Why do we need constrained optimization?

In nonlinear optimization, constraints can be very helpful, for a number of reasons:

Why do we need constrained optimization?

In nonlinear optimization, constraints can be very helpful, for a number of reasons:

- Numerical stability
  - e.g. optimization will crash if it tries a negative value for a variance parameter

Why do we need constrained optimization?

In nonlinear optimization, constraints can be very helpful, for a number of reasons:

- Numerical stability
  - e.g. optimization will crash if it tries a negative value for a variance parameter
- Make results consistent with economic theory
  - e.g. discount factor $\beta \in [0, 1]$ in DDC models, otherwise model is undefined

Why do we need constrained optimization?

In nonlinear optimization, constraints can be very helpful, for a number of reasons:

- Numerical stability
  - e.g. optimization will crash if it tries a negative value for a variance parameter
- Make results consistent with economic theory
  - e.g. discount factor $\beta \in [0, 1]$ in DDC models, otherwise model is undefined
- Simplify the problem
  - e.g. $\beta = 0$ reduces to a static model

Why do we need constrained optimization?

In nonlinear optimization, constraints can be very helpful, for a number of reasons:

- Numerical stability
  - e.g. optimization will crash if it tries a negative value for a variance parameter
- Make results consistent with economic theory
  - e.g. discount factor $\beta \in [0, 1]$ in DDC models, otherwise model is undefined
- Simplify the problem
  - e.g. $\beta = 0$ reduces to a static model
- More quickly solve equilibrium models through a method called MPEC

- How do we do constrained optimization in economics? Lagrangians!

$$\max_x f(x)$$
$$\text{subject to}$$
$$g(x) \leq 0$$

$$\mathcal{L}(x, \lambda) = f(x) - \lambda g(x)$$

- How do we do constrained optimization in economics? Lagrangians!

$$\max_x f(x)$$
$$\text{subject to}$$
$$g(x) \leq 0$$

$$\mathcal{L}(x, \lambda) = f(x) - \lambda g(x)$$

- In the case of optimization, $f(x)$ is our likelihood function; x's are the parameters

- How do we do constrained optimization in economics? Lagrangians!

$$\max_x f(x)$$
$$\text{subject to}$$
$$g(x) \leq 0$$

$$\mathcal{L}(x, \lambda) = f(x) - \lambda g(x)$$

- In the case of optimization, $f(x)$ is our likelihood function; x's are the parameters

- The first-order conditions (FOCs) tell us what the optimal x's are

- How do we do constrained optimization in economics? Lagrangians!

$$\max_{x} f(x)$$
$$\text{subject to}$$
$$g(x) \leq 0$$

$$\mathcal{L}(x, \lambda) = f(x) - \lambda g(x)$$

- In the case of optimization, $f(x)$ is our likelihood function; x's are the parameters
- The first-order conditions (FOCs) tell us what the optimal x's are
- Also must satisfy second-order (SOCs) and Kuhn-Tucker conditions

- How do we do constrained optimization in economics? Lagrangians!

$$\max_x f(x)$$
$$\text{subject to}$$
$$g(x) \leq 0$$

$$\mathcal{L}(x, \lambda) = f(x) - \lambda g(x)$$

- In the case of optimization, $f(x)$ is our likelihood function; x's are the parameters
- The first-order conditions (FOCs) tell us what the optimal x's are
- Also must satisfy second-order (SOCs) and Kuhn-Tucker conditions
- In this case, the SOCs involve looking at the bordered Hessian

How to use JuMP

How to use JuMP

- Let's go through an example of how to estimate an econometric model with `JuMP`

How to use JuMP

- Let's go through an example of how to estimate an econometric model with `JuMP`
- There are four basic components to any `JuMP` model:

How to use JuMP

- Let's go through an example of how to estimate an econometric model with `JuMP`
- There are four basic components to any `JuMP` model:
  1. An optimizer

How to use JuMP

- Let's go through an example of how to estimate an econometric model with `JuMP`
- There are four basic components to any `JuMP` model:
    1. An optimizer
    2. Variables

How to use JuMP

- Let's go through an example of how to estimate an econometric model with `JuMP`
- There are four basic components to any `JuMP` model:
  1. An optimizer
  2. Variables
  3. Constraints

How to use JuMP

- Let's go through an example of how to estimate an econometric model with `JuMP`
- There are four basic components to any `JuMP` model:
  1. An optimizer
  2. Variables
  3. Constraints
  4. Objective function

How to use JuMP

- Let's go through an example of how to estimate an econometric model with `JuMP`
- There are four basic components to any `JuMP` model:
  1. An optimizer
  2. Variables
  3. Constraints
  4. Objective function
- This list is not too different from what goes into `Optim.jl`

Limitations and considerations when using JuMP

Limitations and considerations when using JuMP

- You cannot vectorize the objective function
    - i.e. everything needs to be expressed as a scalar

Limitations and considerations when using JuMP

- You cannot vectorize the objective function
    - i.e. everything needs to be expressed as a scalar
- You cannot use `Distributions.jl` objects in the objective function

Limitations and considerations when using JuMP

- You cannot vectorize the objective function
  - i.e. everything needs to be expressed as a scalar
- You cannot use `Distributions.jl` objects in the objective function
- It is a royal pain to extract the Hessian of the objective function
  - We need the Hessian to conduct statistical inference

Limitations and considerations when using JuMP

- You cannot vectorize the objective function
    - i.e. everything needs to be expressed as a scalar
- You cannot use `Distributions.jl` objects in the objective function
- It is a royal pain to extract the Hessian of the objective function
    - We need the Hessian to conduct statistical inference
- It is very simple to add constraints

Limitations and considerations when using JuMP

- You cannot vectorize the objective function
  - i.e. everything needs to be expressed as a scalar
- You cannot use `Distributions.jl` objects in the objective function
- It is a royal pain to extract the Hessian of the objective function
  - We need the Hessian to conduct statistical inference
- It is very simple to add constraints
- JuMP computes the Hessian of the Lagrangian when constraints are present

Limitations and considerations when using JuMP

- You cannot vectorize the objective function
  - i.e. everything needs to be expressed as a scalar
- You cannot use `Distributions.jl` objects in the objective function
- It is a royal pain to extract the Hessian of the objective function
  - We need the Hessian to conduct statistical inference
- It is very simple to add constraints
- JuMP computes the Hessian of the Lagrangian when constraints are present
- This requires additional processing to obtain correct SEs under constraints