Lab session 3: Hypothesis testing & Grid Search

Adam Theising[*]

February 7th, 2018

Contents

1 Hypothesis testing 1
- 1.1 Student's t-test . . . 2
- 1.2 Wald tests . . . 3
- 1.3 F-stats . . . 4
- 1.4 Delta method . . . 5
- 1.5 Some comments looking ahead . . . 6

2 Grid search 7

Getting started...

- Any questions/comments on assignment 1? On the course more broadly?
- Solutions to assignment 1 should come by end of week.
- Should have received assignment 2 by now. Feel free to drop by office hours/send emails w/ any questions.
- This week in lab, we'll be covering two main subjects: **hypothesis testing** and the basics of **grid searching**.

1 Hypothesis testing

Hypothesis testing is the underlying end goal at the heart of almost everything this course will cover. How many studies have you read in academic or policy settings that simply test whether the effect of X on Y is different than Z? These methods are obviously important to have in your econometric toolbox, so let's start with the basic intuition, and build up to some more complicated tests.

Generally speaking, what is a hypothesis test? One way to think about this, per Bruce Hansen's lecture notes: *hypothesis tests attempt to assess whether there is evidence to contradict a proposed parametric restriction*. Suppose we're estimating a model of the general form $y = f(\mathbf{X}, \boldsymbol{\beta}) + \varepsilon$. One common restriction is a point hypothesis: $\boldsymbol{\beta} = \boldsymbol{\beta_0}$. Such a restriction on our model requires that the parameter(s) take the hypothesized value(s); from here we can assess whether this hypothesis is consistent with our data.

To do this, we can generate a test statistic mapping the data to a decision set. Again quoting from Hansen: *it is convenient to express this mapping as a real-valued function called a test statistic, $T = T((y_1, x_1), \ldots, (y_n, x_n))$, relative to a critical value c. The hypothesis test then consists of the decision rule:*

[*] theising@wisc.edu

1. *Accept* $\mathbb{H}_0$ *(*$\boldsymbol{\beta} = \boldsymbol{\beta_0}$*) if* $T \leq c$.

2. *Reject* $\mathbb{H}_0$ *if* $T > c$.

The formula for test statistics and their associated critical values will vary by application. But the general intuition for hypothesis testing is fairly standard: if the test statistic generated by the data and our hypothesis is smaller than the critical value, we cannot reject the null. With a large test statistic, we can reject the null. Below, I will walk through some standard tests and try to highlight when they might be useful.

1.1 *Student's t-test*

You've presumably all seen the derivation of a t-test in a statistics course.[1] A lightning-fast review of the single parameter case: let's imagine we're in the standard world of multiple linear regression. Then testing the null hypothesis that $\hat{\beta}_j$, the parameter estimate for the j^{th} element of our $\mathbf{X}$ matrix, is equal to β_0, we'd have t-test of general form:

$$t_j \equiv \frac{\hat{\beta}_j - \beta_0}{se(\hat{\beta}_j)} = \frac{\hat{\beta}_j - \beta_0}{\sqrt{\hat{\sigma^2} \cdot (X'X)^{-1}_{jj}}}$$

A (probably) trivial example of this: every time you run a *reg* command in Stata, the output provides you with a vector of t-stats. What are they testing?

So we're post-estimation of our β_j parameter, and have calculated our t-stats for the hypothesized β_0. How do know do we know whether to accept or reject the null? We can refer to the Student t distribution (with $n - k$ DoF) to acquire a critical threshold of $|t_j| \geq 1.96$ at the 5% significance level[2] That is, if $t_j \geq 1.96$, the data generated by the null hypothesis would produce the observed test result less than 5% of the time.

So let's think about a question that should be very similar to a few you answered on the first assignment and t-test your understanding.[3] Let's do an algebraic example here; if you have remaining questions about coding up such a test in MATLAB after we release the answer key to assignment 1, feel free to drop me an e-mail.

Suppose you've just estimated a three parameter linear regression model and obtained the following:

$$\hat{\boldsymbol{\beta}} = \begin{bmatrix} 3.5 \\ -1.5 \\ 2 \end{bmatrix}, \ \text{Var}(\hat{\boldsymbol{\beta}}) = \begin{bmatrix} 1 & 0.25 & -0.4 \\ 0.25 & 2 & 1.2 \\ -0.4 & 1.2 & 1.5 \end{bmatrix}$$

Your adviser has a simple hypothesis for you to test: $\hat{\beta}_1 + 2\hat{\beta}_2 = 1$. How would you structure the t-test?

First, we need to find the standard error of $\hat{\beta}_1 + 2\hat{\beta}_2$. We can begin by calculating that $\text{Var}(\hat{\beta}_1 + 2\hat{\beta}_2) = \text{Var}(\hat{\beta}_1) + 4\text{Var}(\hat{\beta}_2) + 4\text{Cov}(\hat{\beta}_1, \hat{\beta}_2) = 1 + 8 + 1 = 10$. Therefore, we know that $se(\hat{\beta}_1 + 2\hat{\beta}_2) = \sqrt{10}$. Now we can calculate our t-statistic:

$$t = \frac{3.5 + 2(-1.5) - 1}{\sqrt{10}} \approx -0.15$$

The small t-statistic indicates that we cannot reject the null hypothesis.

[1] Though I'd hazard a guess you haven't all read the story of its discovery. Beer lovers, rejoice!

[2] I'll assume you all are up to speed on Type I/II errors and the use of 5% as the scientific standard. See here or here for some recent brouhahas in the broader statistical community over this arbitrary level...

[3] See what I did there?! Sorry...

1.2 *Wald tests*

So now we've seen how to test a single parameter, or single function of parameters. What about the case of simultaneously testing multiple parameters (or multiple functions of parameters)? For this, we turn to the Wald test, a generalization of the t-test that allows for multiple linear restrictions (read: joint hypotheses).

Let's write this out in linear algebra... for the Wald statistic, we can express any set of q linear restrictions on K parameter values as:

$$\underset{q\times K}{\mathbf{R}}\ \underset{K\times 1}{\boldsymbol{\beta}} = \underset{q\times 1}{\mathbf{c}}$$

Since we've already estimated our $\hat{\beta}$s, we also have an estimate for the variance-covariance matrix of the parameters, and can thus define the Wald statistic as:

$$W_n = \underset{1\times q}{\left[\mathbf{R}\hat{\boldsymbol{\beta}} - \mathbf{c}\right]'} \underset{q\times q}{\left[\mathbf{R}Var(\hat{\beta})\mathbf{R}'\right]^{-1}} \underset{q\times 1}{\left[\mathbf{R}\hat{\boldsymbol{\beta}} - \mathbf{c}\right]}$$

where $W_n \sim \chi^2_q$. To check your understanding, you should note that when $q = 1$, $W_n \sim \chi^2_1$. If you've taken some upper-level statistics courses, you might recall that the square of a t-distribution is also distributed χ^2_1. *So in effect, with $q = 1$, $W_n = t^2$, and therefore hypothesis tests based on W_n and $|t|$ are equivalent.* That's right, a t-statistic is essentially just a Wald statistic with a single restriction.

Okay, I'm guessing the linear algebra and abstraction there might be tough to swallow if you haven't seen this before... let's be a bit more concrete. I'll give a simple algebraic example then take it to MATLAB for a simple coding exercise.

Let's say we've run a linear regression with four parameters ($K = 4$), and we wish to test the joint hypotheses $\beta_1 + \beta_2 - \beta_3 = 1$ and $\beta_1 = 10$ - note that we have two restrictions ($q = 2$). So we have:

$$\mathbf{R} = \begin{bmatrix} 1 & 1 & -1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix},\ \boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{bmatrix},\ \mathbf{c} = \begin{bmatrix} 1 \\ 10 \end{bmatrix}$$

which in turn implies

$$\mathbf{R}\boldsymbol{\beta} - \mathbf{c} = \begin{bmatrix} \beta_1 + \beta_2 - \beta_3 - 1 \\ \beta_1 - 10 \end{bmatrix}$$

If you look at the Wald statistic formula, you can see that you now have everything you need to calculate it. For $q = 2$ the 5% critical value of a χ^2_2 is 5.99.[4] Now you have *really* have everything you need for your test.

Let's code up a Wald test in MATLAB. We're going to tackle the second part of Question 9.25 in Hansen's lecture notes. The data set invest on the textbook website contains data on 565 U.S. firms extracted from Compustat for the year 1987 used by Hall and Hall (1993). The variables are (in order):

- Inv_i: Investment to Capital Ratio (multiplied by 100).
- Q_i: Total Market Value to Asset Ratio (Tobin's Q)
- C_i: Cash Flow to Asset Ratio.
- D_i: Long Term Debt to Asset Ratio.

[4] For $q = 1$, 5% critical value is 3.84. Or 1.96^2. Not to beat a point to death, but why is this?

We're going to estimate a linear regression of investment on the other three variables, then test Tobin's q theory of investment which suggests that investment should be predicted by only Q_i. To do this, we will execute a joint hypothesis test that C_i and D_i are equal to 0.

```
clear;                        % command to clear memory
clc;                          % command to clear the command window
[dat,txt,raw] = xlsread('invest.xlsx','Sheet1'); % import data

%% Question 9.25(c)
% Estimate standard linear regression
y = dat(:,1);
x = dat(:,2:4);
coef_names = txt(:, 2:4);
[beta,se,~,~,covb] = basic_ols_proc(x,y,coef_names,1);

% Create restriction and find Wald stat
R_trans = [0, 0, 1, 0; 0, 0, 0, 1]; % This is R'
theta = R_trans*beta; % theta = RBeta - Beta0 (where Beta0 = [0,0])
inner = R_trans*covb*transpose(R_trans); % inner = R'*Var(beta)*R

Wald_stat1 = transpose(theta)*inv(inner)*theta

clear x coef_names beta se covb R_trans theta inner
```

After running the code, we see that the Wald statistic is 24.6961, which is greater than the critical value of 5.99. Thus, we can reject the null hypothesis, and argue that our results contradict Tobin's q theory. Let's now estimate a similar linear regression including quadratic and interaction terms for each of the 3 independent variables. Then, we'll use a Wald test to evaluate the null hypothesis that the joint sum of the 6 additional terms is equal to 0.

```
%% Question 9.25(d)
% Create new variables
Q2 = (dat(:,2)).^2;
C2 = (dat(:,3)).^2;
D2 = (dat(:,4)).^2;
QC = dat(:,2).*dat(:,3);
QD = dat(:,2).*dat(:,4);
CD = dat(:,3).*dat(:,4);

y = dat(:,1);
x = horzcat(dat(:,2:4),Q2,C2,D2,QC,QD,CD);
coef_names = horzcat(txt(:, 2:4),'Q2','C2','D2','QC','QD','CD');
[beta,se,~,~,covb] = basic_ols_proc(x,y,coef_names,1);

% Create restriction and find Wald stat
R_trans = [0, 0, 0, 0, 1, 1, 1, 1, 1, 1]; % This is R'
theta = R_trans*beta; % theta = RBeta - Beta0 (where Beta0 = [0])
inner = R_trans*covb*transpose(R_trans); % inner = R'*Var(beta)*R

Wald_stat2 = transpose(theta)*inv(inner)*theta

clear beta C2 CD coef_names covb D2 inner Q2 QC QD R_trans se theta x y
```

Here, we find the Wald statistic is 1.9052. The critical value is 3.84, so we are unable the reject the null that the addition of the 6 new variables to the regression has a joint sum of zero.

1.3 *F-stats*

This will be short and sweet jaunt into the world of F-tests. You should have performed one in the first assignment. They're an old school mainstay[5], so let's hit the highlights here for completeness.

[5] Some things never go out of style...

An F-test can also be used to test multiple hypotheses, and is similar in nature to the Wald stats above. The main difference: an F stat is compared against the F distribution which incorporates a "small sample" correction. The F-test statistic is given by $F_n \sim F[q, n-k]$ where k is the number of parameters and n is the number of observations. As $n \to \infty$, the F-distribution converges to a χ^2_q distribution, and it can be shown that for a homoskedastic estimate of the variance-covariance matrix, $F_n = \frac{W_n}{q}$. **Main takeaway from this paragraph: F stats and Wald stats are similar.**

The standard application of the F-statistic is expressed in terms of the sum-of-squared-errors from an OLS regression:

$$F_n = \frac{\left(SSE(\tilde{\beta}_{cls}) - SSE(\hat{\beta})\right)/q}{SSE(\hat{\beta})/(n-k)}$$

where $SSE(\tilde{\beta}_{cls})$ is the SSE for a constrained model. A trivial, yet common example again comes from your Stata output after a *reg* command: the F-stat reported is from a constrained model where all the parameters are restricted to equal zero. Thus, this F-stat is testing whether the model has any explanatory power at all.

1.4 *Delta method*

So we've covered a broad set of statistics that can test linear restrictions of interest. But what if you have some interest in testing non-linear functions of parameters? For example, maybe you want to test whether marginal effects (with non-linear terms) are different from zero... Below, let's run through some basic machinery that will be super useful moving forward. We'll walk through an algebraic example as well... will leave coding it up in MATLAB as an exercise to you!

We'll begin with the statement of a general central limit theorem.. Let's suppose the asymptotic distribution (meaning as $n \to \infty$) of a $K \times 1$ set of consistent estimators, $\hat{\beta}$, is jointly normal:

$$\sqrt{n}\left(\underset{K\times 1}{\hat{\boldsymbol{\beta}}} - \underset{K\times 1}{\boldsymbol{\beta_0}}\right) \longrightarrow \mathcal{N}\left(\underset{K\times 1}{\mathbf{0}}, \underset{K\times K}{\text{Var}(\hat{\boldsymbol{\beta}})}\right)$$

This limit theorem provides the basis of the previous hypothesis tests we've covered. The delta method is a theorem that extends this logic to a non-linear framework. Suppose we have a *continuously differentiable* function $g : \mathbb{R}^K \to \mathbb{R}^L$. Then one can prove[6] the delta method holds:

$$\sqrt{n}\left(\underset{L\times 1}{g(\hat{\boldsymbol{\beta}})} - \underset{L\times 1}{g(\boldsymbol{\beta_0})}\right) \longrightarrow \mathcal{N}\left(\underset{L\times 1}{\mathbf{0}}, \underset{L\times L}{\left[\frac{\partial g}{\partial \beta}\right] \text{Var}(\hat{\boldsymbol{\beta}}) \left[\frac{\partial g}{\partial \beta}\right]'}\right)$$

where $\frac{\partial}{\partial \beta} g(\beta)$ is an $(L \times K)$ Jacobian matrix:

$$\frac{\partial}{\partial \beta} g(\beta) = \begin{bmatrix} \frac{\partial}{\partial \beta_1} g_1(\beta) & \cdots & \frac{\partial}{\partial \beta_K} g_1(\beta) \\ \vdots & \ddots & \vdots \\ \frac{\partial}{\partial \beta_1} g_L(\beta) & \cdots & \frac{\partial}{\partial \beta_K} g_L(\beta) \end{bmatrix}$$

So though it might not be immediately clear behind all the matrix notation, this is a *super* useful result. We now have an asymptotic approximation for the variance-covariance structure of *any continuously differentiable function of our estimated parameters*. We can therefore write a general

[6] See Hansen chapter 6.12 for an exquisite treatment. Appendix D of Greene also has a nice exposition.

Wald test statistic as:

$$W_n = \underset{1\times L}{\left[g(\hat{\boldsymbol{\beta}}) - \mathbf{c}\right]'} \underset{L\times L}{\left[\left(\frac{\partial g}{\partial \beta}\right) \text{Var}(\hat{\boldsymbol{\beta}}) \left(\frac{\partial g}{\partial \beta}\right)'\right]^{-1}} \underset{L\times 1}{\left[g(\hat{\boldsymbol{\beta}}) - \mathbf{c}\right]}$$

with $W_n \sim \chi^2_L$. This is a good place to test your understanding: ask yourself what happens if $g(\cdot)$ is just a linear function in $\boldsymbol{\beta}$.

Okay, let's put together everything we've covered so far and take on an analytical example of a non-linear hypothesis.

Imagine we're post-estimation of a two-parameter model, and would like to jointly test whether $\hat{\beta}_1/\hat{\beta}_2 = 1$ and $(\hat{\beta}_2)^2 = 2$. Note that these tests are both non-linear functions of the parameters. Suppose our estimation yielded:

$$\hat{\beta} = \begin{bmatrix} 0.5 \\ 1.5 \end{bmatrix}, \ \text{Var}(\hat{\beta}) = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 2 \end{bmatrix}$$

What do we do? Let's start by determining what our $g(\cdot)$ function is and finding its Jacobian:

$$g(\hat{\beta}) = \begin{bmatrix} \hat{\beta}_1/\hat{\beta}_2 \\ (\hat{\beta}_2)^2 \end{bmatrix}, \ \frac{\partial}{\partial \beta} g(\hat{\beta}) = \begin{bmatrix} 1/\hat{\beta}_2 & -\hat{\beta}_1/(\hat{\beta}_2)^2 \\ 0 & 2\hat{\beta}_2 \end{bmatrix}$$

Note that we have 2 "restrictions" here - i.e. $L = 2$. From this Jacobian, we can the transformed variance-covariance matrix for $g(\hat{\beta})$:

$$\text{Var}(g(\beta)) = \begin{bmatrix} 1/1.5 & -0.5/2.25 \\ 0 & 2 \cdot 1.5 \end{bmatrix} \begin{bmatrix} 1 & -0.5 \\ -0.5 & 2 \end{bmatrix} \begin{bmatrix} 1/1.5 & -0.5/2.25 \\ 0 & 2 \cdot 1.5 \end{bmatrix}' = \begin{bmatrix} 0.691 & -2.3333 \\ -2.3333 & 18 \end{bmatrix}$$

Now, we have all the ingredients to calculate the Wald statistic for our joint hypothesis test:

$$W_n = \left(\begin{bmatrix} 0.5/1.5 \\ 1.5^2 \end{bmatrix} - \begin{bmatrix} 1 \\ 2 \end{bmatrix} \right)' \text{Var}(g(\beta))^{-1} \left(\begin{bmatrix} 0.5/1.5 \\ 1.5^2 \end{bmatrix} - \begin{bmatrix} 1 \\ 2 \end{bmatrix} \right) \approx 1.038$$

This Wald statistic falls well below the 5% critical value of the χ^2_2 distribution: 5.99. Thus, we cannot reject the joint null hypothesis.

1.5 *Some comments looking ahead*

It's important to highlight the fact that we only worked through one variation on hypothesis testing today. Essentially, we covered several variations on the Wald test. Since this is the "workhorse" of hypothesis tests, and probably the one with the most approachable learning curve, it makes sense to start here. But we should note that there are alternatives and that the Wald test has its limitations.

The two main alternatives are likelihood-ratio (LR) and Lagrange-multiplier (LM, or score) tests. I won't discuss them today, but be aware that they exist as the semester continues... Brian will surely bring them into the fold at some point.

One important limitation of Wald tests is their lack of invariance to algebraically equivalent parameterizations of null hypotheses. Following Gregory and Veall (1985) in spirit, consider $\mathbb{H}_0 : \beta_1/\beta_2 - 1 = 0$ and $\mathbb{H}_0 : \beta_1 - \beta_2 = 0$. These hypotheses are algebraic equivalents! While the Wald statistics are also asymptotically identical, the aforementioned paper shows that in finite samples, the statistics can vary widely. This is a fundamental and undesirable problem of using Wald tests for nonlinear hypotheses, so one should exercise caution around them when dealing with small samples.

2 Grid search

Much of the material we're going to cover in this class will rely on optimization algorithms. In class so far, we've seen the Gauss-Newton and Newton-Raphson algorithms, and there are more to come. As you've realized, these algorithms rely heavily on the ability to take functional derivatives in order to linearize non-linear functions - GN is built from a first-order Taylor series of the non-linear regression function, while NR comes from a second-order Taylor series of the SSE function. These algorithms use some nice properties of linearization to yield the optimal parameter values that we desire.

Perhaps you've asked yourself whether there's a more direct way. There is. It's computational brute force - a grid search.[7]

Consider the general problem of maximizing a function $f(\mathbf{x})$ when you have a good idea of the domain of $\mathbf{x}$ (i.e. suppose x is in the bounded space $B \in \mathbb{R}^n$). What if instead of taking first-order and second-order conditions per usual, you found the value of $f(\mathbf{x})$ for many, many points on a grid spanning an n-dimensional grid of the domain of $\mathbf{x}$. Then, whichever point on your grid provides the maximum value of your function also defines the optimal solution to your problem.

Okay, let's be more concrete and solve a simple problem. We will not be minimizing a SSE as required in assignment 2, but hopefully this exercise will be instructive in terms of approaching these kind of problems. Suppose we want to maximize $f(\mathbf{x}) = x_1 - 0.2x_1^2 + x_2 - 0.3x_2^2$. In other words, we want to find the pairing $\mathbf{x}^* = (x_1^*, x_2^*)$ that maximizes this function. To solve, we can estimate a brute-force answer by drawing a grid over the space $x_1, x_2 \in [0, 5]$[8]:

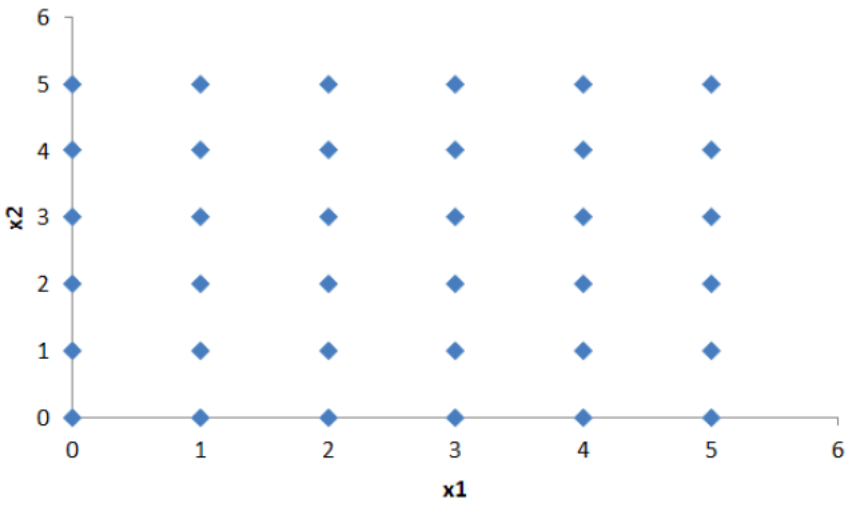

Figure 1: An equally-spaced grid over our domain.

So now you've visualized a grid. Let's run a basic search with uniformly-distanced points between 0 and 5 in MATLAB, and try to determine where the optimum is. *Keep in mind that the analytical solution to this maximization problem is* $x^* = (5/2, 5/3)$.

```
%% Construct a grid from 0 to 5 in both dimensions
m1 = 6; m2 = 6; % Number of points in each direction of grid.
G1 = linspace(0,5,m1);
G2 = linspace(0,5,m2);
grid = ones(m1,m2)*(-1e+10) % ILLUSTRATIVE: draw starting grid
```

[7] H/t: some of what follows comes from lecture notes by A. Parkhomenko at USC.

[8] As we'll see, the domain of 0 to 5 is a "reasonable range" of values on which to define the grid - as mentioned in the problem set, defining a reasonable range is part of the challenge of using this approach.

```
%% Set initial values - don't set too large or you'll never leave.
x1_max = -1e+10; % Value of x1 that maxes function
x2_max = -1e+10; % Value of x2 that maxes function
f_max = -1e+10; % Value of function at its maximum

%% Loop through the grid filling in points
for i = 1:m1 % Loop through every x1 value in grid
    for j = 1:m2 % Loop through every x2 value in grid
        f = G1(i) - 0.2*G1(i)^2 + G2(j) - 0.3*G2(j)^2; % Function value at (i,j)
        grid(i,j) = f % ILLUSTRATIVE: Fill in the grid cell with function value
        if f > f_max; % Replace values below if condition is met.
            f_max = f;
            x1_max = G1(i);
            x2_max = G2(j);
        end
    end
end
fprintf('\nBasic grid search:\n')
fprintf('\nMax of%8.4f attained at x1 =%8.4f and x2 =%8.4f\n',f_max,x1_max,x2_max)
```

If you want to see the guts of what this simple code is doing, open up the *grid* matrix that's been created. This matrix compiles the function's value at every grid point. From here, we can see that the grid search yields a maximum value at $x_1 = 2$ and $x_2 = 2$. How does this compare with our analytical optimum? A few thoughts:

- First of all, our grid is clearly too coarse. We're a fair bit away from the true optimum, largely due to the structure of our grid. How can we improve this?
  - Finer grid - on your own, try an 11×11 grid and compare results.
  - Non-equally spaced grid - again on your own, try a 6×6 grid with unequal spacing. For instance, create varying spacing by transforming your G_1 and G_2 vectors as $G_i = log(3 * G_i + 1)$.[9]
  - Adaptive grid - see Question 3 of Assignment! The idea here is straightforward - run through a coarse grid once, find the maximum, then contract the domain of interest around this max point and iterate again using a finer grid. Repeat until you've converged on a solution. Voila!
- Second, ask yourself (or code on your own time) what would happen your solution if your loop iterated downward - i.e. i and j started from 5 and worked down to 0. What's different and how big of a problem is this?
- Seeing this relatively weak result, you might be asking the big picture question: what are the pros/cons of this approach. Some of my thoughts:
  - Pros: this is easy to code and super intuitive. With the advent of parallel processing and ongoing advances in computing, if you have a decent idea of the neighborhood where your true parameter value(s) live and a small number of parameters to estimate, a grid search is a fast and mindless way to get close.
  - Cons: this approach is not sexy. It also suffers from the same problems that plague optimization algorithms (local min/maxima, badly behaving objective functions). And the curse of dimensionality is a real problem for grid searches - imagine searching for 5 parameters, using 50 grid points for each. You'd be running just over 312 million iterations...

[9] Taking this thought to its logical end, I should mention a cool result that I've seen in the machine learning literature suggesting that "random searches" over your search domain are *guaranteed* to be more efficient than grid searches. This is well, well outside the scope of the course, but if you're interested, the paper is here.