_k) + ln(h(\mathbf{x}))$

$$\Rightarrow a_i(\mathbf{x}) = a_j(\mathbf{x}) \text{ gives us a linear discriminant}$$

25 Lecture 25

25.1 Exponential Family Distribution

Considering The Exponential Family Distribution:

$$p(\phi(x)|\eta_k) = h(\phi(x)) \exp(\eta_k^T \phi(x) - A(\eta_k))(for\ class\ k). \tag{171}$$

25.2 Discrete Feature Space

$\phi(x)$ is the feature space. Considering the case when it is discrete valued.

$$\phi(x) = [attr1, attr2, ..] \tag{172}$$

Let $\phi(x)$ have n attributes and each of the n attributes can take c different (discrete) values.
Total number of possible values of $\phi(x)$ is c^n.

$$\phi(x) = \begin{pmatrix} nval_1 & nval_2 & . & . & . & .nval_n \\ \phi^{(1)}(x) & . & . & . & . & . \\ \phi^{(2)}(x) & . & . & . & . & . \\ \vdots & . & . & . & . & . \\ \vdots & . & . & . & . & . \\ \phi^{(c^n)}(x) & . & . & . & . & . \end{pmatrix} \tag{173}$$

Size of the above table is c^n.

For example, when $n = 3$ and $c = 2$ ($\{0, 1\}$ $\forall$ attributes), table showing all possible values of $\phi(x)$ will be:

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \tag{174}$$

Thus size of the table of $p(\phi^i(x)|\eta_k)$ will also be c^n.

$$p(\phi(x)|\eta_k) = \begin{pmatrix} Probaility & | & Configuration \\ p_1 & | & \phi^{(1)}(x) \\ p_2 & | & \phi^{(2)}(x) \\ : & | & : \\ : & | & : \\ p_{c^n} & | & \phi^{(c^n)}(x) \end{pmatrix} \tag{175}$$

$$where \sum_{\mathbf{i}} p_i = 1 \tag{176}$$

It is clear that $p(\phi(x)|\eta_k) \equiv p(x|\eta_k)$

25.3 Naive Bayes Assumption

As the size(c^n) is exponential in the dimension of feature space, it is not feasible to work with the full table even in a
moderately large dimension feature space.
One possible way out is to approximate the probability distribution so that this size is reduced considerably.
Naive Bayes is one such approximation in which the assumption is:

$$p(\phi(x)|\eta_k) = p(\phi_1(x)|\eta_k)p(\phi_2(x)|\eta_k)\dots p(\phi_n(x)|\eta_k) \tag{177}$$

where, $\phi_i(x)$ denotes the i-th attribute of $\phi(x)$ in the feature space.
Thus, what Naive Bayes Assumption essentially says is that each attribute is independent of other attributes given the class.
So the original probability distribution table of $p(\phi(x)|\eta_k)$ of size c^n will get replaced by n tables
(One per attribute) each of size $cx1$ as follows :

$$p(\phi_j(x)|\eta_k) = \begin{pmatrix} p_{j1} & | & \phi_j(x) = \mu_{j1} \\ p_{j2} & | & \phi_j(x) = \mu_{j2} \\ : & | & : \\ : & | & : \\ p_{jc} & | & \phi_j(x) = \mu_{jc} \end{pmatrix} \tag{178}$$

where $\mu_{j1}..\mu_{jc}$ are c discrete values that $\phi_j(x)$ can take

$$and \sum_{\mathbf{i}} p_{ji} = 1 \tag{179}$$

NOTE:This assumption does NOT say that :
$p(\phi(x)) = p(\phi_1(x)) \dots p(\phi_n(x))$

25.4 Graphical Models

Discussion on Graphical Models was done from the slides [8], [9].

25.5 Graphical Representation of Naive Bayes

$$c = \phi_1(x)\phi_2(x) \dots \phi_n(x) \tag{180}$$

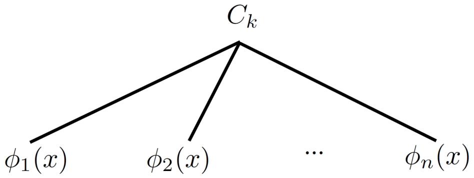


Figure 29: Graphical model representation of Naive Bayes

The fact that there is no Edge between $\phi_1(x)$ and $\phi_2(x)$ denotes Conditional Independence.

$$\begin{aligned}
p(\phi, c) &= p(\phi|c).p(c) \\
&= p(\phi_2|\phi_1, \phi_3, c).p(\phi_1|\phi_3, c).p(\phi_3|c).p(c) \\
&\approx p(\phi_2|\phi_1).p(\phi_1|c).p(\phi_3|c).p(c)(FromFigure29) \\
&= \Pi p(x|\pi(x))(\pi(x) \equiv Set\ of\ Parents\ of\ x)
\end{aligned}$$

25.6 Graph Factorisation

Think of $p(\phi(x), c)$(Factorised) as Specifying a Family of Distributions.
$a \coprod b \mid c$ (a is conditionally independent of b given c) means:

$$p(a|b, c) = p(a|c) \tag{181}$$

$$p(b|a, c) = p(b|c) \tag{182}$$

25.7 Naive Bayes Text Classification

Naive Bayes Text Classification was discussed from the slides [10], [11].

References

[1] "Class notes: Basics of convex optimization,chapter.4."

[2] "Convex Optimization."

[3] "Linear Algebra."

[4] "Bias variance tradeoff." [Online]. Available: http://www.aiaccess.net/English/Glossaries/GlosMod/e_gm_bias_variance.htm

[5] Steve Renals, "Support Vector Machine."

[6] Christopher M. Bishop, "Pattern Recognition And Machine Learning."

[7] Nello Cristianini and John Shawe-Taylor , "An Introduction to Support Vector Machines and Other Kernel-based Learning Methods."

[8] ClassNotes, "Graphical Models,Class Notes."

[9] ——, "Graphical Case Study of Probabilistic Models,Class Notes."

[10] Andrew McCallum,Kamal Nigam, "A Comparison of Event Models for Naive Bayes Text Classification."

[11] Steve Renals, "Naive Bayes Text Classification."