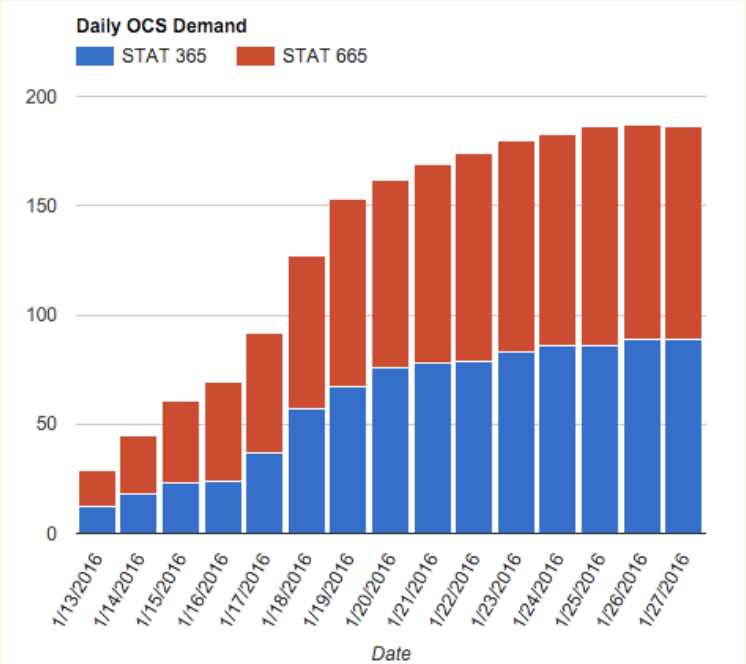
# Lecture 04
# Least squares and classification

27 January 2016

Taylor B. Arnold
Yale Statistics
STAT 365/665

Yale

- Problem set notes:
  - brute force okay for implementation question
  - can use other libraries in the prediction and data analysis questions
  - consider **FNN** (R) or or **sklearn.neighbors** (Python)
- Office hours (more possibly to come):
  - Taylor Arnold  Mondays, 13:00 - 14:15, HH 24, Office 206 (by appointment)
  - Yu Lu  Tuesdays, 10:00-12:00, HH 24, Library
  - Jason Klusowski  Thursdays, 19:00-20:30, HH 24


Daily OCS Demand
STAT 365
STAT 665
200
150
100
50
0
1/13/2016
1/14/2016
1/15/2016
1/16/2016
1/17/2016
1/18/2016
1/19/2016
1/20/2016
1/21/2016
1/22/2016
1/23/2016
1/24/2016
1/25/2016
1/26/2016
1/27/2016
Date

## Ordinary least squares

The multivariate linear regression model is given by:

$$y_i = x_{1,i}\beta_1 + x_{2,i}\beta_2 + \cdots + x_{1,p}\beta_p + \epsilon_i$$

A sample can be re-written in terms of the vector $x_i$ (the vector of covariates for a single observation):

$$y_i = x_i^t \beta + \epsilon_i$$

In matrix notation, we can write the linear model simultaneously for all observations:

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} x_{1,1} & x_{2,1} & \cdots & x_{p,1} \\ x_{1,2} & \ddots & & x_{p,2} \\ \vdots & & \ddots & \vdots \\ x_{1,n} & x_{2,n} & \cdots & x_{p,n} \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{pmatrix}$$

In matrix notation, we can write the linear model simultaneously for all observations:

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} x_{1,1} & x_{2,1} & \cdots & x_{p,1} \\ x_{1,2} & \ddots & & x_{p,2} \\ \vdots & & \ddots & \vdots \\ x_{1,n} & x_{2,n} & \cdots & x_{p,n} \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{pmatrix}$$

Which can be compactly written as:

$$y = X\beta + \epsilon$$

For reference, note the following equation

$$y = X\beta + \epsilon$$

Yields these dimensions:

$$y \in \mathbb{R}^n$$
$$X \in \mathbb{R}^{n \times p}$$
$$\beta \in \mathbb{R}^p$$
$$\epsilon \in \mathbb{R}^n$$

## Least squares

To estimate the least squares solution, which is again the MLE for independent normal errors, we see that:

$$\widehat{\beta} \in \underset{b \in \mathbb{R}^p}{\arg\min} \left\{ ||y - Xb||_2^2 \right\}$$

It will be helpful to re-write the sum of squares as:

$$||y - X\beta||_2^2 \quad = \quad (y - X\beta)^t(y - X\beta)$$

It will be helpful to re-write the sum of squares as:

$$\begin{aligned} ||y - X\beta||_2^2 &= (y - X\beta)^t(y - X\beta) \\ &= (y^t - \beta^t X^t)(y - X\beta) \end{aligned}$$

It will be helpful to re-write the sum of squares as:

$$\begin{aligned} ||y - X\beta||_2^2 &= (y - X\beta)^t(y - X\beta) \\ &= (y^t - \beta^t X^t)(y - X\beta) \\ &= y^t y - y^t X\beta - \beta^t X^t y + \beta^t X^t X\beta \end{aligned}$$

It will be helpful to re-write the sum of squares as:

$$\begin{aligned} ||y - X\beta||_2^2 &= (y - X\beta)^t(y - X\beta) \\ &= (y^t - \beta^t X^t)(y - X\beta) \\ &= y^t y - y^t X\beta - \beta^t X^t y + \beta^t X^t X\beta \\ &= y^t y - 2y^t X\beta + \beta^t X^t X\beta \end{aligned}$$

## Normal Equations

In order to find the minimum of the sum of squares, we take the gradient with respect to $\beta$ and set it equal to zero.

Recall that, for a vector $a$ and symmetric matrix $A$ :

$$\nabla_\beta a^t \beta = a$$
$$\nabla_\beta \beta^t A \beta = 2A\beta$$

## Normal Equations

In order to find the minimum of the sum of squares, we take the gradient with respect to $\beta$ and set it equal to zero.

Recall that, for a vector $a$ and symmetric matrix $A$ :

$$\nabla_\beta a^t \beta = a$$
$$\nabla_\beta \beta^t A \beta = 2A\beta$$

This gives the gradient of the sum of squares as:

$$\begin{aligned}\nabla_\beta ||y - X\beta||_2^2 &= \nabla_\beta \left(y^t y - 2y^t X\beta + \beta^t X^t X\beta\right) \\ &= 2X^t X\beta - 2X^t y\end{aligned}$$

Setting this equal to zero gives a set of $p$ equations called the normal equations:

$$X^t X\widehat{\beta} = X^t y$$

## Maximum or Minimum?

To determine whether the normal equations give a local minimum, maximum, or saddle point, we can calculate the Hessian matrix.

## Maximum or Minimum?

To determine whether the normal equations give a local minimum, maximum, or saddle point, we can calculate the Hessian matrix. This is a $p \times p$ matrix giving every combination of the second partial derivatives:

$$Hf(\beta) = \begin{pmatrix} \frac{\partial^2 f}{\partial \beta_1 \partial \beta_1} & \frac{\partial^2 f}{\partial \beta_1 \partial \beta_2} & \cdots & \frac{\partial^2 f}{\partial \beta_1 \partial \beta_p} \\ \frac{\partial^2 f}{\partial \beta_2 \partial \beta_1} & \ddots & & \frac{\partial^2 f}{\partial \beta_2 \partial \beta_p} \\ \vdots & & \ddots & \vdots \\ \frac{\partial^2 f}{\partial \beta_p \partial \beta_1} & \frac{\partial^2 f}{\partial \beta_p \partial \beta_2} & \cdots & \frac{\partial^2 f}{\partial \beta_p \partial \beta_p} \end{pmatrix}$$

If the Hessian is positive definite ($x^t H x \geq 0$) at a critical point, then the critical point is a local minimum.

Looking at the gradient of the sum of squares:

$$\nabla_\beta ||y - X\beta||_2^2 = 2X^t X\beta - 2X^t y$$

Looking at the gradient of the sum of squares:

$$\nabla_\beta ||y - X\beta||_2^2 = 2X^t X\beta - 2X^t y$$

We can see that the Hessian is simply:

$$H_\beta ||y - X\beta||_2^2 = 2X^t X$$

Looking at the gradient of the sum of squares:

$$\nabla_\beta ||y - X\beta||_2^2 = 2X^t X\beta - 2X^t y$$

We can see that the Hessian is simply:

$$H_\beta ||y - X\beta||_2^2 = 2X^t X$$

Why is this positive definite?

Looking at the gradient of the sum of squares:

$$\nabla_\beta ||y - X\beta||_2^2 = 2X^tX\beta - 2X^ty$$

We can see that the Hessian is simply:

$$H_\beta ||y - X\beta||_2^2 = 2X^tX$$

Why is this positive definite?

$$\begin{aligned} v^t \left(2X^tX\right) v &= 2\left(v^tX^tXv\right) \\ &= 2||Xv||_2^2 \\ &\geq 0 \end{aligned}$$

Back to the normal equations themselves, notice that if the matrix $X^tX$ is invertible, we can 'solve' the normal equations as:

$$\begin{aligned} X^tX\widehat{\beta} &= X^ty \\ \widehat{\beta} &= (X^tX)^{-1}X^ty \end{aligned}$$

Back to the normal equations themselves, notice that if the matrix $X^tX$ is invertible, we can 'solve' the normal equations as:

$$X^tX\widehat{\beta} = X^ty$$
$$\widehat{\beta} = (X^tX)^{-1}X^ty$$

This is not a good way to solve the normal equations numerically, but is a useful theoretical form.

## Ridge regression

The ridge regression estimator is the solution to the following modified least squares optimization problem for some value of $\lambda > 0$.

$$\widehat{\beta}_{ridge} = \arg\min_{b} \left\{ ||y - Xb||_2^2 + \lambda ||b||_2^2 \right\}$$

## Why the ridge penalty?

1. The equation shrinks the coefficients towards zero, adding some bias but reducing the variance of the estimator.

## Why the ridge penalty?

1. The equation shrinks the coefficients towards zero, adding some bias but reducing the variance of the estimator.
2. Using the $\ell_2$-norm keeps the equation rotationally invariant.

## Why the ridge penalty?

1. The equation shrinks the coefficients towards zero, adding some bias but reducing the variance of the estimator.
2. Using the $\ell_2$-norm keeps the equation rotationally invariant.
3. Ridge regression has an analytical solution.

To see this write the criterion as a matrix equation:

$$(y - Xb)^t(y - Xb) + \lambda b^t b = y^t y + b^t X^t X b - 2y^t X b + \lambda b^t b$$

To see this write the criterion as a matrix equation:

$$(y - Xb)^t(y - Xb) + \lambda b^t b = y^t y + b^t X^t X b - 2y^t X b + \lambda b^t b$$

And take its derivative:

$$\frac{\partial}{\partial \beta}\left(y^t y + b^t X^t X b - 2y^t X b + \lambda b^t b\right) = 2X^t X b - 2X^t y + 2\lambda b$$

Setting this to zero yields

$$
\begin{aligned}
2X^tX\widehat{\beta} + 2\lambda\widehat{\beta} &= 2X^ty \\
(X^tX + I_p\lambda)\widehat{\beta} &= X^ty \\
\widehat{\beta} &= (X^tX + I_p\lambda)^{-1} \cdot X^ty
\end{aligned}
$$

Setting this to zero yields

$$
\begin{aligned}
2X^tX\widehat{\beta} + 2\lambda\widehat{\beta} &= 2X^ty \\
(X^tX + I_p\lambda)\widehat{\beta} &= X^ty \\
\widehat{\beta} &= (X^tX + I_p\lambda)^{-1} \cdot X^ty
\end{aligned}
$$

This is a useful analytical form, though as with least squares we would generally not invert the matrix directly but instead use a stable matrix decomposition.

## Computational issues

How can we estimate the regression vector using a technique such as ordinary least squares

$$\widehat{\beta}_{ols} = \arg\min_{b} \left\{ \sum_{i=1}^{n} (y_i - x_i^t b)^2 \right\},$$

When we have a dataset size grows larger than the available memory?

At first glance this seems computationally very difficult as we are trying to minimize a summation with one component per observation.

However, recall that the ordinary least squares solution can be computed by:

$$\widehat{\beta}_{ols} = (X^t X)^{-1} X^t y.$$

Now, assume that the data matrix $X$ is broken by rows into $K$ different chunks:

$$X = \begin{pmatrix} X_{B_1} \\ X_{B_2} \\ \vdots \\ X_{B_K} \end{pmatrix}$$

Now, assume that the data matrix $X$ is broken by rows into $K$ different chunks:

$$X = \begin{pmatrix} X_{B_1} \\ X_{B_2} \\ \vdots \\ X_{B_K} \end{pmatrix}$$

The Gram matrix $X^t X$ can then be computed by summing up the Gram matrices of the individual chunks:

$$X^t X = \sum_{i=1}^{K} X_{B_i}^t X_{B_i}.$$

If the vector $y$ is broken in to the same set of chunks, and corresponding blocks are stored next to one another, such as:

$$X = \begin{pmatrix} X_{B_1} \\ X_{B_2} \\ \vdots \\ X_{B_K} \end{pmatrix}, \quad y = \begin{pmatrix} y_{B_1} \\ y_{B_2} \\ \vdots \\ y_{B_K} \end{pmatrix}$$

The exact same technique works for computing the correlations $X^t y$.

$$X^t y = \sum_{i=1}^{K} X_{B_i}^t y_{B_i}.$$

So, we can compute the least squares solution:

$$\widehat{\beta}_{ols} = (X^t X)^{-1} X^t y$$

by only working with chunks of the data. Specifically:

$$\widehat{\beta}_{ols} = \left( \sum_{i=1}^{K} X_{B_i}^t X_{B_i} \right)^{-1} \cdot \left( \sum_{i=1}^{K} X_{B_i}^t y_{B_i} \right)$$

This means that we can either work in parallel or with a single process that only reads a small chunk of the data into memory at any given time.

The whole operation only requires, at most, memory for and transmission of $K \cdot (p^2 + p)$ values.

By applying the summation iteratively via *folds*, this can be done by only holding $2(p^2 + p)$ values in memory.

More generally, the following 6 quantities can be easily computed over chunks of the data set:

$$\begin{aligned}
\text{Gram matrix} &= X^t X \\
\text{correlation vector} &= X^t y \\
\text{column sums} &= X^t \mathbf{1}_n \\
\text{response sums} &= y^t \mathbf{1}_n \\
\text{response variance} &= y^t y \\
\text{sample size} &= \mathbf{1}_n^t \mathbf{1}_n
\end{aligned}$$

It turns out that these alone are sufficient to calculate many classical and modern estimation techniques.

**Ordinary least squares**

Not only can we solve ordinary least squares,

$$||y - X\beta||_2^2 = y^t y + \beta^t X^t X \beta + 2 y^t X \beta,$$

But we can also calculate an estimator of the noise variance:

$$\widehat{\sigma^2} = \frac{1}{n-p}\left(y^t y + \beta^t X^t X \beta + 2 y^t X \beta\right)$$

And compute standard errors:

$$\text{S.E}(\widehat{\beta}_j) = \sqrt{\widehat{\sigma^2}(X^t X)^{-1}_{jj}}$$

**Ridge regression**

The objective function in Ridge regression is also easily written in terms of these quantities:

$$||y - X\beta||_2^2 + \lambda||\beta||_2^2 = y^t y + \beta^t X^t X \beta + 2y^t X\beta + \lambda\beta^t\beta$$

With similar formulas for standard errors.

## Classification problems

Many (most?) of the tasks we'll consider this semester are actually classification tasks. That is, the values $y_i$ that we are trying to predict are class labels rather than a continuous response.

## Classification problems, cont.

When we have two classes, we can encode these with numerical values. For example, either

$$y_i = \begin{cases} 0 \\ 1 \end{cases}$$

Or,

$$y_i = \begin{cases} -1 \\ +1 \end{cases}$$

## Classification problems, cont.

Methods for continuous responses can be used as is (whether the theory extends to discrete data is a different matter entirely). The values of $\widehat{y}$ can be interpreted as probabilities. If we need to estimate the actually class label of $y$, we can use some threshold:

$$\widehat{y}_i^{class} = \begin{cases} 0, & \widehat{y}_i < \alpha \\ 1, & \text{else} \end{cases}$$

For some cutoff value $\alpha$.

## Classification problems, cont.

Methods for continuous responses can be used as is (whether the theory extends to discrete data is a different matter entirely). The values of $\widehat{y}$ can be interpreted as probabilities. If we need to estimate the actually class label of $y$, we can use some threshold:

$$\widehat{y}_i^{class} = \begin{cases} 0, & \widehat{y}_i < \alpha \\ 1, & \text{else} \end{cases}$$

For some cutoff value $\alpha$.

Some classification methods such as knn can directly produce class estimates; for example, simply using whichever class label is most common near $x_{new}$.

## Classification problems, cont.

How can we evaluate how well a classification algorithm works, say when doing cross validation? Mean squared error on the predicted probabilities can work well in many cases; using statistical deviance (based on the log-likelihood) can also make sense in certain contexts.

Another method, which we will largely use in this course, is to instead evaluate the actually class predictions themselves using the **misclassification rate**. On the validation set this can be written as:

$$MCR(k) = \frac{\#\left\{\widehat{y}_i^{class} \neq y_i\right\}_{i \in V}}{\#V}$$

## Multiclass classification

It will also be common that we will have values of $y_i$ that can come from more than two classes. For example, tagging a word as a part of speech or labeling an image in the CIFAR-10 corpus.

## Multiclass classification

It will also be common that we will have values of $y_i$ that can come from more than two classes. For example, tagging a word as a part of speech or labeling an image in the CIFAR-10 corpus.

A few techniques such as knn can directly solve the multiclass problem, but unlike the two class problem, we cannot directly code the multiclass problem as a regression problem. There are two generic methods for overcoming this deficiency: one-vs.-rest and one-vs.-one classification.

## Multiclass classification

It will also be common that we will have values of $y_i$ that can come from more than two classes. For example, tagging a word as a part of speech or labeling an image in the CIFAR-10 corpus.

A few techniques such as knn can directly solve the multiclass problem, but unlike the two class problem, we cannot directly code the multiclass problem as a regression problem. There are two generic methods for overcoming this deficiency: one-vs.-rest and one-vs.-one classification.

Like classification, there are several techniques for evaluating the fit of a multiclass problem. Misclassification rate is also again the simplest and can be defined exactly as it was before.