Basics of Modeling

Miles Chen, PhD

Department of Statistics

Week 1 Wednesday

UCLA

Section 1

Models and Machine Learning

What is Machine Learning?

Wikipedia:

Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to effectively perform a specific task without using explicit instructions, relying on patterns and inference instead.

Big idea:

We fit models to data. People still have to make many decisions regarding the specification of the model, but a lot of decisions are guided by the data and the algorithms used. The decisions made by the algorithm is the "machine learning."

What is a model?

A model is a simplified representation of something complex.

A statistical model is a simplified representation of a (likely complex) real-world process that generates data.

Statistical Models

A statistical model is a simplified representation of a (likely complex) real-world process that generates data.

The process that generates data in the real world is complicated and there are many things that ultimately determine the outcome.

A statistical model is a simplified version of the relationships between variables in the real world. It includes only a subset of variables related to the outcome and always includes an error term to account for everything else.

“All models are wrong, but some are useful” - George Box

There are two primary uses for statistical models:

- We can use models to make accurate predictions.
- We can use models to gain understanding of the relationships between variables.

We often have to make a choice to prioritize one of these goals over the other.

An Example from the textbook

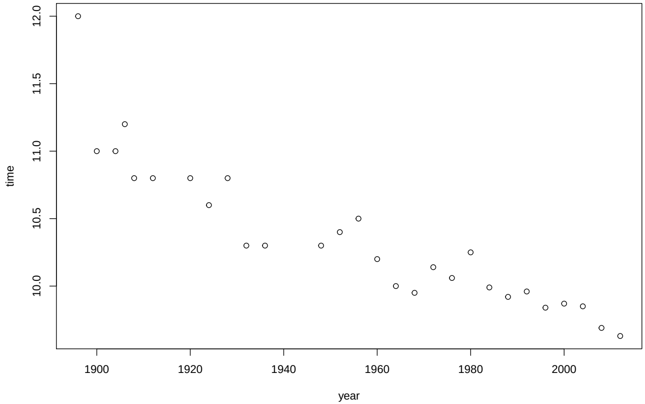
Data: Olympic Mens 100m race

outcome: The gold medal time at the Olympics for men's 100m track and field event.

Real world: There are many things that ultimately factor into the winning time.

We will make a simple model.

```r
url <- "https://raw.githubusercontent.com/sdrogers/fcmlcode/master/R/data/olympics/male100.csv
olympic <- read.csv(url, header = FALSE, col.names = c("year","time"))
plot(olympic)
```



A Linear Model (stuff you already know)

Our model is something like this: race times have gotten faster over time.

So if we know the year of the Olympic games, we can predict the winning race time.

A Linear model implies the amount by which we adjust our prediction remains a constant through time.

If I predict the time improves by 0.05 seconds from 1920 to 1924, I will also predict the time improves by 0.05 seconds from 2000 to 2004.

This will clearly not work for predicting values in the distant future (could have a negative time), but the model could still be useful

Fitting the model

The linear model fits a straight line to the data.

It has two parameters to estimate: slope and intercept.

Fitting a model is the process of estimating the model parameters.

We want to estimate the value of the parameters (slope and intercept) so the resulting model does the best job of fitting the data.

The objective function

When we fit a model to our data, we want the **best** parameter estimates. **How we define the word "best" is very important and is carried out through our choice of an objective function.**

For Ordinary Least Squares regression, we define *best* as the model that *minimizes the total squares of the residuals*. (residual = actual value - predicted value)

This objective is given by the following function:

$$RSS = \sum_{n=1}^{N} (y_n - \hat{y_n})^2$$

Because this *objective function* is one we try to minimize, we also call it a **loss function** or **cost function**. (In general people try to minimize their losses and costs.)

Machine Learning and Optimization

Many machine learning models (including linear regression) fit parameters by trying to minimize or maximize an objective function.

Thus **many machine learning problems can be reframed as optimization problems.**

Although linear regression is review for you, we start here to get you familiar with many of the ideas and concepts in Machine Learning.

I’ll try to match the textbook’s notation

In the text, the actual values of the data are denoted as t_n

The values predicted by the model (with paramters w_0, w_1) are $f(x_n; w_0, w_1)$.

The text also uses mean squared error for the loss function rather than total squared error. (As we are just scaling the total squared error by 1/N, the solution that minimizes MSE will be the same solution that minimizes RSS.)

So we can rewrite the loss function as

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^{n} (y_n - \hat{y_n})^2 = \frac{1}{N} \sum_{n=1}^{N} (t_n - f(x_n; w_0, w_1))^2$$

Side note: RMSE

A very common loss function the people optimize is the root mean squared error (RMSE):

$$RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(t_n - f(x_n))^2}$$

For Linear Regression, the parameters that optimize RSS are equal to the parameters that optimize MSE and RMSE.

(The square-root function is monotonic. So if $a > b$, then $\sqrt{a} > \sqrt{b}$ for all values $a > b > 0$. Thus the values that minimize MSE will be the same values that minimize RMSE.)

The linear model

The linear model has two parameters:

- intercept: w_0
- slope: w_1

The value predicted by the model is

$$f(x_n; w_0, w_1) = w_0 + w_1 x_n$$

If we plug this into the Loss function, we get:

$$\mathcal{L} = \frac{1}{N}\sum_{n=1}^{N}(t_n - f(x_n; w_0, w_1))^2 = \frac{1}{N}\sum_{n=1}^{N}(t_n - (w_0 + w_1 x_n))^2$$

Minimizing the loss function with Calculus

We treat our observed data (the x and t) as fixed. The value of our Loss function depends on the two model parameters: w_0 and w_1. We want to find the values of the arguments w_0 and w_1 which will minimize the loss function for our observed data. These estimates are called $\hat{w}_0$ and $\hat{w}_1$.

$$\hat{w}_0, \hat{w}_1 = \underset{w_0, w_1}{\arg\min} \frac{1}{N} \sum_{n=1}^{N} (t_n - (w_0 + w_1 x_n))^2$$

To accomplish this we'll start by expanding the Loss function:

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^{N} (w_1^2 x_n^2 + 2 w_1 x_n (w_0 - t_n) + w_0^2 - 2 w_0 t_n + t_n^2)$$

We'll then find the partial derivatives with respect to w_0 and w_1. We set those partial derivatives equal to zero, and solve for w_0 and w_1.

Minimizing the loss function with Calculus

$$\frac{\partial \mathcal{L}}{\partial w_1} = 2w_1 \frac{1}{N}\left(\sum_{n=1}^{N} x_n^2\right) + \frac{2}{N}\left(\sum_{n=1}^{N} x_n(w_0 - t_n)\right)$$

$$\frac{\partial \mathcal{L}}{\partial w_0} = 2w_0 + 2w_1 \frac{1}{N}\left(\sum_{n=1}^{N} x_n\right) - \frac{2}{N}\left(\sum_{n=1}^{N} t_n\right)$$

We set the partial derivatives equal to 0 and solve for w_0 and w_1 to get our estimates $\hat{w}_0$ and $\hat{w}_1$

I skipped a bunch of steps for these slides. The full details of the work is provided in section 1.1.4 in the text.

Linear Algebra

Because we are dealing with multiple observations, we have a summation term in the Loss Function and a summation term in the derivatives, which can become annoying to deal with. Using matrices helps as the summation is implicit in the matrix multiplication.

If it's been a while since you took Linear Algebra, you'll want to spend some time reviewing it.

I highly recommend the following Youtube playlist from 3Blue1Brown (a channel that everyone should subscribe to):

https://www.youtube.com/playlist?list=PLZHQObOWTQDPD3MizzM2xVFitgF8hE_ab

Section 2

OLS regression with Matrix Notation

OLS regression with Matrix Notation

We define our column matrix of parameters $\mathbf{w}$:

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$

A single observation x_n is:

$$\mathbf{x}_n = \begin{bmatrix} 1 \\ x_n \end{bmatrix}$$

And thus the value predicted by the model can be written as:

$$f(x_n; w_0, w_1) = w_0 + w_1 x_n = \mathbf{w}^T \mathbf{x}_n$$

The matrix representing all the input data $\mathbf{X}$ and the column matrix of outcomes $\mathbf{t}$ are:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_N^T \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_N \end{bmatrix}, \mathbf{t} = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{bmatrix}$$

With these matrices, we can show that the Loss function becomes:

$$\mathcal{L} = \frac{1}{N}(\mathbf{t} - \mathbf{X}\mathbf{w})^T(\mathbf{t} - \mathbf{X}\mathbf{w})$$

The column matrix of values predicted by the function $f(x_n; w_0, w_1)$ are:

$$\mathbf{X}\mathbf{w} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_N \end{bmatrix} \times \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = \begin{bmatrix} w_0 + w_1 x_1 \\ w_0 + w_1 x_2 \\ \vdots \\ w_0 + w_1 x_N \end{bmatrix}$$

Sometimes it can be confusing whether to write $\mathbf{X}\mathbf{w}$ or $\mathbf{w}\mathbf{X}$. To make sure I have the order of matrices right, I like to write matrix dimensions underneath the matrices:

$$\underset{N\times 2}{\mathbf{X}} \times \underset{2\times 1}{\mathbf{w}} = \underset{N\times 1}{\mathbf{X}\mathbf{w}}$$

When doing matrix multiplication, the inner dimensions must always match.

Side note: Matrix Operations in R

```r
x <- olympic$year
N = length(x)
ones <- rep(1, N)
X <- cbind(ones, x)
dim(X)
```

```
## [1] 28  2
```

```r
w <- matrix(1, nrow = 2, ncol = 1)
dim(w)
```

```
## [1] 2 1
```

Side note: Matrix Operations in R

```r
dim(X %*% w)
```

```
## [1] 28  1
```

```r
dim(w %*% X)
```

```
## Error in w %*% X: non-conformable arguments
```

This error means that the dimensions don't align properly to do a matrix operation. You can often figure out the problem by writing out the matrices by hand and writing the dimensions underneath them to make sure they align properly.

Back to OLS regression with Matrix Notation

The column matrix of residual is:

$$\mathbf{t} - \mathbf{X}\mathbf{w} = \begin{bmatrix} t_1 - (w_0 + w_1 x_1) \\ t_2 - (w_0 + w_1 x_2) \\ \vdots \\ t_N - (w_0 + w_1 x_N) \end{bmatrix}$$

$$\underset{N\times 1}{\mathbf{t}} - \underset{N\times 1}{\mathbf{X}\mathbf{w}} = \underset{N\times 1}{(\mathbf{t} - \mathbf{X}\mathbf{w})}$$

The sum of squared errors can be expressed as:

$$
\begin{aligned}
&= \sum_{n=1}^{N} (t_n - (w_0 + w_1 x_n))^2 \\
&= \begin{bmatrix} t_1 - (w_0 + w_1 x_1) & t_2 - (w_0 + w_1 x_2) & \cdots & t_N - (w_0 + w_1 x_N) \end{bmatrix} \times \begin{bmatrix} t_1 - (w_0 + w_1 x_1) \\ t_2 - (w_0 + w_1 x_2) \\ \vdots \\ t_N - (w_0 + w_1 x_N) \end{bmatrix} \\
&= (\mathbf{t} - \mathbf{X}\mathbf{w})^T (\mathbf{t} - \mathbf{X}\mathbf{w})
\end{aligned}
$$

We can now write our loss function compactly with matrix notation, which produces a scalar (1x1) value:

$$
\underset{1\times 1}{\frac{1}{N}} \underset{1\times N}{(\mathbf{t} - \mathbf{X}\mathbf{w})^T} \underset{N\times 1}{(\mathbf{t} - \mathbf{X}\mathbf{w})} = \underset{1\times 1}{\mathcal{L}}
$$

We can expand the sum of squared errors as follows:

$$
\begin{aligned}
&= (\mathbf{t} - \mathbf{X}\mathbf{w})^T(\mathbf{t} - \mathbf{X}\mathbf{w}) \\
&= (\mathbf{t}^T - (\mathbf{X}\mathbf{w})^T)(\mathbf{t} - \mathbf{X}\mathbf{w}) \\
&= (\mathbf{t}^T - \mathbf{w}^T\mathbf{X}^T)(\mathbf{t} - \mathbf{X}\mathbf{w}) \\
&= \underset{(1\times N)\times(N\times 1)}{\mathbf{t}^T\mathbf{t}} - \underset{(1\times N)\times(N\times 2)\times(2\times 1)}{\mathbf{t}^T\mathbf{X}\mathbf{w}} - \underset{(1\times 2)\times(2\times N)\times(N\times 1)}{\mathbf{w}^T\mathbf{X}^T\mathbf{t}} + \underset{(1\times 2)\times(2\times N)\times(N\times 2)\times(2\times 1)}{\mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w}} \\
&= \underset{(1\times 1)}{\mathbf{t}^T\mathbf{t}} - \underset{(1\times 1)}{\mathbf{t}^T\mathbf{X}\mathbf{w}} - \underset{(1\times 1)}{\mathbf{w}^T\mathbf{X}^T\mathbf{t}} + \underset{(1\times 1)}{\mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w}} \\
&= \mathbf{t}^T\mathbf{t} - 2\mathbf{w}^T\mathbf{X}^T\mathbf{t} + \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w}
\end{aligned}
$$

We can include $1/N$ to return it to our Loss function:

$$\mathcal{L} = \frac{1}{N}(\mathbf{t}^T\mathbf{t} - 2\mathbf{w}^T\mathbf{X}^T\mathbf{t} + \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w})$$

On the previous slide, the terms $\mathbf{t}^T\mathbf{X}\mathbf{w}$ and $\mathbf{w}^T\mathbf{X}^T\mathbf{t}$ are transposes of each other and are scalars. So they are the same value and can be combined.

Minimizing the Loss with vector calculus

Now that we have defined our loss function in matrix notation, we still want to find the value of $\mathbf{w}$ that will minimize the loss. Like before, this involves taking the derivative (now with respect to the vector $\mathbf{w}$), setting the derivative equal to 0, and then solving for $\mathbf{w}$.

Thus, we must delve into doing vector Calculus.

Review: Derivatives

Let's say we have a function of one variable that produces a scalar output (real-valued input is mapped to real-valued output):

$$f : \mathbb{R} \to \mathbb{R}$$

The derivative of this function tells us how the function changes with respect to that variable. The derivative is:

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

Which can be rearranged to say that the derivative at a $f'(x)$ satisfies the following:

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

The Gradient

Let's say we have a function of a vector that produces a scalar output (n-dimensional real-valued input is mapped to one real-valued output):

$$f : \mathbb{R}^n \to \mathbb{R}$$

The gradient $\nabla f(\mathbf{x})$ is the multidimensional derivative and it tells us how the function changes with respect to each component in the vector. It satisfies the following equation:

$$\lim_{\|\mathbf{h}\| \to 0} \frac{\|f(\mathbf{x}+\mathbf{h}) - f(\mathbf{x}) - \nabla f(\mathbf{x}) \cdot \mathbf{h}\|}{\|\mathbf{h}\|} = \mathbf{0}$$

The Gradient

Again, we have a function of a vector that produces a scalar ($f : \mathbb{R}^n \to \mathbb{R}$).

The input to the function is a vector:

$$\mathbf{w} = (w_1, w_2, ...w_n)^T$$

The gradient of the function with respect to $\mathbf{w}$ at $\mathbf{w}$ is itself a vector:

$$\nabla f(\mathbf{w}) = \frac{\partial f}{\partial \mathbf{w}} = \left(\frac{\partial f}{\partial w_1}, \frac{\partial f}{\partial w_2}, \cdots, \frac{\partial f}{\partial w_n} \right)^T$$

The Gradient

Like the derivative, the gradient represents the slope of the tangent of the graph of the function. More precisely, the gradient points in the direction of the greatest rate of increase of the function, and its magnitude is the slope of the graph in that direction.

Useful gradient relations

Important: The gradient will always have the same dimensions as the input vector.

In the following examples, $\mathbf{w}$ and $\mathbf{w}$ are 2 x 1 matrices.

$\underset{(1\times1)}{f(\mathbf{w})}$	$\frac{\partial f}{\partial \mathbf{w}}$
$\mathbf{w}^\mathbf{T}\mathbf{x}$ $(1\times2)\times(2\times1)$	$\mathbf{x}$ (2×1)
$\mathbf{x}^\mathbf{T}\mathbf{w}$ $(1\times2)\times(2\times1)$	$\mathbf{x}$ (2×1)
$\mathbf{w}^\mathbf{T}\mathbf{w}$ $(1\times2)\times(2\times1)$	$\mathbf{2w}$ (2×1)
$\mathbf{w}^\mathbf{T}\mathbf{C}\mathbf{w}$ $(1\times2)\times(2\times2)\times(2\times1)$	$\mathbf{2Cw}$ $(2\times2)\times(2\times1)$

Back to OLS Regression Loss (Matrix notation)

Earlier, we found the total OLS regression loss function to be:

$$\mathcal{L} = \frac{1}{N}\mathbf{t}^T\mathbf{t} - \frac{2}{N}\mathbf{w}^T\mathbf{X}^T\mathbf{t} + \frac{1}{N}\mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w}$$

We will take the derivative of the loss with respect to the vector $\mathbf{w}$, set the derivative equal to the 0 vector, and then solve for $\mathbf{w}$.

If $\mathbf{w}$ is a 2 x 1 vector, then ∇f must also be 2 x 1.

$$\begin{aligned}\underset{(2\times 1)}{\frac{\partial \mathcal{L}}{\partial \mathbf{w}}} &= \underset{(2\times 1)}{\mathbf{0}} - \underset{(2\times N)\times(N\times 1)}{\frac{2}{N}\mathbf{X}^T\mathbf{t}} + \underset{(2\times N)\times(N\times 2)\times(2\times 1)}{\frac{2}{N}\mathbf{X}^T\mathbf{X}\mathbf{w}} \\ &= \underset{(2\times 1)}{\frac{-2}{N}\mathbf{X}^T\mathbf{t}} + \underset{(2\times 1)}{\frac{2}{N}\mathbf{X}^T\mathbf{X}\mathbf{w}}\end{aligned}$$

Solving for the $\mathbf{w}$ that minimizes the OLS Loss

Set the gradient equal to 0 and solve for $\mathbf{w}$:

$$\begin{aligned}
\mathbf{0} &= \frac{-2}{N}\mathbf{X}^T\mathbf{t} + \frac{2}{N}\mathbf{X}^T\mathbf{X}\mathbf{w} \\
\mathbf{X}^T\mathbf{t} &= \mathbf{X}^T\mathbf{X}\mathbf{w} \\
\underset{(2\times N)(N\times 2)(2\times N)(N\times 1)}{(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{t}} &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{X}\mathbf{w} \\
\underset{(2\times 1)}{(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{t}} &= \mathbf{w}
\end{aligned}$$

OLS Result

The estimate of $\mathbf{w}$ that minimizes the loss function of OLS regression is $\hat{\mathbf{w}}$:

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