Introduction to Optimization

Econ 5970: Data Science for Economists
Prof. Tyler Ransom
University of Oklahoma

Spring 2018

1 What is "optimization"?

The word "optimization" generally means "the action of making the best or most effective use of a situation or resource" (Google's dictionary). In data science, this could refer to two different things:

1. Streamlining some sort of automated process
   (a) e.g. "We *optimized* the web scraping script so that it now runs in half the time and has 40% fewer lines of code."
2. Finding the optimum of an objective function (or "selecting the best element from some set of available alternatives," according to Wikipedia)
   (a) e.g. "Our preferred model estimate is the set of parameters that *optimizes* the sum of the squared distance between the data and the model." [in this case, "optimize" would mean "minimize"]

The **second** of these two definitions is what you should think of when you hear "optimization" in a data science context.

2 Ways to optimize

There are many different ways we can optimize a particular objective function. The two most common ways—which we will discuss today—are known as **minimum distance estimation** and **maximum likelihood estimation**. These are two different but related ways of expressing the objective function.

In any optimization problem, the data scientist must also choose an *algorithm* with which to optimize, if the optimum cannot be expressed in a math formula. The three most popular optimization algorithms are:

1. gradient descent

2. stochastic gradient descent

3. L-BFGS

We'll talk about these algorithms a bit later in the course.

3 Math: the classical linear model

Let's suppose we have the classical linear model:

$$y_i = x_i'\beta + \varepsilon_i \tag{1}$$

where y_i is the outcome for observation (equiv. training example) i, x_i are the covariates (equiv. features) of observation i, β are parameters to be estimated, and ε_i is the error term.

We can rewrite (1) in vector form:

$$y = X\beta + \varepsilon \tag{2}$$

where now y is a vector (i.e. column of data), X is a matrix (i.e. a table of data), β are exactly the same as before, and ε is an (unobservable) error term vector.

3.1 Optimization method 1: least squares

We can define our objective function so that we minimize the (squared) distance between $X\beta$ and y. In this sense, minimizing the distance will give us the best fit (i.e. optimum) of the model given the data. In formal terms, we want to

$$\begin{aligned} &\min_\beta \sum_i \left(y_i - x_i'\beta\right)^2 \\ =&\min_\beta \sum_i \varepsilon_i^2 \\ =&\min_\beta \varepsilon'\varepsilon \\ =&\min_\beta \left(y - X\beta\right)' \left(y - X\beta\right) \end{aligned} \tag{3}$$

Note that the four equations above are all equivalent. We can conserve on notation by utilizing vector and matrix notation (e.g. $\varepsilon'\varepsilon$ is the dot product of the ε vector with itself).[1]

[1] A related objective function to the one above is to minimize the *absolute value* of the distance between y and $X\beta$. This model is known as *least absolute deviations* and the parameter estimates at the optimum correspond to the *median regression* model rather than the OLS regression model (the linear prediction of which passes through the mean).

3.1.1 Using calculus to solve for the least squares estimator

We can solve this optimization problem using calculus:

$$\begin{aligned}
&\min_{\beta} (y - X\beta)' (y - X\beta) \\
=&\min_{\beta} \left(y'y - \beta' X'y - y'X\beta - \beta' X'X\beta\right)
\end{aligned}$$

How do we use calculus to find an optimum? Take first-order conditions and set them equal to zero:

$$\begin{aligned}
[\beta] : &- \frac{\partial}{\partial \beta} \beta' X'y - \frac{\partial}{\partial \beta} y'X\beta - \frac{\partial}{\partial \beta} \beta' X'X\beta \\
0 = &- X'y - X'y - 2\left(X'X\right)\hat{\beta} \\
0 = &- 2X'y - 2\left(X'X\right)\hat{\beta} \\
X'y = &\left(X'X\right)\hat{\beta} \\
\hat{\beta} = &\left(X'X\right)^{-1} X'y
\end{aligned} \tag{4}$$

which is known as the *OLS estimator*.

3.1.2 Checking second-order conditions

How do we know if our optimum is a minimum or a maximum? We need to check that the second-order conditions are satisfied. For a *minimum*, we need the second derivative to be *positive*. Taking the derivative of our first-order condition with respect to beta, we get:

$$\begin{aligned}
0 < &\frac{\partial^2}{\partial \beta \partial \beta'} (y - X\beta)' (y - X\beta) \\
0 < &\frac{\partial}{\partial \beta'} \left[-X'y - X'y - 2\left(X'X\right)\hat{\beta}\right] \\
0 < &-2\left(X'X\right) \\
\left(X'X\right) > &0
\end{aligned} \tag{5}$$

This condition is satisfied so long as the following conditions hold:

1. X has more rows than columns (i.e. $N \geq K$, where N is the number of observations and K is the number of columns in X)
2. X cannot have perfectly collinear columns

In linear algebra parlance, if these conditions are met, then we say that the matrix $(X'X)$ has *full rank*, or that $(X'X)$ is *nonsingular*.

3.2 Optimization method 2: maximum likelihood

Another option for finding the optimum of our linear model is by using what is called maximum likelihood. Let's review what maximum likelihood is and then derive it for this simple linear model.

3.2.1 Maximum likelihood estimation of the mean and variance of the normal distribution

Suppose we have N observations of a random variable, call it X, which we know to be independently and identically distributed (*iid*) according to the Normal distribution. This means that, for any given observation x_i, we will have the following probability density function:

$$f\left(x_i; \mu, \sigma\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\left(x_i - \mu\right)^2}{2\sigma^2}\right) \tag{6}$$

where we put μ, σ on the right hand side of the semicolon to denote that we are interested in estimating these parameters from our N observations of variable X.

Why do we care about the probability density function? We want to use our data to find what the mean and variance would be if we assume that our collected data follows a normal distribution.

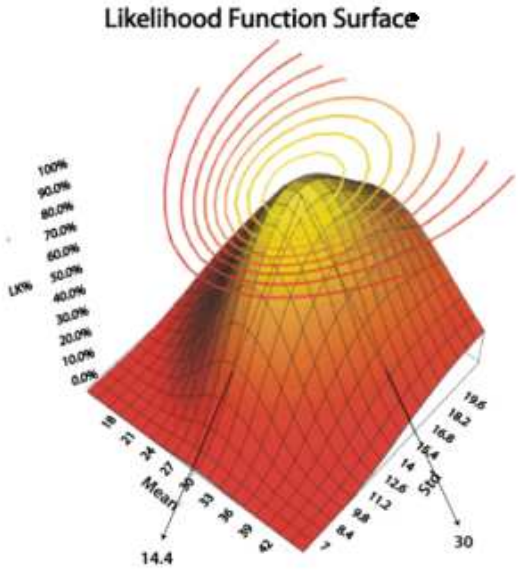
The likelihood function We define what is called the *likelihood function*, which in plain terms is the likelihood that—given our data and assumption of normal distribution of that data—the mean is equal to μ and the standard deviation is equal to σ.

$$\begin{aligned}\mathcal{L}\left(\mu, \sigma; x_1, x_2, \ldots, x_N\right) =& \prod_{i=1}^{N} f\left(x_i; \mu, \sigma\right) \\ =& f\left(x_1; \mu, \sigma\right) f\left(x_2; \mu, \sigma\right) \cdots f\left(x_N; \mu, \sigma\right)\end{aligned} \tag{7}$$

where here $\prod$ is the Greek capital π, which is known as the *product operator*. $\prod$ is the multiplication analog of the summation operator $\sum$. We can do a little bit of simplifying:

$$\begin{aligned}\mathcal{L}\left(\mu, \sigma; x_1, x_2, \ldots, x_N\right) =& \prod_{i=1}^{N} f\left(x_i; \mu, \sigma\right) \\ =& f\left(x_1; \mu, \sigma\right) f\left(x_2; \mu, \sigma\right) \cdots f\left(x_N; \mu, \sigma\right) \\ =& \left[\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\left(x_1 - \mu\right)^2}{2\sigma^2}\right)\right] \times \cdots \times \left[\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\left(x_N - \mu\right)^2}{2\sigma^2}\right)\right]\end{aligned} \tag{8}$$

Figure 1: Illustration of a likelihood function



Source: reliawiki.org

If you recall properties of exponents, we can simplify (8) as follows:

$$\begin{aligned}\mathcal{L}\left(\mu,\sigma;x_1,x_2,\ldots,x_N\right) &= \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^N \exp\left(-\frac{\sum_{i=1}^N\left(x_i-\mu\right)^2}{2\sigma^2}\right)\\ &= \left(\frac{1}{\left(\sqrt{2\pi\sigma^2}\right)^N}\right)\exp\left(-\frac{\sum_{i=1}^N\left(x_i-\mu\right)^2}{2\sigma^2}\right)\end{aligned} \tag{9}$$

What does the likelihood function look like? See Figure 1 for an example where there are two parameters (e.g. μ and σ^2 in the example above).

The log likelihood function Recall that, when ordering a set, one can take a *monotonic transformation* of the set and preserve the ordering. It turns out that we can apply such a transformation to (9) by taking the natural logarithm (ln) of both sides. This leaves us

with:

$$\begin{aligned}
\ln \mathcal{L}\left(\mu, \sigma; x_1, x_2, \ldots, x_N\right) &= \ln \left[\left(\frac{1}{\left(\sqrt{2\pi\sigma^2}\right)^N} \right) \exp\left(-\frac{\sum_{i=1}^N \left(x_i - \mu\right)^2}{2\sigma^2} \right) \right] \\
\ell\left(\mu, \sigma; x_1, x_2, \ldots, x_N\right) &= \ln \left(\frac{1}{\left(\sqrt{2\pi\sigma^2}\right)^N} \right) + \ln \left(\exp\left(-\frac{\sum_{i=1}^N \left(x_i - \mu\right)^2}{2\sigma^2} \right) \right) \\
&= \underbrace{\ln\left(1\right)}_{=0} - \ln\left(\left(2\pi\sigma^2\right)^{N/2} \right) - \frac{1}{2\sigma^2} \sum_{i=1}^N \left(x_i - \mu\right)^2 \\
&= -\frac{N}{2} \ln\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2} \sum_{i=1}^N \left(x_i - \mu\right)^2 \qquad (10)
\end{aligned}$$

Equation (10) is known as the *log likelihood function*. Our goal is find the estimated μ and σ that maximize this function.[2]

Finding the maximum of the log likelihood function To find the maximum likelihood estimates (MLEs) of the log likelihood function, we take the first derivative of $\ell\left(\mu, \sigma; x_1, \ldots, x_N\right)$ with respect to μ and σ:

$$\begin{aligned}
\frac{\partial}{\partial \mu} \ell\left(\mu, \sigma; x_1, x_2, \ldots, x_N\right) &= \frac{\partial}{\partial \mu} \left[-\frac{N}{2} \ln\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2} \sum_{i=1}^N \left(x_i - \mu\right)^2 \right] \\
&= -\frac{1}{2\sigma^2} \frac{\partial}{\partial \mu} \sum_{i=1}^N \left(x_i - \mu\right)^2 \quad \ldots \text{ (need to use the chain rule)} \\
0 &= -\frac{2 \cdot (-1)}{2\sigma^2} \sum_{i=1}^N \left(x_i - \hat{\mu}\right) \qquad (11) \\
0 &= \frac{1}{\sigma^2} \left(\sum_{i=1}^N x_i - N\hat{\mu} \right) \\
\hat{\mu} &= \frac{1}{N} \sum_{i=1}^N x_i
\end{aligned}$$

Now with respect to σ^2 (it's easier than with respect to σ):

[2] Because of monotonicity, the maximum of the likelihood function will be the same as the maximum of the log likelihood function.

$$
\begin{aligned}
\frac{\partial}{\partial \sigma^2} \ell\left(\mu, \sigma; x_1, x_2, \ldots, x_N\right) &= \frac{\partial}{\partial \sigma^2}\left[-\frac{N}{2} \ln\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2} \sum_{i=1}^{N} \left(x_i - \mu\right)^2\right] \\
&= \frac{\partial}{\partial \sigma^2}\left[-\frac{N}{2} \ln\left(2\pi\right) - \frac{N}{2} \ln\left(\sigma^2\right) - \frac{1}{2\sigma^2} \sum_{i=1}^{N} \left(x_i - \mu\right)^2\right] \\
&= -\frac{N}{2\sigma^2} - \frac{\partial}{\partial \sigma^2}\left[\frac{1}{2\sigma^2} \sum_{i=1}^{N} \left(x_i - \mu\right)^2\right] \\
0 &= -\frac{N}{2\hat{\sigma}^2} + \frac{1}{2\left(\hat{\sigma}^2\right)^2} \sum_{i=1}^{N} \left(x_i - \mu\right)^2 \qquad (12) \\
0 &= -\frac{N}{2\hat{\sigma}^2} + \frac{1}{2\hat{\sigma}^4} \sum_{i=1}^{N} \left(x_i - \mu\right)^2 \\
0 &= \frac{1}{2\hat{\sigma}^2}\left[\frac{1}{2\hat{\sigma}^2} \sum_{i=1}^{N} \left(x_i - \mu\right)^2 - N\right] \\
\hat{\sigma}^2 &= \frac{1}{N} \sum_{i=1}^{N} \left(x_i - \mu\right)^2
\end{aligned}
$$

So our Maximum Likelihood Estimate (MLE) is

$$
\begin{aligned}
\hat{\mu} &= \frac{1}{N} \sum_{i=1}^{N} x_i \\
\hat{\sigma}^2 &= \frac{1}{N} \sum_{i=1}^{N} \left(x_i - \mu\right)^2
\end{aligned}
$$

and those formulas should look very familiar to you.

Second-order conditions For a maximum, we want the second order conditions to be *negative* to ensure that we indeed have a maximum. It turns out that these conditions are satisfied for the MLE of the normal distribution.

3.3 MLE optimization for linear regression

Going back to the linear model, let's now assume that $\varepsilon \overset{iid}{\sim} N\left(0, \sigma^2\right)$. We can write down a likelihood function for this model, which is going to look similar to the likelihood function we wrote before, but with a slight tweak—instead of targeting μ as a parameter, we'll be targeting β:

$$
\begin{aligned}
\mathcal{L} &= \prod_i f(\varepsilon_i) \\
&= \prod_i \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\varepsilon_i^2}{2\sigma^2}\right) \\
&= \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n \exp\left(-\frac{\sum_i \varepsilon_i^2}{2\sigma^2}\right) \\
&= \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n \exp\left(-\frac{\varepsilon'\varepsilon}{2\sigma^2}\right) \quad \text{, and, taking logs,} \\
\ell(y, X; \beta, \sigma) &= -\frac{n}{2}\ln(2\pi) - n\ln(\sigma) - \frac{1}{2\sigma^2}(y - X\beta)'(y - X\beta)
\end{aligned}
$$

Our objective is to maximize the likelihood above with respect to β and σ:

$$
\begin{aligned}
&\max_{\beta,\sigma} -\frac{n}{2}\ln(2\pi) - n\ln(\sigma) - \frac{1}{2\sigma^2}(y - X\beta)'(y - X\beta) \\
\frac{\partial \ell}{\partial \beta} &= \frac{1}{2\sigma^2}\frac{\partial}{\partial \beta}(y - X\beta)'(y - X\beta) \\
0 &= \frac{1}{2\sigma^2}\left[-2X'y - 2\left(X'X\right)\hat{\beta}\right] \\
\hat{\beta} &= \left(X'X\right)^{-1}X'y \\
\frac{\partial \ell}{\partial \sigma} &= -\frac{n}{\sigma} + \frac{(y - X\beta)'(y - X\beta)}{\sigma^3} \\
0 &= -\frac{n}{\hat{\sigma}} + \frac{(y - X\beta)'(y - X\beta)}{\hat{\sigma}^3} \\
\frac{n}{\hat{\sigma}} &= \frac{(y - X\beta)'(y - X\beta)}{\hat{\sigma}^3} \\
\hat{\sigma}^2 &= \frac{(y - X\beta)'(y - X\beta)}{n}
\end{aligned}
$$

Second-order conditions for a maximum require that the matrix below be *negative definite*[3]:

$$
\begin{bmatrix} \frac{\partial^2 \ell}{\partial \beta \partial \beta'} & \frac{\partial^2 \ell}{\partial \beta \partial \sigma} \\ \frac{\partial^2 \ell}{\partial \sigma \partial \beta'} & \frac{\partial^2 \ell}{\partial \sigma^2} \end{bmatrix} = \begin{bmatrix} -\frac{X'X}{\hat{\sigma}^2} & \frac{X'X\hat{\beta} - X'y}{\hat{\sigma}^3} \\ \frac{\hat{\beta}'X'X - y'X}{\hat{\sigma}^3} & \frac{n}{\hat{\sigma}^2} - \frac{3\left(y - X\hat{\beta}\right)'\left(y - X\hat{\beta}\right)}{\hat{\sigma}^4} \end{bmatrix} \tag{13}
$$

3.4 Vocabulary

closed-form solution A solution to an optimization problem that can be expressed as a mathematical formula. For example, $\hat{\beta} = (X'X)^{-1}X'y$ is a closed-form solution to

[3] Negative definite is how we tell that a matrix is "negative."

the parameters of the classical linear model. This solution holds for either ordinary least squares (OLS) minimization or maximum likelihood estimaiton (MLE)

Most optimization problems do *not* have a closed-form solution, in which case we will need a computer to iteratively "guess and check" different candidate parameters to find the minimum or maximum of our objective function.

We will talk about these in the next couple of classes.

gradient vector The gradient vector is the vector of first derivatives of the objective function. For MLE problems, it is the vector

$$\begin{bmatrix} \frac{\partial \ell}{\partial \beta} \\ \frac{\partial \ell}{\partial \sigma} \end{bmatrix} = \begin{bmatrix} \frac{1}{2\sigma^2}\left[-2X'y - 2\left(X'X\right)\hat{\beta}\right] \\ -\frac{n}{\hat{\sigma}} + \frac{(y-X\beta)'(y-X\beta)}{\hat{\sigma}^3} \end{bmatrix}$$

hessian matrix The hessian matrix is the matrix of second derivatives of the objective function. For MLE problems, it is the matrix listed in (13).

Why do we need to know these three vocabulary words? Because we need to understand how the computer finds the optimum of our objective function. For the 99% of problems that don't have a closed-form solution, the computer tries to find the values of the parameters that set the gradient vector equal to zero. Depending on the algorithm, it will also check that the hessian matrix is negative definite (if we are maximizing) or positive definite (if we are minimizing).

4 Math: logistic regression

The above examples optimized the classical linear model, which is a model where the dependent variable is continuous and the optimization has a closed-form solution.

Now let's look at a case where the dependent variable y is binary and (without loss of generality) takes on values 0 and 1.

The appropriate statistical distribution to model a 0/1 outcome is known as the *Bernoulli distribution*. In this distribution, $y = 1$ with probability p and $y = 0$ with probability $1 - p$. You can think of a Bernoulli distribution as modeling a weighted coin flip, where heads occurs with probabiliy p.

The probability density function (pdf) of the Bernoulli distribution is

$$f\left(y; p\right) = p^y \left(1 - p\right)^{1-y}$$

Now let's return to our classical linear model, but where y is now binary instead of continuous. As before, we have (in matrix form)

$$y = X\beta + \varepsilon$$

Now we want to think about the conditions under which $y = 1$ (which occurs with probability p).

$$\begin{aligned}
\Pr(y = 1) &= \Pr(y > 0) \\
&= \Pr(X\beta + \varepsilon > 0) \\
&= \Pr(X\beta > -\varepsilon) \\
&= \Pr(-\varepsilon < X\beta) \\
&= \Pr(\varepsilon < X\beta) \quad \text{if } \varepsilon \text{ has a symmetric distribution} \\
&= F(X\beta)
\end{aligned}$$

where $F(\cdot)$ denotes the cumulative distribution function of ε.

4.1 MLE of the logistic regression model

If we assume that ε is drawn from the logistic distribution (instead of the normal distribution as in (8)), then we get

$$\begin{aligned}
F(x) &= \frac{1}{1 + e^{-x}} \\
&= \frac{e^x}{1 + e^x}
\end{aligned}$$

This is known as the **logistic probability function**, or the **sigmoid function**.

If we go back to the previous set of equations and think about our Bernoulli parameter p, we have that $p = \Pr(y = 1) = F(X\beta) = \frac{\exp(X\beta)}{1+\exp(X\beta)}$.

4.1.1 Logit likelihood function

Combining the above results, we get our likelihood function:

$$\begin{aligned}
\mathcal{L}(y, X; \beta) &= \prod_i p_i^{y_i} (1 - p_i)^{1-y_i} \\
&= \prod_i \left(\frac{\exp(X\beta)}{1 + \exp(X\beta)} \right)^{y_i} \left(1 - \frac{\exp(X\beta)}{1 + \exp(X\beta)} \right)^{1-y_i} \\
&= \prod_i \left(\frac{\exp(X\beta)}{1 + \exp(X\beta)} \right)^{y_i} \left(\frac{1}{1 + \exp(X\beta)} \right)^{1-y_i} \\
\ell(y, X; \beta) &= \sum_i y_i \ln \left(\frac{\exp(X\beta)}{1 + \exp(X\beta)} \right) + (1 - y_i) \ln \left(\frac{1}{1 + \exp(X\beta)} \right) \qquad (14)
\end{aligned}$$

4.1.2 Logit first order conditions

The first order conditions of (14) are:

$$\begin{aligned}
\frac{\partial \ell}{\partial \beta} =& \frac{\partial \ell}{\partial \beta} y \left[X\beta - \ln\left(1 + \exp\left(X\beta\right)\right)\right] - \left(1 - y\right)\left[\ln\left(1 + \exp\left(X\beta\right)\right)\right] \\
0 =& \frac{\partial \ell}{\partial \beta} y \left[X\beta\right] - \ln\left(1 + \exp\left(X\beta\right)\right) \\
0 =& X'y - \left[\frac{1}{1 + \exp\left(X\beta\right)} X \exp\left(X\beta\right)\right] \\
0 =& X'y - \left[X \frac{\exp\left(X\beta\right)}{1 + \exp\left(X\beta\right)}\right] \\
0 =& X'y - X'p \\
0 =& X'\left(y - p\right) \\
0 =& X'\left(y - \frac{\exp\left(X\beta\right)}{1 + \exp\left(X\beta\right)}\right)
\end{aligned} \tag{15}$$

Now we need to solve for β:

$$X'y = X'\left(\frac{\exp\left(X\beta\right)}{1 + \exp\left(X\beta\right)}\right)$$

But this does not have a closed-form solution! Hence, we need to use numerical methods such as gradient descent or L-BFGS to iteratively "guess and check" various values of the β vector that solve (15).