10 Linear Regression

10.1 Simple Linear Regression

We consider the problem of dependence of a variable on another variable. For example output values depend almost linearly on input values. Let us consider $(x_i, y_i), i = 1, ..., n$ datapoints. We want to fit a line that best matches the input values x_i to the corresponding output values y_i. Our model is

$$f(x) = mx + b.$$

To get the line that best fits the data we consider the objective function

$$J = \sum_{i=1}^{n} (mx_i + b - y_i)^2.$$

The mean values of the inputs and outputs $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ and $\overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ are coordinates of a point on the line $y = mx + b$, therefore we have

$$b = \overline{y} - m\overline{x}.$$

To find the best parameter value m we need to look for critical points of the function

$$J = \sum_{i=1}^{n} (m(x_i - \bar{x}) - (y_i - \bar{y}))^2.$$

The condition $\frac{dJ}{dm} = 0$ gives

$$= 2\sum_{i=1}^{n} (m(x_i - \bar{x}) - (y_i - \bar{y})) \cdot (x_i - \bar{x}) = 0$$

and by direct calculation we get

$$m = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}.$$

If the variable y_i depends approximately linearly on x_i then the line $y = mx + b$ will be called the regression line and the above method is called linear regression. If y_i does not depend linearly on x_i then the line will be just a random line through a cloud of points. To measure the linear dependence between variables we use (Pearson's) correlation coefficient

$$r = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n} (y_i - \bar{y})^2}}.$$

A value of $r \approx 1$ $(r < 1)$ means positive correlation, value of $r \approx -1$ $(-1 < r)$ means negative correlation, while $r \approx 0$ means there is no correlation between the variables.

10.2 Multilinear regression

Let us consider the following problem. Estimate house prices based on recent sales

Price	No.Bed	No.Bath	Size	Lot Size	Year Built
320000	2	1.75	941	0	1995
250000	1	1	754	0	1988
340000	3	2.25	1209	110077	1999
316000	2	2.5	1175	479497	1999
300000	2	2.25	1173	0	1999

We define the estimate as a linear function

$$h(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + ... + w_n x_n$$

or

$$h(\mathbf{x}) = \mathbf{w}^{\mathbf{T}}\mathbf{x}$$

where we use the affine trick (set $x_0 = 1$)

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ ... \\ w_n \end{bmatrix}, \ \mathbf{x} = \begin{bmatrix} 1 \\ x_1 \\ ... \\ x_n \end{bmatrix}$$

The machine learning problem here is to learn the weights that give a correct estimate. More precisely:

Given training data $(\hat{\mathbf{x}}_k, \hat{y}_k)$, $k = 1, ..., m$, (please note that each $\hat{\mathbf{x}}_k$ is a vector of $n+1$ components) find the weights $w_0, w_1, ..., w_n$ such that

$$h(\hat{\mathbf{x}}_k) = \hat{y}_k, \ k = 1, ..., m$$

It should also output correct result on new data. It is easy to see that the problem is overdetermined.

To make sure that we have a solvable problem we can formulate the problem as an optimization problem, that of minimizing the error

$$J(\mathbf{w}) = \sum_{k=1}^{m} (h(\hat{\mathbf{x}}_k) - \hat{y}_k)^2$$

10.3 Preliminaries

Gradient and Hessian

- Gradient of a function $f : \mathbb{R}^n \to \mathbb{R}$

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ ... \\ \frac{\partial f}{\partial x_n} \end{bmatrix}$$

- Hessian

$$
\mathbf{H}_{\mathbf{x}} f = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & ... & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ ... & ... & ... \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & ... & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}
$$

Convex Optimization

- For a convex function $J(\mathbf{w})$ a local minimum is also a global minimum
- Asume that we found a critical point of $J(\mathbf{w})$, i.e.,

  $$\nabla_{\mathbf{w}} J(\mathbf{w}) = 0,$$

  where the Hessian is positive semidefinite

  $$\mathbf{H}_{\mathbf{x}} J(\mathbf{w}) \geq 0,$$

  then we found the global minimum of $J(\mathbf{w})$.

Gradient and Hessian Properties

Lemma 1. *We have*

$$\nabla_{\mathbf{x}}(\mathbf{b}^{\mathbf{T}}\mathbf{x}) = \mathbf{b}$$

$$\nabla_{\mathbf{x}}(\mathbf{x}^{\mathbf{T}}\mathbf{b}) = \mathbf{b}$$

Lemma 2. *If* $\mathbf{A}$ *is symmetric then*

$$\nabla_{\mathbf{x}}(\mathbf{x}^{\mathbf{T}}\mathbf{A}\mathbf{x}) = 2\mathbf{A}\mathbf{x}$$

$$\mathbf{H}_{\mathbf{x}}(\mathbf{x}^{\mathbf{T}}\mathbf{A}\mathbf{x}) = 2\mathbf{A}$$

10.4 The normal equation

We rewrite J in terms of matrix operations using

$$
\mathbf{X} = \begin{bmatrix} ... & \hat{\mathbf{x}}_1 & ... \\ ... & ... & ... \\ ... & \hat{\mathbf{x}}_m & ... \end{bmatrix}, \ \mathbf{y} = \begin{bmatrix} \hat{y}_1 \\ ... \\ \hat{y}_m \end{bmatrix}
$$

$$
J(\mathbf{w}) = \sum_{k=1}^{m} (h(\hat{\mathbf{x}}_k) - \hat{y}_k)^2 = (\mathbf{X}\mathbf{w} - \mathbf{y})^{\mathbf{T}}(\mathbf{X}\mathbf{w} - \mathbf{y}) = \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 .
$$

Theorem 1. *The global minimum of*

$$J(\mathbf{w}) = (\mathbf{X}\mathbf{w} - \mathbf{y})^{\mathbf{T}}(\mathbf{X}\mathbf{w} - \mathbf{y})$$

is attained for the weights given by

$$\mathbf{w} = (\mathbf{X}^{\mathbf{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathbf{T}}\mathbf{y}$$

It is known that the matrix $\mathbf{X^T X}$ is symmetric and positive semidefinite (i.e., has nonnegative eigenvalues). In order to make it invertible, it is sufficient to have all positive eigenvalues. If eventually $\mathbf{X^T X}$ has some zero eigenvalues, we can use a regularization $\mathbf{X^T X}+\varepsilon\mathbf{I}$, with $\mathbf{I}$ being the identity matrix of correct dimension and $\varepsilon > 0$. The matrix $\mathbf{X^T X}+\varepsilon\mathbf{I}$ is invertible and

$$\mathbf{w} = (\mathbf{X^T X} + \varepsilon\mathbf{I})^{-1}\mathbf{X^T y}$$

minimizes

$$J(\mathbf{w}) = \|\mathbf{Xw} - \mathbf{y}\|^2 + \varepsilon\|\mathbf{w}\|^2 .$$

Formal derivation

- We are trying to solve the problem

  $$\mathbf{Xw} = \mathbf{y}.$$

  Multiply from the left by $\mathbf{X^T}$ and we get the normal equation

  $$\mathbf{X^T X w} = \mathbf{X^T y}.$$

  Multiply from the left by $(\mathbf{X^T X})^{-1}$ and we get

  $$\mathbf{w} = (\mathbf{X^T X})^{-\mathbf{1}}\mathbf{X^T y}.$$

- This is only a formal derivation but easy to remember!

10.5 Statistical Interpretation

- We will predict house prices as

  $$\hat{y}_k = \mathbf{w^T}\hat{\mathbf{x}}_k + \varepsilon_k$$

  where the error is Normally distributed

  $$\varepsilon_k \sim N(0, \sigma^2).$$

  We define the likelyhood of the parameter value to be $\mathbf{w}$ based on the data as

  $$L(\mathbf{w}) = L(\mathbf{w}; \mathbf{X}, \mathbf{y}) = P(\mathbf{y}|\mathbf{X}, \mathbf{w}).$$

- We can write the likelihood as

  $$L(\mathbf{w}) = \prod_{k=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(\mathbf{w^T}\hat{\mathbf{x}}_k - \hat{y}_k\right)^2}{2\sigma^2}\right)$$

  The log likelihood is

  $$l(\mathbf{w}) = \ln L(\mathbf{w}) = C - \frac{1}{2\sigma^2}\sum_{k=1}^{m}\left(\mathbf{w^T}\hat{\mathbf{x}}_k - \hat{y}_k\right)^2 = C - \frac{1}{2\sigma^2}J(\mathbf{w})$$

  Maximum Likelihood Estimate (MLE) for $l(\mathbf{w})$ is exactly at the minimum of $J(\mathbf{w})$.

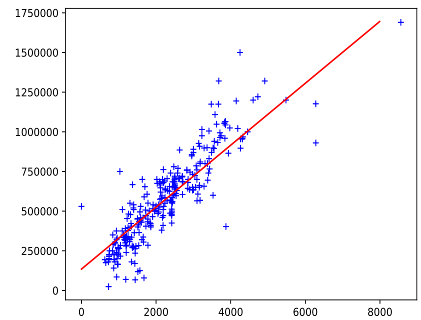

Figure 1: Estimation of home price based on living area

10.6 Examples

We can now solve the problem of estimating house prices

1. Polynomial approximation for given data. Assume you have to predict a function $y = f(x)$ using a polynomial of degree n.

$$h(x) = w_0 + w_1 x + ... + w_n x^n.$$

   Use least squares method to find a polynomial of best fit.

2. Application: yield to temperature dependence for a plants http://openmv.net/info/bioreactor-yields

3. Use a bivariate quadratic polynomial to fit a surface in $\mathbb{R}^3$

4. Project. Predict student loan debt at colleges based on data available at College scorecard data: https://collegescorecard.ed.gov/data/