Machine learning notes

Francesco Boi

1 Preliminary definitions

1.1 Trace of a matrix

The **trace** of a square matrix $\mathbf{A}$, denoted as $\mathbf{Tr}(\mathbf{A})$ is the sum of diagonal elements:

$$\mathbf{Tr}(\mathbf{A}) = \sum_{d=1}^{p} A_{dd} \tag{1}$$

It follows that $\mathrm{Tr}(\mathbf{I_d}) = p$. Also $\mathbf{Tr}(\mathbf{AB}) = \mathbf{Tr}(\mathbf{AB})$ and $\mathbf{Tr}(\mathbf{x}^T\mathbf{x}) = \mathbf{x}^T\mathbf{x}$ the latter being a scalar.

1.2 Expectation

Definition 1.1. Expectation Let X be a random variable with a finite number of outcomes $x_1, x_2, \cdots, x_k$ occurring respectively with probabilities $p_1, p_2, \cdots, p_k$. The expectation value is the summation of each outcome times its probability.

$$E[X] = \sum_{k} x_k \cdot p_k \tag{2}$$

In case of an infinite number of outcomes the summation is replaced with the integral:

$$E[X] = \int x \cdot p(x)\, dx \tag{3}$$

1.3 Variance

Definition 1.2. Variance The variance of a random variable X is the expected value of the squared deviation from the mean of X:

$$\mathrm{Var}(X) = E\left[(X - \mu)^2\right] \tag{4}$$

Variance can be expressed in another way recalling $\mu = \mathrm{E}[\mathrm{X}]$:

$$\mathrm{Var}(\mathrm{X}) = \mathrm{E}\left[(\mathrm{X} - \mu)^2\right] = \mathrm{E}\left[(\mathrm{X} - \mathrm{E}[\mathrm{X}])^2\right] = \tag{5}$$

$$= \mathrm{E}\left[\mathrm{X}^2 - 2 \cdot \mathrm{X} \cdot \mathrm{E}[\mathrm{X}] + \mathrm{E}[\mathrm{X}]^2\right] \tag{6}$$

Using the linearity property:

$$\mathrm{Var}(\mathrm{X}) = \mathrm{E}\left[(\mathrm{X} - \mu)^2\right] = \tag{7}$$

$$= \mathrm{E}\left[\mathrm{X}^2\right] - 2 \cdot \mathrm{E}[\mathrm{X} \cdot \mu] + \mathrm{E}\left[\mu^2\right] \tag{8}$$

$$= \mathrm{E}\left[\mathrm{X}^2\right] - 2 \cdot \mu \cdot \mathrm{E}[\mathrm{X}] + \mu^2 \tag{9}$$

$$= \mathrm{E}\left[\mathrm{X}^2\right] - \mu^2 = \mathrm{E}\left[\mathrm{X}^2\right] - \mathrm{E}[\mathrm{X}]^2 \tag{10}$$

$$\tag{11}$$

Definition 1.3. Median For any probability distribution on the real line $\mathbb{R}$ with cumulative distribution function F, regardless of whether it is any kind of continuous probability distribution, in particular an absolutely continuous distribution (which has a probability density function), or a discrete probability distribution, a median is by definition any real number m that satisfies the inequalities:

$$\mathrm{P}(\mathrm{x} \leq \mathrm{m}) \geq \frac{1}{2} \quad \text{and} \quad \mathrm{P}(\mathrm{x} \leq \mathrm{m}) \geq \frac{1}{2} \tag{12}$$

$$\int_{-\infty}^{\mathrm{m}} \mathrm{F}(\mathrm{x})\mathrm{dx} \geq \frac{1}{2} \quad \text{and} \quad \int_{\mathrm{m}}^{\infty} \mathrm{F}(\mathrm{x})\mathrm{dx} \geq \frac{1}{2} \tag{13}$$

1.4 Median as the minimizer of L_1 norm

Assume that S is a finite set, with say k elements. Line them up in order, as $\mathrm{s}_1 < \mathrm{s}_2 < \cdots < \mathrm{s}_\mathrm{k}$.

If k is even there are (depending on the exact definition of median) many medians. $|\mathrm{x} - \mathrm{s}_\mathrm{i}|$ is the **distance** between x and s_i, so we are trying to minimize the sum of the distances. For example, we have k people who live at various points on the x-axis. We want to find the point(s) x such that the sum of the travel distances of the k people to x is a minimum.

Imagine that the s_i are points on the x-axis. For clarity, take $\mathrm{k} = 7$. Start from well to the left of all the s_i, and take a tiny step, say of length ϵ, to the

right. Then you have gotten ϵ closer to every one of the s_i, so the sum of the distances has decreased by 7ϵ.

Keep taking tiny steps to the right, each time getting a decrease of 7ϵ. This continues until you hit s_1. If you now take a tiny step to the right, then your distance from s_1 increases by ϵ, and your distance from each of the remaining s_i decreases by ϵ. So there is a decrease of 6ϵ, and an increase of ϵ, for a net decrease of 5ϵ in the sum.

This continues until you hit s_2. Now, when you take a tiny step to the right, your distance from each of s_1 and s_2 increases by ϵ, and your distance from each of the five others decreases by ϵ, for a net decrease of 3ϵ.

This continues until you hit s_3. The next tiny step gives an increase of 3ϵ, and a decrease of 4ϵ, for a net decrease of ϵ.

This continues until you hit s_4. The next little step brings a total increase of 4ϵ, and a total decrease of 3ϵ, for an increase of ϵ. Things get even worse when you travel further to the right. So the minimum sum of distances is reached at s_4, the median.

The situation is quite similar if k is even, say $k = 6$. As you travel to the right, there is a net decrease at every step, until you hit s_3. When you are between s_3 and s_4, a tiny step of ϵ increases your distance from each of s_1, s_2, and s_3 by ϵ. But it decreases your distance from each of the three others, for no net gain. Thus any x in the interval from s_3 to s_4, including the endpoints, minimizes the sum of the distances. In the even case, Some people prefer to say that any point between the two "middle" points is a median. So the conclusion is that the points that minimize the sum are the medians. Other people prefer to define the median in the even case to be the average of the two "middle" points. Then the median does minimize the sum of the distances, but some other points also do.

In formulas consider two x_i's x_1 and x_2,

- $$x_1 \leq a \leq x_2 \tag{14}$$

$$\sum_{i=1}^{2} |x_i - a| = |x_1 - a| + |x_2 - a| = a - x_1 + x_2 - a = x_2 - x_1 \tag{15}$$

•

$$a < x_1$$
$$\sum_{i=1}^{2} |x_i - a| = x_1 - a + x_2 - a = x_1 + x_2 - 2a \geq x_1 + x_2 - 2x_1$$
$$= x_2 - x_1 \tag{16}$$

•

$$a \geq x_2$$
$$\sum_{i=1}^{2} |x_i - a| \quad = -x_1 + a - x_2 + a = -x_1 - x_2 + 2a \geq -x_1 - x_2 + 2x_2 =$$
$$= x_2 - x_1 \tag{17}$$

$\Longrightarrow$ for any two x_i's the sum of the absolute values of the deviations is minimum when $x_1 \leq a \leq x_2$ or $a \in [x_1, x_2]$.

When n is odd,

$$\sum_{i=1}^{n} |x_i - a| = |x_1 - a| + |x_2 - a| + \cdots + \left|x_{\frac{n-1}{2}} - a\right| + \left|x_{\frac{n+1}{2}} - a\right| + \\ + \left|x_{\frac{n+3}{2}} - a\right| + \cdots + |x_{n-1} - a| + |x_n - a| \tag{18}$$

consider the intervals $[x_1, x_n], [x_2, x_{n-1}], [x_3, x_{n-2}], \ldots, \left[x_{\frac{n-1}{2}}, x_{\frac{n+3}{2}}\right]$. If a is a member of all these intervals. i.e, $\left[x_{\frac{n-1}{2}}, x_{\frac{n+3}{2}}\right]$,

using the above theorem, we can say that all the terms in the sum except $\left|x_{\frac{n+1}{2}} - a\right|$ are minimized. So

$$\sum_{i=1}^{n} |x_i - a| = (x_n - x_1) + (x_{n-1} - x_2) + (x_{n-2} - x_3) + \cdots + \\ + \left(x_{\frac{n+3}{2}} - x_{\frac{n-1}{2}}\right) + \left|x_{\frac{n+1}{2}} - a\right| = \left|x_{\frac{n+1}{2}} - a\right| + \text{costant} \tag{19}$$

To minimize also the term $\left|x_{\frac{n+1}{2}} - a\right|$ it is clear we have to choose $a = x_{\frac{n+1}{2}}$ to get 0 but this is the definition of the median.

$\Longrightarrow$ When n is odd,the median minimizes the sum of absolute values of the deviations.

When n is even,

$$\begin{aligned}\sum_{i=1}^{n} |x_i - a| &= |x_1 - a| + |x_2 - a| + \cdots + |x_{\frac{n}{2}} - a| + \\ &+ |x_{\frac{n}{2}+1} - a| + \cdots + |x_{n-1} - a| + |x_n - a|\end{aligned} \tag{20}$$

If a is a member of all the intervals $[x_1, x_n], [x_2, x_{n-1}], [x_3, x_{n-2}], \ldots, \left[x_{\frac{n}{2}}, x_{\frac{n}{2}+1}\right]$, i.e, $a \in \left[x_{\frac{n}{2}}, x_{\frac{n}{2}+1}\right]$,

$$\sum_{i=1}^{n} |x_i - a| = (x_n - x_1) + (x_{n-1} - x_2) + (x_{n-2} - x_3) + \cdots + \left(x_{\frac{n}{2}+1} - x_{\frac{n}{2}}\right) \tag{21}$$

$\Longrightarrow$ When n is even, any number in the interval $[x_{\frac{n}{2}}, x_{\frac{n}{2}+1}]$, i.e, including the median, minimizes the sum of absolute values of the deviations. For example consider the series:$2, 4, 5, 10$, median, $M = 4.5$.

$$\sum_{i=1}^{4} |x_i - M| = 2.5 + 0.5 + 0.5 + 5.5 = 9$$

If you take any other value in the interval $\left[x_{\frac{n}{2}}, x_{\frac{n}{2}+1}\right] = [4, 5]$, say 4.1

$$\sum_{i=1}^{4} |x_i - 4.1| = 2.1 + 0.1 + 0.9 + 5.9 = 9$$

Taking for example 4 or 5 yields the same result:

$$\sum_{i=1}^{4} |x_i - 4| = 2 + 0 + 1 + 6 = 9$$

$$\sum_{i=1}^{4} |x_i - 5| = 3 + 1 + 0 + 5 = 9$$

This is because when summing the distance from a to the two middle points, you end up with the distance between them: $a - x_{\frac{n}{2}} + (x_{\frac{n}{2}+1} - a) = x_{\frac{n}{2}+1} - x_{\frac{n}{2}}$

For any value outside the interval $\left[x_{\frac{n}{2}}, x_{\frac{n}{2}+1}\right] = [4, 5]$, say 5.2

$$\sum_{i=1}^{4} |x_i - 5.2| = 3.2 + 1.2 + 0.2 + 4.8 = 9.4$$

2 Statistical Decision Theory

2.1 Expected prediction error

Let $X \in \mathbb{R}^p$ denote a real valued random input vector, and $Y \in \mathbb{R}^p$ a real valued random output variable, with joint distribution $Pr(X, Y)$. We seek a function $f(X)$ for predicting Y given values of the input X. This theory requires a loss function$L(Y, f(X))$ for penalizing errors in prediction, and by far the most common and convenient is squared error loss: $L(Y, f(X)) = (Y - f(X))^2$.

The estimated prediction error is

$$EPE(f) = \mathbf{E}\left[(Y - f(X))^2\right] = \int (y - f(x))^2 \, p(x, y) dxdy \tag{22}$$

$$= \int_x \int_y (y - f(x))^2 \, p(x, y) dxdy \tag{23}$$

$$\tag{24}$$

Recalling $p(x, y) = p(y|x)p(x)$:

$$\begin{aligned} \mathrm{EPE}(f) &= \int_x \int_y (y - f(x))^2 \, p(y|x)p(x)dxdy && (25) \\ &= \int_x \int_y \left((y - f(x))^2 \, p(y|x)dy \right) p(x)dx && (26) \\ &= \int_x \mathbf{E}_{Y|X} \left[(y - f(X))^2 \, |X = x \right] p(x)dx && (27) \\ &= \mathbf{E}_X \left[\mathbf{E}_{Y|X} \left[(y - f(X))^2 \, |X = x \right] \right] && (28) \end{aligned}$$

So to minimize the prediction error:

$$f(x) = \arg\min_c \mathbf{E}_{Y|X} \left[(y - c(x))^2 \, |X = x \right] \quad \Rightarrow f(x) = \mathbf{E} \left[Y|X = x \right] \qquad (29)$$

The *Nearest Neighbour* algorithm, which assigns labels to points by counting and averaging the labels of the points belonging to a given neighbourhood:

$$\hat{f}(x) = \mathrm{Ave} \left(y_i | x_i \in N_k(x) \right) \qquad (30)$$

where $N_k(x)$ contains the k points closest to x. This presents **two approximations**

- expectation is approximated by averaging
- conditioning at a point is relaxed to conditioning on some region centred at the target point.

With k sufficiently large, the average gets more stable and with large N the points will be more likely close to x. If $k, N \to \infty$ with $k/N \to 0$ the average becomes the expectation and we have the best possible estimator.

Unfortunately often we do not have so much data and some other times we might want exploit the supposed structure of data (linear, polynomial etc.).

However there is even a bigger problem. When there are too many dimensions (i.e., p is large). Consider a uniformly distributed input in a p dimensional unit hypercube. Consider a hypercube neighbourhood around the target point capturing a fraction r of the total observations distributed

among the unit hypercube. The edge of the neighbour hypercube will be $e_p(r) = r^{1/p}$. In 10 dimensions, using a neighborhood capturing 1% of the data we have $r(0.01) = 0.63$ so we must use 63% of the total data for one target.

On the contrary the linear regression is a **model-based approach**, i.e., one assumes that the function $f(x)$ is approximately linear:

$$f(x) \approx x^T \beta \tag{31}$$

Putting this in the EPE equation:

$$f(x) = \arg\min_c \mathbf{E}_{Y|X} \left[(y - c(x))^2 \, | X = x \right] \tag{32}$$

$$= \arg\min_\beta \mathbf{E}_{Y|X} \left[\left(y - x^T \beta \right)^2 | X = x \right] \tag{33}$$

$$\Rightarrow \beta = \left[\mathbf{E} \left[X \cdot X^T \right] \right]^{-1} \cdot \mathbf{E} \left[X \cdot Y \right] \tag{34}$$

The minimum of a quadratic function is given by deriving and setting its derivative to 0.

If instead of a L_2 loss function we use L_1

$$\mathrm{EPE}(f) = \mathbf{E} \left[|Y - f(X)| \right] = \int |y - f(x)| \, p(x, y) dx dy \tag{35}$$

$$= \int_x \int_y |y - f(x)| \, p(x, y) dx dy \tag{36}$$

$$\tag{37}$$

Recalling $p(x, y) = p(y|x)p(x)$:

$$\mathrm{EPE}(f) = \int_x \int_y |y - f(x)| \, p(y|x) p(x) dx dy \tag{38}$$

$$= \int_x \int_y \left| (y - f(x)| \, p(y|x) dy \right) p(x) dx \tag{39}$$

$$= \int_x \mathbf{E}_{Y|X} \left[|y - f(X)| \, | X = x \right] p(x) dx \tag{40}$$

$$= \mathbf{E}_X \left[\mathbf{E}_{Y|X} \left[|y - f(X)| \, | X = x \right] \right] \tag{41}$$

$$f(x) = \arg\min_{c} \mathbf{E}_{Y|X} \left[|y - c(x)| \, |X = x \right] \tag{42}$$

and as already seen in 1.4, the minimizer for the sum of distances is the median.

2.1.1 Loss function for categorical variables

For categorical output variables $\mathbb{G}_k$ we have:

$$\text{EPE} = \mathbf{E}\left[L\left(G, \hat{G}(X) \right) \right] = \mathbf{E}_x \sum_{k=1}^{K} L\left[\mathbb{G}_k, \hat{G}(X) \right] \Pr\left(\mathbb{G}_k | X \right) \tag{43}$$

where the expectation is again taken with respect to the joint distribution $\Pr(G, X)$. Conditioning again we can write:

$$\text{EPE} = \mathbf{E}_x \sum_{k=1}^{K} L\left[\mathbb{G}_k, \hat{G}(X) \right] \Pr\left(\mathbb{G}_k | X \right) \tag{44}$$

where the integral over y has been substituted with the summation due to the categorical nature of the variable.

The minimizer is given by:

$$\hat{G}(x) = \arg\min_{g \in \mathbb{G}} \sum_{k=1}^{K} L\left(\mathbb{G}_k, g \right) \Pr\left(\mathbb{G}_k | X = x \right) \tag{45}$$

Often the *zero-one loss function* is used for categorical variables and the above simplifies to:

$$\hat{G}(x) = \arg\min_{g \in \mathbb{G}} \left[1 - \Pr\left(g | X = x \right) \right] = \arg\max_{g \in \mathbb{G}} \left[\Pr\left(g | X = x \right) \right] \tag{46}$$

This is known as *Bayes classifier* because **it classifies to the most probable class, using conditional discrete probability distribution**.

Note: When models or loss functions use additional parameters that penalize complexity (Lasso, Ridge and others) we cannot use the training data to determine these parameters, since we would pick those that gave interpolating fits with zero residuals but it will be unlikely to predict future data.

2.2 Bias-Variance trade-off

$$\begin{aligned} \mathrm{EPE(f)} &= \mathbf{E}\left[(\mathrm{Y} - \mathrm{f(X)})^2\right] = && (47)\\ &= \mathbf{E}\left[\mathrm{Y}^2\right] - 2\mathbf{E}\left[\mathrm{Y}\right]\mathbf{E}\left[\mathrm{f(X)}\right] + \mathbf{E}\left[\mathrm{f(X)}^2\right] && (48)\\ &= \mathrm{Y}^2 - 2\mathrm{Y}\mathbf{E}\left[\mathrm{f(X)}\right] + \mathbf{E}\left[\mathrm{f(X)}^2\right] && (49) \end{aligned}$$

Recalling

$$\begin{aligned} &\mathrm{BIAS}\left(\mathrm{Y}, \mathbf{E}\left[\mathrm{f(X)}\right]\right) = \left|\mathrm{Y} - \mathbf{E}\left[\mathrm{f(X)}\right]\right| && (50)\\ &\Rightarrow \mathrm{BIAS}\left(\mathrm{Y}, \mathrm{f(X)}\right)^2 = \left(\mathrm{Y} - \mathbf{E}\left[\mathrm{f(X)}\right]\right)^2 && (51)\\ &= \mathrm{Y}^2 - 2\mathrm{Y}\mathbf{E}\left[\mathrm{f(X)}\right] + \mathbf{E}\left[\mathrm{f(X)}\right]^2 && (52)\\ & && (53) \end{aligned}$$

EPE can be expressed using also the Variance definition in 1.3

$$\begin{aligned} \mathrm{EPE(f)} &= \mathrm{Y}^2 - 2\mathrm{Y}\mathbf{E}\left[\mathrm{f(X)}\right] + \mathbf{E}\left[\mathrm{f(X)}^2\right] + \mathbf{E}\left[\mathrm{f(X)}\right]^2 - \mathbf{E}\left[\mathrm{f(X)}\right]^2 && (54)\\ &= \mathrm{BIAS(Y, f(X)}^2 + \mathrm{Var(Y, f(x))} && (55) \end{aligned}$$

The bias is given by the distance of our predictions from real points. Complex models have more degrees of freedom and are able to fit closer real points hence they to tend to have low bias. However, they present higher variance. On the contrary simple models (i.e., linear) have lower variance but higher bias.

The error due to variance is the amount by which the prediction, over one training set, differs from the expected predicted value, over all the training sets. As with bias, you can repeat the entire model building process multiple times. To paraphrase Manning et al (2008), variance measures how inconsistent are the predictions from one another, over different training sets, not whether they are accurate or not. A learning algorithm with low bias must

be "flexible" so that it can fit the data well. But if the learning algorithm is too flexible, it will fit each training data set differently, and hence have high variance.

3 Linear Regression Models

We start from the *univariate linear regression*, i.e., each output consists of a single value while the input is a vector of values.

3.1 Univariate linear regression

Univariate means single output, i.e, y is a vector. The basic form is

$$\mathrm{f(X)} = \beta_0 + \sum_{\mathrm{j–1}}^{\mathrm{p}} \mathrm{X_j}\beta_\mathrm{j} \tag{56}$$

The most popular estimation method for a linear model is the least square:

$$\mathrm{RSS}(\beta) = \sum_{\mathrm{i=1}}^{\mathrm{p}} (\mathrm{y_i} - \mathrm{f(x_i)})^2 = (\mathbf{y} - \mathbf{X}\beta)^{\mathrm{T}} (\mathbf{y} - \mathbf{X}\beta) \tag{57}$$

where $\mathbf{X}$ is a $\mathrm{N} \times (\mathrm{p}+1)$ matrix with each row being an input vector, $\mathbf{y}$ a N vector (we must have N input-output pairs, in this case the output is considered mono-dimensional).

By minimizing we get:

$$\hat{\beta} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1} \mathbf{X}^{\mathrm{T}}\mathrm{Y} \tag{58}$$

Geometrically we are projecting y onto the hyperplane spanned by X and the projections is referred to as $\hat{\mathrm{y}}$:

$$\hat{\mathrm{y}} = \mathbf{X}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1} \mathbf{X}^{\mathrm{T}}\mathrm{Y} = \mathrm{HY} \tag{59}$$

where H is called the *hat* matrix.

3.2 Equivalence of Ordinary least squares and maximum likelihood

We are using an additive model, assuming a Gaussian white noise:

$$\mathrm{y} = \beta^{\mathrm{T}}\mathbf{x} + \epsilon \tag{60}$$

$$\epsilon \sim \mathcal{N}(0, \sigma^2) \tag{61}$$

Adding a constant to a Gaussian random variable is equivalent to another Gaussian random variable with the mean shifted:

$$\Pr(\mathrm{y}) \sim \mathcal{N}(\beta^{\mathrm{T}}\mathbf{x}, \sigma^2) \tag{62}$$

Considering the matrix $\mathbf{X}$ and the output vector $\mathbf{y}$ representing the training set, used to estimate the coefficients, we have:

$$\Pr(\mathbf{y}|\mathbf{X}, \beta, \sigma^2) = \prod_{\mathrm{i}=1}^{\mathrm{N}} \Pr(\mathrm{y_i}|\mathbf{X}, \beta, \sigma^2) = \prod_{\mathrm{i}=1}^{\mathrm{N}} \mathcal{N}(\beta^{\mathrm{T}}\mathbf{x}, \sigma^2) \tag{63}$$

where we have assumed each observation is independent. A product of univariate Gaussian can be rewritten as a multivariate Gaussian:

$$\begin{aligned}
\Pr(\mathbf{y}|\mathbf{X}, \beta, \sigma^2) &= \prod_{\mathrm{i}=1}^{\mathrm{N}} \mathcal{N}(\beta^{\mathrm{T}}\mathbf{x}, \sigma^2) = \prod_{\mathrm{i}=1}^{\mathrm{N}} \frac{1}{(2\pi)^{\frac{1}{2}}\sigma} \mathrm{e}^{-\frac{\left(\mathrm{y_i}-\beta^{\mathrm{T}}\mathbf{x_i}\right)^2}{2\sigma^2}} = \\
&= \frac{1}{(2\pi)^{\frac{\mathrm{p}}{2}}\sigma} \prod_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{e}^{-\sum_{\mathrm{i}=1}^{\mathrm{N}} \frac{\left(\mathrm{y_i}-\beta^{\mathrm{T}}\mathbf{x_i}\right)^2}{2\sigma^2}} = \\
&= \frac{1}{(2\pi)^{\frac{\mathrm{p}}{2}}\sigma|\mathbf{I}|} \mathrm{e}^{-\frac{1}{2}(\mathbf{y}-\beta^{\mathrm{T}}\mathbf{X})^{\mathrm{T}}(\sigma^2\mathbf{I})^{-1}(\mathbf{y}-\beta^{\mathrm{T}}\mathbf{X})} = \mathcal{N}(\beta^{\mathrm{T}}\mathbf{X}, \sigma^2\mathbf{I})
\end{aligned} \tag{64}$$

If the variables are not independent the more general form is:

$$\mathcal{N}(\beta^{\mathrm{T}}\mathbf{X}, \mathbf{\Sigma}) = \frac{1}{(2\pi)^{\frac{\mathrm{p}}{2}}|\Sigma|^{\frac{1}{2}}} \mathrm{e}^{-\frac{1}{2}(\mathbf{y}-\beta^{\mathrm{T}}\mathbf{X})^{\mathrm{T}}\Sigma^{-1}(\mathbf{y}-\beta^{\mathrm{T}}\mathbf{X})} \tag{65}$$

Definition of likelihood The quantity $\Pr(\mathrm{y}|\mathbf{x}, \beta, \sigma^2)$ is called **likelihood** and tell us how much it is likely the outcome $\mathrm{y_i}$ in the dataset given the input $\mathbf{x}$ and the parameters.

A different approach to find a model that fits the data is to maximize the *likelihood* of the whole dataset:

$$\mathrm{L} = \Pr(\mathbf{y}|\mathbf{X}, \beta, \sigma^2) = \frac{1}{(2\pi)^{\frac{\mathrm{p}}{2}}|\Sigma|^{\frac{1}{2}}} \mathrm{e}^{-\frac{1}{2}(\mathbf{y}-\beta^{\mathrm{T}}\mathbf{X})^{\mathrm{T}}\Sigma^{-1}(\mathbf{y}-\beta^{\mathrm{T}}\mathbf{X})} \tag{66}$$

Actually maximizing the likelihood is equivalent to maximizing its logarithmic, the *log-likeilhood*:

$$\begin{aligned}\log \mathrm{L} &= \sum_{\mathrm{i}=1}^{\mathrm{N}} \log\left[\frac{1}{(2\pi)^{\frac{1}{2}}\sigma} \mathrm{e}^{-\frac{\left(\mathbf{y}_{\mathrm{i}}-\beta^{\mathrm{t}}\mathbf{x}_{\mathrm{i}}\right)^2}{2\sigma^2}}\right] = \sum_{\mathrm{i}=1}^{\mathrm{N}} -\frac{1}{2}\log 2\pi - \log\sigma - \frac{\left(\mathrm{y}_{\mathrm{i}} - \beta^{\mathrm{T}}\mathbf{x}_{\mathrm{i}}\right)^2}{2\sigma^2} = \\ &= -\frac{1}{2}\log 2\pi - \log\sigma - \frac{1}{2\sigma^2}\sum_{\mathrm{i}=1}^{\mathrm{N}}\left(\mathrm{y}_{\mathrm{i}} - \beta^{\mathrm{T}}\mathbf{x}_{\mathrm{i}}\right)^2\end{aligned} \tag{67}$$

As already done for OLS, taking the derivative w.r.t. β and setting it to 0:

$$\begin{aligned}&\frac{\partial \log \mathrm{L}}{\partial \beta} = -\frac{1}{2\sigma^2}(-2)\left(\mathbf{y}_{\mathrm{i}} - \beta^{\mathrm{T}}\mathbf{x}_{\mathrm{i}}\right) = 0 \\ &\Rightarrow \left(\mathbf{y} - \beta^{\mathrm{T}}\mathbf{X}\right) = 0 \Rightarrow \beta = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}\end{aligned} \tag{68}$$

This is the same solution of the OLS: the two models are equivalent.
The two models are equivalent assuming a normal distribution.

3.3 Expectation of the parameter estimation: unbiased estimator

Computing the expectation of $\hat{\beta}$:

$$\begin{aligned}&\mathbf{E}_{\Pr(\mathbf{y}|\mathbf{X},\beta,\sigma^2)}[\hat{\mathrm{w}}] = \sum \hat{\mathrm{w}}\Pr(\mathbf{y}|\mathrm{X}, \beta, \sigma^2) = \\ &= \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\sum \mathbf{y}\Pr(\mathbf{y}|\mathrm{X}, \beta, \sigma^2) = \\ &= \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{E}_{\Pr(\mathbf{y}|\mathbf{X},\beta,\sigma^2)}[\mathbf{y}] = \\ &= \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{X}\beta = \beta\end{aligned} \tag{69}$$

This is an **unbiased estimator**.

Now let us calculate the covariance matrix:

$$\begin{aligned}
&\mathrm{Cov}\left[\hat{\beta}\right] = \\
&= \mathbf{E}_{\Pr(\mathbf{y}|\mathbf{X},\beta,\sigma^2)}\left[\left(\hat{\beta} - \mathbf{E}_{\Pr(\mathbf{y}|\mathbf{X},\beta,\sigma^2)}\left[\beta\right]\right)\left(\hat{\beta} - \mathbf{E}_{\Pr(\mathbf{y}|\mathbf{X},\beta,\sigma^2)}\left[\beta\right]\right)^{\mathrm{T}}\right] = \\
&= \mathbf{E}_{\Pr(\mathbf{y}|\mathbf{X},\beta,\sigma^2)}\left[\left(\hat{\beta} - \beta\right)\left(\hat{\beta} - \beta\right)^{\mathrm{T}}\right] = \\
&= \mathbf{E}_{\Pr(\mathbf{y}|\mathbf{X},\beta,\sigma^2)}\left[\hat{\beta}\hat{\beta}^{\mathrm{T}}\right] - \beta\beta^{\mathrm{T}}
\end{aligned} \tag{70}$$

and

$$\begin{aligned}
&\mathbf{E}_{\Pr(\mathbf{y}|\mathbf{X},\beta,\sigma^2)}\left[\hat{\beta}\hat{\beta}^{\mathrm{T}}\right] = \\
&= \mathbf{E}_{\Pr(\mathbf{y}|\mathbf{X},\beta,\sigma^2)}\left[\left(\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}\right)\left(\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}\right)^{\mathrm{T}}\right] = \\
&= \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{E}_{\Pr(\mathbf{y}|\mathbf{X},\beta,\sigma^2)}\left[\mathbf{y}\mathbf{y}^{\mathrm{T}}\right]\mathbf{X}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}
\end{aligned} \tag{71}$$

and recalling from 64 $\Pr(\mathbf{y}) \sim \mathcal{N}(\beta^{\mathrm{T}}\mathbf{X}, \sigma^2\mathbf{I})$

$$\begin{aligned}
&\mathrm{Cov}\left[\mathbf{y}\right] = \sigma^2\mathbf{I} = \\
&= \mathbf{E}_{\Pr(\mathbf{y}|\mathbf{X},\beta,\sigma^2)}\left[\mathbf{y}\mathbf{y}^{\mathrm{T}}\right] - \mathbf{E}_{\Pr(\mathbf{y}|\mathbf{X},\beta,\sigma^2)}\left[\mathbf{y}\right]\mathbf{E}_{\Pr(\mathbf{y}|\mathbf{X},\beta,\sigma^2)}\left[\mathbf{y}^{\mathrm{T}}\right]
\end{aligned} \tag{72}$$

Rearranging

$$\begin{aligned}
&\Rightarrow \mathbf{E}_{\Pr(\mathbf{y}|\mathbf{X},\beta,\sigma^2)}\left[\mathbf{y}\mathbf{y}^{\mathrm{T}}\right] = \sigma^2\mathbf{I} + \mathbf{E}_{\Pr(\mathbf{y}|\mathbf{X},\beta,\sigma^2)}\left[\mathbf{y}\right]\mathbf{E}_{\Pr(\mathbf{y}|\mathbf{X},\beta,\sigma^2)}\left[\mathbf{y}^{\mathrm{T}}\right] = \\
&= \sigma^2\mathbf{I} + \mathbf{E}_{\Pr(\mathbf{y}|\mathbf{X},\beta,\sigma^2)}\left[\mathbf{X}\beta\right]\mathbf{E}_{\Pr(\mathbf{y}|\mathbf{X},\beta,\sigma^2)}\left[\beta^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\right] = \\
&= \sigma^2\mathbf{I} + \mathbf{X}\beta\beta^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}
\end{aligned} \tag{73}$$

Substituting:

$$\begin{aligned}
&\mathbf{E}_{\Pr(\mathbf{y}|\mathbf{X},\beta,\sigma^2)}\left[\hat{\beta}\hat{\beta}^{\mathrm{T}}\right] = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{X}\left(\sigma^2\mathbf{I} + \mathbf{X}\beta\beta^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\right)\mathrm{X}^{\mathrm{T}}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}} = \\
&= \sigma^2\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}
\end{aligned} \tag{74}$$

and finally variance-covariance matrix of the least square parameters is

$$\mathrm{Cov}\left[\hat{\beta}\right] = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\sigma^2 \tag{75}$$

$$\mathrm{Var}(\hat{\beta}) = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\sigma^2 \tag{76}$$

3.4 Noise variance estimation

We can find an estimation of the noise variance from the maximum likelihood model using the same procedure used to find the parameters, i.e., taking the derivative and equating it to 0:

$$\begin{aligned}
&\frac{\partial \log \mathrm{L}}{\partial \sigma} = \sum_{\mathrm{i}=1}^{\mathrm{N}} -\frac{1}{\sigma} + \frac{1}{\sigma^3}\left(\mathrm{y_i} - \mathbf{x}_{\mathrm{i}}^{\mathrm{T}}\beta\right) = 0 \Rightarrow \frac{\mathrm{N}}{\sigma} + \frac{1}{\sigma^3}\sum_{\mathrm{i}=1}^{\mathrm{N}}\left(\mathrm{y_i} - \mathbf{x}^{\mathrm{T}}\beta\right) = 0 \\
&\Rightarrow \hat{\sigma^2} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}\left(\mathrm{y_i} - \mathbf{x}_{\mathrm{i}}^{\mathrm{T}}\beta\right)
\end{aligned} \tag{77}$$

This can be re-expressed as

$$\begin{aligned}
&\Rightarrow \hat{\sigma^2} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}\left(\mathrm{y_i} - \mathbf{x}^{\mathrm{T}}\hat{\beta}\right) = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}\left(\mathrm{y_i} - \mathbf{x}_{\mathrm{i}}^{\mathrm{T}}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}\right) = \\
&= \frac{1}{\mathrm{N}}\left(\mathbf{y} - \mathbf{X}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}\right)^{\mathrm{T}}\left(\mathbf{y} - \mathbf{X}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}\right) = \\
&= \mathbf{y}^{\mathrm{T}}\mathbf{y} - 2\mathbf{y}^{\mathrm{T}}\mathbf{X}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y} + \mathbf{y}^{\mathrm{T}}\mathbf{X}\cancel{\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{X}}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y} = \\
&= \mathbf{y}^{\mathrm{T}}\mathbf{y} - \mathbf{y}^{\mathrm{T}}\mathbf{X}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y} = \left(\mathbf{y}^{\mathrm{T}}\mathbf{y} - \mathbf{y}^{\mathrm{T}}\mathbf{X}\hat{\beta}\right)
\end{aligned} \tag{78}$$

Taking the expectation w.r.t. $\Pr(\mathbf{y}|\mathbf{X}, \beta, \sigma^2)$:

$$\begin{aligned}
&\mathbf{E}_{\Pr(\mathbf{y}|\mathbf{X},\beta,\sigma^2)}\left[\hat{\sigma}^2\right] = \frac{1}{\mathrm{N}}\mathbf{E}_{\Pr(\mathbf{y}|\mathbf{X},\beta,\sigma^2)}\left[\mathbf{y}^{\mathrm{T}}\mathbf{y} - \mathbf{y}^{\mathrm{T}}\mathbf{X}\hat{\beta}\right] = \\
&= \frac{1}{\mathrm{N}}\mathbf{E}_{\Pr(\mathbf{y}|\mathbf{X},\beta,\sigma^2)}\left[\mathbf{y}^{\mathrm{T}}\mathbf{y}\right] - \frac{1}{\mathrm{N}}\mathbf{E}_{\Pr(\mathbf{y}|\mathbf{X},\beta,\sigma^2)}\left[\mathbf{y}^{\mathrm{T}}\mathbf{X}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}\right]
\end{aligned} \tag{79}$$

Suppose $\mathbf{t} \sim \mathcal{N}(\mu, \mathbf{\Sigma})$, then $\mathbf{E}_{\mathrm{p}(\mathbf{t})}\left(\mathbf{t^T A t}\right) = \mathbf{Tr}(\mathbf{A\Sigma}) + \mu^{\mathbf{T}}\mathbf{A}\mu$ with $\mu = \mathbf{X}\beta$

$$
\begin{aligned}
&\mathbf{E}_{\Pr(\mathbf{y}|\mathbf{X},\beta,\sigma^2)}\left[\hat{\sigma}^2\right] = \frac{1}{\mathrm{N}}\mathbf{E}_{\Pr(\mathbf{y}|\mathbf{X},\beta,\sigma^2)}\left[\mathbf{y}^{\mathrm{T}}\mathbf{y}\right] + \\
&-\frac{1}{\mathrm{N}}\mathbf{E}_{\Pr(\mathbf{y}|\mathbf{X},\beta,\sigma^2)}\left[\mathbf{y}^{\mathrm{T}}\mathbf{X}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}\right] = \\
&= \frac{1}{\mathrm{N}}\mathbf{E}_{\Pr(\mathbf{y}|\mathbf{X},\beta,\sigma^2)}\left[\mathbf{y}^{\mathrm{T}}\mathbf{I}_{\mathrm{N}}\mathbf{y}\right] + \\
&-\frac{1}{\mathrm{N}}\mathbf{E}_{\Pr(\mathbf{y}|\mathbf{X},\beta,\sigma^2)}\left[\mathbf{y}^{\mathrm{T}}\mathbf{X}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}\right] = \\
&= \frac{1}{\mathrm{N}}\left(\mathbf{Tr}\left(\sigma^2\mathbf{I}_{\mathrm{N}}\right) + \beta^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\beta\right) + \\
&-\frac{1}{\mathrm{N}}\left(\mathbf{Tr}\left[\sigma^2\mathbf{X}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\right] + \beta^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{X}\beta\right) = \\
&= \frac{1}{\mathrm{N}}\left(\mathrm{N}\sigma^2 + \beta^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\beta\right) - \frac{1}{\mathrm{N}}\left(\sigma^2\mathbf{Tr}\left[\mathbf{X}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\right] + \beta^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\beta\right) = \\
&= \sigma^2 + \cancel{\frac{1}{\mathrm{N}}\beta^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\beta} - \frac{\sigma^2}{\mathrm{N}}\mathbf{Tr}\left[\mathbf{X}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\right] \cancel{-\frac{1}{\mathrm{N}}\beta^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathrm{X}\beta} = \\
&= \sigma^2 - \frac{\sigma^2}{\mathrm{N}}\mathbf{Tr}\left[\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{X}\right] = \sigma^2 - \frac{\sigma^2}{\mathrm{N}}\mathbf{Tr}\left[\mathbf{I}_{\mathrm{p}}\right] \\
&\Rightarrow \mathbf{E}_{\Pr(\mathbf{y}|\mathbf{X},\beta,\sigma^2)}\left[\hat{\sigma}^2\right] = \sigma^2\left(1 - \frac{\mathrm{p}}{\mathrm{N}}\right)
\end{aligned}
\tag{80}
$$

where lastly we have used the product property of the trace (see 1.1).

Normally $\mathrm{p} < \mathrm{N}$ hence the estimate of the variance is smaller than the true variance, so this estimator is **biased**. The estimate gets closer to the real value when p/N is small, i.e., assuming p is fixed, increasing the samples used.

This result might be strange. First of all notice from 77 that the closer the model gets to the data, the smaller $\hat{\sigma}^2$. By definition the parameter estimates are the ones that minimise the noise and hence hs, hence when using the true parameters we would get equal or higher variance.

To estimate the true variance we can use the following formula:

$$\hat{\sigma}^2 = \frac{1}{\mathrm{N} - \mathrm{p} - 1} \sum_{\mathrm{i}=1}^{\mathrm{N}} (\mathrm{y_i} - \hat{\mathrm{y}}_\mathrm{i})^2 \tag{81}$$

$\mathrm{N} - \mathrm{p} - 1$ makes this estimation unbiased i.e., $\mathbf{E}[\hat{\sigma}^2] = \sigma^2$.

3.5 Interpretation of covariance

Consider a covariance matrix of size 2×2 and suppose the first diagonal element, corresponding to the variable $\hat{\beta}_0$ is much bigger than the second one corresponding to $\hat{\beta}_1$. This means that we can change $\hat{\beta}_0$ a little without affecting too much the model. On the contrary if the variance is small, small changes will affect significantly the model. Sometimes this happens when one variable has a much higher absolute value.

If the values on the off-diagonals are negative, then when increasing one coefficient i.e., $\hat{\beta}_0$, the other must be decreased to have the line to pass as close as possible to all points. For example in 2D, increasing $\hat{\beta}_0$ reduces the coefficient value: if it is positive, the line will be "more horizontal", if negative it becomes steeper, "more vertical".

3.6 Z-score

Let us assume data were really generated by a linear model but were corrupted by Gaussian noise with 0 mean and variance σ^2:

$$\mathrm{Y} = \beta_0 + \sum_{\mathrm{i}}^{\mathrm{p}} \beta_\mathrm{i} \mathrm{X_i} + \epsilon \tag{82}$$

where $\epsilon \sim \mathrm{N}(0, \sigma^2)$. The estimated parameters will still be a normal distribution:

$$\hat{\beta} \sim \mathcal{N}(\beta, (\mathbf{X}^\mathrm{T}\mathbf{X})^{-1}\sigma^2) \tag{83}$$

Definition 3.1. Z-score A Z-score is a numerical measurement used in statistics of a value's relationship to the mean (average) of a group of values,

measured in terms of standard deviations from the mean. If a Z-score is 0, it indicates that the data point's score is identical to the mean score. A Z-score of 1.0 would indicate a value that is one standard deviation from the mean. Z-scores may be positive or negative, with a positive value indicating the score is above the mean and a negative score indicating it is below the mean. Z-scores are measures of an observation's variability

$$\mathrm{z_j} = \frac{\mathrm{x} - \mu}{\sigma} \tag{84}$$

In machine learning the z-value is the regression coefficient divided by its standard error. It is also sometimes called the z-statistic. If the z-value is too big in magnitude (i.e., either too positive or too negative), it indicates that the corresponding true regression coefficient is not 0 and the corresponding X-variable matters. A good rule of thumb is to use a cut-off value of 2 which approximately corresponds to a two-sided hypothesis test with a significance level of $\alpha = 0.05$.

Z-values are computed as the test statistic for the hypothesis test that the true corresponding regression coefficient β is 0. In hypothesis testing, we assume the null hypothesis is true, and then see if the data provide evidence against it. So in this case, we assume β is 0. That is, we assume the expectation of the fitted regression coefficient $\hat{\beta}$ is 0:

$$\mathrm{z_j} = \frac{\hat{\beta}_\mathrm{j}}{\hat{\sigma}\sqrt{\mathrm{v_j}}} \tag{85}$$

where the denominator is the variance of the parameter (from 83) with $\mathrm{v_j}$ being the diagonal element of $(\mathrm{X^T X})^{-1}$

Theorem 1 (**The Gauss-Markov theorem**)**.** Among all linear unbiased estimators, the least square estimates of the parameters are the ones having smallest variance and consequently from 54 is the one with the smallest mean squared error (the bias-squared term for unbiased estimator is by definition 0).

Proof. Let $\hat{\beta} = \mathrm{Cy}$ be another linear estimator of β with $\mathrm{C} = (\mathrm{X}^{\mathrm{T}}\mathrm{X})^{-1}\mathrm{X}^{\mathrm{T}} + \mathrm{D}$

$$\mathbf{E}\left[\hat{\beta}\right] = \mathbf{E}\left[\mathrm{Cy}\right] = \mathbf{E}\left[\left(\left(\mathrm{X}^{\mathrm{T}}\mathrm{X}\right)^{-1}\mathrm{X}^{\mathrm{T}} + \mathrm{D}\right)(\mathrm{X}\beta + \epsilon)\right] = \tag{86}$$

$$= \left(\left(\mathrm{X}^{\mathrm{T}}\mathrm{X}\right)^{-1}\mathrm{X}^{\mathrm{T}}\mathrm{X}\beta + \mathrm{DX}\beta\right) + \cancel{\left(\left(\mathrm{X}^{\mathrm{T}}\mathrm{X}\right)^{-1}\mathrm{X}^{\mathrm{T}} + \mathrm{D}\right)\mathbf{E}[\epsilon]} = \tag{87}$$

$$= (\beta + \mathrm{DX}\beta) = (\mathrm{I} + \mathrm{DX})\,\beta \tag{88}$$

$$\tag{89}$$

where $\mathbf{E}[\epsilon] = 0$.

To be an unbiased estimator $\mathrm{DX} = 0$, then

$$\begin{aligned}
\mathrm{Var}(\tilde{\beta}) &= \mathrm{Var}(\mathrm{Cy}) = \mathrm{C}\mathrm{Var}(\mathrm{y})\mathrm{C}^{\mathrm{T}} = \sigma^2\mathrm{C}\mathrm{C}^{\mathrm{T}} = \\
&= \sigma^2\left(\left(\mathrm{X}^{\mathrm{T}}\mathrm{X}\right)^{-1}\mathrm{X}^{\mathrm{T}} + \mathrm{D}\right)\left(\left(\mathrm{X}^{\mathrm{T}}\mathrm{X}\right)^{-1}\mathrm{X}^{\mathrm{T}} + \mathrm{D}\right)^{\mathrm{T}} = \\
&= \sigma^2\left(\left(\mathrm{X}^{\mathrm{T}}\mathrm{X}\right)^{-1}\mathrm{X}^{\mathrm{T}} + \mathrm{D}\right)\left(\mathrm{X}\left(\mathrm{X}^{\mathrm{T}}\mathrm{X}\right)^{-1} + \mathrm{D}^{\mathrm{T}}\right) = \\
&= \sigma^2(\cancel{\left(\mathrm{X}^{\mathrm{T}}\mathrm{X}\right)^{-1}\mathrm{X}^{\mathrm{T}}\mathrm{X}}\left(\mathrm{X}^{\mathrm{T}}\mathrm{X}\right)^{-1} + \left(\mathrm{X}^{\mathrm{T}}\mathrm{X}\right)^{-1}\mathrm{X}^{\mathrm{T}}\mathrm{D}^{\mathrm{T}} + \\
&\quad + \mathrm{DX}\left(\mathrm{X}^{\mathrm{T}}\mathrm{X}\right)^{-1} + \mathrm{D}\mathrm{D}^{\mathrm{T}}) = \\
&= \sigma^2(\cancel{\left(\mathrm{X}^{\mathrm{T}}\mathrm{X}\right)^{-1}\mathrm{X}^{\mathrm{T}}\mathrm{X}}\left(\mathrm{X}^{\mathrm{T}}\mathrm{X}\right)^{-1} + \left(\mathrm{X}^{\mathrm{T}}\mathrm{X}\right)^{-1}(\mathrm{DX})^{\mathrm{T}} + \\
&\quad + \mathrm{DX}\left(\mathrm{X}^{\mathrm{T}}\mathrm{X}\right)^{-1} + \mathrm{D}\mathrm{D}^{\mathrm{T}})
\end{aligned}$$

$$\begin{aligned}
&\mathrm{DX} = 0 \\
\Rightarrow\ &\mathrm{Var}(\tilde{\beta}) = \sigma^2\left(\left(\mathrm{X}^{\mathrm{T}}\mathrm{X}\right)^{-1} + \cancel{\left(\mathrm{X}^{\mathrm{T}}\mathrm{X}\right)^{-1}(\mathrm{DX})^{\mathrm{T}}} + \cancel{\mathrm{DX}\left(\mathrm{X}^{\mathrm{T}}\mathrm{X}\right)^{-1}} + \mathrm{D}\mathrm{D}^{\mathrm{T}}\right) \\
&\mathrm{Var}(\tilde{\beta}) = \mathrm{Var}(\hat{\beta}) + \sigma^2\mathrm{D}\mathrm{D}^{\mathrm{T}}
\end{aligned} \tag{90}$$

□

3.7 Orthogonalization

Normally inputs are not perpendicular but can be orthogonalized. The goal is to define a new orthogonal basis for the data. The procedure, named

Grand-Schmidt procedure is the following: When inputs are correlated,

initialize $z_0 = x_0 = \mathbf{1}$ where $\mathbf{1}$ is a vector of all ones;
For $j = 1, \cdots, p$ regress x_j on $z_0, \cdots, z_{p-1}$ to produce the coefficients
$\hat{\gamma}_{lj} = \frac{\langle z_l, x_j \rangle}{\langle z_l, z_l \rangle}$ for $l = 0, \cdots, j-1$;
Calculate the residual vectors as $z_j = x_j - \sum_{k=0}^{j-1} \hat{\gamma}_{kj} z_k$;
Regress $\mathbf{y}$ on the residual z_p to get the estimate $\hat{\beta}_p = \frac{\langle z_p, y \rangle}{\langle z_p, z_p \rangle}$

the residual will be close to zero generating instabilities in the coefficients $\hat{\beta}_j$ and the z-score will be small.

The algorithm in matrix form is

$$\mathbf{X} = \mathbf{Z\Gamma}$$

where z_j are the column vectors of Z and Γ is upper triangular. Introducing the diagonal matrix D with j-th diagonal element $D_{jj} = ||z_j||$ we get:

$$\mathbf{X} = \mathbf{ZD^{-1}D\Gamma} = \mathbf{QR} \tag{91}$$

where Q is an orthogonal, $Q^T Q = I$, $N \times (p+1)$ matrix and R is a $(p+1)(p+1)$ upper triangular matrix.

The least square solution becomes

$$\hat{\beta} = R^{-1} Q^T y \hat{y} = Q Q^T y \tag{92}$$

3.8 Multivariate output

We can rewrite the equation in matrix form:

$$Y = XB + E \tag{93}$$

where Y is a $N \times K$ matrix, X is $N \times (p+1)$, B is $(p+1) \times K$, E has the same dimensions of Y. The root squared error is

$$\begin{aligned} RSS(B) &= \sum_{k=1}^{K} \sum_{i=1}^{N} (y_{ik} - f(x_i))^2 = \mathrm{tr}\left[(Y - XB)^T (Y - XB) \right] \\ \Rightarrow B &= \left(X^T X \right)^{-1} X^T Y \end{aligned} \tag{94}$$

Multiple outputs do not affect one another's least square estimates.

3.9 Subset selection

Least square estimates sometimes show low bias but high variance resulting in a non-satisfactory prediction accuracy. This can be improved by setting some coefficients to 0 to sacrifice a little of bias to reduce significantly the variance.

Other times it is useful to reduce the size of inputs for easiness of interpretation or for computation purposes.

3.9.1 Best subset selection

This algorithm finds for each $k \in 0, 1, 2, \cdots p$, the subset of size k that gives the smallest residual sum of squares. Note that if a variable is in the best subset of size m, it might not be in the subsets of larger size (and of course neither in the smallest ones).

3.9.2 Forward stepwise selection

Searching for all the subsets is too computationally intensive (and infeasible for $p > 40$). The *forward stepwise* algorithm starts with the intercept and then sequentially adds to the model the predictor that most improves the fit. QR decomposition can be exploited to choose the next candidate.

This algorithm is a greedy sub-optimal algorithm. Statistically it will have lower variance but higher bias.

3.9.3 Backward stepwise selection

Backward stepwise selection starts with the full model and sequentially removes the predictor having least impact on the model, i.e., having the smallest $Z-$score.

Note: this algorithm can only be applied when $N > p$ while *forward stepwise* can always be used.

3.9.4 Implementations

[From ESLII pg. 60] Some software packages implement hybrid stepwise-selection strategies that consider both forward and backward moves at each step, and select the "best" of the two. For example in the R package the step function uses the AIC criterion for weighing the choices, which takes proper

account of the number of parameters fit; at each step an add or drop will be performed that minimizes the AIC score.

Other more traditional packages base the selection on F-statistics, adding "significant" terms, and dropping "non-significant" terms. These are out of fashion.

3.9.5 Forward stagewise regression

As forward stepwise, it starts with the intercept. At each step the algorithm identifies the variable most correlated with the residual and computes the linear regression coefficient of the residual on this chosen variable and then adds it to the current coefficient for that variable. This is continued till none of the variables have correlation with the residual.

At each step only one coefficient is updated by a small step so that the number of steps is bigger than p. This slow-fitting pays in high dimensions.

Note: forward stagewise is very competitive in

3.10 Shrinkage methods

Subset selection methods either keep or remove a predictor. It has higher variance. Shrinkage or regularization methods are more continuous. They force the model to keep the weights as small as possible.

3.10.1 Ridge regression

Ridge regression shrinks the coefficients by imposing a penalty on their size. The ridge coefficients minimize a penalized residual sum of squared errors

$$\hat{\beta}^{\text{ridge}} = \arg\min_{\beta} \left\{ \sum_{\text{i}=1}^{\text{N}} \left(\text{y}_\text{i} - \beta_0 - \sum_{\text{j}=1}^{\text{p}} \text{x}_{\text{ij}} \beta_\text{j} \right)^2 + \lambda \sum_{\text{j}=1}^{\text{p}} \beta_\text{j}^2 \right\} \tag{95}$$

where λ is a parameter controlling the amount of shrinkage: the bigger the value the greater the amount of shrinkage. This concept is also used in

the Neural Networks. Another way to express 95 is the following:

$$\begin{aligned}\hat{\beta}^{\text{ridge}} &= \arg\min_{\beta} \left\{ \sum_{\text{i}=1}^{\text{N}} \left(\text{y}_\text{i} - \beta_0 - \sum_{\text{j}=1}^{\text{p}} \text{x}_{\text{ij}}\beta_\text{j} \right)^2 \right\} \\ \text{subject to} &\sum_{\text{j}=1}^{\text{p}} \beta_\text{j}^2 \leq \text{t}\end{aligned} \tag{96}$$

where there is a one-to-one correspondence between λ and t. Note that the penalization term does not consider β_0 otherwise the procedure will depend on the chosen origin for Y.

This algorithm solves the problem of high variance in case of correlated inputs when big coefficients of correlated variables can be cancelled out. With a constraints on the coefficients this problem is alleviated.

The coefficients are not preserved when the input is scaled. Generally inputs are standardized before applying the algorithm.

$$\text{RSS}(\lambda, \beta) = (\text{y} - \text{X}\beta)^\text{T} (\text{y} - \text{X}\beta) + \lambda\beta^\text{T}\beta \tag{97}$$

$$\hat{\beta}^{\text{ridge}} = \left(\text{X}^\text{T}\text{X} + \lambda\text{I}\right)^{-1} \text{X}^\text{T}\text{y} \tag{98}$$

Now even if X is not full rank, the problem is non singular (the inverse exists). In case of orthonormal inputs, the ridge coefficients are the same of least square but scaled: $\hat{\beta}^{\text{ridge}} = \frac{\hat{\beta}}{1+\lambda}$.

The parameter λ can also be derived assuming a prior distribution $\text{y}_\text{i} \sim \text{N}\left(\beta_0 + \text{x}_\text{i}^\text{T}\beta, \sigma^2\right)$ and the parameters β_j are distributed as $\text{N}\left(0, \tau^2\right)$. Then from 95 $\lambda = \frac{\sigma^2}{\tau^2}$.

Applying the SVD decomposition of the matrix $\mathbf{X} = \mathbf{UDV^T}$ where X is N × p, U is N × pand V is p × p, the latter two both orthogonal with the columns of U spanning the column space of X and the columns of V spanning the row space of X. D is a p × p diagonal matrix with the elements $\text{d}_1 \geq \text{d}_2 \geq \cdots \geq \text{d}_\text{p} \geq 0$ called singular value decomposition of X. If any $\text{d}_\text{j} = 0$ then X is singular.

The least squares equation can be rewritten as

$$\mathbf{X}\hat{\beta}^{\text{ls}} = \mathbf{X}\left(\mathbf{X^T X}\right)^{-1}\mathbf{X^T y} = \mathbf{UU^T}\text{y} \tag{99}$$

In case of the ridge regression, the coefficients are

$$\begin{aligned}\mathbf{X}\hat{\beta}^{\text{ridge}} &= \mathbf{X}\left(\mathbf{X^T X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X^T y} = \mathbf{U U^T y} = \\ &= \sum_{\text{j–1}}^{\text{p}} \mathbf{u}_{\text{j}} \frac{\text{d}_{\text{j}}^2}{\text{d}_{\text{j}}^2 + \lambda} \mathbf{u_j}^{\text{T}} \text{y}\end{aligned} \tag{100}$$

where u_{j} are the column vectors. So ridge regression first computes the coordinates of y with respect to the orthonormal basis $\mathbf{U}$, it then shrinks those coordinates since $\lambda \geq 0$. A greater amount of shrinkage is applied to the coordinates of basis vector with smaller d_{j}, corresponding to elements with small variance.

3.10.2 Lasso regression

The lasso regression is similar to ridge regression but it uses a L_1 penalization instead of L_2.

$$\hat{\beta}^{\text{lasso}} = \arg\min_{\beta} \left\{ \sum_{\text{i=1}}^{\text{N}} \left(\text{y}_{\text{i}} - \beta_0 - \sum_{\text{j=1}}^{\text{p}} \text{x}_{\text{ij}}\beta_{\text{j}} \right)^2 + \lambda \sum_{\text{j=1}}^{\text{p}} |\beta_{\text{j}}| \right\} \tag{101}$$

or equivalently

$$\begin{aligned}\hat{\beta}^{\text{lasso}} &= \arg\min_{\beta} \left\{ \sum_{\text{i=1}}^{\text{N}} \left(\text{y}_{\text{i}} - \beta_0 - \sum_{\text{j=1}}^{\text{p}} \text{x}_{\text{ij}}\beta_{\text{j}} \right)^2 \right\} \\ &\text{subject to} \sum_{\text{j=1}}^{\text{p}} |\beta_{\text{j}}| \leq \text{t}\end{aligned} \tag{102}$$

If $\text{t} > \text{t}_0 = \sum_1^{\text{p}} |\beta_{\text{j}}|$ where β_{j} are the least square coefficients, then the lasso coefficients are the same of the least squares ones. If $\text{t} = \text{t}_0/2$ then the least square coefficients are shrunk by 50% on average. Making t sufficiently will cause some of the coefficients to be exactly 0.

In 3.10.2, the blue areas show the constraints for the estimates of ridge and lasso regressions, while the ellipses are contours of the residual sum of squares, centered at β^{ls}, in case of only two coefficients. For the lasso one of the coefficients is 0 when it hits one of the corner. In higher dimensions

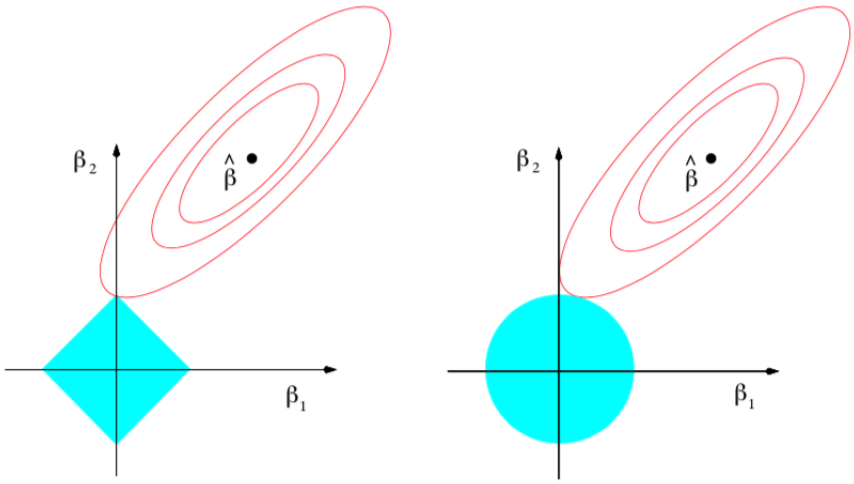


Figure 1: Constraints of ridge and lasso regression.

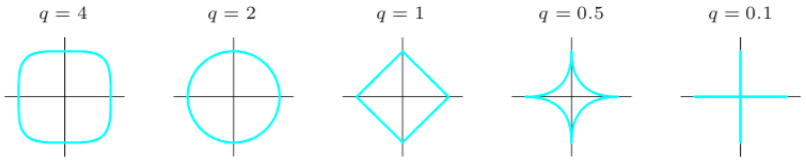


Figure 2: Shapes of different penalization factors.

the figure becomes a rhomboid with many corners (and faces) so it becomes easy to hit a corner.

Also other penalization factors can be chosen, 3.10.2

3.10.3 Least angle regression

It is similar to forward stepwise. At first it identifies the variable most correlated with the response but instead of fitting it completely, it moves the variable toward its least squares value. As soon as another value gets correlated to the residual as the previous variable, the process is paused. The second variable joins the active set and their coefficients are moved together in a way that keeps their correlations tied and decreasing. The process is continued until all the variables are in the model. After $\min(N-1, p)$ steps

standardize the predictors to have 0 mean and unit norm;
$\mathcal{A}_k \leftarrow 0$;
$r \leftarrow y - \bar{y}$, $\beta_i = 0$ for $i \neq 0$;
while $|\mathcal{A}_k| < p$ **do**
  find the predictor most correlated to r;
  insert the predictor in the active set $\mathcal{A}_k$;
  move the coefficients of the predictors in the active set to the direction defined by their joint least squares coefficient of the current residual, i.e.,

$$\delta_k = \left(X_{\mathcal{A}_k}^T X_{\mathcal{A}_k}\right)^{-1} X_{\mathcal{A}_k}^T r_k \tag{103}$$

  (where $\mathcal{A}_k$ is the current active set of variables) until some other competitor x_l has as much correlation with the current residual;
  $r_k \leftarrow y - X_{\mathcal{A}_k}\beta_{\mathcal{A}_k}$
end

we we arrive at the full least squares solution. The coefficient profile evolves as

$$\beta_{\mathcal{A}_k} = \beta_{\mathcal{A}_k} + \alpha\delta_k \tag{104}$$

If to the LAR algorithm we add the following rule i.e.,

If a non-zero coefficient hits 0*, drop its variable from the active set of variables and recompute the current joint least squares direction.*

we get the same coefficient path of the lasso and this is called LAR(lasso). So this become an efficient solution to compute the Lasso problem, especially with $N >> p$ since Lasso can take more than p steps while LAR require p steps and it is efficient since it requires the same complexity as that of a single least squares fit using the p predictors.

3.11 Derived input directions methods

When a large number of inputs is present, often there is a high correlation among them. In this case is convenient to regress on a new set inputs obtained form a linear combination of the original input.

3.11.1 Principal component analysis

First input must be standardized since this analysis depends on the scaling. The principal components are defined as

$$z_i = Xv_i \tag{105}$$

where v_i are the column vectors of V from the SVD decomposition of $X = UDV^T$ (recall that z_m are orthogonal). The algorithm then regress X on $z_1, \cdots, z_M$ for $M \leq p$ and we have:

$$\hat{y}^{pcr}_{(M)} = \bar{y}\mathbf{1} + \sum_{m=1}^{M} \hat{\theta}_m z_m \tag{106}$$

where $\hat{\theta}_m = \langle z_m, y\rangle / \langle z_m, z_m\rangle$. Since z_m are linear combination of the original predictors x_j, we can express the solution as

$$\hat{\beta}^{pcr}(M) = \sum_{m=1}^{M} \hat{\theta}_m v_m \tag{107}$$

With $M = p$ we get the usual least squares. Principal component analysis discards the $p - M$ smallest eigenvalue components.

The value M is suggested by cross-validation.

3.11.2 Partial least squares

This technique use a set of linear combinations of y in addition to X for the construction. It is not scale invariant so each x_j must be standardized.

Since PLS use y to construct its directions, its solution path is not linear in y. It seeks direction with high variance and high correlation with the response while PCR only with high variance.

3.12 Multioutput shrinkage and selection

To apply selection and shrinkage methods in the multiple output case, one could apply a univariate technique individually to each outcome or si- multaneously to all outcomes, i.e., different λ in Ridge or Lasso can be used for each output or the same value can be adopted.

standardize x_j to 0 mean and 1 variance;
$\hat{y}^{(0)} \leftarrow \bar{y}\mathbf{1}$;
$x_j^{(0)} \leftarrow x_j$;
for $m = 1, \cdots, p$ **do**
  $z_m = \sum_{j=1}^{p} \hat{\phi}_{mj} x_j^{(m-1)}$ where $\hat{\phi}_{mj} = \langle x_j^{(m-1)}, y \rangle$;
  $\hat{\theta}_m = \frac{\langle z_m, y \rangle}{\langle z_m, z_m \rangle}$;
  $\hat{y}^m = \hat{y}^{(m-1)} + \hat{\theta}_m z_m$;
  orthogonalize each x_j^{m-1} w.r.t.
    $z_m : x_j^{(m)} = x_j^{(m-1)} - \frac{\langle z_m, x_j^{(m-1)} \rangle}{\langle z_m, z_m \rangle} z_m$, $j = 1, \cdots, p$;
end
Output the sequence of fitted vectors $\{\hat{y}^m\}_1^p$. Since z_{l1}^m are linear in x_j, so is $\hat{y}^{(m)} = X\hat{\beta}^{pls}(m)$. These coefficients can be recovered from the sequence of PLS transformations.

3.13 Other derived algorithms

3.13.1 Incremental forward stagewise

$r \leftarrow y$;
$\beta_i = 0$ for $i \neq 0$;
find the (standardized) predictor x_j most correlated with the residual ;
$\beta_j \leftarrow \beta_j + \delta_j$ where $\delta_j = \epsilon \text{sign}\left[\langle x_j, r \rangle\right]$ and $\epsilon > 0$ small;
$r \leftarrow r - \delta_j x_j$;
Repeat the steps many times until the residuals are uncorrelated with the predictors.

3.13.2 The Dantzig selector

...

3.13.3 The Grouped Lasso

...

4 Linear Classification

For classification problem, in this section we assume the classification boundaries are linear, i.e., in the input hyperspace the points belonging to different classes can be separated by hyperplanes.

4.1 Linear regression of an Indicator matrix

Suppose we have K classes. For a single output we build a vector $\mathbf{Y} = (\mathrm{Y}_1, \cdots, \mathrm{Y}_\mathrm{k})$ where $\mathrm{Y}_\mathrm{k} = 1$ if the class it belongs is the k class. As output we will have the matrix $\mathbf{Y}$ of 0 and 1 with each row having a single 1. We fit a linear regression model to each of the columns of $\mathbf{Y}$ simultaneously:

$$\hat{\mathbf{Y}} = \mathbf{X}\left(\mathbf{X}^\mathrm{T}\mathbf{X}\right)^{-1}\mathbf{X}^\mathrm{T}\mathbf{Y}$$

We get a coefficient vector for each response column y_k, and hence a $(\mathrm{p}+1)\times\mathrm{K}$ matrix $\hat{\mathbf{B}} = \left(\mathbf{X}^\mathrm{T}\mathbf{X}\right)^{-1}\mathbf{X}^\mathrm{T}\mathbf{Y}$,where $\mathbf{X}$ will have $\mathrm{p}+1$ columns with a leading column of 1 for the intercept.

Suppose we are given a new input x. Then the classification problem becomes:

1. compute the output $\hat{\mathrm{f}}(\mathrm{x})^\mathrm{T} = (1, \mathrm{x}^\mathrm{T})\hat{\mathbf{B}}$ which is a k vector identify the largest component $\hat{\mathrm{G}}(\mathrm{x}) = \underset{\mathrm{k}\in\mathcal{G}}{\arg\max}\, \hat{\mathrm{f}}_\mathrm{k}(\mathrm{x})$

With this approach we are basically estimating a conditional expectation, i.e., given the inputs x what is the probability the output is of class k? Mathematically $\mathbf{E}\left(\mathrm{Y}_\mathrm{k}|\mathrm{X}=\mathrm{x}\right) = \Pr\left(\mathrm{G}=\mathrm{k}|\mathrm{X}=\mathrm{x}\right)$

Although the linear model guarantees $\sum_{\mathrm{k}\in\mathcal{G}}\hat{\mathrm{f}}_\mathrm{k} = 1$, as long as there is an intercept in the model, $\hat{\mathrm{f}}_\mathrm{k}(\mathrm{x})$ can be negative or bigger than one, especially when making predictions outside the hull of training data. Although this fact, this approach still work in many cases.

An important limitation is when $\mathrm{K} \geq 3$. Even if the classes can still be separated by more than one linear boundaries, linear regression cannot find linear boundaries (because they are more than one?).

If quadratic regression might solve the problem, but a general rule is that if $k \geq 3$ classes are lined up (their centroids are in the same line), a polynomial term with degree up to $k-1$ is needed to solve the problem, and since the direction is arbitrary, cross-products terms might be needed too.

4.2 Linear Discriminant analysis

For optimal classification we have to know the class posteriors $\Pr(G|X)$. Let π_k be the prior probability of class k with $\sum_k^K \pi_k = 1$. Suppose $f_k(x)$ is the class conditional density. From the Bayes theorem (?) we get

$$\Pr(G = k|X = x) = \frac{\Pr(X = x|G = k)\Pr(G = k)}{\Pr(X = x)} \tag{108}$$

$\Pr(G = k)$ is the prior probability π_k, $\Pr(X = x|G = k)$ is the class conditional density while the denominator can be rewritten using the **Total Probability Theorem** as

$$\Pr(X = x) = \sum_{l=1}^{K} \Pr(X = x|G = l)\Pr(G = l) = \sum_{l=1}^{K} \Pr(X = x|G = l)\pi_l \tag{109}$$

$$\Rightarrow \Pr(G = k|X = x) = \frac{f_k(x)\pi_k}{\sum_{l=1}^{K} f_l(x)\pi_l} \tag{110}$$

The goodness of classification mostly rely on $f_k(x)$ and many techniques use models for class densities:

- linear and quadratic discriminant analysis use Gaussian densities;
- mixtures of Gaussians allow for non-linear boundaries;
- Naive Bayes models assume that each class density is a product of marginal densities i.e., inputs are conditionally independent in each class.

Modelling each class density as a multivariate Gaussian we have

$$f_k(x) = \frac{1}{(2\pi)^{\frac{p}{2}}|\sum_k|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)} \tag{111}$$

Linear Discriminant analysis assumes equal covariance matrices for all classes. Taking as comparison between two classes the log-ratio, we have

$$\begin{aligned}
\log\frac{\Pr(\mathrm{G}=\mathrm{k}|\mathrm{X}=\mathrm{x})}{\Pr(\mathrm{G}=\mathrm{l}|\mathrm{X}=\mathrm{x})} &= \log\frac{\mathrm{f_k(x)}\pi_\mathrm{k}}{\mathrm{f_l(x)}\pi_\mathrm{l}} = \log\frac{\pi_\mathrm{k}}{\pi_\mathrm{l}} + \log\frac{\mathrm{f_k(x)}}{\mathrm{f_l(x)}} = \\
&= \log\frac{\pi_\mathrm{k}}{\pi_\mathrm{l}} + \log\frac{\mathrm{e}^{-\frac{1}{2}(\mathrm{x}-\mu_\mathrm{k})^\mathrm{T}\Sigma^{-1}(\mathrm{x}-\mu_\mathrm{k})}}{\mathrm{e}^{-\frac{1}{2}(\mathrm{x}-\mu_\mathrm{l})^\mathrm{T}\Sigma^{-1}(\mathrm{x}-\mu_\mathrm{l})}} = \\
&= \log\frac{\pi_\mathrm{k}}{\pi_\mathrm{l}} - \frac{1}{2}(\mathrm{x}-\mu_\mathrm{k})^\mathrm{T}\Sigma^{-1}(\mathrm{x}-\mu_\mathrm{k}) - \left(-\frac{1}{2}\right)(\mathrm{x}-\mu_\mathrm{l})^\mathrm{T}\Sigma^{-1}(\mathrm{x}-\mu_\mathrm{l}) = \\
&= \log\frac{\pi_\mathrm{k}}{\pi_\mathrm{l}} - \frac{1}{2}\left[(\mathrm{x}-\mu_\mathrm{k})^\mathrm{T}\Sigma^{-1}(\mathrm{x}-\mu_\mathrm{k}) + (\mathrm{x}-\mu_\mathrm{l})^\mathrm{T}\Sigma^{-1}(\mu_\mathrm{l}-\mathrm{x})\right] = \\
&= \log\frac{\pi_\mathrm{k}}{\pi_\mathrm{l}} - \frac{1}{2}\left[(2\mathrm{x}-\mu_\mathrm{k}-\mu_\mathrm{l})^\mathrm{T}\Sigma^{-1}(\mu_\mathrm{l}-\mu_\mathrm{k})\right] = \\
&= \log\frac{\pi_\mathrm{k}}{\pi_\mathrm{l}} + \frac{1}{2}\mathrm{x}^\mathrm{T}\Sigma^{-1}(\mu_\mathrm{k}-\mu_\mathrm{l}) - \frac{1}{2}(\mu_\mathrm{k}+\mu_\mathrm{l})^\mathrm{T}\Sigma^{-1}(\mu_\mathrm{k}-\mu_\mathrm{l}) =
\end{aligned} \tag{112}$$

which is linear in x, so all decision boundaries are linear (i.e., they are hyperplanes in p dimensions). If the common covariance matrix is spherical, i.e., $\Sigma = \sigma^2\mathrm{I}$ and the class priors are equal, each boundary that separates two classes is the perpendicular bisector of the segment joining the centroids of the two classes.

The linear discriminant functions of each class are

$$\delta_\mathrm{k}(\mathrm{x}) = \mathrm{x}^\mathrm{T}\Sigma^{-1}\mu_\mathrm{k} - \frac{1}{2}\mu_\mathrm{k}^\mathrm{T} + \log\pi_\mathrm{k} \tag{113}$$

We do not know the parameters of the Gaussian distribution and we must estimate them from the training data:

$$\hat{\pi}_\mathrm{k} = \frac{\mathrm{N_k}}{\mathrm{N}} \tag{114}$$

$$\hat{\mu}_\mathrm{k} = \sum_{\mathrm{g_i}=\mathrm{k}} \frac{\mathrm{x_i}}{\mathrm{N_k}} \tag{115}$$

$$\hat{\Sigma} = \sum_{\mathrm{k}=1}^{\mathrm{K}} \frac{(\mathrm{x_i}-\hat{\mu}_\mathrm{k})(\mathrm{x_i}-\hat{\mu}_\mathrm{k})^\mathrm{T}}{(\mathrm{N}-\mathrm{K})} \tag{116}$$

Note that LDA does not use Gaussian assumption for the features.

4.2.1 Decision rule

Consider two classes 1 and 2. LDA classifies to class 1 if

$$\mathrm{x}^{\mathrm{T}}\hat{\mathbf{\Sigma}}^{-1}\hat{\mu}_1 - \frac{1}{2}\hat{\mu}_1^{\mathrm{T}}\hat{\mathbf{\Sigma}}^{-1}\hat{\mu}_1 + \log\hat{\pi}_1 > \mathrm{x}^{\mathrm{T}}\hat{\mathbf{\Sigma}}^{-1}\hat{\mu}_2 - \frac{1}{2}\hat{\mu}_2^{\mathrm{T}}\hat{\mathbf{\Sigma}}^{-1}\hat{\mu}_2 + \log\hat{\pi}_2 \quad (117)$$

to class 2 if $<$ holds. In such case there is a correspondence between LDA and linear regression classification if the two classes are coded with +1 and –1. In this case the coefficient vector from least squares is proportional to the LDA direction. However unless $\mathrm{N}_1 = \mathrm{N}_2$, the intercepts are different and so are the decision rules.

With more than 2 classes, linear regression is not able to classify correctly while LDA does.

4.3 Quadratic Discriminant analysis

If we do not assume equal covariance, the squared term in x does not cancel out and we get quadratic discriminant functions:

$$\delta_{\mathrm{k}}(\mathrm{x}) = -\frac{1}{2}\log|\Sigma_{\mathrm{k}}| - \frac{1}{2}\left(\mathrm{x}-\mu_{\mathrm{k}}\right)^{\mathrm{T}}\mathbf{\Sigma}_{\mathrm{k}}^{-1}\left(\mathrm{x}-\mu_{\mathrm{k}}\right) + \log\pi_{\mathrm{k}} \quad (118)$$

The decision boundaries between two classes are quadratic functions.

Note: QDA does not differ much from LDA applied the enlarged quadratic polynomial input space but generally QDA is preferred in this case.

The estimates are similar but the covariance matrix must be estimated for each class. When p is large, this means a dramatic increase in the number of parameters, considering we only need the differences $\delta_{\mathrm{k}}(\mathrm{x})-\delta_{\mathrm{l}}(\mathrm{x})$. LDA needs $(\mathrm{K}-1)\times(\mathrm{p}+1)$ parameters, while QDA needs $(\mathrm{K}-1)\times(\mathrm{p}\,(\mathrm{p}+3)/2+1)$.

4.4 Regularized discriminant analysis

This method shrinks the separate covariances of QDA towards a common covariance as in LDA. In a way it is similar to ridge regression. The regularized covariance matrices have the form:

$$\hat{\mathbf{\Sigma}}_{\mathrm{k}}(\alpha) = \alpha\hat{\mathbf{\Sigma}}_{\mathrm{k}} + (1-\alpha)\hat{\mathbf{\Sigma}} \quad (119)$$

with $\alpha \in [0,1]$, the two extremes being LDA and QDA. α can be chosen on the validation data or by cross-validation.

Similarly we can allow $\hat{\mathbf{\Sigma}}$ to be shrunk toward the scalar covariance:

$$\hat{\mathbf{\Sigma}}(\gamma) = \gamma\hat{\mathbf{\Sigma}} + (1-\gamma)\hat{\sigma}^2\mathrm{I} \tag{120}$$

with $\gamma \in [0,1]$ so we get a more general family of covariances

$$\hat{\mathbf{\Sigma}}_\mathrm{k}(\alpha,\gamma) = \alpha\hat{\mathbf{\Sigma}}_\mathrm{k} + (1-\alpha)\left(\gamma\hat{\mathbf{\Sigma}} + (1-\gamma)\,\hat{\sigma}^2\mathrm{I}\right) \tag{121}$$

4.5 Computation

Computation of LDA and QDA is simplified by diagonalizing the covariance matrices with the singular value decomposition $\hat{\mathbf{\Sigma}}_\mathrm{k} = \mathrm{U_k D_k U_k^T}$. The terms in 118 become

$$\begin{aligned}
&(\mathrm{x}-\mu_\mathrm{k})^\mathrm{T}\,\mathbf{\Sigma}_\mathrm{k}^{-1}\,(\mathrm{x}-\mu_\mathrm{k}) = (\mathrm{x}-\mu_\mathrm{k})^\mathrm{T}\left(\mathrm{U_k D_k U_k^T}\right)^{-1}(\mathrm{x}-\mu_\mathrm{k}) = \\
&= (\mathrm{x}-\mu_\mathrm{k})^\mathrm{T}\,\mathrm{U_k D_k^{-1} U_k^T}\,(\mathrm{x}-\mu_\mathrm{k}) = \left[\mathrm{U_k^T}\,(\mathrm{x}-\mu_\mathrm{k})\right]^\mathrm{T}\mathrm{D_k^{-1}}\left[\mathrm{U_k^T}\,(\mathrm{x}-\mu_\mathrm{k})\right]
\end{aligned} \tag{122}$$

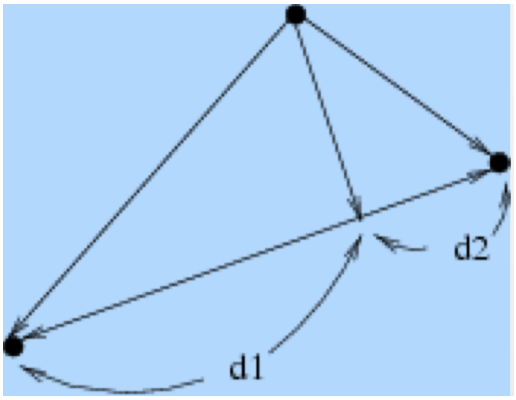
$$\log|\hat{\mathbf{\Sigma}}_\mathrm{k}| = \sum_\mathrm{l}\log \mathrm{d_{kl}} \tag{123}$$

Considering the above steps, LDA classifier can be seen as performing the following steps:

- sphere the data w.r.t. the common covariance estimate: $\mathrm{X}* \leftarrow \mathrm{D}^{-\frac{1}{2}}\mathrm{U}^\mathrm{T}\mathrm{X}$. The common covariance estimate fof X$*$ will now be the identity.
- Classify to the closest centroid in the transformed space, modulo the effect of the class prior probability π_k(?)

4.6 Regularized-rank linear discriminant analysis

Consider K = 2 with two centroids and the input is 2, i.e., input points are on a plane. Given an input point, for classification purposes what matters is not the distance in the p space of such point from the two centroids but rather the distance from the two centroids of the projection of this point on the line joining them (4.6). So basically instead of using the 2 dimensions, we are calculating the distance in one dimension, a line. If K = 3 then the points are projected onto a plane (2d), of course in this case it is convenient if p > 2.



More generally the K centroids in p-dimensional input, lie in an affine subspace of dimensions $\leq K-1$ and if p > is much larger than K this will be a considerable drop in dimension.

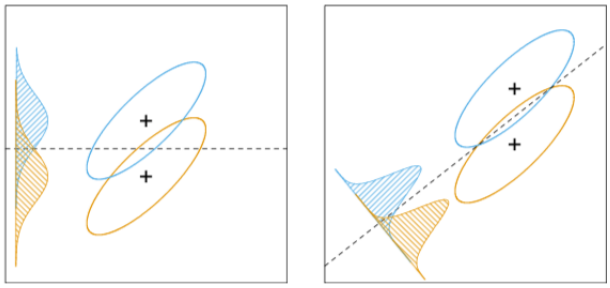
If $K > 3$ we can look for a $L < K-1$ dimensional subspace optimal for LDA. Fisher defined *optimal* such that the projected centroids were spread out as much as possible in terms of variance. This problem, finding the principal component subspaces of the centroids, involves the following steps:

- compute the $k \times p$ matrix of class centroids M and the common covariance matrix W for within-class covariance;
- compute $M* = MW^{-\frac{1}{2}}$ using the eigen-value decomposition;
- compute B∗, the covariance matrix of M∗ (B for between class covariance) and its eigenvalue decomposition $B^* = V^* D_B V^{*T}$. The columns v_l^* from the first to the last define the coordinates of optimal subspaces.

The lth discriminant variable is given by $Z_l = v_l^T X$ with $v_l = W^{-\frac{1}{2}} v_l^*$. Although the direction joining the centroids separates the means as much as possible(maximizes the between class covariance), there is an overlap between the projected classes due to the nature of covariances. Taking the covariances into account reduce the overlap and that is what we are doing (4.6).

The between-class variance Z is $a^T Ba$ and the within class variance is $a^T Wa$, with $B + W = T$, the total covariance matrix of X. Fisher's problem maximizes the *Rayleigh quotient*:

$$\max_a \frac{a^T Ba}{a^T Wa} \tag{124}$$



This is a generalized eigenvalue problem, with a given by the largest eigenvalue. Similarly one can find the next direction a_2, orthogonal in W to a_1, such that $a_2^T B a_2 / a_2^T W a_2$ is maximized; the solution is $a_2 = v_2$, and so on. a_l are the *discriminant coordinates* or *canonical variates*, different from discriminant functions.

The reduced subspaces can be used both for visualization and classification by limiting the distance between centroids to the chosen subspace. However, when doing this, due to the Gaussian classification, a correction factor of $\log \pi_k$ is needed. The misclassification is given by the overlapping area in 4.6 between the two densities. When both classes have the same priors π_k as in the figure, the optimal cut-point is the midway between projected means, if not the cut-point is moved towards the smaller class to have a better error rate.

For 2 classes one can derive the linear rule using LDA, and then choosing the cut-point to minimize misclassification error.

4.7 Logistic regression

The idea behind logistic regression is to still exploit a linear model $x^T \beta$ but having its output representing a probability, i.e., constrained between 0 and 1. This part is performed using the sigmoid function

$$p = \sigma(q) = \frac{1}{1 + e^{-q}} \tag{125}$$

Inverting the terms we get

$$q = -\log \frac{1-p}{p} = \log \frac{p}{1-p} \tag{126}$$

126 is called **logit function**. As q increases to ∞, the output of the sigmoid gets closer to 1; instead when it diverges to $-\infty$ we get 0. Suppose we have just 2 classes or equivalently a binary classifier that tells the probability of an event to happen. This is equivalent of having 2 classes: $\mathrm{Y_n} = 1$ when the event happens and $\mathrm{Y_n} = 0$ when it does not. We can express the probabilities output by our classifier as:

$$\begin{aligned} \mathrm{P}(\mathrm{G} = 1|\mathrm{x_n}, \beta) &= \frac{1}{1 + \mathrm{e}^{-\beta^{\mathrm{T}}\mathbf{x}_{\mathrm{new}}}} \\ \mathrm{P}(\mathrm{G} = 0|\mathrm{x_n}, \beta) &= 1 - \frac{1}{1 + \mathrm{e}^{-\beta^{\mathrm{T}}\mathbf{x}_{\mathrm{new}}}} = \frac{\mathrm{e}^{-\beta^{\mathrm{T}}\mathbf{x}_{\mathrm{new}}}}{1 + \mathrm{e}^{-\beta^{\mathrm{T}}\mathbf{x}_{\mathrm{new}}}} \end{aligned} \tag{127}$$

These single equations can be combined in a single equation:

$$\mathrm{P}(\mathrm{G} = \mathrm{g}|\mathrm{x_n}, \beta) = \mathrm{P}(\mathrm{G} = 1|\mathrm{x_n}, \beta)^{\mathrm{g}}\mathrm{P}(\mathrm{G} = 0|\mathrm{x_n}, \beta)^{\mathrm{g}-1} \tag{128}$$

Taking the log-ratio between the two probabilities we have:

$$\log \frac{\mathrm{P}(\mathrm{Y_n} = 0|\mathrm{x_n}, \beta)}{\mathrm{P}(\mathrm{Y_n} = 1|\mathrm{x_n}, \beta)} = \log \frac{\frac{\mathrm{e}^{-\beta^{\mathrm{T}}\mathbf{x}_{\mathrm{new}}}}{\cancel{1+\mathrm{e}^{-\beta^{\mathrm{T}}\mathbf{x}_{\mathrm{new}}}}}}{\frac{1}{\cancel{1+\mathrm{e}^{-\beta^{\mathrm{T}}\mathbf{x}_{\mathrm{new}}}}}} = -\beta^{\mathrm{T}}\mathbf{x}_{\mathrm{new}} \tag{129}$$

So we are using lines (or hyperplanes) to separate the two classes.

4.7.1 Multinomial logistic regression: more than 2 classes

Now suppose we have more than two classes and we still want to separate those classes with linear functions, which means we will have a hyperplane separating two classes. We have K possible outcomes. We can think to run $\mathrm{K}-1$ independent binary logistic regression in which one class is chosen as **pivot**, generally the one corresponding to class K, and then the other $\mathrm{K}-1$ are separately regressed against the pivot:

$$\begin{aligned} \log \frac{\Pr(\mathrm{G} = 1|\mathrm{X} = \mathrm{x})}{\Pr(\mathrm{G} = \mathrm{K}|\mathrm{X} = \mathrm{x})} &= \beta_{10} + \beta_1^{\mathrm{T}}\mathrm{x} \\ \log \frac{\Pr(\mathrm{G} = 2|\mathrm{X} = \mathrm{x})}{\Pr(\mathrm{G} = \mathrm{K}|\mathrm{X} = \mathrm{x})} &= \beta_{20} + \beta_2^{\mathrm{T}}\mathrm{x} \\ &\vdots \\ \log \frac{\Pr(\mathrm{G} = \mathrm{K}-1|\mathrm{X} = \mathrm{x})}{\Pr(\mathrm{G} = \mathrm{K}|\mathrm{X} = \mathrm{x})} &= \beta_{(\mathrm{K}-1)0} + \beta_{(\mathrm{K}-1)}^{\mathrm{T}}\mathrm{x} \end{aligned} \tag{130}$$

The model uses the last class as the denominator but this choice is arbitrarily and the estimates are equivalent under this choice.

Summing the probability of each class we must get 1:

$$\begin{aligned}
&\sum_{l=1}^{K} \Pr\left(G = l|X = x\right) = 1 \Rightarrow \Pr\left(G = K|X = x\right) + \sum_{l=1}^{K-1} \Pr\left(G = l|X = x\right) = 1 \\
&\Rightarrow \Pr\left(G = K|X = x\right) + \sum_{l=1}^{K-1} \Pr\left(G = K|X = x\right) e^{\beta_{l0} + \beta_l^T x} = 1 \\
&\Rightarrow \Pr\left(G = K|X = x\right) = \frac{1}{1 + \sum_{l=1}^{K-1} e^{\beta_{l0} + \beta_l^T x}}
\end{aligned} \tag{131}$$

So we can re-express the probabilities as:

$$\Pr\left(G = k|X = x\right) = \frac{e^{\beta_{k0} + \beta_k^T x}}{1 + \sum_{l=1}^{K-1} e^{\beta_{l0} + \beta_l^T x}} \tag{132}$$

Softmax function We do not have anymore the sigmoid function, instead we have used another function named **softmax** that as the sigmoid takes any real value as input and outputs a value between 0 and 1. The difference mostly relies on the denominator used for normalization factor.

4.7.2 Fitting logistic regression

We want to find the best parameters for the chosen model according to some criteria. First let us apply the Bayes theorem:

$$\Pr\left(\beta|\mathbf{y}, \mathbf{X}\right) = \frac{\Pr\left(\mathbf{y}|\beta, \mathbf{X}\right) \Pr(\beta)}{\Pr(\mathbf{y}|\mathbf{X})} \tag{133}$$

This formula tells we want to find the coefficients given some input and output data. Let us analyze each term:

- $\Pr(\mathbf{y}|\mathbf{X})$(**marginal likelihood**): It can be expressed as

$$\Pr(\mathbf{y}|\mathbf{X}) = \int \Pr\left(\mathbf{y}|\beta, \mathbf{X}\right) \Pr(\beta) \mathrm{d}\beta \tag{134}$$

Let us express $\Pr(\mathrm{G}=\mathrm{k}|\mathrm{X}=\mathrm{x})=\mathrm{p_k}(\mathrm{x},\theta)$ with $\theta=\left\{\beta_{10},\beta_1^{\mathrm{T}},\cdots,\beta_{(\mathrm{K}-1)0},\beta_{\mathrm{K}-1}^{\mathrm{T}}\right\}$ and let us define the log-likelihood

$$\mathrm{l}(\theta)=\sum_{\mathrm{i}=1}^{\mathrm{N}}\log \mathrm{p}_{\mathrm{g_i}}\left(\mathrm{x_i},\theta\right) \tag{135}$$

Consider just two classes with responses 0, 1 and let $\mathrm{p}_1(\mathrm{x},\theta)=\mathrm{p}(\mathrm{x},\theta)$ and $\mathrm{p}_2(\mathrm{x},\theta)=1-\mathrm{p}(\mathrm{x},\theta)$. Recall that having two classes $\sum_{\mathrm{l}=1}^{\mathrm{K}-1}\mathrm{e}^{\beta_{\mathrm{l}0}+\beta_{\mathrm{l}}^{\mathrm{T}}\mathrm{x_i}}=\mathrm{e}^{\beta_{10}+\beta_1^{\mathrm{T}}\mathrm{x_i}}$. The log-likelihood can be written as:

$$\begin{aligned}
&\mathrm{l}(\beta)=\sum_{\mathrm{i}=1}^{\mathrm{N}}\left\{\mathrm{y_i}\log \mathrm{p}\left(\mathrm{x_i},\beta\right)+(1-\mathrm{y_i})\log\left(1-\mathrm{p}\left(\mathrm{x_i},\beta\right)\right)\right\}=\\
&=\sum_{\mathrm{i}=1}^{\mathrm{N}}\left\{\mathrm{y_i}\log\frac{\mathrm{e}^{\beta_{10}+\beta_1^{\mathrm{T}}\mathrm{x_i}}}{1+\sum_{\mathrm{l}=1}^{\mathrm{K}-1}\mathrm{e}^{\beta_{\mathrm{l}0}+\beta_{\mathrm{l}}^{\mathrm{T}}\mathrm{x_i}}}+(1-\mathrm{y_i})\log\left(1-\frac{\mathrm{e}^{\beta_{10}+\beta_1^{\mathrm{T}}\mathrm{x_i}}}{1+\sum_{\mathrm{l}=1}^{\mathrm{K}-1}\mathrm{e}^{\beta_{\mathrm{l}0}+\beta_{\mathrm{l}}^{\mathrm{T}}\mathrm{x_i}}}\right)\right\}=\\
&=\sum_{\mathrm{i}=1}^{\mathrm{N}}\left\{\mathrm{y_i}\log\frac{\mathrm{e}^{\beta_{10}+\beta_1^{\mathrm{T}}\mathrm{x_i}}}{1+\mathrm{e}^{\beta_{10}+\beta_1^{\mathrm{T}}\mathrm{x_i}}}+(1-\mathrm{y_i})\log\left(1-\frac{\mathrm{e}^{\beta_{10}+\beta_1^{\mathrm{T}}\mathrm{x_i}}}{1+\mathrm{e}^{\beta_{10}+\beta_1^{\mathrm{T}}\mathrm{x_i}}}\right)\right\}=\\
&=\sum_{\mathrm{i}=1}^{\mathrm{N}}\mathrm{y_i}\left(\log\mathrm{e}^{\beta_{10}+\beta_1^{\mathrm{T}}\mathrm{x_i}}-\log\left(1+\mathrm{e}^{\beta_{10}+\beta_1^{\mathrm{T}}\mathrm{x_i}}\right)\right)+(1-\mathrm{y_i})\log\frac{1}{1+\mathrm{e}^{\beta_{10}+\beta_1^{\mathrm{T}}\mathrm{x_i}}}=\\
&=\sum_{\mathrm{i}=1}^{\mathrm{N}}\mathrm{y_i}\left(\log\mathrm{e}^{\beta_{10}+\beta_1^{\mathrm{T}}\mathrm{x_i}}-\log\left(1+\mathrm{e}^{\beta_{10}+\beta_1^{\mathrm{T}}\mathrm{x_i}}\right)\right)-(1-\mathrm{y_i})\log\left(1+\mathrm{e}^{\beta_{10}+\beta_1^{\mathrm{T}}\mathrm{x_i}}\right)=\\
&=\sum_{\mathrm{i}=1}^{\mathrm{N}}\mathrm{y_i}\left(\beta_{10}+\beta_1^{\mathrm{T}}\mathrm{x_i}\right)-\log\left(1+\mathrm{e}^{\beta_{10}+\beta_1^{\mathrm{T}}\mathrm{x_i}}\right)
\end{aligned} \tag{136}$$

Here $\beta=[\beta_{10},\beta_1]$. To maximize the log-likelihood we set the derivative to 0:

$$\frac{\partial\ell(\beta)}{\partial\beta}=\sum_{\mathrm{i}=1}^{\mathrm{N}}\mathrm{x_i}(\mathrm{y_i}-\mathrm{p}(\mathrm{x_i},\beta)=0 \tag{137}$$

Using the *Newton-Raphson* algorithm that requires the second-derivative:

$$\frac{\partial^2 \ell(\beta)}{\partial \beta \partial \beta^{\mathrm{T}}} = -\sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{x_i} \mathrm{x_i^T} \mathrm{p}(\mathrm{x_i}, \beta)(1 - \mathrm{p}(\mathrm{x_i}, \beta) \tag{138}$$

$$\Rightarrow \beta^{\mathrm{new}} = \beta^{\mathrm{old}} - \left(\frac{\partial^2 \ell\,(\beta)}{\partial \beta \partial \beta^{\mathrm{T}}} \right)^{-1} \frac{\partial \mathrm{l}(\beta)}{\partial \beta} \tag{139}$$

Using the matrix notation,

$$\frac{\partial \ell(\beta)}{\partial \beta} = \mathrm{X}^{\mathrm{T}}(\mathrm{y} - \mathrm{p}) \tag{140}$$

$$\frac{\partial^2 \ell(\beta)}{\partial \beta \partial \beta^{\mathrm{T}}} = -\mathrm{X}^{\mathrm{T}} \mathrm{W} \mathrm{X} \tag{141}$$

So the Newton step becomes

$$\begin{aligned} \beta^{\mathrm{new}} &= \beta^{\mathrm{old}} + (\mathrm{X}^{\mathrm{T}} \mathrm{W} \mathrm{X})^{-1} \mathrm{X}^{\mathrm{T}} (\mathrm{y} - \mathrm{p}) = \\ &= (\mathrm{X}^{\mathrm{T}} \mathrm{W} \mathrm{X})^{-1} \mathrm{X}^{\mathrm{T}} \mathrm{W} \left(\mathrm{X} \beta^{\mathrm{old}} + \mathrm{W}^{-1} (\mathrm{y} - \mathrm{p}) \right) = \\ &= \left(\mathrm{X}^{\mathrm{T}} \mathrm{W} \mathrm{X} \right)^{-1} \mathrm{X}^{\mathrm{T}} \mathrm{W} \mathrm{z} \end{aligned} \tag{142}$$

with

$$\mathrm{z} = \mathrm{X} \beta^{\mathrm{old}} + \mathrm{W}^{-1} (\mathrm{y} - \mathrm{p}) \tag{143}$$

sometimes known as the adjusted response. This algorithm is known as **iteratively reweighted least squares (IRLS)** since each iteration solves the weighted least square problem:

$$\beta^{\mathrm{new}} \leftarrow \arg\min_{\beta} (\mathrm{z} - \mathrm{X}\beta)^{\mathrm{T}} \mathrm{W} (\mathrm{z} - \mathrm{X}\beta) \tag{144}$$

$\beta = 0$ seems a good starting value. Convergence is never guaranteed but typically the algorithm does converge, since the log-likelihood is concave but overshooting can occur. In the rare cases that the log-likelihood decreases, step size halving will guarantee convergence.

For $\mathrm{K} > 2$ we still use the iteration procedure but we will have a $\mathrm{K} - 1$ vector response and a non-diagonal weight matrix per observation. In this case it is better to work with the vector θ directly.

4.7.3 Usage

LR is used as data analysis tool where the goal is to understand the role of the input variables in explaining the outcome. Typically many models are fit in a search for a parsimonious model involving a subset of the variables, possibly with some interactions terms.

It is widely used in biostatistical applications where binary responses (two classes) occur quite frequently. For example, patients survive or die, have heart disease or not, or a condition is present or absent.

4.8 Regularized Logistic regression

We can use the L_1 penalty for variable selection and shrinkage:

$$\underset{\beta_0,\beta_1}{\arg\max} \left\{ \sum_{i=1}^{N} y_i \left(\beta_0 + \beta^T x_i\right) - \log\left(1 + e^{\beta_0+\beta^T x_i}\right) - \lambda \sum_{j=-1}^{p} |\beta_j| \right\} \quad (145)$$

This function is concave and can be solved using a nonlinear programming method.

4.9 Logistic vs LDA

The difference between the models relies on how the linear coefficients are estimated. The logistic regression model is more general since it makes less assumptions.

LDA is not robust to outliers since observations far from the decision boundary are used to estimate the common covariance matrix, while they are scaled down in the Logistic regression.

4.10 Perceptron learning algorithm

It tries to find a separate hyperplane by minimizing the distance of misclassified points to the decision boundary. If a response $y_i = 1$ is misclassified, then $x_i^T\beta + \beta_0 < 0$, and the opposite for a misclassified response with $y_i = -1$. The goal is to minimize

$$D(\beta, \beta_0) = -\sum_{i \in \mathcal{M}} y_i \left(x_i^T \beta + \beta_0\right) \quad (146)$$

where $\mathcal{M}$ is the set of misclassified points. The gradient is

$$\frac{\partial \mathrm{D}(\beta, \beta_0}{\partial \beta} = -\sum_{\mathrm{i} \in \mathcal{M}} \mathrm{y_i x_i} \tag{147}$$

$$\frac{\partial \mathrm{D}(\beta, \beta_0}{\partial \beta_0} = -\sum_{\mathrm{i} \in \mathcal{M}} \mathrm{y_i} \tag{148}$$

where the algorithm uses the stochastic gradient descent where the coefficients are updated by the gradient value weighted by a step ρ. There are many problems though:

- when data are separable, there are many solutions which depend on the starting value;
- many steps might be required;
- when data are not separable, the algorithm will not converge;

4.11 Best separating hyperplanes

The optimal separating hyperplane separates the two classes and maximizes the distance to the closest point from either class (Vapnik, 1996). Not only does this provide a unique solution to the separating hyperplane problem, but by maximizing the margin between the two classes on the training data, this leads to better classification performance on test data.

Suppose M is the distance. Then

$$\max_{\beta, \beta_0, ||\beta||=1} \mathrm{M} \tag{149}$$

$$\mathrm{y_i}(\mathrm{x_i^T}\beta + \beta_0) \geq \mathrm{M} \tag{150}$$