Noter til dynamic programming

# Structural estimation of Markov decision process

## Introduction

The general idea of the Markov decision process is to establish to optimal way in which to make sequential decisions under uncertainty.

It has been extensively used in theoretical studies, as it is able to solve most economic problems involving choices made over time.

Before considering if it is possible to estimate the agents beliefs and preferences, it should be considered if it is logically possible.

It then goes through the different limitations and possibilities of the Markov decision process and the discrete decision process.

## Solving MDP’s via dynamic programming, a brief overview.

Det er meget teoretisk, tror det er mere egnet til opslag, end til at læse alene.

# Dynamic economic quantative methods and application

## Chapter 1: overview

This chapter is mostly about what they will go through throughout the book, and why they do so

## Chapter 2: theory of dynamic programming

### Indirect utility

This part is just about how they are generally optimization problems, which can be used for firms and consumers to maximize their utility

### Cake eating example

If there was a cake, some person could choose the optimal consumption saving, providing the highest amount of utility for the user. The cake is assumed not to neither grow nor decay.

We start by analyzing the cake eating problem through a lens of a finite problem.

The lifetime utility of the cake is written as

Where is the discount factor, and

#### Direct attack

Based on formula (1) we might look at the problem through wanting to maximize the wealth and the consumption

### Some extensions to the cake eating problem

#### Infinite horizon

This chapter starts out talking about the infinite horizon of the basic structure. It states, that it is dependent on the very standard formula meaning that we wish to maximize the consumption and cake size throughout all periods. Further, we wish to gain as much utility as possible.

We are also given the usual equation which means that the wealth(cake) in next period is given as the wealth in this period minus what is being consumed.

This is then recreated to make the formula

which is the dynamic programming problem, stating that the value of the infinite horizon problem is dependent on the utility of consumption plus the discounted value of the wealth minus the consumption.

In the formentioned equation, the state variable is the size of the cake W, the control variable is c, which is chosen. Those two combined creates the transition equation W’=W-c

We can then move the equation to depend on the next period to make it easier algebraically.

To find the first order condition we would first differentiate the important one written above

##### Example

We are now given an example, where

Given the results from the t-period problem, the general form of the equation is

This then goes on to become

We then find the first order condition, which is

This is then inserted into the equation above

We then collect all the terms that include the ln(W)

Herfra kan man se at hvilket når man isolerer for B giver

#### Taste shocks

En af de store fordele ved dynamisk programmering, er at man relativt nemt kan tilføje usikkerheder til sine udregninger. Til at udregne ”taste shocks” bruger man til at påsætte værdien af disse taste schock.

Man bliver nødt til at udregne hvor meget disse ”taste shocks” kommer til at betyde i fremtiden, samt hvor meget det betyder i nuværende periode, for at finde det optimale forbrug. Man sætter her i første omgang til at være i diskrete værdier altså eller . Værdien af disse baseres derefter også på sandsynligheden for at det bliver den ene eller den anden af de to. Det opgøres derefter i en matrice, som kaldes **Transition matrix** som jeg antager ser således ud

Man løser herefter førsteordens bettingelserne.

#### Discrete choice

Nu antager vi at kagen skal spises i en periode. Vi tilføjer dertil muligheden for at kagen kan vokse (mindske) med raten . Disse typer valg bruges blandt andet til at finde ud af hvornår man bør tage et job, hvornår man bør slå sig ned…

### General formulation

#### Non-stochastic case

Forestil dig et infinite horizon optimerinsproblem med en payoff function for period t givet som

Agentens payoff over perioden skrives som

For at maximere payoff skal man enten finde førsteordensbettingelserne eller man kan løse

Dette skulle derefter løses for at finde den optimale ”policy function”

#### Stochastic dynamic programming

I denne udregning tilføjer vi stochastiske shocks ligesom vi gjorde tilbage i økonometri når vi lavede en ar model.

### Conclusion

Simply states that the chapter is mostly about how the different theories work, and how they are a cornerstone of how dynamic programming works. This is to say also that there will be worked further on all the theories, which will be more empirically oriented.

## Chapter 3: Numerical analysis

### 3.1 Overview

This tells how this chapter will go over the numerical analysis aspect of dynamic programming, which will be used a lot during the remainder of this book. We will especially go over **Value function iteration(VFI)** Which we have also already used in class.

### 3.2 Stochastic cake eating problem

The stochastic cake eating problem is written as the function

With

In this equation there are two state variables which is the size of the cake, and y, which is the endowment of cake in the following periods.

We start by analysing in the situation, where the function for y is iid, meaning independent and identically distributed, meaning that each y doesn’t tell anything about the next periods y.

The consumer would in this case only care about the amount which can be eating given as the function

We can then rewrite the original problem to only be based on the x making it

When programming a value iteration function one should go through 4 steps

1. Choosing a functional form of the utility function
2. Discretizing the state and control variables
3. Building a computer code to perform value iteration functions
4. Evaluating the value and the policy function.

#### 3.2.1 Value function iteration

##### Functional form and parameterization

Here we are given the function

We assume for now that the growth of the cake is equal to the discounting factor of the discounting value.

##### State and control space

When we wish to make these calculations we cannot do it on a continuous time, as that we demand infinite computation. We can however closely estimate is using some vector of different values, with more values equaling more precise answer but also more computational time.

##### Value function iteration and policy function

Here it informs about how the values are iterated over a number of times until the difference between this period and the next is below a certain threshold. This is something you pick yourself, again with a trade-off between computational time and precision.

This is also based on an initial guess for the correct value for the consumption function, where a better guess will decrease the computation time.

#### 3.2.2 Policy function iteration

Sometimes the value functions can be a bit slow to converge, as it will only converge with the value there are therefore created faster methods, herunde the policy function iteration, which is about

#### 3.2.3 Projection methods

##### Solving for the policy rule

##### Collocation methods

##### Finite element methods

### 3.3 Stochastic discrete cake eating problem

#### 3.3.1 Value function iterations

### 3.4 Extension and conclusion

#### 3.4.1 Larger state spaces

# Savings and liquidity constraints

## Introduction

The paper seems to be people who have some liquidity constraint, as they do not make more than enough money to go by, why they are less able to invest an gain traction throughout their lifetime. This is also partly because it shown te be very hard to borrow for increased consumption, since theis would in theory worsen the families economic situation.

He seems to further investigate how different kinds of incomes and borrowing opportunities result in different levels of investments, and also how they at times result in less than optimal savings strategies.

## Savings and liquidity constraints with basic income

### The basic model

This chapter seemed to provide further basis that there was a problem if the consumer weren’t able to save above a certain threshold, they wouldn’t be able top have sustainable savings, and would then be regressing to their average income.

### Stationary serially stationary income

In this model, there is assumed an autoreggresive model, meaning that an increase in income in one period, would mean that there is a higher income in the next and vice verca.

## Savings and liquidity constraints with nonstationary income

Here he studies the savings and liquidity with nonstationary income, meaning that the income doesn’t have to go back to some mean.

### independent and identically distributed growth

Nothing I really understood.

### autocorrelated growth and the cycle

### individual behavior, noisy income and individual behavior

# Questions and answers

What is the Markov decision process?

* The Markov decision process is a maximization problem, where we wish to optimize the utility based on how we decide to react to the state space. This can then be done either in a finite horizon perspective or an infinite horizon.

What are the differences between the infinite and the finite horizon dynamic programming problems?

* The finite horizon is solved using dynamic programming, but where you start with the ending period, and then go back to see how we should do in the first period.
* With the infinite horizon however, there wouldn’t be the same constraints, as you would have infinite time in all periods. This is solved by guessing I think, though I am not entirely sure.

What does the v mean in the bellman equation

* It seems to imply the value with which we are attempting to maximize.
* V is the value function – not entirely sure what that exactly means
* Maybe it is the asset price?
* Value or the sum of all future values after accounting for the discounting.