1 Derive the normal equation

Given a data **X**,**y** and assuming a probabilistic model given by $y = \beta^T x + \epsilon$ where $\epsilon \sim Normal(0, \sigma^2)$, show that the β that maximizes the probability of obtaining the data is given by: $\beta = (X^T X)^{-} X^T y$.

1.1 Solution

1.1.1 Feature space

Assume m training examples $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), ..., (x^{(m)}, y^{(m)})$ with n features $x_1, x_2, ..., x_n$. At index 0, let $x_0^{(1)}, x_0^{(2)}, ..., x_0^{(i)}, ..., x_0^{(n)}$ all equal 1. Therefore, if there are n features and a 0th index, there will be $n+1$ feature vectors.

Let X be the design matrix of $n+1$ feature vectors $x_{n+1}^{(i)}$ where $x^{(i)}$ denotes the i-th $n+1$-dimensional feature vector contained within X.

Thus, each row of the matrix X is filled by $(x^{(1 \text{ to } m)})^T$, making X an $m \times (n+1)$-dimensional matrix of all the features of the training data:

$$x^{(i)} = \begin{bmatrix} x_0^{(i)} \\ x_1^{(i)} \\ x_2^{(i)} \\ . \\ x_n^{(i)} \end{bmatrix} \in \mathrm{I\!R}^{\mathrm{n+1}} \qquad \mathrm{X} = \begin{bmatrix} x_0^{(1)} & x_1^{(1)} & x_2^{(1)} & . & . & . & x_n^{(1)} \\ x_0^{(2)} & x_1^{(2)} & x_2^{(2)} & . & . & . & x_n^{(2)} \\ . & . & . & . & . & . & . \\ x_0^{(i)} & x_1^{(i)} & x_2^{(i)} & . & . & . & x_n^{(i)} \\ . & . & . & . & . & . & . \\ x_0^{(m)} & x_1^{(m)} & x_2^{(m)} & . & . & . & x_n^{(m)} \end{bmatrix}$$

Let y be the vector of true values of the above described training examples.

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

1.1.2 Hypothesis function

Given the hypothesis function:

$$h_\beta(x) = \beta_0 x_0 + \beta_1 x_1 + ... + \beta_n x_n$$

(Recall that $x_0 = 1$.)

The above hypothesis function can be represented using matrix notation. The regression coefficients of hypothesis function $h_\beta(x)$ can be represented as an

$n+1$-dimensional vector:

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ . \\ . \\ . \\ \beta_n \end{bmatrix} \in \mathrm{I\!R}^{\mathrm{n+1}}$$

Similarly, each of the m training examples is an $n+1$-dimensional vector $\begin{bmatrix} x_0^{(i)} \\ x_1^{(i)} \\ \vdots \\ x_n^{(i)} \end{bmatrix}$ with $x_0^{(i)} = 1$ to allow for a convenient vector multiplication.

Thus, the hypothesis function for each x_i, $h_\beta(x_i)$, can be written as:

$$h_\beta(x_i) = \beta^T x + \epsilon$$

where β and x_i are $n+1$-dimensional vectors, and ϵ is the normally distributed error for each observation.

1.1.3 Training error

The training error for the above generalized example could be expanded algebraically as:

$$Error\ (\epsilon) = \begin{bmatrix} y_1 - (\beta_0 x_0^{(1)} + \beta_1 x_1^{(1)} + \beta_2 x_2^{(1)} + ... + \beta_n x_n^{(1)}) \\ y_2 - (\beta_0 x_0^{(2)} + \beta_1 x_1^{(2)} + \beta_2 x_2^{(2)} + ... + \beta_n x_n^{(2)}) \\ y_3 - (\beta_0 x_0^{(3)} + \beta_1 x_1^{(3)} + \beta_2 x_2^{(3)} + ... + \beta_n x_n^{(3)}) \\ ... \\ ... \\ y_m - (\beta_0 x_0^{(m)} + \beta_1 x_1^{(m)} + \beta_2 x_2^{(m)} + ... + \beta_n x_n^{(m)}) \end{bmatrix} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ . \\ . \\ \epsilon_m \end{bmatrix} \in \mathrm{I\!R}^{\mathrm{m}}$$

This can be simplified using matrix notation and the matrices that were defined above. $y_{1 \text{ to } m}$ in the matrix above correspond to the matrix y of true values. The $\beta_{0 \text{ to } n}$ correspond to the matrix β of regression coefficients. The $x_{0 \text{ to } n}$ correspond to the design matrix X of the $m \times (n+1)$-dimensional feature space.

Using matrix addition and multiplication, the above matrix simplifies to:

$$Error\ (\epsilon) = y_{m \times 1} - X_{m \times n+1} \beta_{n+1} \in \mathrm{I\!R}^{\mathrm{m}}$$

(The subscripts denote the dimensions of these matrices for convenience.)

1.1.4 Deriving the cost function

The likelihood of obtaining the model parameters from the data is given by:

$$\mathcal{L}(\beta | y, X) = \Pr(X | \beta)$$

where $\mathcal{L}$ is the likelihood, X is the design matrix (i.e., the data), and β is the vector of model parameters.

The probability of the data X given the model parameters β is the joint probability of each individual data point:

$$\Pr(X|\beta) = \prod_{i=1}^{m} \Pr(y_i|x_i, \beta)$$

It is given that $\epsilon \sim Normal(0, \sigma)$. As the noise ϵ is additive, the linearity condition implies that $\Pr(y_i|x_i, \beta) \sim Normal(\beta^T x_i, \sigma_\epsilon^2)$.

The goal is to find a set of model parameters β that maximize the likelihood. Taking the logarithm of both sides helps simplify the equation. Since the logarithm is a monotonic function, the maximum of the log-likelihood occurs at the same value of β as the maximum of the likelihood. Thus, taking the ln of both sides:

$$\begin{aligned}
\ln \mathcal{L}(\beta) &= \ln \prod_{i=1}^{m} \Pr(y_i|x_i, \beta) \\
&= \sum_{i=1}^{m} [\ln \Pr(y_i|x_i, \beta)] \\
&= \sum_{i=1}^{m} \left[-\frac{1}{2\sigma_\epsilon^2}(y_i - \beta^T x_i)^2 - \ln\left(\sqrt{2\pi\sigma_\epsilon^2}\right) \right]
\end{aligned}$$

The last step follows because $\Pr(y_i|x_i\beta)$ is a Gaussian probability density as noted above.

As the goal is to maximize the above likelihood (or more precisely, log-likelihood) in terms of the model parameters, the above terms that do not depend on β (i.e., $-\frac{1}{2\sigma_\epsilon^2}$, $-\ln\sqrt{2\pi\sigma_\epsilon^2}$) can be ignored. Thus, the optimization problem can be written as:

$$\ln \mathcal{L}(\beta) = \sum_{i=1}^{m} (y_i - \beta^T x_i)^2$$

This is the sum of least squares!

1.1.5 Rewriting the cost function using matrices

The goal is to minimize the least-squares cost function:

$$J(\beta_{0\ldots n}) = \frac{1}{2m} \sum_{i=1}^{m} (y_i - h_\beta(x^{(i)}))^2$$

where, as above, $x^{(i)}$ is the i-th sample from a set of m samples and $y^{(i)}$ is the i-th true value.

Because the term $h_\beta(x^{(i)}) - x^{(i)} = \epsilon^{(i)}$, i.e.,

$$J(\beta_{0...n}) = \frac{1}{2m}\sum_{i=1}^{m}(y_i - h_\beta(x^{(i)}))^2 = \frac{1}{2m}\sum_{i=1}^{m}\epsilon_i^2$$

another way of stating this problem is minimizing the sum of the squared errors in the *Error* vector ϵ, i.e., $\epsilon^T \times \epsilon$. Concretely,

$$\sum_{i=1}^{m}\epsilon_i^2 = \begin{bmatrix}\epsilon_1 & \epsilon_2 & \epsilon_3 & . & . & \epsilon_m\end{bmatrix} * \begin{bmatrix}\epsilon_1\\ \epsilon_2\\ \epsilon_3\\ .\\ .\\ \epsilon_m\end{bmatrix} = \epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2 + ... + \epsilon_m^2$$

As above, $\epsilon = y - X\beta$. Thus,

$$J(\beta_{0...n}) = \frac{1}{2m}(y - X\beta)^T(y - X\beta)$$

Ignoring the constant $\frac{1}{2m}$,

$$\begin{aligned} J(\beta_{0...n}) &= (y^T - (X\beta^T))(y - X\beta) \\ &= y^Ty - y^TX\beta - (X\beta)^Ty + (X\beta)^TX\beta \end{aligned}$$

Take the transpose of the second term in the above equation $(y^TX\beta)^T = (X\beta)^Ty$. Thus,

$$\begin{aligned} J(\beta_{0...n}) &= y^Ty - (X\beta)^Ty - (X\beta)^Ty + (X\beta)^TX\beta \\ &= y^Ty - 2(X\beta)^Ty + (X\beta)^TX\beta \end{aligned}$$

Distribute the transpose in the last term and the final equation for $J(\beta_{0...n})$ is:

$$J(\beta_{0...n}) = y^Ty - 2(X\beta)^Ty + \beta^TX^TX\beta$$

1.1.6 Minimizing the cost function

In order to find the minimum of the cost function, the derivative of $J(\beta_{0...n})$ must be taken and then set to zero:

$$\frac{\partial J}{\partial \beta} = 0$$

To simplify the operations, the derivative of each term of $J(\beta_{0...n})$ will be taken separately. y^Ty will be ignored given it has no β terms and the derivative of a constant is 0.

$$J(\beta_{0...n}) = P(\beta_{0...n}) + Q(\beta_{0...n}) + y^Ty$$

$$P(\beta_{0...n}) = \beta^TX^TX\beta \qquad (1)$$

$$Q(\beta_{0...n}) = -2(X\beta)^Ty \qquad (2)$$

1.1.6.1 Differentiate $P(\beta_{0...n})$

$$P(\beta_{0...n}) = \beta^T X^T X \beta$$

Importantly, the product $X^T_{n+1\times m} X_{m\times n+1}$ is a square, symmetrical $n+1\times n+1$-dimensional matrix. For convenience Z will be substituted for $X^T X$. Therefore, $P(\beta_{0...n})$ can be rewritten as: $\beta^T Z \beta$ where Z is the square, symmetrical matrix defined above.

For the case where a scalar α is given by

$$\alpha = x^T A x$$

where x is $n \times 1$, A is $n \times n$, and A does not depend on x:

$$\begin{aligned}
\alpha &= \sum_{j=1}^{n}\sum_{i=1}^{n} a_{ij}x_i x_j \\
\frac{\partial \alpha}{\partial x_k} &= \sum_{j=1}^{n} a_{kj}x_j + \sum_{i=1}^{n} a_{ik}x_i && \text{for the } k\text{th element of } x \\
\frac{\partial \alpha}{\partial x} &= x^T A^T + x^T A && \text{for all } k = 1, 2, ...n \\
&= x^T(A^T + A)
\end{aligned}$$

For the special case where A is a symmetrical matrix, $A^T = A$, therefore $(A^T + A) = 2A$ and $\frac{\partial \alpha}{\partial x} = 2x^T A$.

Back to the problem at hand, Z was noted to be a square, symmetrical matrix. Therefore,

$$\begin{aligned}
P(\beta_{0...n}) &= \beta^T Z \beta \\
\frac{\partial P}{\partial \beta} &= 2\beta^T Z \\
&= 2\beta^T X^T X && \text{substituting } X^T X \text{ for } Z \\
&= 2(\beta^T X^T X)^T && \text{take the transpose} \\
&= 2X^T X \beta
\end{aligned}$$

1.1.6.2 Differentiate $Q(\beta_{0...n})$

$$
\begin{aligned}
Q(\beta_{0...n}) &= -2(X\beta)^T y \\
&= -2\left(\begin{bmatrix} x_0^{(1)} & x_1^{(1)} & x_2^{(1)} & . & . & . & x_n^{(1)} \\ x_0^{(2)} & x_1^{(2)} & x_2^{(2)} & . & . & . & x_n^{(2)} \\ . & . & . & . & . & . & . \\ x_0^{(i)} & x_1^{(i)} & x_2^{(i)} & . & . & . & x_n^{(i)} \\ . & . & . & . & . & . & . \\ x_0^{(m)} & x_1^{(m)} & x_2^{(m)} & . & . & . & x_n^{(m)} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ . \\ . \\ . \\ \beta_n \end{bmatrix}\right)^T \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ y_m \end{bmatrix} \\
&= -2\left(\begin{bmatrix} \beta_0 x_0^{(1)} + \beta_1 x_1^{(1)} + \beta_2 x_2^{(1)} + ... + \beta_n x_n^{(1)} \\ \beta_0 x_0^{(2)} + \beta_1 x_1^{(2)} + \beta_2 x_2^{(2)} + ... + \beta_n x_n^{(2)} \\ ... \\ \beta_0 x_0^{(i)} + \beta_1 x_1^{(i)} + \beta_2 x_2^{(i)} + ... + \beta_n x_n^{(i)} \\ ... \\ \beta_0 x_0^{(m)} + \beta_1 x_1^{(m)} + \beta_2 x_2^{(m)} + ... + \beta_n x_n^{(m)} \end{bmatrix}\right)^T \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ y_m \end{bmatrix}
\end{aligned}
$$

$$
\begin{aligned}
Q(\beta_{0...n}) = -2[\, & y_1(\beta_0 x_0^{(1)} + ... + \beta_n x_n^{(1)}) \\
& + y_2(\beta_0 x_0^{(2)} + ... + \beta_n x_n^{(2)}) + ... + y_m(\beta_0 x_0^{(m)} + ... + \beta_n x_n^{(m)})\,]
\end{aligned}
$$

Rearranging the above using sums:

$$
\begin{aligned}
Q(\beta_{0...n}) &= -2\sum_{r=1}^{m} y_r(\beta_0 x_0^{(r)} + ... + \beta_n x_n^{(r)}) && (3) \\
&= -2\sum_{r=1}^{m} y_r \sum_{s=1}^{n} \beta_s x_s^{(r)} && (4)
\end{aligned}
$$

Using equation (3) above to differentiate:

$$
\frac{\partial Q}{\partial \beta} = -2\sum_{r=1}^{m} y_r(\beta_0 x_0^{(r)} + ... + \beta_n x_n^{(r)})\partial\beta
$$

This can be rewritten as a series of partial derivatives:

$$\begin{aligned}
\frac{\partial Q}{\partial \beta_0} &= -2(x_0^{(1)}y_1 + x_1^{(1)}y_1 + ... + x_n^{(1)}y_m) \\
\frac{\partial Q}{\partial \beta_1} &= -2(x_0^{(2)}y_1 + x_1^{(2)}y_1 + ... + x_n^{(2)}y_m) \\
\frac{\partial Q}{\partial \beta_2} &= -2(x_0^{(3)}y_1 + x_1^{(3)}y_1 + ... + x_n^{(3)}y_m) \\
&... \\
\frac{\partial Q}{\partial \beta_n} &= -2(x_0^{(m)}y_1 + x_1^{(m)}y_1 + ... + x_n^{(m)}y_m)
\end{aligned}$$

This can be collapsed as a vector of partial derivatives:

$$\begin{bmatrix} \frac{\partial Q}{\partial \beta_0} \\ \frac{\partial Q}{\partial \beta_1} \\ \frac{\partial Q}{\partial \beta_2} \\ . \\ . \\ \frac{\partial Q}{\partial \beta_n} \end{bmatrix} = -2 \left(\begin{bmatrix} x_0^{(1)} & x_1^{(1)} & x_2^{(1)} & . & . & x_n^{(1)} \\ x_0^{(2)} & x_1^{(2)} & x_2^{(2)} & . & . & x_n^{(2)} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ x_0^{(m)} & x_1^{(m)} & x_2^{(m)} & . & . & x_n^{(m)} \end{bmatrix} \right)^T \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ y_m \end{bmatrix}$$

In other words,

$$\frac{\partial Q}{\partial \beta_0} = -2\frac{\partial (X\beta)^T y}{\partial \beta_0} = -2X^T y$$

Putting this all together,

$$\begin{aligned}
J(\beta_{0...n}) &= P(\beta_{0...n}) + Q(\beta_{0...n}) \\
\frac{\partial J}{\partial \beta} &= \frac{\partial P}{\partial \beta} + \frac{\partial Q}{\partial \beta_0} \\
&= 2X^T X\beta - 2X^T y = 0 \qquad \text{and solve for } \beta \\
2X^T X\beta &= 2X^T y \\
X^T X\beta &= X^T y \qquad \text{multiply both sides by } (X^T X)^- \\
\beta &= (X^T X)^- X^T y \qquad \blacksquare
\end{aligned}$$

2 Show that Regularized Linear Regression has a Bayesian interpretation

Given data $\mathbf{X}$,$\mathbf{y}$ and assuming a linear model $y = \beta^T x$ with a prior distribution over β given by a normal distribution with mean 0, show that the β that maximizes the probability of having obtained the data is given by:

$$\beta = (X^T X + \lambda I)^- X^T y$$

where λ depends on the variance of the prior distribution.

2.1 Solution

2.1.1 Using Bayes' theorem to rephrase maximum likelihood estimation

The maximum likelihood estimator discussed **2.1.4** can be related to the most probable Bayes estimator given a uniform prior distribution. The maximum *a posteriori* estimate is the vector of parameters β that maximize the probability of β given the data. Using Bayes' theorem to write this:

$$\Pr(\beta|x_1, x_2, ..., x_n) = \frac{h(x_1, x_2, ..., x_n|\beta)\Pr(\beta)}{\Pr(x_1, x_2, ..., x_n)}$$

where $\Pr(\beta)$ is the prior distribution for the parameters β and $\Pr(x_1, x_2, ..., x_n)$ is the probability of obtaining the data. The denominator is independent of β, so the Bayesian estimator is obtained by maximizing $h(x_1, x_2, ..., x_n|\beta)\Pr(\beta)$ with respect to β.

In the derivation of the cost function in **2.1.4**, the Bayesian estimator could be considered to correspond to the maximum likelihood estimator for a uniform prior distribution of βs given by $\Pr(\beta) \sim Uniform(0, \beta)$. In other words, it is solving for $\Pr(data|\beta)$, i.e., the probability of obtaining the data given the parameters β.

In contrast to finding $\Pr(data|\beta)$, here the problem is written as finding $\Pr(\beta|data)$, which is derived using Bayes' theorem and **prior** knowledge (i.e., a prior) of the distribution of β.

In **Exercise 3**, the prior distribution of the β vector is given as $\Pr(\beta) \sim Normal(0, \beta)$. Therefore, the Bayesian estimator can be rewritten as:

$$\Pr(\beta|data) = \frac{\Pr(data|\beta) * prior}{\Pr(data)}$$
$$\Pr(\beta|x_1, x_2, ..., x_n) = h(x_1, x_2, ..., x_n|\beta)\Pr(\beta)$$

again ignoring the denominator because it is independent of β.

2.1.2 Maximum *a posteriori* estimator

The maximum *a posteriori* extimator is then given by:

$$\mathcal{L}(\beta|X) = \Pr(X|\beta)\Pr(\beta)$$

where $\mathcal{L}$ is the likelihood, X is the design matrix (i.e., the data), β is the vector of model parameters, and $\Pr(\beta)$ is given by the normal distribution as described above.

The likelihood of the model parameters given the data is the joint probability of each individual data point multiplied by the prior:

$$\mathcal{L}(\beta|X) = \Pr(\beta)\prod_{i=1}^{m}\Pr(y_i|x_i, \beta)$$

As before, the log-likelihood is easier to work with:

$$\ln \mathcal{L}(\beta|X) = \ln \left[\Pr(\beta) \prod_{i=1}^{m} \Pr(y_i|x_i, \beta) \right] \tag{1}$$

$$= \ln \Pr(\beta) + \sum_{i=1}^{m} \ln \left[\Pr(y_i|x_i, \beta) \right] \tag{2}$$

From **2.1.4** above, the second term $\sum_{i=1}^{m} \ln \left[\Pr(y_i|x_i, \beta) \right]$ is the sum of squared residuals $\sum_{i=1}^{m} (y_i - \beta^T x_i)^2$.

To gain an intuition of how the first term can be written as a sum, assume the parameters β are distributed normally and independently around the origin with variance σ_β^2, as given:

$$\begin{aligned}
\Pr(\beta) &= \prod_{i=0}^{n} \Pr(\beta_i) \\
&= \frac{1}{2\pi\sigma_\beta^2} \exp\left(- \frac{\sum_{i=0}^{n} \beta_i^2}{2\sigma_\beta^2} \right) \\
&= \frac{1}{2\pi\sigma_\beta^2} \exp\left(- \frac{\beta^T \beta}{2\sigma_\beta^2} \right) \qquad \text{written as a vector} \\
\ln \Pr(\beta) &= -\frac{1}{2\sigma_\beta^2} \beta^T \beta \qquad \text{taking the ln() of both sides}
\end{aligned}$$

Plugging this back into equation (2) above and rewriting the sum of least squares in matrix form (see **2.1.5**), we obtain the below. Recall that the objective can be multiplied by any scalar without affecting the optimum:

$$\begin{aligned}
\ln \mathcal{L}(\beta|X) &= -\frac{1}{2\sigma_\beta^2} \beta^T \beta - \left(\frac{1}{2\sigma_\epsilon^2} (y - X\beta)^T (y - X\beta) \right) \\
&= -\frac{\sigma_\epsilon^2}{\sigma_\beta^2} \beta^T \beta - (y - X\beta)^T (y - X\beta)
\end{aligned}$$

Rather than maximizing the above function, the signs can be reversed and the function minimized:

$$\begin{aligned}
\ln \mathcal{L}(\beta|X) &= \frac{\sigma_\epsilon^2}{\sigma_\beta^2} \beta^T \beta + (y - X\beta)^T (y - X\beta) \qquad \text{set } \frac{\sigma_\epsilon^2}{\sigma_\beta^2} = \lambda \\
&= \beta^T X^T X \beta + \lambda \beta^T \beta - 2\beta^T X^T y
\end{aligned}$$

The first and third terms' partial derivatives with respect to β were proven above in **Exercise 2**:

$$2X^T X\beta - 2X^T y$$

The second term's partial derivative with respect to β is:

$$2\lambda\beta$$

Putting this all together:

$$\frac{\partial \mathcal{L}}{\partial \beta} = 2X^T X\beta - 2X^T y + 2\lambda\beta$$

Set the derivative equal to 0 to minimize, and then solve for β. The λ term, recall, depends on the variance of the prior distribution:

$$\begin{aligned}
0 &= X^T X\beta - X^T y + \lambda\beta \\
X^T y &= X^T X\beta + \lambda\beta \\
X^T y &= (X^T X + \lambda I)\beta \\
\beta &= (X^T X + \lambda I)^{-} X^T y \qquad \blacksquare
\end{aligned}$$