TA Session 3

Bruno Amat

November 6, 2024

Outline

Numerical Optimization Methods

INTRODUCTION

We have 3 ways to Optimize a function:

- Grid Search
- Gradient Method
- Non-Gradient Methods

Grid Search

The idea of the Grid Search Model is to apply the objective function to all elements inside a grid to determine the maximum or minimum value.

This method is ideal for an objective function that depends on only one or two parameters, or to check if you have really achieved the minimum/maximum by fixing other parameters.

First, create a grid that contains a large set, for example, $[-10, 10]$. After finding the minimum, then use the same number of elements on a smaller grid, such as $[0, 1]$. Perform this algorithm until you are satisfied.

Also, plotting the objective function is very helpful to find its characteristics.

Gradient Methods

The idea of the Gradient Methods is to use the gradient of a function to determine its minimum or maximum.

Instead of calculating the maximum by using all possible inputs like in the grid search, we can use the gradient of the function and find its roots.

Hence, let $Q(\theta)$ be the objective function and g its gradient. We want to find θ^* such that $g(\theta^*) = 0$.

Gradient Methods

A possible way to find the max/min is to use the function's Taylor Expansion.

We are going to iterate the following expression:

$$\hat{\theta}_{t-1} = \hat{\theta}_t - \hat{H}(\hat{\theta}_{t-1})^{-1} g(\hat{\theta})$$

where $H(.)$ is the Hessian matrix of the gradient method. We are going to iterate this function until $g(\hat{\theta}) = 0$.

To determine if its a minimum or maximum, we look at the Hessian matrix (positive - maximum — negative - minimum).

Computing the gradient numerically

The gradient can be approximated by using:

$$g(\hat{\theta}) = \frac{Q(\hat{\theta} + he_j) - Q(\hat{\theta} - he_j)}{2h}$$

where e_j is a vector that contains zeros in all elements except j, h is a very small number.

Be careful with the value of h.

NON-GRADIENT METHODS

It is also possible to determine the maximum/minimum by using a algorithm that does not require the gradients.

- **SAN:** Iteratively explores the solution space, allowing it to accept worse solutions with a decreasing probability as the algorithm progresses
- **Nelder Mead:** The Nelder-Mead method is an iterative optimization which uses a Simplex. The simplex is modified in each iteration by replacing its worst-performing vertex with a new vertex that reflects the behavior of the function

Bootstrap

MOTIVATION

Sometimes we use estimators which there is no close form variance formula.

Therefore, we need an approach to estimate it. This approach is the Bootstrap.

ALGORITHM

First we estimate our model. Let it be:

$$y = g(x_i, \theta) + \epsilon$$

We are going to obtain $\hat{\theta}$. However, we cannot compute its variance analytically. Hence we will use a bootstrap algorithm.

Algorithm

1) Resample your dataset with replacement. That is select randomly observations to re-construct your dataset and let that it can take repeated observations.

2) Re-estimate your model and obtain a new parameter. Store this parameter in a vector.

3) repeat steps 1 and 2, b times.

ALGORITHM

After obtaining the vector $\hat{\Theta}^B = \{\hat{\theta}^{B_1}, ..., \hat{\theta}^{B_b}\}$ you will calculate the variance of $\hat{\theta}$ by;

$$\hat{V}_{\hat{\theta}}^{boot} = \sum_{i=1}^{b} (\hat{\theta}^{B_1} - \hat{\theta})'(\hat{\theta}^{B_1} - \hat{\theta})$$

Using the variance you can compute the standard errors and the estimators confidence interval.

Discrete choice models - Fixed Coefficient

INTRODUCTION

I will explain the discrete choice model by an example.

- Heterogeneous goods Bertrand model
- There are 4 chocolates bars in the market: 1) Dark chocolate with peanut butter, 2) Chocolate with peanut butter, 3) Dark chocolate without peanut butter and 4) Chocolate with peanut butter.
- Individual can choose which chocolate bar they want to buy based on their preference.

INTRODUCTION

Our data is given by:

- which chocolate bar the individual bought
- chocolate bar prices
- product characteristics

The model

Suppose that the individual j chooses the Dark chocolate bar without peanut butter (DC).

Hence, by revealed preference, we know that at current prices, he prefers this chocolate bar over the other ones. Thus,

$$u(DC) \geq u(y) \ \ \forall y \neq DC \tag{1}$$

The model

Suppose that the individual j utility function for the product characteristics (x) and the good i is given by:

$$u_j(x) = u(x) + \epsilon_{j,i}$$
$$u_j(x) = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} - \alpha p_i + \epsilon_{j,i}$$

where $\epsilon_{j,i}$ follows a extreme value type 1, $x_{1,.}$ refers to Dark or Normal chocolate and $x_{2,.}$ refers to with peanut butter or without it. p_i refers to good i price.

THE MODEL

We want to estimate the parameters $\beta_0, \beta_1, \beta_2, \alpha_0$. According to our data, we already have information about the share of each good. Moreover, each good share on the model can be computed by:

$$P(u(DC) \geq u(y) \forall y \neq DC)$$

Tip for the next step: Look at the error distribution.

Discrete choice models - Random Coefficient

BLP

The BLP model (random coefficient) has the same objective of the previous one: Estimate the demand under a Bertrand competition with heterogeneous goods.

However, it is superior because:

- Uses only market-level price and quantity data
- treat prices as endogenous variables
- more realistic elasticities

BLP

I will present a slightly different model when compared to Nevo's Practitioner's Guide.

The utility function is given by:

$$U_{ij} = \sum x_{jk}\beta_{ik} + \epsilon_j + \zeta_{ik} \text{ where } \beta_{ik} = \bar{\beta}_k + \beta_k^0 z_i + \beta^k v_i$$

where x_{jk} is characteristic k of good j, z_i is demographic characteristics. Note that there is more heterogeneity than the previous model ($\epsilon_j, \zeta_{ik}, v_i$).

Let $\sum x_{jk}\bar{\beta}_k + \epsilon_j = \delta_j$, $\theta_1 = \bar{\beta}_k$, $\theta_2 = [\beta_k^0, \beta^k]$.

Step 1: Market Share

Guessing δ_j, θ_2. Compute the market share using numerical integration:

$$s_j = \int I(u_ij > u_it) dF(\epsilon_i, z_i, v_i)$$

Draw nS consumers and verify which good they choose. Calculate the emprical market share

$$\tilde{s}_j = \frac{1}{nS} \sum_{i=1}^{nS} I(u_ij > u_it)$$

Inversion: $\tilde{s}_j(\delta_j, x, p, \theta_2) = s \rightarrow \delta_j = \tilde{s}_j^{-1}(s, x, p, \theta_2)$

Step 2: Inversion

To guess θ_2, we need to find δ as a function of the market share.

$$\delta_t^{h+1} = \delta_t^h + log(s_t) - log(s(\delta_t^h, x, p, \theta_2))$$

BLP proved that it is a contraction

STEP 3: GMM

We know that

$$\delta_j = x_{ij}\beta + \alpha P_{ji} + \zeta_{ik}(\theta)$$
$$\zeta_{ik}(\theta) = x_{ij}\beta + \alpha P_{ji} - \delta_j$$

Note that $\beta \in \theta_1$ and $\delta_j = \tilde{s}_j^{-1}(s, x, p, \theta_2)$. Hence, we can estimate θ_1 and θ_2 by a GMM estimation.

Step 3: GMM

Let Z be the instruments matrix (costs, number of other products with similar prices). Our GMM function is given by:

$$\min_{\theta} \zeta(\theta)Z'WZ\zeta(\theta)$$

We can rewrite the model as:

$$\min_{\theta_1}[\min_{\theta_2} \zeta(\theta)Z'WZ\zeta(\theta)]$$