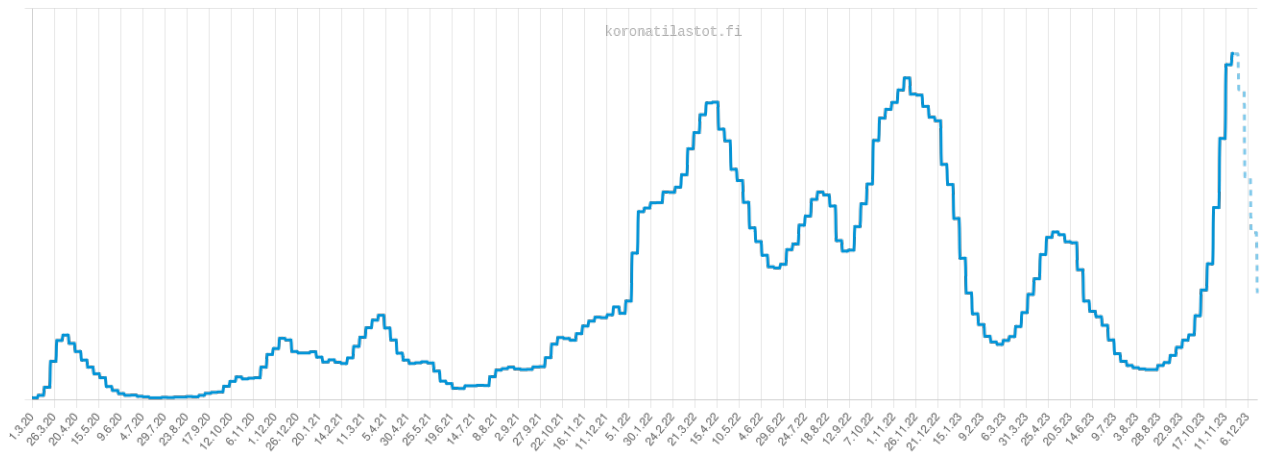# Decisions in the Time of Koala Ebola

Number of people hospitaized due to Corona 1.3.2020–6.12.2023 (source `www.koronatilastot.fi`).

Tommi Sottinen

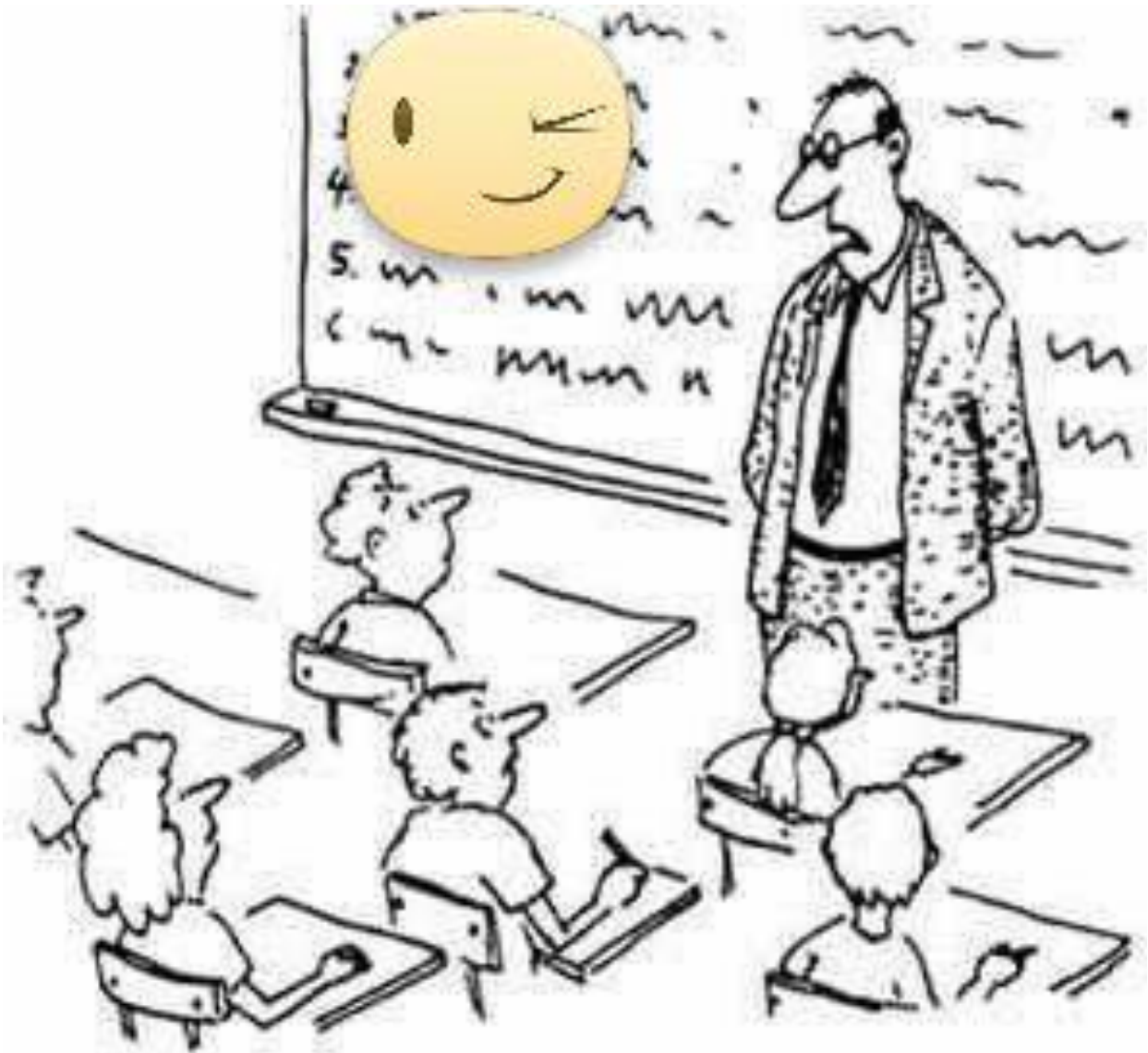`tommi.sottinen@uwasa.fi`

`https://www.uwasa.fi/~tsottine/orms2020/en/dtke.pdf`

May 16, 2024

Preface

"I expect you all to be independent, innovative, critical thinkers who will do exactly as I say!"

Welcome to the course **ORMS2020 Decisions under Uncertainty**! The course will be completed by doing one or two assignments. There is no final exam. There are no weekly exercises. There is no compulsory attendance of any kind. There are weekly guidance groups where help for the assignments are given. Attending the groups is recommended but not mandatory.

In this course we will use the GNU Octave software. At the writing of these notes, May 16, 2024, the latest stable version of GNU Octave is 9.1.0. If you have an older version of GNU Octave installed, it should work just fine. Also, if you want to use Matlab instead of GNU Octave, that should also work just fine.

Remark The PDF driver of some versions of the GNU Octave does not work. In such case you can use, e.g., JPG format for you pictures.

Remark For your assignments and for some examples in these notes you will need to have the home made GNU Octave functions: `scaling.m` and `dmatrix.m` in your current directory. They can be downloaded from the course web directory

> `https://lipas.uwasa.fi/~tsottine/orms2020/en/`.

or via the following direct links

- `https://lipas.uwasa.fi/~tsottine/orms2020/en/scaling.m`.
- `https://lipas.uwasa.fi/~tsottine/orms2020/en/dmatrix.m`.

I wish to thank PhLic Saara Lehto for helping naming the names in this booklet.

Vaasa May 16, 2024
T.S.

Tutorial Videos

All the tutorial videos (mkv format) of the course can be found in the course web directory
`https://lipas.uwasa.fi/~tsottine/orms2020/en/`.
The tutorial videos are titled CxEy.mkv for Chapter x Episode y. In this booklet, there are eight chapters and each chapter has three to six sections. Not each booklet section has a dedicated video, but some episodes span several sections. So the episodes and sections do not match. Sorry for the confusion!

Below are also the direct links to the videos

Part 0 Welcome to the course!

> **Chapter 0** Introductory Material
> - `https://lipas.uwasa.fi/~tsottine/orms2020/en/C0S1.mp4` (10 min)
> - `https://lipas.uwasa.fi/~tsottine/orms2020/en/C0S2.mp4` (10 min)

Part I Leading examples

> **Chapter 1** Brianna Brisk's Rain Gear
> - `https://lipas.uwasa.fi/~tsottine/orms2020/en/L1E1.mp4` (25 min)
> - `https://lipas.uwasa.fi/~tsottine/orms2020/fi/L1E2.mp4` (28 min)
> - `https://lipas.uwasa.fi/~tsottine/orms2020/fi/L1E3.mp4` (23 min)
> - `https://lipas.uwasa.fi/~tsottine/orms2020/en/L3E1.mp4` (21 min)
> - `https://lipas.uwasa.fi/~tsottine/orms2020/en/L3E1.mp4` (21 min)
>
> **Chapter 2** Eugene Equally-Brisk's Synchronized Rain Gear
> - `https://lipas.uwasa.fi/~tsottine/orms2020/en/L2E1.mp4` (19 min)
> - `https://lipas.uwasa.fi/~tsottine/orms2020/en/L2E2.mp4` (15 min)
> - `https://lipas.uwasa.fi/~tsottine/orms2020/en/L2E3.mp4` (11 min)
> - `https://lipas.uwasa.fi/~tsottine/orms2020/en/L3E1.mp4` (21 min)
> - `https://lipas.uwasa.fi/~tsottine/orms2020/en/L3E1.mp4` (21 min)

Part II Decision Matrix Techniques

> **Chapter 3** Decision Matrices and Decision Rules
> - `https://lipas.uwasa.fi/~tsottine/orms2020/en/L3E1.mp4` (21 min)
> - `https://lipas.uwasa.fi/~tsottine/orms2020/en/L3E1.mp4` (21 min)
> - `https://lipas.uwasa.fi/~tsottine/orms2020/en/L3E1.mp4` (21 min)
> - `https://lipas.uwasa.fi/~tsottine/orms2020/en/L3E1.mp4` (21 min)
> - `https://lipas.uwasa.fi/~tsottine/orms2020/en/L3E1.mp4` (21 min)
>
> **Chapter 4** Decision Matrix Calculations with GNU Octave

- https://lipas.uwasa.fi/~tsottine/orms2020/en/L4E1.mp4 (10 min)
- https://lipas.uwasa.fi/~tsottine/orms2020/en/L4E1.mp4 (10 min)
- https://lipas.uwasa.fi/~tsottine/orms2020/en/L4E1.mp4 (10 min)
- https://lipas.uwasa.fi/~tsottine/orms2020/en/L3E1.mp4 (21 min)

Part III Examples for Assignments

Chapter 5 Stefan Student's Fall
- https://lipas.uwasa.fi/~tsottine/orms2020/en/L5E1.mp4 (40 min)
- https://lipas.uwasa.fi/~tsottine/orms2020/en/L5E1.mp4 (40 min)
- https://lipas.uwasa.fi/~tsottine/orms2020/en/L5E1.mp4 (40 min)
- https://lipas.uwasa.fi/~tsottine/orms2020/en/L3E1.mp4 (21 min)

Chapter 6 Catherine Cato's Scientific Report
- https://lipas.uwasa.fi/~tsottine/orms2020/en/L6E1.mp4 (13 min)
- https://lipas.uwasa.fi/~tsottine/orms2020/en/L6E1.mp4 (13 min)
- https://lipas.uwasa.fi/~tsottine/orms2020/en/L6E1.mp4 (13 min)
- https://lipas.uwasa.fi/~tsottine/orms2020/en/L3E1.mp4 (21 min)

Part IV Assigments

Chapter 7 Assignment 1: Your Fall
- https://lipas.uwasa.fi/~tsottine/orms2020/en/L7E1.mp4 (13 min)

Chapter 8 Assignment 2: Ministry Report
- https://lipas.uwasa.fi/~tsottine/orms2020/en/L7E1.mp4 (13 min)

Contents

Part I

Leading Examples

Chapter 1

Brianna Brisk's Rain Gear

Problem Description

Brianna Brisk walks every day to her office and back. The journey is 10 km. Brianna has to decide what kind of gear to put on in the morning. The travel to work is not a problem, since the weather is obvious. The travel back from work is a problem, because Brianna lives is Vaasa where the weather can change a lot during a day.

Brianna has to decide between the following three alternatives: normal gear, an umbrella, or a rain coat. If the weather is dry it is annoying to carry an umbrella and a rain coat is always uncomfortable. If the weather is rainy but calm, a rain coat is a good choise but a bit sweaty: an umbrella would be better. If the weather is both rainy and windy, an umbrella is mostly useless and a rain coat is the best choice.

Decision Matrix

We build the decision matrix, a.k.a. the reward matrix, for Brianna.

Brianna mus choose between the following three alternatives (or actions)

a_1 = Normal gear.
a_2 = Umbrella.
a_3 = Rain coat.

Brianna decides to consider her altarnatives in the following threee scenarios:

s_1 = Dry weather.
s_2 = Rainy, but calm weather.
s_3 = Rainy and windy weather.

Because Brianna is weather forecast critical she does not bother to contemplate the probabilities of different weather scenarios. This means she does not care about the probabilities

p_1 = Probability of dry weather after the working day.
p_2 = Probability of rainy but calm weather after the working day.
p_3 = Probability of rainy and windy weather after the working day.

Brianna measures the consequences of her actions, a.k.a. her rewards, as subjective comforts that are numbers in the scale $0 \ldots 100$ (per cent). These rewards are based on the following facts:

- Normal gear is a good choice for dry weather, but the worst possible choice for rainy weather.
- Carrying an umbrella is slightly annoying and an umbrella is useless in windy weather.
- A rain coat is a safe choice for any weather, but it is slightly uncomfortable.

Because comfort is, in addition of being subjective, a very nebulous concept, Brianna measures it only on a four-point scale:

100 = Very comfortable.
80 = Relatively comfortable.
20 = Relatively uncomfortable.
0 = Very uncomfortable.

After contemplating, Brianna is left with the decision matrix (reward matrix)

	Weather		
Gear	Dry	Rain and calm	Rain and wind
Normal gear	100	0	0
Umbrella	80	80	0
Rain coat	20	20	20

Brianna, like all of us, makes her decision on an emotional basis. Brianna's feelings tells her to decide a_2 = Umbrella.

Next, Brianna will be a bit more analytical and consequently she may change her decision, or maybe not. If not, then she may have at least found a good argument for her decision, or at least a good excuse.

Decision Rules and Decision

Intuititvely, or based on her feelings, Brianna has made the choice a_2 = Umbrella. We obtain the decision matrix with value function, or decision function, V^{int}. Here 1 denotes the alternative we choose and 0 denotes an alternative we do not choose. We have also boldfaced the choice.

	Weather			Value
Gear	Dry	Rain and calm	Rain and wind	V^{int}
Normal gear	100	0	0	0
Umbrella	80	80	0	**1**
Rain coat	20	20	20	0

Here, formally, V^{int} is a decision rule or a value function that associates every alternative a_i with a value V_i^{int}.

In addition to her intuitive choice, Brianna wants to analyze her decision; its background and values. As background, Brianna needs to decide how she reacts to uncertainty. Brianna decides that she wants to minimize her regret.

Minimizing regret is a kind of if ifs and buts thinking. Regret is what you get if you do not choose the best possible choice for a given scenario. Formally, if R_{ij} is the reward associated with the altrenative a_i if the scenario s_j occurs, then the regret with the chosen alternative a_i and occurred scenario s_j is

$$K_{ij} = \max_i R_{ij} - R_{ij}.$$

Note that here $\max_i R_{ij}$ is the best possible reward that we could have obtained if we only knew that scenario s_j would occur.

So, Brianna Brisk has the following regret matrix:

Gear	Dry	Rain and calm	Rain and wind
		Regret due to weather	
Normal gear	0	80	20
Umbrella	20	0	20
Rain coat	80	60	0

Brianna wants to minimize the maximal regret. The maximal regrets V_i^{mr} for the alternatives a_i are

Gear	Dry	Rain and calm	Rain and wind	V^{mr}
		Regret due to weather		**Value**
Normal gear	0	80	20	80
Umbrella	20	0	20	**20**
Rain coat	80	60	0	80

So, in order to minimize her maximal regret, Brianna Brisk has to choose the alternative $a_2 =$ Umbrella.

It turned out that Brianna Brisk is consistent: her intuitive choice was the same as the choice arising from the minimax regret rule.

Sensitivity Analysis

Brianna's choice was based on her subjective comforts

$100 =$ Very comfortable.
 $80 =$ Relatively comfortable.
 $20 =$ Relatively uncomfortable.
 $0 =$ Very uncomfortable.

Here the comfort scale 100, 80, 20, 0 is very arbitrary. Why precisely these numbers? The extremes 0 and 100 are of course normal. They just fix the range. But the middle numbers 80 and 20 are very arbitrary. By keeping the scale symmteric, we replace the numbers 80 and 20 with a free parameter c in the following way (and replace 100 (%) with 1):

$$1 = \text{Very comfortable.}$$
$$c = \text{Relatively comfortable.}$$
$$1-c = \text{Relatively uncomfortable.}$$
$$0 = \text{Very uncomfortable.}$$

Note that the parameter c is in the interval $0.5\ldots1$. The parameter c tells us how Brianna weighs the extremes of her comfort. For $c = 0.5$ the alternatives relatively comfortable and relatively uncomfortable are the same. For $c = 1$ very comfortable and relatively comfortable are the same; ditto for relatively uncomfortable and very uncomfortable.

By using the parameter c we get the following decision matrix augmented with the intuitive choice V_i^{int}:

	Weather			Value
Gear	Dry	Rain and calm	Rain and wind	V^{int}
Normal gear	1	0	0	0
Umbrella	c	c	0	1
Rain coat	$1-c$	$1-c$	$1-c$	0

The parametrized regret matrix for Brianna and its maximal regrets is (recall that $c \geq 0.5$)

	Regret due weather			Value
Gear	Dry	Rain and calm	Rain and wind	V^{mr}
Normal gear	0	c	$1-c$	c
Umbrella	$1-c$	0	$1-c$	$\mathbf{1-c}$
Rain coat	c	$2c-1$	0	c

Because $c \geq 0.5$, we have $1-c \leq c$. Similarly, $c \geq 2c-1$. We see that Brianna Brisk's minimax regret choice is not at all sensitive on how she measures the extremes of her comfort.

Chapter 2

Eugene Equally-Brisk's Synchronized Rain Gear

Problem Description

Eugene Equally-Brisk is Brianna Brisk's spouse and collegue. He walks every day to his office with Brianna. Eugene's decision problem is the same as Brianna's, i.e., what gear to use for the walk: normal gear, an umbrella or a rain coat.

A new flavor in Eugene's problem is the goal to synchronize the gear with Brianna, i.e, is is possible to find a compromise so that both Brianna and Eugene use the same gear.

Eugene models the scenarios almost like Brianna. In addition to dry, rain and calm, rain and wind, there is warm weather. It is well known that it is never warm when it rains in Vaasa. This leaves Eugene with four scenarios (see below).

Decision Matrix

Eugene has the same alternatives as Brianna:

a_1 = Normal gear.
a_2 = Umbrella.
a_3 = Rain coat.

Eugene's scenarios are an extended version of Brianna's scenarios. We emphasize this with tree-like enumeration

$s_{1,1}$ = Dry warm weather.
$s_{1,2}$ = Dry cool weather.
s_2 = Rainy and calm weather.
s_3 = Rainy and windy weather.

Unlike Brianna, Eugene is not weather forecast critical. He is very interested in the probabilities

$p_{1,1}$ = Probability of dry and warm weather after the working day.
$p_{1,2}$ = Probability of dry and cool weather after the working day.
p_2 = Probability of rainy and calm weather after the working day.
p_3 = Probability of rainy and windy weather after the working day.

Decision Rules and Decision

Just like Brianna, Eugene basis his decision on subjective comfort. Unlike Brianna, Eugene is a extremist in the sense that everything is either very comfortable (value 1) or very uncomfortable (value 0) for him. The comfort matrix (reward matrix, decision matrix) for Eugene is

Gear	Weather			
	Dry and warm	Dry and cool	Rain and calm	Rain and wind
Normal gear	1	1	0	0
Umbrella	0	0	1	0
Rain coat	0	1	0	1

Intuitively, based on his feelings, Eugene thinks like Brianna $a_2 =$ Umbrella is a good choice. So, Eugene's reward matrix with intuitive values is

Gear	Weather				Value
	$s_{1,1}$	$s_{1,2}$	s_2	s_3	V^{int}
a_1	1	1	0	0	0
a_2	0	0	1	0	**1**
a_3	0	1	0	1	0

(We used the symbols a_i and s_j here for marginal reasons.)

Just like Brianna, Eugene want to understand his intuitive decision, its background and values. Eugene decides that he values uncertainty by using expected values: the value of an alternative a_i is its probability-weighted sum:

$$V_i^{\text{ev}} = \sum_j R_{ij} p_j,$$

where p_j is the probability of the scenario s_j, the summing index j runs through all the scenarios, and R_{ij} is the reward associated with the alternative a_i if the scenario s_j occurs. In the language of probability theory this means that a_i is a random variable and Eugene's value is the expectation

$$
\begin{aligned}
V_i^{\text{ev}} &= \mathbb{E}[a_i] \\
&= \sum_j a_i(s_j) \mathbb{P}[s_j] \\
&= \sum_j R_{ij} p_j.
\end{aligned}
$$

(We use square brackets for $\mathbb{E}$ and $\mathbb{P}$ purely for aesthetic reasons. There is nothing profound in this.)

In order to use the expected value decision rule, Eugene must know the probabilities p_j. In general, estimating probabilities is a difficult task in practice. Let us assume that these probabilities are simply given by the weather forecast as

$p_{1,1} = 0.05$
$p_{1,2} = 0.45$

$$p_2 = 0.25$$
$$p_3 = 0.25$$

(So, for example, the probability of rain is $p_2 + p_3 = 0.50$, the probability of dry and warm weather is on $p_{1,1} = 0.05$, and the probability for rain and calm is $p_2 = 0.25$.)

Calculating the value function V^{ev} is straightforward but cumbersome. We obtain the values

$$
\begin{aligned}
V_1^{ev} &= 1 \cdot p_{1,1} + 1 \cdot p_{1,2} + 0 \cdot p_2 + 0 \cdot p_3 \\
&= 0.05 + 0.45 \\
&= 0.50, \\
V_2^{ev} &= 0 \cdot p_{1,1} + 0 \cdot p_{1,2} + 1 \cdot p_2 + 0 \cdot p_3 \\
&= 0.25, \\
V_3^{ev} &= 0 \cdot p_{1,1} + 1 \cdot p_{1,2} + 0 \cdot p_2 + 1 \cdot p_3 \\
&= 0.45 + 0.25 \\
&= 0.70.
\end{aligned}
$$

Consequently, the Eugene's decision matrix augmented with the intuitive choice and the expected values is

| | Weather | | | | Value | |
Gear	$s_{1,1}$ (0.05)	$s_{1,2}$ (0.45)	s_2 (0.25)	s_3 (0.25)	V^{int}	V^{ev}
a_1	1	1	0	0	0	0.50
a_2	0	0	1	0	**1**	0.25
a_3	0	1	0	1	0	**0.70**

We see that Eugene's feeling based choice a_2 = Umbrella is not consistent with the expected value choice a_3 = Rain coat. Now, Eugene is left with two choices: either stick with his intuitive choice, or change the choice to the maximal expected value choice. Eugene decides to be consistent with expected value and change his choice to a_3 = Rain coat.

Combining Decision Rules

Brianna Brisk and Eugene Equally-Brisk chose different alternatives. Or course, nothing prevents then to choose different gears for their walks, but for some reason they want to choose the same gear. So, the problem is to find a reasonable compromise.

There were three alternatives: a_1, a_2 and a_3. Brianna chose the alternative a_2 and Eugene chose the alternative a_3. Now one might think that we only need to choose between the alternatives a_2 and a_3. This is not true, however. It dould be that the alternative a_3 is simply repulsive for Brianna and the alternative a_2 is equally repulsive for Eugene. Furthermore, it could in principle be that the alternative a_1 could be a reasonable alternative for both Brianna and Eugene. So, we need to be careful in finding the compromise.

Brianna Brisk and Eugene Equally-Brisk both considered subjective comfort. This is not important in finding the compromise. We can base the compromise purely on their respective value functions $V(B)$ (for Brianna) and $V(E)$ (for Eugene). All that has to be taken care of that both $V(B)$ and $V(E)$ are the same type and in the same scale. By type we mean either maximization or minimization. Let us transform both $V(B)$ and $V(E)$ to be of maximization

type. Now, $V(E)$ is the expected value, which is already maximization type. So, there is nothing to transform here. However, $V(B)$ is regret type, which is minimization type. Thus, we have to transform it into maximization type. The transformation is simple: we transform

$$V_i(B) \quad \longleftarrow \quad -V_i(B).$$

Now $V(B)$ is maximization type.

Next, we have to ensure that both $V(B)$ and $V(E)$ are in the same scale. To do so we apply to both of them the affine transformation

$$V_i \quad \longleftarrow \quad \frac{V_i - \min_k V_k}{\max_k V_k - \min_k V_k}.$$

Indeed, this transformation does not change the preferences for either Brianna or Eugene, but it ensures that the value functions are on the interval $0\ldots1$ with maximal value 1 and minimal value 0. By not changing the preferences we mean that if $V_{i_1} \le V_{i_2}$ before the affine transformation, so is $V_{i_1} \le V_{i_2}$ after the affine transformation

Remark If the value function V is constant, then we have a division-by-zero problem at our hand. To avoid this, we set $V \equiv 1$ for constant value functions V.

The next GNU Octave code will be used later to automatically scale value functions.

```
1  function V_scaled = scaling(V)
2  %% Funktion V_scaled = scaling(V) returns the scaled vector
3  %% V_scaled = (V−min(V))/(max(V)−min(V)) if max(V) > min(V), and the constant
4  %% vector V_scaled = [1 1 1 ... 1]' otherwise.  The vector V_scaled is a
5  %% column vector.
6
7      %% Make V to be a coulumn vector.
8      if size(V,1) < size(V,2)
9          V = V';
10     end
11     %% V is a constant vector.
12     if max(V) == min(V)
13         V_scaled = ones(length(V),1);
14     %% V is not a constant vector.
15     else
16         V_scaled = (V−min(V)) / (max(V)−min(V));
17     end
18 end
```

https://www.uwasa.fi/~tsottine/orms2020/en/scaling.m

Let us than recall Brianna's regret matrix with the associated maximal regret value function

Gear	Weather s_1	s_2	s_3	Value $V^{mr}(B)$
a_1	0	80	20	80
a_2	20	0	20	**20**
a_3	80	60	0	80

Thus all that matters are the maximal regret values are

$$\begin{aligned}
V_1^{mr}(B) &= 80, \\
V_2^{mr}(B) &= 20, \\
V_3^{mr}(B) &= 80.
\end{aligned}$$

15

Similarly, for Eugene, all that matters are the expected values

$$
\begin{aligned}
V_1^{\text{ev}}(E) &= 0.50, \\
V_2^{\text{ev}}(E) &= 0.25, \\
V_3^{\text{ev}}(E) &= 0.70.
\end{aligned}
$$

Let us denote $V_i^{\text{mj}} = -V_i^{\text{mr}}$ (mj = maximize joy). Then, for Brianna,

$$
\begin{aligned}
V_1^{\text{mj}}(B) &= -80, \\
V_2^{\text{mj}}(B) &= -20, \\
V_3^{\text{mj}}(B) &= -80.
\end{aligned}
$$

That is, we just transformed Brianna's value function into a maximization form. Next, we transform Brianna's joy values onto the interval $0\ldots 1$ by using the affine transformation. We obtain

$$
\begin{aligned}
V_1(B) &= \frac{-80 - (-80)}{-20 - (-80)} \\
&= 0, \\
V_2(B) &= \frac{-20 - (-80)}{-20 - (-80)} \\
&= 1, \\
V_3(B) &= \frac{-80 - (-80)}{-20 - (-80)} \\
&= 0.
\end{aligned}
$$

Similarly, the affine transformation to Euqene yields

$$
\begin{aligned}
V_1(E) &= \frac{0.50 - 0.25}{0.70 - 0.25} \\
&= 0.56, \\
V_2(E) &= \frac{0.25 - 0.25}{0.70 - 0.25} \\
&= 0, \\
V_3(E) &= \frac{0.70 - 0.25}{0.70 - 0.25} \\
&= 1.
\end{aligned}
$$

Now, Brianna's and Eugene's value functions are of the same type and on the same scale. Thus it makes sense to compare them. The comparisons are give in the following table:

Gear	Values	
	$V(B)$	$V(E)$
a_1	0	0.56
a_2	1	0
a_3	0	1

Now, we can combine Brianna's and Eugene's values by choosing a weight w from the interval $0\ldots1$ and set a new value function $V(w)$ to be

$$V_i(w) \;=\; wV_i(B) + (1-w)V_i(E)$$

This means that Brianna's point of view gets the weight w and Eugene's point of view gets the complementary weight $1-w$. In the next table we have calculated the compromise value function $V(w)$ with weights $w = 0.75, 0.50, 0.25$

			Values		
Gear	$V(B)$	$V(E)$	V(0.75)	V(0.50)	V(0.25)
a_1	0	0.56	0.14	0.28	0.42
a_2	1	0	**0.75**	**0.50**	0.25
a_3	0	1	0.25	**0.50**	**0.75**

We see that in the equal weight situation $w = 0.5$ both the alternatives a_2 and a_3 are equally optimal. If Brianna gets a greater weight in the compromise value function, then Brianna's optimal alternative a_2 will be compromise optimal, and if Eugene gets greater weight then Eugene's optimal alternative a_3 will be compromise optimal.

Sensitivity Analysis

The analysis of $V(w)$ in the end of the previous section was far from complete. Also, we could consider the sensitivity of the compromise value function if the Brianna's comfort matrix has parameter c. This is what we do this, i.e., we look how sensitive the problem is with respect to the parameters w and c.

Brianna's regret matrix and maximal regrets are

	Regret due weather			Value
Gear	Dry	Rain and calm	Rain and wind	$V^{\mathrm{mr}}(B;c)$
Normal gear	0	c	$1-c$	c
Umbrella	$1-c$	0	$1-c$	$\mathbf{1-c}$
Rain coat	c	$2c-1$	0	c

By transforming into maximization by

$$V_i \;\longleftarrow\; -V_i$$

and by affinely scaling

$$V_i \;\longleftarrow\; \frac{V_i \;-\; \min_k V_k}{\max_k V_k \;-\; \min_k V_k},$$

we obtain the values

$$
\begin{aligned}
V_1(B;c) &= \frac{-c-(-c)}{-(1-c)-(-c)} \\
&= 0, \\
V_2(B;c) &= \frac{-(1-c)-(-c)}{-(1-c)-(-c)} \\
&= 1, \\
V_3(B;c) &= \frac{-c-(-c)}{-(1-c)-(-c)} \\
&= 0.
\end{aligned}
$$

We see that the parameter c disappears in the scaling. So, all that is left is the compromise weight w.

Now,

$$
\begin{aligned}
V_1(w) &= 0.56(1-w), \\
V_2(w) &= w, \\
V_3(w) &= (1-w).
\end{aligned}
$$

We see that a_1 is never the optimal choice. Also, a_2 is optimal if and oly if $w \geq 0.5$, and a_3 is optimal if and only if $w \leq 0.5$.

Part II

Decision Matrix Techniques

Chapter 3

Decision Matrices and Rules

Static Setting

We only consider decision problems that are **static** in their nature: the decision is done only once and then we see the consequences. The decisions do not affect the scenarios or their probabilities. In principle there is no time dimension or cause and effect.

The simplest way to understand decision matrix technique is the following: first Tu Ipse (you self) chooses an alternative a_i from the set $\{a_i ; i \in I\}$, and then Fortuna Brevis (the goddess of chance) chooses a scenario s_j from the set $\{s_j ; j \in J\}$ with probability p_j. In this game Tu Ipse gets the reward R_{ij}.

Remark In the language of probability theory we may say that the scenarios form the probability space $\Omega = \{s_j ; j \in J\}$. The alternatives a_i, $i \in I$, are random variables $a_i \colon \Omega \to \mathbb{R}$ with values $a_i(s_j) = R_{ij}$ with probabilities

$$\mathbb{P}\big[a_i = R_{ij}\big] \;=\; p_j,$$

or equally,

$$p_j \;=\; \mathbb{P}[s_j].$$

The **value function** defining a decision rule is a function $V \colon I \to \mathbb{R}$ can in principle be anything that associates the alternative a_i with a value V_i (some value functions can be silly). The **decision rule** is either minimizing or maximizing the value function. So, strictly speaking decision rules and value functions are different things, but we will use them synonymously anyway.

If the decision rule is of maximization form, it means that the decision maker will choose such an alternative $a^* = a_{i*}$, for which

$$V_{i*} \;=\; \max_i V_i.$$

In other words

$$i^* \;=\; \operatorname*{argmax}_i V_i.$$

It should be noted that there may be many different optimal alternatives a^*. If there are many optimizers, then just one is chosen at will.

If the decision rule is of minimization form, it means that the decision maker will choose such an alternative $a^* = a_{i*}$, for which

$$V_{i*} = \min_i V_i.$$

In other words

$$i^* = \operatorname*{argmin}_i V_i.$$

Again, the minimizer a^* may not be unique. In the case of non-uniqueness one optimizer is chosen at will.

Non-Stochastic Decision Rules

Formally a decision rule (which we equate with the value function V) depends on the decision problem via the reward matrix R and the probabilities p for the scenarios (columns of the reward matrix). If the decision rule is independent of the scenario probabilities p we call the decision rule **non-stochastic**. (Stochastic means random in modern English, it comes from the ancient Greek word that means to aim or to guess).

Below we will briefly introduce some popular non-stochastic decision rules.

Optimist is Gladstone Gander, who thinks that Fortuna Brevis is his friend: whichever alternative a_i the optimist chooses, Laby Fortuna will choose the best possible scenario $s_j = s_j(i)$ for the chosen alternative a_i. Formally this means that the optimist's value function to be maximized is

$$V_i^{\mathrm{opt}} = \max_j R_{ij}$$

Because the optimist's selection $a^* = a_{i*}$ satisfies

$$V_{i*}^{\mathrm{opt}} = \max_i \max_j R_{ij},$$

is the optimist rule also called the **maximax rule**.

Pessimist is Donald Duck, who thinks Fortuna Brevis is his enemy: whichever alternative a_i the pessimist chooses, Lady Fortuna will choose the worst possible scenario $s_j = s_j(i)$ for the chosen alternative a_i. Formally this means that the pessimist's value function to be maximized is

$$V_i^{\mathrm{pess}} = \min_j R_{ij}.$$

Because the pessimist's selection $a^* = a_{i*}$ satisfies

$$V_{i*}^{\mathrm{pess}} = \max_i \min_j R_{ij},$$

is the pessimist rule also called the **maximin rule**.

The optimist and the pessimist are in some sense natural opposites. To form something in between, we can take a weight w from the interval $0 \ldots 1$ and combine the optimist and the

pessimist rules. This is called the **Hurwicz rule**. Formally in Hurwicz rule the value function to be maximized is

$$V_i^{\text{Hur}}(w) = w \cdot \max_j R_{ij} + (1-w) \cdot \min_j R_{ij}$$
$$= w \cdot V_i^{\text{opt}} + (1-w) \cdot V_i^{\text{pess}}.$$

The free parameter w here is called the **degree of optimism** of the decision maker. (Of course in practice almost no-one knows his/her/their degree of optimism, but it can be estimated in certain ways. To explain these ways are beyond the scope of this course.)

Regret averse decision maker who wants to minimize the maximal regret for all scenarios. The old wisdom says that "it pays to be pessimist, for you shall have no regrets". This is not true! The regret averse rule is different to pessimist rule. To apply the regret averse rule one has to first calculate the **regret matrix** $K = [K_{ij}]_{i \in I, j \in J}$. Here K_{ij} tells how much the decision a_i regrets us when the scenario s_j occurs. Formally

$$K_{ij} = \max_i R_{ij} - R_{ij}.$$

So, R_{ij} is the reward for choosing a_i if s_j occurs and $\max_i R_{ij}$ is the best possible reward if s_j occurs. Of course, we do not know which s_j occurs, but the regret is there, just the same. The regret averse decision maker wants to minimize the maximal possible regret. So, the regret averse value function to be minimized is

$$V_i^{\text{mr}} = \max_j K_{ij}$$
$$= \max_j \left[\max_i R_{ij} - R_{ij} \right].$$

The regret for the optimal decision $a^* = a_{i*}$ is therefore

$$V_{i*}^{\text{mr}} = \min_i \max_j K_{ij}$$
$$= \min_i \max_j \left[\max_i R_{ij} - R_{ij} \right].$$

Consequently, the regret averse rule is also called the **minimax regret rule**.

The rules we have seen so far — the optimist, the pessimist, Hurwicz, and regret averse – were all non-stochastic. They did not care about the probabilities p_j, $j \in J$, of the scenarios s_j, $j \in J$. This is both a good and a bad thing. The good thing about it is that we do not have to estimate the probabilities, which can be difficult or outright impossible. The bad thing is that they treat all the scenarios equally even if some scenarios may be typical and other scenarios may be extremely rare.

Finally, we present a decision rule that could be called **semi-stochastic**. This is the so-called **Laplace rule**. It is based on the idea that all the scenarios s_j, $j \in J$, are — more or less – equally likely. Suppose that there are m scenarios. Then the value function to be maximized in the Laplace rule is the scenario-wise average

$$V_i^{\text{Lap}} = \frac{1}{m} \sum_j R_{ij}.$$

The optimal decision in the Laplace rule satisfies

$$V_{i*}^{\text{Lap}} = \frac{1}{m} \max_i \sum_j R_{ij}.$$

Consequently omitting the constant m, the Laplace rule is also called the **maxsum rule**.

Stochastic Decision Rules

Unlike the non-stochastic rules, the **stochastic rules** depend not only on the reward matrix R but also on the scenario probabilities p.

By far the most natural stochastic rule is the **expected value rule**, which is to maximize the value function

$$V_i^{\text{ev}} \;=\; \sum_j R_{ij} p_j.$$

In the language of probability theory this rule meas that the value of the random variable a_i in the decision making is its **expectation**:

$$
\begin{aligned}
V_i^{\text{ev}} \;&=\; \mathbb{E}[a_i] \\
&=\; \sum_j a_i(s_j) \mathbb{P}[s_j] \\
&=\; \sum_j R_{ij} p_j.
\end{aligned}
$$

The optimal decision $a^* = a_{i*}$ in the expectation rule satisfies

$$V_{i*}^{\text{ev}} \;=\; \max_i \mathbb{E}[a_i].$$

Consequently, the expected value rule is sometimes called the **maxiE rule**.

Remark If all the scenarios s_j, $j \in J$, are equally probable, then the expected value rule is simply the Laplace rule.

Remark The expected value rule is also called the **risk neutral rule**. Indeed, the decision maker who follows the expected value rule is indifferent between the following two choices:

(i) a certain reward of 1 €.
(ii) with probability 0.5 a reward of 2 € and with probability 0.5 nothing.

Most of us are probably indifferent with these choices, or indeed indifferent of the game altogether. The gist of the question is in scaling. A risk neutral decision maker is also indifferent between the choices

(i) a certain reward of 100 000 €.
(ii) with probability 0.5 a reward of 200 000 € and with probability 0.5 nothing.

Now most of us will probably choose the alternative (i). This choice corresponds to **risk aversion**. Choosing (ii) would correspond to **risk seeking**.

To model risk aversion and seeking one usually uses **utility functions**. The idea is that we still use expected value as the decision rule but we do not apply it to the rewards R, but to the so-called utilities U that are calculated from the reward matrix R by using a so-called utility function $U_{ij} = u(R_{ij})$. Here $u \colon \mathbb{R} \to \mathbb{R}$. If the **marginal utility**, i.e. the derivative u' of the utility function u is decreasing, is the decision maker risk averse. In the same way, if u' is increasing the decision maker risk seeking. In general, the utility function of a decision maker can exhibit both risk averse and risk seeking behavior for different values of the reward.
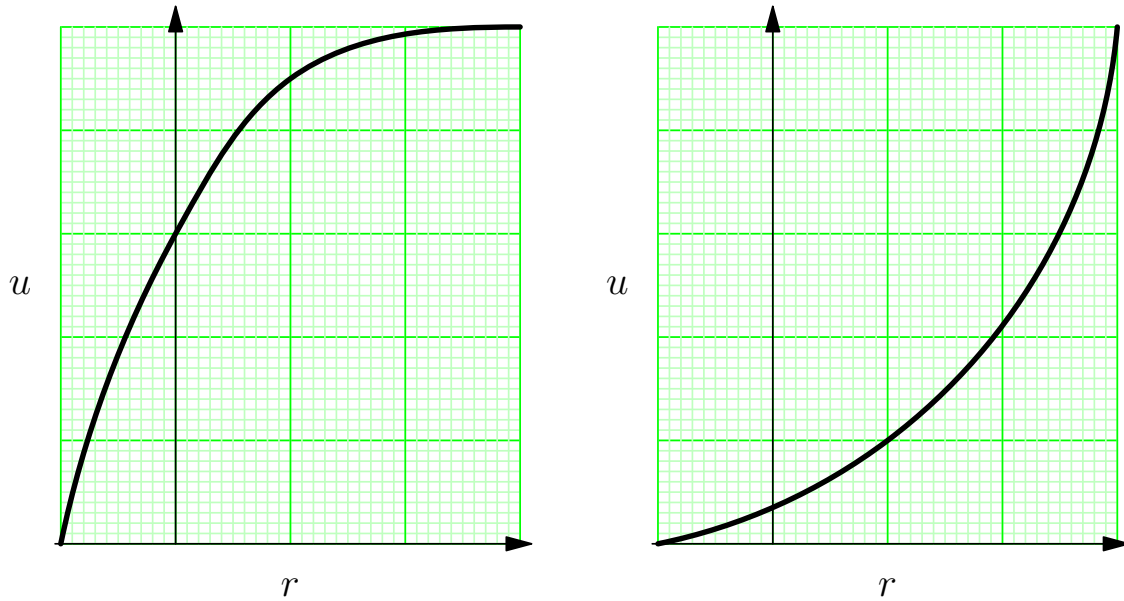
Risk averse utility function on the left and risk seeking utility function on the right.

If the utility function u is given then the value function to be maximized associated with the **utility rule** is

$$V_i^{\text{util}} \;=\; \sum_j u\left(R_{ij}\right) p_j.$$

In the language of the probability theory this means that the value of the random variable a_i is its expected utility:

$$
\begin{aligned}
V_i^{\text{util}} \;&=\; \mathbb{E}\left[u(a_i)\right] \\
&=\; \sum_j u\left(a_i(s_j)\right) \mathbb{P}[s_j] \\
&=\; \sum_j u\left(R_{ij}\right) p_j \\
&=\; \sum_j U_{ij}\, p_j.
\end{aligned}
$$

The value of the optimal decision $a^* = a_{i^*}$ in the utility rule is

$$V_{i^*}^{\text{util}} \;=\; \max_i \mathbb{E}\left[u(a_i)\right].$$

Therefore the utility rule is sometimes called the **maxiEU rule**.

The expected utility rule is the gold standard of decision making. This follows from the remarkable results of Von Neumann and Morgenstern, who showed that a decision maker is rational if and only if he/she/they follow an expected utility rule. To go into details of that theory is beyond the scope of this course.

Now we are faced with the problem of finding the utility function u. As in the case of the degree of otimism for the Hurwicz rule, almost no-one know his/her/their utility function. A one solution is to build a parametrized model for the utility u and then provide a way to estimate the parameter. This is what we do next.

We assume that the decision maker is risk averse, and his/her/their absolute risk aversion is constant. This is also called the CARA assumption, CARA = **Constant Absolute Risk Aversion**. This means that the risk aversion

$$-\frac{u''(r)}{u'(r)} = \text{constant.}$$

This is a second order differential equation that can be solved in many ways. It turns out that the solution is

$$u(r) = 1 - e^{-r/r_0},$$

where r_0 is a free parameter. We call this parameter r_0 the **risk tolerance**. To estimate the risk tolerance we note that it is the value of indifference for the following game: what is the amount of modey r_0 € that you are indifferent with the alternatives

(i) You get nothing.
(ii) You get r_0 € with probability 0.5, but lose $r_0/2$ € with probability 0.5.

It is obvious that the more risk averse the decision maker is, the smaller the risk tolerance r_0 is. The expected value rule is a limiting case of the CARA rule, where the risk tolerance is the limit $r_0 = \infty$.

Below is a picture that illustrates the CARA utility function with different risk tolerances r_0.
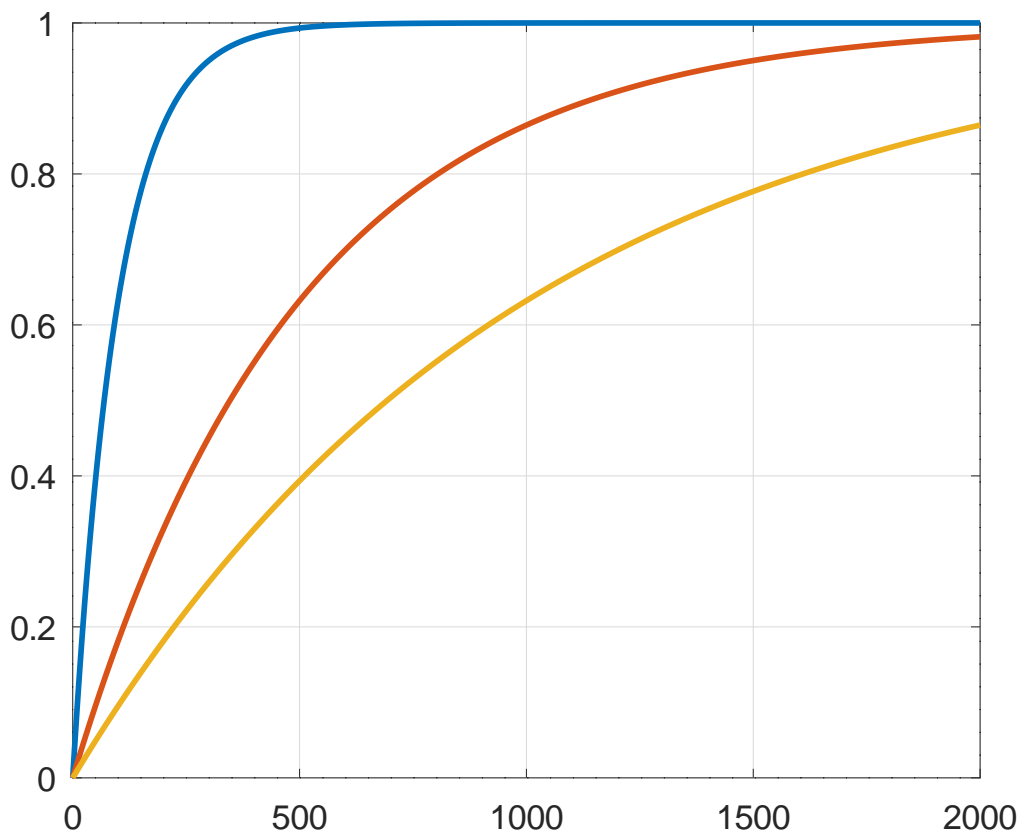
CARA function with $r_0 = 100$ (blue), $r_0 = 500$ (red) and $r_0 = 1000$ (yellow).

Combining Decision Problems

To combine two decision problems we only need their respective value functions. However, the value functions must be of the same type (maximization or minimization: we choose maximization in this course). A minimization type value function can be transformed into a maximization type value problem with the simple transformation

$$V_i \quad \longleftarrow \quad -V_i$$

Also the value functions must be on the same scale. We use the affine scale transformation

$$V_i \quad \longleftarrow \quad \frac{V_i - \min_k V_k}{\max_k V_k - \min_k V_k}.$$

to scale the value function on the interval $0 \ldots 1$ so that the best value will be 1 and the worst value will be (if possible) 0. (If $\max_k V_k = \min_k V_k$, we have a division-by-zero problem. In this case we set $V_i = 1$ for all i.) We have a home-made GNU Octave function `scaling` that does the affine scaling automatically. The m-file `scaling.m` defining the function `scaling` can be downloaded here:

https://www.uwasa.fi/~tsottine/orms2020/en/scaling.m

Most of our m-files in this booklet uses the affine scaling function `scaling`. So, it is a good idea to download the file `scaling.m` to your GNU Octave working directory, if you have not done so already. Another home-made GNU Octave function that is used repeatedly in the m-files in this booklet is `dmatrix` that solves decision matrix problems automatically. The function `dmatrix` is defined in the m-file `dmatrix.m` that can be downloaded here:

https://www.uwasa.fi/~tsottine/orms2020/en/dmatrix.m

Let V^1 and V^2 be two value functions that are both of the maximization type and both are affinely scaled to the same scale. Then we can choose a **compromise parameter** w from the interval $0 \ldots 1$ to combine the value functions V^1 and V^2 into a new combined, or compromise, value function $V(w)$:

$$V(w) \quad = \quad wV^1 + (1-w)V^2.$$

So, the combined value function $V(w)$ is a compromise of V^1 and V^2, where V^1 is given the weight w and V^2 is given the weight $1-w$.

Chapter 4

Decision Matrix Calculus with GNU Octave

Installing GNU Octave

GNU Octave is an open source version of the celebrated matrix calculus software Matlab. It can be downloaded from the address https://octave.org/download. In making these lectures the latest (stable) version of GNU Octave is 9.1.0, but older versions should work just fine.

Installing GNU octave to Windows and Linux is simple. MacOS installation is, however, a bit complicated. The easiest way to install GNU Octave is to install an old version 6.2.0 via Octave.app: https://octave-app.org/Download.html.

You can also use GNU Octave online in the address https://octave-online.net/. I do not recommend the online version.

Once the GNU Octave is installed, you should have (on your desktop) two icons: Octave CLI and Octave GUI. CLI stands for Command Line Interface and GUI stands for Graphical User Interface. I strongly recommend using the GUI. After lauching the Octave GUI you should have a screen like this (I am using version 7.1.0, so your screen may be slight different):
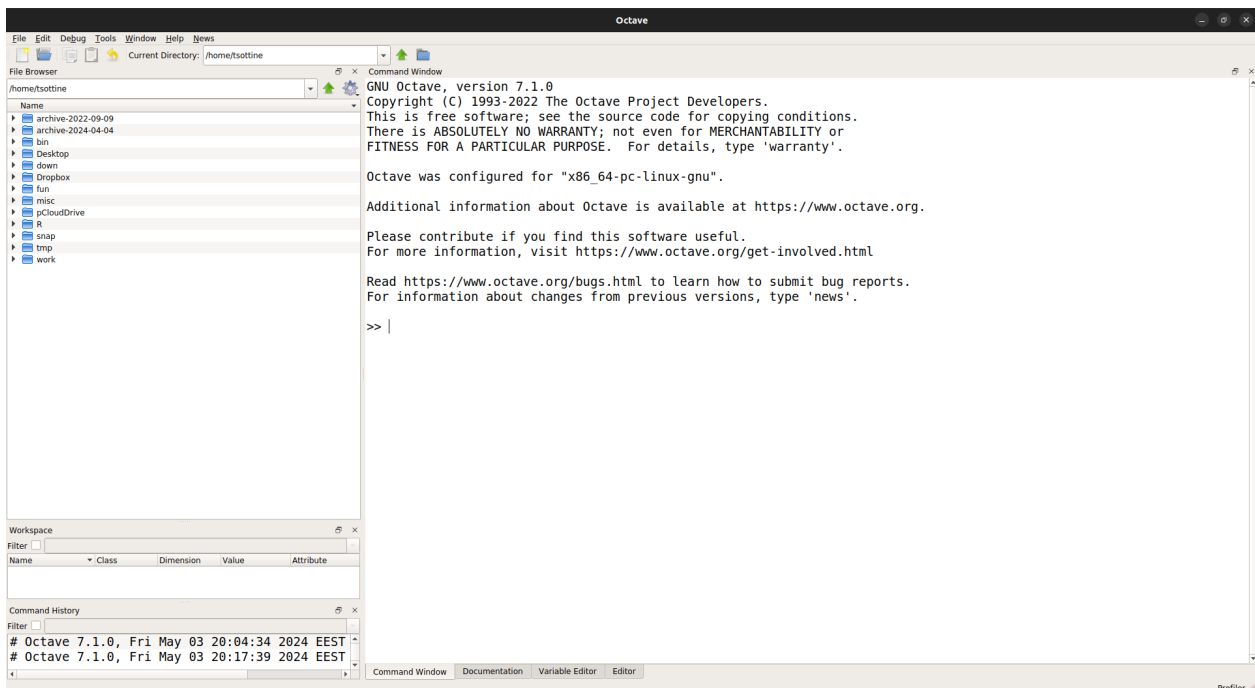
This course is not a programming course as such, and it is not strictly mandatory to use GNU Octave. You are free to use any tool you wish, like Excel (if you are a masochist), Matlab,

Python, Julia, C, pen and paper, or whatever. It is my opinion that GNU Octave is the best and easiest tool for the purposes of this course, however. Also, I have made a function `dmatrix` that calculates decision problems with GNU Octave automatically.

The Octave m-files of this course are written in such a way that they should also work with Matlab, if you want to use Matlab instead of GNU Octave. I have not tested them with Matlab, though.

Function `dmatrix` for Decision Matrices

The m-file `dmatrix.m` defines a function

```
dresult = dmatrix(dproblem),
```

that is solves decision matrix problems. (In the m-files of this booklet we usually write `dp` and `dr` instead of `dproblem` and `dresult`.)

The function `dmatrix` takes in a structure type variable `dproblem` that defines the decision matrix problem. It has the following fields:

`drule` is a string that determines the decision rule to be used. The supported decision rules are

`"opt"`	optimist
`"pess"`	pessimist
`"Hur"`	Hurwicz rule
`"regret"`	regret aversion
`"Lap"`	Laplace rule
`"ev"`	expected value rule
`"cara"`	constant absolute risk aversion rule

The field `drule` is obviously mandatory. Note also that small and capital letters are not the same for GNU Octave.

`dmat` is the decision matrix, or the reward matrix whose rows are the alernatives and the columns are the scenarios. The field `dmat` is obviously mandatory.

`odegree` the the degree of optimism in the Hurwicz rule. This field is only needed if the decision rule is the Hurwicz rule.

`prob` is a **column vector** giving the probabilities of the scenarios. This field is needed only in the case for the decision rules `"ev"` and `"cara"`.

`rtol` is the risk tolerance only needed with the `"cara"` rule.

The function `dmatrix` returns a structure type variable `dresult` that always has the following fields:

`dvalues` is a vector giving the values of the decisions.
`doptim` is the index of the optimal decision.

In addition, if the decision rule is `"regret"` the `dresult` has the field `rmat` that contains the regret matrix, and if the decision rule is `"cara"` then the `dresult` has the field `umat` that contains the utility matrix.

The reason to use structure type variables is that in GNU Octave one does not have to define fields that are unnecessary. Also, if one defines fields that are not used, it does not cause any problems.

Below is the listing for the function file `dmatrix.m`:

```
1  function dresult = dmatrix(dproblem)
2  %% Function dresult = dmatrix(dproblem) returns a struct type variable dresult,
3  %% whose field dvalues tells the values of the decisions and the field doptim
4  %% tells the index of the optimal decision.  If the decision rule if minimax
5  %% regret, i.e. "regret", then the output field rmat will be the regret matrix.
6  %% If the decision rule is constant absolute risk aversion, or "cara", then the
7  %% output field umat returns the utility matrix.
8  %%
9  %% The argument dproblem is also a struct variable.  Its fields are:
10 %%
11 %% dmat is the decision (or reward) matrix.
12 %%
13 %% drule is the string that describes the decision rule.  Supported rules are
14 %% "opt" (optimist), "pess" (pessimist), "Hur" (Hurwicz), "regret" (minimax
15 %% regret), "Lap" (Laplace), "ev" (expected value), "cara" (constant absolute
16 %% risk aversion).
17 %%
18 %% odegree is the degree of optimism in Hurwicz rule.
19 %%
20 %% prob is the column vector containing the probabilites of the scenarios,
21 %%
22 %% rtol is the risk tolerance related to the "cara" rule.
23
24     %% Lyhennysmerkintoja
25     R = dproblem.dmat;         %% Decision matrix
26     s = dproblem.drule;        %% Decision rule
27     I = 1:size(R,1);           %% R's rows
28     J = 1:size(R,2);           %% R's columns
29
30     if strcmp(s,"opt")
31         dresult.dvalues = max(R') ';
32         [tmp, dresult.doptim] = max(dresult.dvalues);
33     elseif strcmp(s,"pess")
34         dresult.dvalues = min(R') ';
35         [tmp, dresult.doptim] = max(dresult.dvalues);
36     elseif strcmp(s,"Hur")
37         w = dproblem.odegree;
38         dresult.dvalues = w*max(R') ' + (1-w)*min(R') ';
39         [tmp, dresult.doptim] = max(dresult.dvalues);
40     elseif strcmp(s,"regret")
41         %% Build the regret matrix
42         for i=I
43             for j=J
44                 K(i,j) = max(R(:,j)) - R(i,j);
45             end
46         end
47         dresult.dvalues = max(K') ';
48         [tmp, dresult.doptim] = min(dresult.dvalues);
49         dresult.rmat = K;
50     elseif strcmp(s,"Lap")
51         dresult.dvalues = (1/size(R,2))*sum(R') ';
52         [tmp, dresult.doptim] = max(dresult.dvalues);
53     elseif strcmp(s,"ev")
```

```
54        p = dproblem.prob;
55        dresult.dvalues = R*p;          %% p has to be a clolumn vector!
56        [tmp, dresult.doptim] = max(dresult.dvalues);
57    elseif strcmp(s,"cara")
58        p = dproblem.prob;
59        r0 = dproblem.rtol;
60        %% Build the utility matrix
61        for i=I
62            for j=J
63                U(i,j) = 1 - exp(-R(i,j)/r0);
64            end
65        end
66        dresult.dvalues = U*p;          %% p has to be a column vector!
67        [tmp, dresult.doptim] = max(dresult.dvalues);
68        dresult.umat = U;
69    end
70 end
```

https://www.uwasa.fi/~tsottine/orms2020/en/dmatrix.m

We will explain the inner workings of the function dmatrix in a video lecture. It is, however, not necessary to understand how the function dmatrix works. It is enough to know how to use it.

In the next sections and chapters we will demonstrate how to use the function dmatrix via examples.

Finally, a word of warning: The function dmatrix has no input checking whatsoever. It works with the maxim bullshit in – bullshit out. So, if the input variable dproblem is not properly defined, the output variable dresult may be nonsense! Caveat emptor!

Using Function `dmatrix` for Brianna Brisk and Eugene Equally-Brisk

We investigate Brianna's and Eugene's examples of Chapter 1 and Chapter 2 by using the function dmatrix. Note that to use these examples you must have the m-file dmatrix.m downloaded in your GNU Octave current directory. You will also need the m-file scaling.m in you current directory. Both can be downloaded from the course web directory

https://lipas.uwasa.fi/~tsottine/orms2020/en/.

or via the following direct links

- https://lipas.uwasa.fi/~tsottine/orms2020/en/scaling.m.
- https://lipas.uwasa.fi/~tsottine/orms2020/en/dmatrix.m.

Brianna Brisk's Decision

The m-file example_bb.m calculates Brianna's optimal decision with mid-comport degree $c = 80\%$. The actual calculations are done in the lines 10–22. The lines 1–9 are simply comments that are there for the readers' comfort. Finally, in the lines 24–41 we have listed the output of the m-file itself.

The m-file example_bb.m is listed below:

```
 1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 2 %%
 3 %% FILE: example_bb.m
 4 %%
 5 %% Brianna Brisk's problem with the function dmatrix.
 6 %%
 7 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 8
 9 %% Mid-comfort degree
10 c = 0.8;
11
12 %% Decision matrix
13 R = [  1     0     0;
14        c     c     0;
15      1-c   1-c   1-c];
16
17 %% Decision problem (dp) for the function dmatrix
18 dp.dmat = R;
19 dp.drule = "regret";
20
21 %% Decision result (dr)
22 dr = dmatrix(dp)
23
24 %% Running the file we get the output dr
25 %%>> example_bb
26 %%dr =
27 %%
28 %%  scalar structure containing the fields:
29 %%
30 %%    dvalues =
31 %%
32 %%       0.8000
33 %%       0.2000
34 %%       0.8000
35 %%
36 %%    doptim = 2
37 %%    rmat =
38 %%
39 %%            0    0.8000    0.2000
40 %%       0.2000         0    0.2000
41 %%       0.8000    0.6000         0
```

In the example_bb.m the first comment block 1–7 is simply a title. The comment sign "%" beginning a line makes the line a comment meaning that the GNU Octave interpreter will simply ignore it. These lines are there only for the readers' convenience. The empty lines are also simply ignored by the GNU Octave interpreter. Thus the line 8 is there only for the readers' convenience. Ditto for the comment the line 9. In the line 10 we set the mid-comfort degree to be $0.8 = 80\%$. This can be changed to different values if one wants to experiment. The empty line 11 and the comment line 12 are there for the readers' convenience. The lines 13–15 define the the reward or decision matrix R. The empty line 16 and the comment line 17 are there only for readers' convenience. The lines 18–19 define the required fields for the decision problem structure variable which we call dp (for decision problem). The empty line 20 and the comments line 21 are there only for the readers' convenience (again). The line 22 is the beef. There the decision problem dp is solved with the function dmatrix and the solution is

stored in the structure variable `dr` (for decision result). Lines 23 and 24 are there only for the readers' convenience. Finally, the commented lines 25–41 show how the result of running the script m-file `example_bb` should look like. Compare these results with the Brianna's results from Chapter 1.

Eugene Equally-Brisk's Decision

Eugene Equally-Bisk's solution is given in the m-file `example_ee.m` listed below.

```
 1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 2 %%
 3 %% FILE: example_ee.m
 4 %%
 5 %% Eugene Equally-Brisk's solution with dmatrix.
 6 %%
 7 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 8
 9 %% Decision matrix
10 R = [ 1    1    0    0;
11        0    0    1    0;
12        0    1    0    1];
13 %% Probabilities for scenarios.
14 p = [0.05   0.45   0.25   0.25]';
15
16 %% Decision matrix and its solution
17 dp.dmat = R;
18 dp.drule = "ev";
19 dp.prob = p;
20 dr = dmatrix(dp)
21
22 %% Running the file we get the output dr
23 %% >> example_ee
24 %% dr =
25 %%
26 %%   scalar structure containing the fields:
27 %%
28 %%     dvalues =
29 %%
30 %%        0.5000
31 %%        0.2500
32 %%        0.7000
33 %%
34 %%     doptim = 3
```

The actual calculations are made in the lines 10–20. All the rest is either comments for the readers' convenience (lines 1–9), or listing the result of the m-file (lines 22–34). To be more precise, the lines 1–9 are there only for readers' convenience. In the lines 10–12 the decision, or reward, matrix `R` is given. The line 13 is just a comment line. In the line 14 the probabilities `p` for scenarios are given as a column matrix. Lines 15–16 are there only for the readers' convenience. In the lines 17–19 the decision problem (dp) is defined. The beef is the line 20, where the decision problem (dp) is solved by using the function `dmatrix` and the solution is stored in the variable `dr` (decision result). The lines 21–22 are there for the readers' convenience. Finally, in the lines 23–34 we show the output of calling the m-file `example_ee.m`

Branna's and Eugene's Compromise Decision

The compromise calculations for Brianna and Eugene are given in the m-file `example_bbee.m` that is listed below

```matlab
 1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 2 %%
 3 %% FILE: example_bbee.m
 4 %%
 5 %% Brianna's and Eugene's compromise solution by using dmatrix.
 6 %%
 7 %% NOTE: In addition to dmatrix.m this script uses scaling.m.
 8 %%
 9 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
10
11 %% Brianna's problem with scaling
12 c = 0.8;
13 RB = [  1     0     0;
14          c     c     0;
15        1-c   1-c   1-c];
16 dpB.drule = "regret";
17 dpB.dmat = RB;
18 drB = dmatrix(dpB);
19 VB = drB.dvalues;
20 VB = -VB;
21 VB = scaling(VB);
22
23 %% Eugene's problem with scaling
24 RE = [ 1   1   0   0;
25        0   0   1   0;
26        0   1   0   1];
27 p = [0.05   0.45   0.25   0.25]';
28 dpE.drule = "ev";
29 dpE.dmat = RE;
30 dpE.prob = p;
31 drE = dmatrix(dpE);
32 VE = drE.dvalues;
33 VE = scaling(VE);
34
35 %% Compromise decision with Briann weight w=0.1, 0.2, ... , 1
36 w = 1:-0.1:0';
37 for i=1:length(w)
38    V(:,i) = w(i)*VB + (1-w(i))*VE;
39    [tmp, compromise(i)] = max(V(:,i)');
40 end
41
42 %% Running the file and checking the output compromise
43 %%>> example_bbee
44 %%>> compromise
45 %%compromise =
46 %%
47 %%    2   2   2   2   2   2   3   3   3   3   3
```

https://www.uwasa.fi/~tsottine/orms2020/en/example_bbee.m

Let us analyze the script file `example_bbee.m`. As always, the first commented block, here the lines 1–9, are there just for the readers' convenience. The same is true for the empty line 10 and for the comment line 11. In the line 12 the mid-comfort degree c is set to $0.8 = 80\%$.

The decision matrix for Brianna, `RB` is given in the lines 13–15. Then the decision problem for Brianna `dpB` is given in the lines 16–21. Not that here we scale the Brianna's decision values `VB` in the line 21. This requires the home made funxtion `scaling.m`. The lines 22–23 are there only for the readers' convenience. In the lines 24–26 the decision matrix for Eugene `RE` is given. In the line 27 the probabilities `p` for the scenarios are given. In the lines 28–30 the decision problem (`dpE`) of Eugene is given. In the line 31 the Eugene's decision problem `dpE` is solved with the function `dmatrix` and the solution is stored in the variable `drE` (decision result for Eugene). In the line 32 the decision values are stored in the variable `VE` (values for Eugene), which is scaled in the line 33. The lines 41–42 a the for the readers' convenience. Finally, in the lines 43–47 we run the script file and check its result variable `compromise`.

Visualizing Sensitivity for Sheldon Share

In this section we show how to visualize decision data when there are two free (or uncertain) sensitivity parameters. In your assignments in chapters 7 and 8 (and in the related examples in chapters 5 and 6) you should always have two sensitivity parameters, so pay attention.

In the example of this section we encounter the Koala Ebola with is an infectious disease that may evolve into a global pandemic. We will encounter the Koala Ebola in all our later examples and assignments also. So far, the Koala Ebola has only been encountered in Outdonesia. Unfortunately, Outdonesia is a popular tourist attraction and the Koala Ebola has penetrated the entire land. Outdonesia has not enforced any kind of quarantine yet, so it may be only a matter of time before we may have a global pandemic at our hand.

Sheldon Share has $c = 100\,000$ € extra money to invest. Sheldon does not believe in diversification. Instead he will invest in one of he following alternatives:

a_1 = Invest in a retail company stock.
a_2 = Invest in a retail company call-options.
a_3 = Invest in an airline company stock.
a_4 = Invest is an airline company call-options.

Sheldon thinks that the following scenarios are relevant to his choise:

s_1 = Koala Ebola runs rampant.
s_2 = Koala Ebola is contained, but the economy is in recess.
s_3 = Koala Ebola is contained and the economy is booming.

Sheldon assumes the probabilities for the scenarios above to be

$$
\begin{aligned}
p_1 &= p_0, \\
p_2 &= q_0, \\
p_3 &= q_0.
\end{aligned}
$$

Here p_0 is an unknown uncertainty parameter and (necessarily) $q_0 = (1 - p_0)/2$. The reason to have p_2 and p_3 with equal probabilities is the so-called maximum entropy principle: we have no reason to assume the probabilities to be different in any special way, so we assumed them to be the same. Recall also that the probability p_0 is of course not fixed, it is a sensitivity parameter.

Sheldon Share has consulted the "best investing experts" and they have given him the following decision matrix (values are in returns)

	Return (prob)		
Choice	s_1 (p_0)	s_2 (q_0)	s_3 (q_0)
a_1	80	-50	10
a_2	120	-100	-100
a_3	-60	20	200
a_4	-100	-100	400

Sheldon is risk averse with CARA type utility function

$$u(r) \;=\; 1 - \mathrm{e}^{-r/r_0}.$$

Unfortunately, Sheldon does not know his risk tolerance r_0.

Remark Because Sheldon Share uses a utility rule, he should not calculate with returns but with outcomes. Therefore his decision matrix is actually

	Outcome (prob)		
Choice	s_1 (p_0)	s_2 (q_0)	s_3 (q_0)
a_1	$1.8c$	$0.5c$	$1.1c$
a_2	$2.2c$	$0.0c$	$0.0c$
a_3	$0.4c$	$1.2c$	$3.0c$
a_4	$0.0c$	$0.0c$	$5.0c$

where $c = 100\,000$ € is the amount of money to be invested.

Sheldon has two uncertain parameters, p_0 and r_0 that he does not know how to estimate. Therefore, he treats them as sensitivity parameters and plots how his optimal decision changes as the two parameters change. This is done in the m-file `example_shsh.m` below.

```
 1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 2 %%
 3 %% FILE: example_shsh.m
 4 %%
 5 %% Sheldon Share has c euros extra money to be invested.  He contenplates
 6 %% between the following four alternatives:
 7 %%
 8 %% a1 = retail stock
 9 %% a2 = retail call-option
10 %% a3 = airliner stock
11 %% a4 = airliner call-option
12 %%
13 %% Sheldon believes the following scenarios are relevant to his alternatives:
14 %%
15 %% s_1 = Koala Ebola runs rampant.
16 %% s_2 = Koala Ebola is contained, but the economy is in recess.
17 %% s_3 = Koala Ebola is contained and the economy is booming.
18 %%
19 %% Sheldon assumes that the associated probabilities are:
20 %%
21 %% p1 = p0.
22 %% p2 = q0.
23 %% p3 = q0.
24 %%
25 %% Here p0 is an unknown sensitivity parameter and q0 = (1-p0)/2.
26 %%
```

```matlab
27 %% Sheldon uses the CARA rule, but is uncertain of the related risk tolerance
28 %% r0.  Thus, he treats r0 also as a sensitivity parameter.
29 %%
30 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
31
32 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
33 %% Calculations and defititions for visualization.
34 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
35
36 %% Money to be invested and the decition matrix
37 c = 100000;
38 R = [ 1.8*c    0.5*c    1.1*c ;
39       2.2*c    0.0*c    0.0*c ;
40       0.4*c    1.2*c    3.0*c ;
41       0.0*c    0.0*c    5.0*c ];
42
43 %% N is the sampling frequency for p0 and r0.  The bigger the N the better the
44 %% picture, but the longer the calculations take.  For N=100 the calculations
45 %% take 1.5 seconds.  For N=500 the calculations take 35 seconds.
46 N = 500;
47 p0samples = linspace(0, 1, N);          %% N equidistant samples from 0...1
48 r0samples = linspace(1, 10*c ,N);       %% N equidistant samples from 0...10c
49
50 %% Constant fields for decision problem dp.
51 dp.dmat = R;
52 dp.drule = "cara";
53
54 %% Optimal choices with different parameter values p0 and r0 are stored in the
55 %% matrix z.
56 for i = 1:N
57     x = p0samples(i);
58     dp.prob = [x (1-x)/2 (1-x)/2]';
59     for j = 1:N
60         y = r0samples(j);
61         dp.rtol = y;
62         dr = dmatrix(dp);
63         z(j,i) = dr.doptim;
64     end
65 end
66
67 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
68 %% Plotting the sensitivity picture
69 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
70
71 %% Contour picture of z
72 contourf(p0samples, r0samples, z);
73
74 %% Names for axes and their labels
75 set(gca, "fontsize", 22);
76 xlabel("p_0");
77 ylabel("r_0");
78
79 %% Change the colors of the picture
80 red  = [1 0 0];                                %% RGB coded color
81 green = [0 1 0];                               %% RGB coded color
82 colormap([red; 0.5*red; green; 0.5*green]);    %% Colormap with four values
83
84 %% Plot the labels for the alternative (we printed z with N=10 to check which
```

```
85 %% color corresponded to which alternative and we noted that the y axis is
86 %% reverted in z compared to the picture)
87 text(0.05, 9.50*c, "a_4", "fontsize", 24);
88 text(0.30, 9.50*c, "a_3", "fontsize", 24);
89 text(0.60, 9.50*c, "a_1", "fontsize", 24);
90 text(0.85, 9.50*c, "a_2", "fontsize", 24);
91 %orient("landscape")
92 %print("sheldon_picture.pdf")
```

https://www.uwasa.fi/~tsottine/orms2020/en/example_shsh.m

Let us explain the code in the file example_shsh.m in detail.

The lines 1–30 is a comment block that explains the problem. GNU Octave interpreter will simply omit this block. Indeed, GNU Octave will always omit what comes after the comment sign "%" in the line. Empty lines are also omitted in the interpretation.

The next empty line 31 there is only for the readers' convenience.

The comment lines 32–34 is there only for the readers' convenience. They explain that in the next code block preliminary calculations are done for the visualization.

The Lines 35–36 are again only for the readers' convenience.

In line the 37 we set the money to be invested c as 100 0000. This can be changed if we want to experiment.

The decision matrix, or the reward matrix, is given in the lines 38–41.

The empty line 42 and the comment lines 43–45 are there for the readers' convenience.

In the line 46 we set the sampling frequency N for the plot. We use a small number N=10 or N=100 when we test the code and a big number N=500 for the final picture.

In the line 47 we define p0samples to be a vector of N values equidistant from the interval $0 \ldots 1$. They are the sampled values of the sensitivity parameter p0. Similarly, in the line 48, the vector r0samples will be a equidistant values from the interval $0 \ldots 10c$ representing the sampled values of r0.

Lines 49–50 are there only for the readers' convenience.

In the lines 51–52 we fill in the constant (with respect to p0 and r0) field for the decision problem structure variable dp.

The lines 53–55 are there for the readers' convenience only. Finally, in the lines 56–65 we have two nested for loops for p0 (outer loop) and r0 (inner loop). In the inner loop the decision problem is solved for given fixed values of p0 and r0 and the optimal decisions are stored in the matrix z (note that the y-axis in z is reversed in the contourf picture below).

Again, the lines 66–71 are there for the readers' convenience.

The actual picture is drawn in the line 72. Here the function call contourf(p0samples, r0samples, z) plots a filled contour plot of the matrix z with x sampled in p0samples and y sampled in r0samples.

Again, the lines 73–74 are there for the readers' convenience only.

In the line 75 we set the font size for the axis to be 22. (here gca stands for Get Current Axis).

In the lines 76–77 the labels for the x and y axis are given.

Once again the lines 78–79 are there for the readers' convenience only.

In the lines 80–82 we change the colors of the contour plot. Line 80 defines the color read and line 81 defines the color green. In the line 82 the rotating color map for the contour plot is set by red, dark red, green, dark green.

37

Yet again, the lines 83–86 are there for the readers' convenience only.

The lines 87–90 print the labels for the alternatives in the contour picture. The function `text` takes the first two arguments as the x and y position of the text. The third argument is the text itself, and the rest of the arguments are formatting for the text.

Finally, if we uncomment the lines 91–92 then the contour plot is printed in the landscape in the file sheldon_picture.pdf. (Note that in some systems the PDF driver is broken. In this case you should use, e.g., JPG picture. This means changing the line 91 to `print("sheldon_picture.jpg")`. You may also have to remove the orientation line `orient("landscape")`.)

The output of the file `example_shsh.m` with sampling frequency $N = 500$ is printed below:
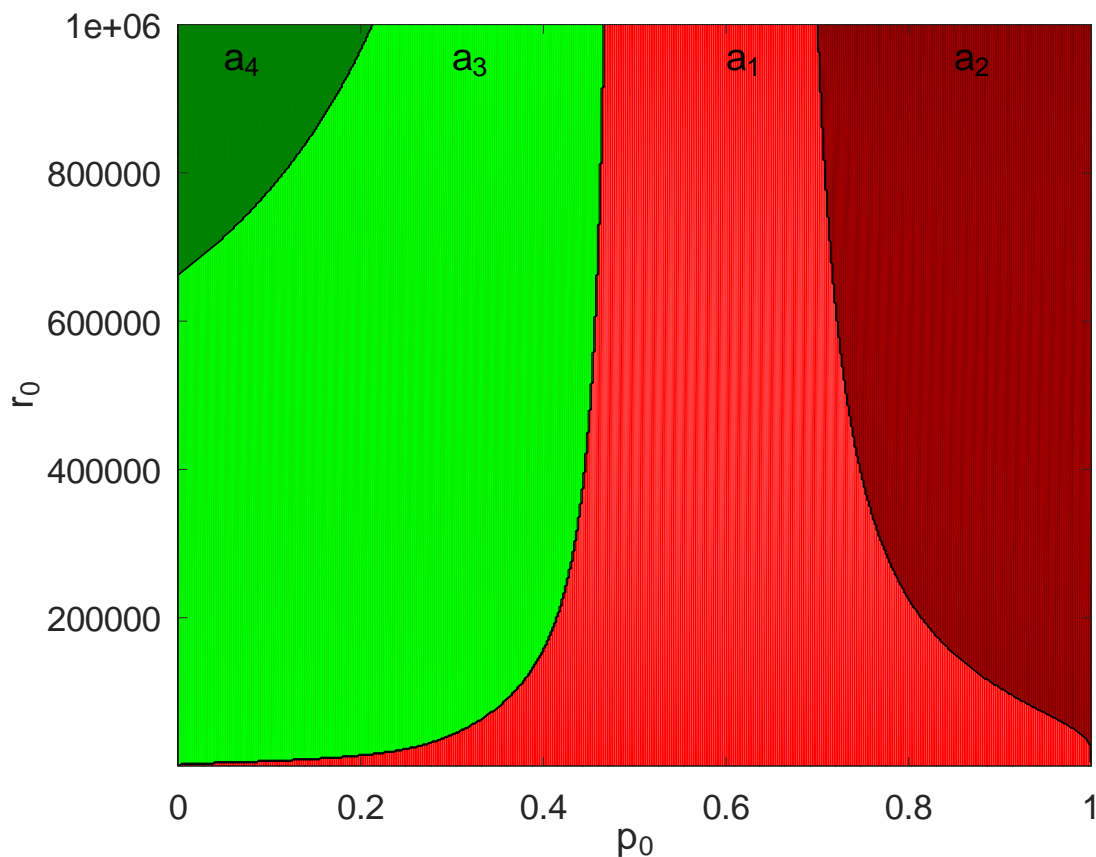
The interpretation of the picture should be obvious. For example, in the case of extreme risk aversion r_0 (near zero risk tolerance) the optimal alternative is a_1. As risk tolerance increases, all the alternative become more-or-less equally optimal depending on the Koala Ebola probability p_0.

Part III

Examples for Assignments

Chapter 5

Sophie Sophomore's Fall

This chapter gives an example "Sophie Sophomore's Fall" that aims to help you for your first assignment "Your fall" given in Chapter 7

Problem Description

Sophie Sophomore is a second year technology student in the University of Vaasa. It is February and Sophie has to decide what to do in the next Fall. What affects Sophie's choice is the pending Koala Ebola pandemic. So far, the Koala Ebola has only been encountered in Outdonesia. Unfortunately, Outdonesia is a popular tourist attraction and the Koala Ebola has penetrated the entire land. Outdonesia has not enforced any kind of quarantine yet, so it may be only a matter of time before we may have a global pandemic at our hand.

Sophie Sophomore contemplates between the following alternatives:

a_1 = Student exchange to China.
a_2 = Concentrate on her studies full-time.
a_3 = Mostly studies and working for a food delivery company in the evenings.
a_4 = Little studies in the evenings and working full-time for a multinational corporation.

Sophie Sophomore thinks that the following yes/no questions are relevant to her alternatives:

- Koala Ebola is pandemic or not.
- University is in distant teaching or in normal teaching.
- World economy is in recess or recovering.

In principle, with these three yes/no questions we get $2^3 = 8$ different scenarios. However, the distant teaching in the university occurs only if Koala Ebola is pandemic. Therefore we are left with the following six scenarios:

s_1 = Koala Ebola is pandemic, the university is in distant teaching, and the world economy is in recess.
s_2 = Koala Ebola is pandemic, the university is in distant teaching, and the world economy is recovering.
s_3 = Koala Ebola is pandemic, the university is in normal teaching, and the world economy is in recess.

s_4 = Koala Ebola is pandemic, the university is in normal teaching, and the world economy is recovering.

s_5 = Koala Ebola has been defeated, and the world economy is in recess.

s_6 = Koala Ebola has been defeated, and the world economy is recovering.

Let p_0 be the unknown Koala Ebola pandemic probability. The other probabilities for the scenarios are assumed to be

$$
\begin{aligned}
p_1 &= p_0/4 \\
p_2 &= p_0/4 \\
p_3 &= p_0/4 \\
p_4 &= p_0/4 \\
p_5 &= (1-p_0)/2 \\
p_6 &= (1-p_0)/2.
\end{aligned}
$$

The reasoning behind these probabilities is that the Koala Ebola pandemic probability $p_0 = p_1 + p_2 + p_3 + p_4$ is unknown, but the pandemic scenarios s_1, s_2, s_3, s_4 are equally probable. Ditto for the non-pandemic scenarios s_5 and s_6

Decision Matrices and Rules

Sophie Sophomore decides to practice multi-objective optimization. This means that she considers two decision matrices (or objectives) R^1 and R^2. To make things interesting, Sophie has chosen two opposite objectives. The objectives are

- Earned credit points in one year (reward).
- Remaining expected time to graduation after one year (punishment).

There is a lot of uncertainty involved in these objectives. Nevertheless, Sophie will estimate the decision matrices and take the values as they are. In other words, there will be no uncertainty parameters involved in the decision matrices.

Sophie's net worth is zero: no assets, no debt. Sophie also lives for free: her spouse takes care of the living expenses. Sophie gets student allowance of 252.76 € per month[1] and a 5 000 € scholarship if she decides to go to China.

The values in Sophie's decision matrices are based on the following observations:

- During the recess the living costs are lower.
- During the Koala Ebola pandemic the living costs are lower.
- Living costs in China are lower than in Finland, but in China one must pay rent.
- In a multinational corporation the last hired is the first fired.
- Working for a multinational corporation increases the cost of living.
- During a pandemic there is more work for food delivery.

---

[1]This amount has changed since the writing of these notes. In your assignment you should use the current student allowance.

More precise calculations are given in the m-file `example_so.m` that is listed in the Appendix A: Calculations.

Here are Sophie Sophomore's decision matrices (first R^1 for net worth and then R^2 for remaining time to graduate)

| Choice | Net Worth Pandemic/Distant Teaching/Recess Probabilities | | | | | |
	$p_0/4$	$p_0/4$	$p_0/4$	$p_0/4$	$(1-p_0)/2$	$(1-p_0)/2$
Exchange to China	3633	3033	3633	3033	3333	3033
Full-Time Studying	–567	–2967	–567	–2967	–1767	–2967
Food delivery	8073	5673	3753	1353	393	–807
Full-Time Working	–2850	3750	–2850	6000	450	10500

| Choice | Time to Graduation Pandemic/Distant Teaching/Recess Probabilities | | | | | |
	$p_0/4$	$p_0/4$	$p_0/4$	$p_0/4$	$(1-p_0)/2$	$(1-p_0)/2$
Exchange to China	2	2	2	2	2	2
Full-Time Studying	2	2	1	1	1	1
Food Delivery	2	2	2	2	2	2
Full-Time Working	3	4	3	4	3	5

Sophie Sophomore decides to use the expected value rule. Since she has two decision matrices this means that Sophie has two value functions

$$V_i^1 = \sum_{j=1}^{6} R_{ij}^1 p_j \quad \text{and} \quad V_i^2 = \sum_{j=1}^{6} R_{ij}^2 p_j.$$

Since the p_j's are determined by the Koala Ebola pandemic probability p_0 we will write explicitly $V^1(p_0)$ for V^1 and $V^2(p_0)$ for V^2. This emphasizes that fact that p_0 is a sensitivity parameter.

Another sensitivity parameter arises from the fact that we have to find a compromise between the value functions $V^1(p_0)$ and $V^2(p_0)$. We could in principle weigh them equally, in which case we would have the compromise value function $V = 0.5\bar{V}^1(p_0) + 0.5\bar{V}^2(p_0)$, where the barred value functions are affinely scaled versions of the corresponding un-barred value functions. Instead of equal weighing, we make the weight w a second sensitivity parameter. So, have the two-parameter value function

$$V(p_0, w) = w\bar{V}^1(p_0) + (1-w)\bar{V}^2(p_0)$$

where both p_0 and w are sensitivity parameters from the interval $0\ldots1$, and the barred value functions are affinely scaled:

$$\bar{V}^1(p_0) = \frac{V^1(p_0) - \min_k V_k^1(p_0)}{\max_k V_k^1(p_0) - \min_k V_k^1(p_0)},$$

$$\bar{V}^2(p_0) = \frac{V^2(p_0) - \min_k V_k^2(p_0)}{\max_k V_k^2(p_0) - \min_k V_k^2(p_0)},$$

if the value functions are not constant, so that we have no division-by-zero problem. In the constant case, the affine scaling is set to be the constant value function $\equiv 1$.

Sensitivity Analysis

We analyze Sophie Sophomore's decision given by the value function

$$V(p_0, w) \;=\; w\bar{V}^1(p_0) + (1-w)\bar{V}^2(p_0)$$

as the sensitivity parameters p_0 and w alternate in the interval $0\ldots 1$. We obtain the following picture (see Appendix A, m-file `example_so.m`). Recall that

- a_1 = Student exchange to China.
- a_2 = Concentrate on the studies full-time.
- a_3 = Mostly studies and working for a food delivery company in the evenings.
- a_4 = Little studies in the evenings and working full-time for a multinational corporation.
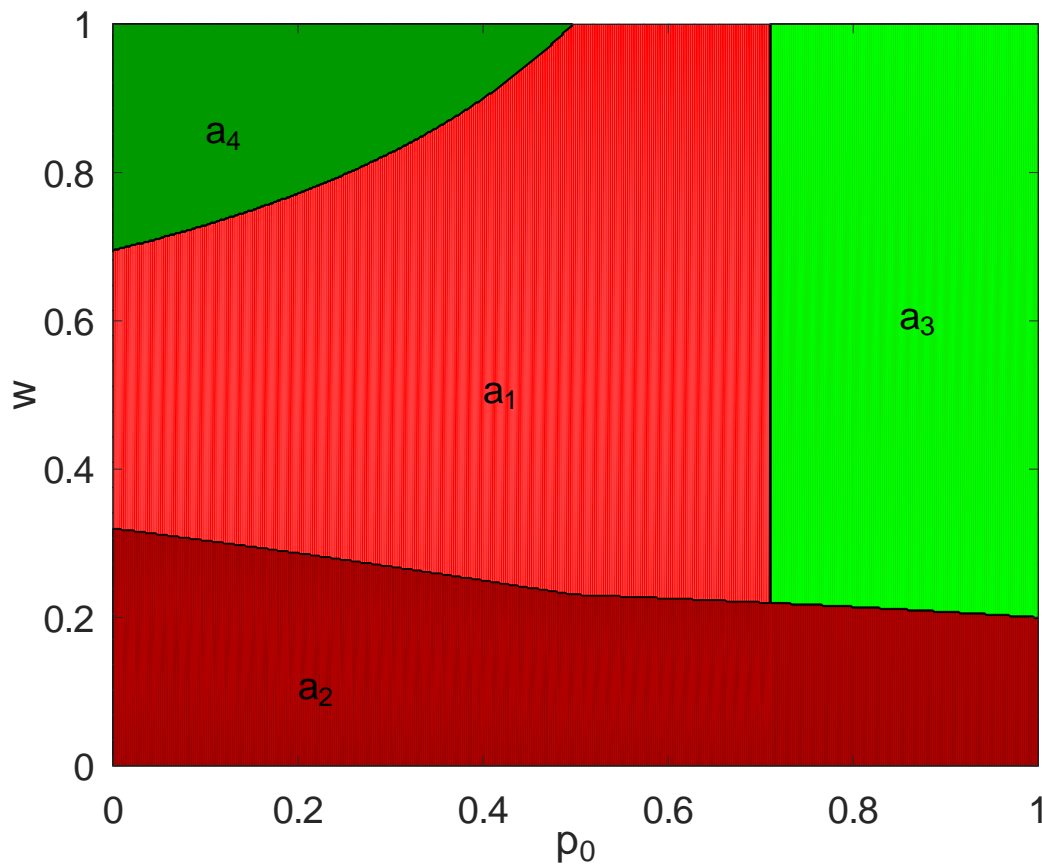
The picture says more than thousand words. Anyway, in some words we see that the alternative a_2 (full-time studies) is optimal if most of the weight (w is low) is given to remaining time to graduation (objective R^2). If most weight is given to net worth (w is high, objective R^1), then the optimal solution is very sensitive to the Koala Ebola pandemic probability p_0. If the Koala Ebola pandemic probability p_0 is small and the net worth weight w is high, then a_4 (full-time work) is optimal. If the Koala Ebola pandemic probability is high and the net worth weight is relatively high, then a_3 (food delivery) is generally a good choice. Finally, a_1 (student exchange to China) is generally a good choice if the Koala Ebola pandemic probability is not too high and net worth is given some reasonable weight.

Susan Suit's Fall

Susan Suit is Sophie Sophomore's spouse. Susan is faced with the same problem as Sophie. The difference is that Susan's decision matrix (reward matrix, objective) is her (monthly) salary after three years. As a decision rules, Susan uses expected value and regret aversion. The alternatives and the scenarios for Susan are the same as those of Sophie.

Remark For the compromise problem between Susan and Sophie to make sense only the alternatives have to be the same. It in not necessary that the scenarios are the same also.

Susan has two sensitivity parameters. First, there is the weight w that tells how the expected value rule and the regret aversion rule are combined. Second, there is the Koala Ebola pandemic probability p_0.

Susan has as single decision matrix (reward matrix, objective) of the expected monthly salaries after three years. Susan has come up (somehow) with the following decision matrix

Choice	Monthly Salary after 3 Years Pandemic/Distant Teaching/Recess Probabilities					
	$p_0/4$	$p_0/4$	$p_0/4$	$p_0/4$	$(1-p_0)/2$	$(1-p_0)/2$
Exchange to China	3000	3500	2800	3500	2800	3500
Full-Time Studying	2500	2700	3300	3800	3500	4000
Food Delivery	3000	3200	2000	2500	2000	2500
Full-Time Working	3000	3200	2500	4000	3000	3500

This decision matrix above is based on the following observations:

- Quick graduation increases salary.
- Recess lowers salary.
- Exchange in China increases salary, but prolongs graduation.
- Full-time working increases salary, but prolongs graduation.
- Distant teaching and food delivery work well together.
- Distant teaching suits well exchange students
- On site teaching works badly for full-time working.

Recall the affine scaling

$$\bar{V} = \frac{V - \min_k V_k}{\max_k V_k - \min_k V_k}$$

if $\max_k V_k > \min_k V_k$, and

$$V \equiv 1$$

otherwise.

Susan's decision rule is

$$V(p_0, w) = w\bar{V}^1(p_0) + (1-w)\bar{V}^2,$$

where $\bar{V}^1(p_0)$ the the affinely scaled expected value rule given the Koala Ebola pandemic probability p_0 that is a sensitivity parameter from the interval $0\ldots 1$, and $\bar{V}^2$ is the affinely scaled regret averse decision rule turned into a maximization form which is independent of the sensitivity parameter p_0. The combination weight w is a sensitivity parameter from the interval $0\ldots 1$.

We obtain the following picture (see Appendix A, m-file `example_su.m`). Recall that

a_1 = Student exchange to China.
a_2 = Concentrate on the studies full-time.
a_3 = Mostly studies and working for a food delivery company in the evenings.
a_4 = Little studies in the evenings and working full-time for a multinational corporation.
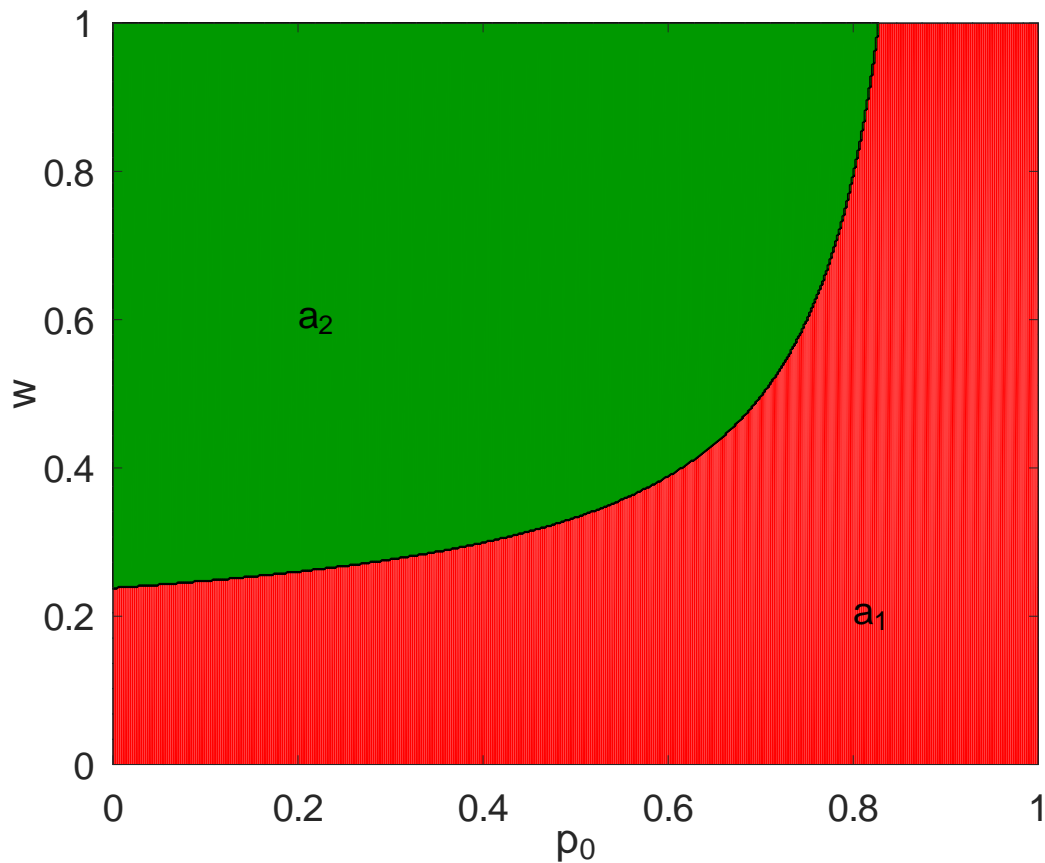
The picture says more than thousand words. We see that the alternative a_2 (full-time studies) is best if the weight w for the decision rule for the expected value is relatively high and the Koala Ebola pandemic probability is not too high. Otherwise, the alternative a_1 (Exchange to China) is optimal. The alternatives a_3 (food delivery) and a_4 (full-time working) are never optimal.

Sophie's and Susan's Compromise Sensitivity Analysis

Sophie Sophomore and Susan Suit want to find a compromise alternative that both can accept. They decide to consider the problem with two sensitivity parameters (since paper is two dimensional). Obvious parameters are p_0, the Koala Ebola pandemic probability, and w that is the weight Sophie's value function has in the compromise value function. To form the compromise value function $V(p_0, w)$ we need the individual values functions $V(So; p_0)$ and $V(Su; p_0)$ for Sophie and Susan. Then

$$V(p_0, w) = wV(So, p_0) + (1-w)V(Su; p_0)$$

where we have affinely scaled the individual value functions $V(So; p_0)$ and $V(Su; p_0)$ and they are both in maximization form.

We assume that in their internal calculations both Sophie and Susan use the equal weight $w = 0.5$ in combining their internal value functions. Then (see Appendix A: Calculations, m-file `example_sosu.m`) we obtain the sensitivity picture
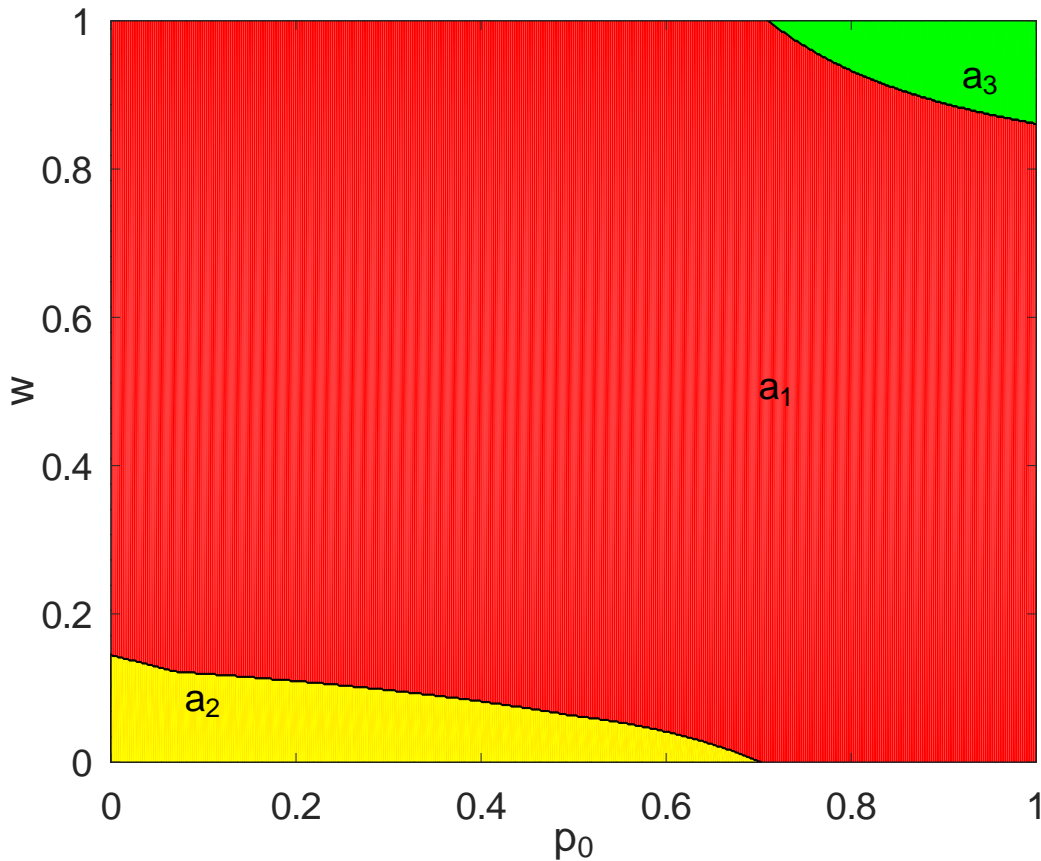
Recall that

a_1 = Student exchange to China.
a_2 = Concentrate on the studies full-time.
a_3 = Mostly studies and working for a food delivery company in the evenings.
a_4 = Little studies in the evenings and working full-time for a multinational corporation.

Again, the picture says more than thousand words. In some words, though, it seems that a_1 (exchange to China) is most commonly the optimal compromise choice. If we weigh Sophie highly and the Koala Ebola pandemic probability is relatively low, then the alternative a_2 (full-time studies) is the optimal compromise. If we weigh Susan highly and the Koala Ebola pandemic probability is high, then the alternative a_3 (food delivery) is the optimal compromise. The alternative a_4 (full-time work) is never an optimal compromise.

Appendix A: Calculations

Below is listed the m-file for Sophie Sophomore.

```
1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %%
3  %% FILE: example_so.m
4  %%
5  %% Sophie Sophomore's fall.
6  %%
7  %% Sophie's alternatives
8  %%
9  %% a1 = Student exchange to China.
10 %% a2 = Concentrate on the studies full-time.
11 %% a3 = Mostly studies and working for a food delivery company in the evenings.
12 %% a4 = Little studies and working full-time for a multinational corporation.
13 %%
14 %% Sophie's scenarios
15 %%
16 %% s1 = Koala Ebola is pandemic, the university is in distant teaching,
17 %%      and the wofdd economy is in recess.
18 %% s2 = Koala Ebola is pandemic, the university is in distant teaching,
19 %%      and the wofdd economy is recovering.
20 %% s3 = Koala Ebola is pandemic, the university is in normal teaching,
21 %%      and the wofdd economy is in recess.
22 %% s4 = Koala Ebola is pandemic, the university is in normal teaching,
23 %%      and the wofdd economy is recovering.
24 %% s5 = Koala Ebola has been defeated, and the wofdd economy is in recess.
25 %% s6 = Koala Ebola has been defeated, and the wofdd economy is recovering.
26 %%
27 %% The probabilities for the scenarios are
28 %%
29 %% p1 = p0/4
30 %% p2 = p0/4
31 %% p3 = p0/4
32 %% p4 = p0/4
33 %% p5 = (1-p0)/2
34 %% p6 = (1-p0)/2
35 %%
36 %% The Koela Ebola pandemic paramter p0 is a sensitivity parameter.
37 %%
38 %% Sophie uses the expected value rule with two decision matrices (objectives)
39 %%
40 %% - Net worth after one year (R1).
41 %% - Remaining time to graduation (R2)
42 %%
43 %% The compromise weigth w for the objectives is a sensitivity parameter.
44 %%
45 %% The file uses the functions dmatrix and scaling.
46
47 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
48 %% Decision matrices
49 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
50
51 %% Net worth after one year.
52
53 sa = 12*252.76;        %% Yearly student allowance.
```

```matlab
54 cs = 5000;              %% China scholarship.
55 cm = 2000;              %% China moving costs.
56 cf = 12*500;            %% Yearly living costs in Finland.
57 cc = 12*250;            %% Yearly living costs in China.
58
59 sc = sa+cs-cm;          %% Yearly income in China.
60 sf = sa;                %% Yearly income if Finland.
61
62 fd = 12*200*0.9;        %% Yearly food delivery salary.
63 ft = 12*2500*0.75;      %% Yearly full-time salary in a multinational corporation.
64
65 R1 = [
66 sc-0.8*cc        sc-cc        sc-0.8*cc        sc-cc        sc-0.9*cc        sc-cc      ;
67 sf-0.6*cf        sf-cf        sf-0.6*cf        sf-cf        sf-0.8*cf        sf-cf      ;
68 sf+4*fd-0.6*cf sf+4*fd-cf   sf+2*fd-0.6*cf sf+2*fd-cf   sf+fd-0.8*cf   sf+fd-cf;
69 0.3*ft-1.6*cf   0.7*ft-2*cf 0.3*ft-1.6*cf 0.8*ft-2*cf 0.5*ft-1.8*cf ft-2*cf
70 ];
71
72 %% Remaining time to graduation after one year.
73
74 R2 = [
75 2  2  2  2  2  2;
76 2  2  1  1  1  1;
77 2  2  2  2  2  2;
78 3  4  3  4  3  5
79 ];
80
81 R2 = -R2;              %% Turn R2 into a maximization form.
82
83 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
84 %% Sensitivity visualization
85 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
86
87 %% Sampling frequency.  The bigger the better the picrure, but the longer the
88 %% calculations take.  With frequency N=500 the calculations take 5 seconds.
89 N=500;
90
91 %% Calculations
92
93 %% N equidistant samples from the interval 0..1 for both p0 and w.
94 p0samples = linspace(0,1,N);
95 wsamples = linspace(0,1,N);
96
97 %% Constant (w.r.t. p0 and w) fields for dp1 and dp2: decision rule and the
98 %% decision matrices.
99 dp1.drule = "ev";
100 dp1.dmat = R1;
101 dp2.drule = "ev";
102 dp2.dmat = R2;
103
104 %% Calculete optimal decisions z(j,i)=z(w,p0) in nested for loops for p0 and w.
105 for i=1:length(p0samples)
106     p0 = p0samples(i);
107     q0 = 1-p0;
108     dp1.prob = [p0/4 p0/4 p0/4 p0/4 q0/2 q0/2]';
109     dp2.prob = [p0/4 p0/4 p0/4 p0/4 q0/2 q0/2]';
110     dr1 = dmatrix(dp1);
111     dr2 = dmatrix(dp2);
```

```
112      V1 = scaling(dr1.dvalues);
113      V2 = scaling(dr2.dvalues);
114      for j=1:length(wsamples)
115          w = wsamples(j);
116          V = w*V1 + (1-w)*V2;
117          [tmp, z(j,i)] = max(V);
118      end
119 end
120
121 %% Picture drawing
122
123 %% Draw the picture with contourf(x,y,z).
124 contourf(p0samples, wsamples, z)
125
126 %% Set the axes labels fontsize to 24. (gca stands for get current axes.)
127 set(gca, "fontsize", 24);
128
129 %% Draw the x and y axes labels.
130 xlabel("p_0")
131 ylabel("w")
132
133 %% Define a RGB coded colormap for the colors in the picture.
134 red   = [1 0 0];
135 green = [0 1 0];
136 colormap([red; 0.6*red; green; 0.6*green])
137
138 %% Set the alternative labels in the picture.
139 text(0.40,0.50, "a_1", "fontsize", 24)
140 text(0.20,0.10, "a_2", "fontsize", 24)
141 text(0.85,0.60, "a_3", "fontsize", 24)
142 text(0.10,0.85, "a_4", "fontsize", 24)
143
144 %% Uncommet the following lines to draw the picture in a landcape paper in
145 %% PDF format to the file so_picture.pdf. Your system may not suppor the PDF
146 %% format format. In that case you can try, e.g., the JPG format. This means
147 %% naming the print file as so_picture.jpg. You may also want to leave the
148 %% orient line commented.
149 %orient("landscape")
150 %print("so_picture.pdf")
```

$$\text{https://www.uwasa.fi/}\sim\text{tsottine/orms2020/en/example\_so.m}$$

Below is listed the m-file for Susan Suit.

```
 1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 2 %%
 3 %% FILE: example_su.m
 4 %%
 5 %% Sophie Sophomore's spouse Susan Suit contemplates what to do next Fall.
 6 %% Susan's alterantives and scenarios are the same as Sophie's:
 7 %%
 8 %% Alternatives:
 9 %%
10 %% a1 = Student exchange to China.
11 %% a2 = Concentrate on the studies full-time.
12 %% a3 = Mostly studies and working for a food delivery company in the evenings.
13 %% a4 = Little studies and working full-time for a multinational corporation.
14 %%
```

```matlab
%% Scenarios:
%%
%% s1 = Koala Ebola is pandemic, the university is in distant teaching,
%%      and the wofdd economy is in recess.
%% s2 = Koala Ebola is pandemic, the university is in distant teaching,
%%      and the wofdd economy is recovering.
%% s3 = Koala Ebola is pandemic, the university is in normal teaching,
%%      and the wofdd economy is in recess.
%% s4 = Koala Ebola is pandemic, the university is in normal teaching,
%%      and the wofdd economy is recovering.
%% s5 = Koala Ebola has been defeated, and the wofdd economy is in recess.
%% s6 = Koala Ebola has been defeated, and the wofdd economy is recovering.
%%
%% The probabilities for the scenarios are
%%
%% p1 = p0/4
%% p2 = p0/4
%% p3 = p0/4
%% p4 = p0/4
%% p5 = (1-p0)/2
%% p6 = (1-p0)/2
%%
%% The Koala Ebola pandemic paramter p0 is a sensitivity parameter.
%%
%% Susan bases her decisions on excpectd value rule and regret aversion rule.
%% Susan's decision matrix is her expected monthy salary after three years-
%% The combination weight for expected value and regret arversion is a
%% sensitivity parameter w.
%%
%% The file uses the functions dmatrix and scaling.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Decision matrix
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

R = [
3000   3500   2800   3500   2800   3500;
2500   2700   3300   3800   3500   4000;
3000   3200   2000   2500   2000   2500;
3000   3200   2500   4000   3000   3500
];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Visualization
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Calculations.

%% Constant fields for decision problem 1 (expected value)
dp1.dmat = R;
dp1.drule = "ev";

%% Decision problem 2 (regret aversion)
dp2.dmat = R;
dp2.drule = "regret";
dr2 = dmatrix(dp2);
V2 = dr2.dvalues;
V2 = -V2;                          %% Maximization form.
```

```
73  V2 = scaling(V2);                      %% Affine scaling.
74
75
76  %% Otteistustaajuus.  Mita isompi sen parempi, mutta laskeminen kestaa.
77  %% Jos N=500 laskeminen kestaa noin 20 sekuntia.
78  N=500;
79
80  %% Calculations.
81  p0samples = linspace(0,1,N);
82  wsamples = linspace(0,1,N);
83
84  for i=1:length(p0samples)
85      p0 = p0samples(i);
86      q0 = 1-p0;
87      dp1.prob = [p0/4 p0/4 p0/4 p0/4 q0/2 q0/2]';
88      dr1 = dmatrix(dp1);
89      V1 = scaling(dr1.dvalues);
90      for j=1:length(wsamples)
91          w = wsamples(j);
92          V = w*V1 + (1-w)*V2;
93          [tmp, z(j,i)] = max(V);
94      end
95  end
96
97  %% The picture. See m-file example_so.m for more detailed comments.
98  contourf(p0samples, wsamaples, z)
99  set(gca, "fontsize", 24);
100 xlabel("p_0")
101 ylabel("w")
102 red   = [1 0 0];
103 green = [0 1 0];
104 colormap([red; 0.6*red; green; 0.6*green])
105 text(0.80,0.20, "a_1", "fontsize",24)
106 text(0.20,0.60, "a_2", "fontsize",24)
107 %% Uncomment the following lines to produce landscape PDF picture.  Note that
108 %% in some systems the PDF driver does not work.  In that case use, e.g., the
109 %% JPG driver.  This means changing the fie extension to .jpg.  You may also
110 %% what to leave the orient line commented in that case.
111 %orient("landscape")
112 %print("su_picture.pdf")
```

Below is listed the m-file for Sophie Sophomore's and Susan Suit's compromise.

```
1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %%
3  %% FILE: example_sosu.m
4  %%
5  %% Sophie Sophomore's and Susan Suit's compromise solution
6  %%
7  %% Alternatives:
8  %%
9  %% a1 = Student exchange to China.
10 %% a2 = Concentrate on the studies full-time.
11 %% a3 = Mostly studies and working for a food delivery company in the evenings.
12 %% a4 = Little studies and working full-time for a multinational corporation.
13 %%
```

```matlab
14 %% Scenarios:
15 %%
16 %% s1 = Koala Ebola is pandemic, the university is in distant teaching,
17 %%      and the wofdd economy is in recess.
18 %% s2 = Koala Ebola is pandemic, the university is in distant teaching,
19 %%      and the wofdd economy is recovering.
20 %% s3 = Koala Ebola is pandemic, the university is in normal teaching,
21 %%      and the wofdd economy is in recess.
22 %% s4 = Koala Ebola is pandemic, the university is in normal teaching,
23 %%      and the wofdd economy is recovering.
24 %% s5 = Koala Ebola has been defeated, and the wofdd economy is in recess.
25 %% s6 = Koala Ebola has been defeated, and the wofdd economy is recovering.
26 %%
27 %% The probabilities for the scenarios are
28 %%
29 %% p1 = p0/4
30 %% p2 = p0/4
31 %% p3 = p0/4
32 %% p4 = p0/4
33 %% p5 = (1-p0)/2
34 %% p6 = (1-p0)/2
35 %%
36 %% The Koala Ebola pandemic paramter p0 is a sensitivity parameter.  The weight
37 %% w comibing Sophie's and Susan's value functions is a sentitivity parameter.
38 %%
39 %% The file uses the functions dmatrix and scaling.
40
41 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
42 %% Decision matrices
43 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
44
45 %% Sophie's matrix 1
46
47 sa = 12*252.76;        %% Yearly student allowance.
48 cs = 5000;             %% China scholarship.
49 cm = 2000;             %% China moving costs.
50 cf = 12*500;           %% Yearly living costs in Finland.
51 cc = 12*250;           %% Yearly living costs in China.
52
53 sc = sa+cs-cm;         %% Yearly income in China.
54 sf = sa;               %% Yearly income if Finland.
55
56 fd = 12*200*0.9;       %% Yearly food delivery salary.
57 ft = 12*2500*0.75;     %% Yearly full-time salary in a multinational corporation.
58
59 RSo1 = [
60 sc-0.8*cc        sc-cc        sc-0.8*cc        sc-cc        sc-0.9*cc        sc-cc     ;
61 sf-0.6*cf        sf-cf        sf-0.6*cf        sf-cf        sf-0.8*cf        sf-cf     ;
62 sf+4*fd-0.6*cf sf+4*fd-cf  sf+2*fd-0.6*cf sf+2*fd-cf  sf+fd-0.8*cf    sf+fd-cf;
63 0.3*ft-1.6*cc    0.7*ft-2*cc 0.3*ft-1.6*cc    0.8*ft-2*cc 0.5*ft-1.8*cc ft-2*cc
64 ];
65
66 %% Sophie's matrix 2
67
68 RSo2 = [
69 2   2   2   2   2   2;
70 2   2   1   1   1   1;
71 2   2   2   2   2   2;
```

```
72  3   4   3   4   3   5
73  ];
74
75  RSo2 = -RSo2;              %% Maximization form!
76
77  %% Susan's decision matrix
78
79  RSu = [
80  3000   3500   2800   3500   2800   3500;
81  2500   2700   3300   3800   3500   4000;
82  3000   3200   2000   2500   2000   2500;
83  3000   3200   2500   4000   3000   3500
84  ];
85
86  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
87  %% Visualization.
88  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
89
90  %% Decision problems' constant fields (independent of p0 and w)
91  dpSo1.dmat = RSo1;
92  dpSo1.drule = "ev";
93  dpSo2.dmat = RSo2;
94  dpSo2.drule = "ev";
95
96  dpSu1.dmat = RSu;
97  dpSu1.drule = "ev";
98  dpSu2.dmat = RSu;
99  dpSu2.drule = "regret";
100
101 %% Decision problem dpSu2 can be calculated completely, since it is independent
102 %% pf p0 and w.
103 drSu2 = dmatrix(dpSu2);
104 VSu2 = drSu2.dvalues;
105 VSu2 = -VSu2;                        %% Regret is a minimization form.
106 VSu2 = scaling(VSu2);
107
108
109 %% Sampling frequeny.  The bigge the better picure, but the longer the
110 %% calculations take.  For N=500 the calculations take 5 seconds.
111 N = 500;
112
113 %% Sample points for x (p0samples) and y (wsamples) axes.
114 p0samples = linspace(0,1,N);
115 wsamples = linspace(0,1,N);
116
117 for i=1:length(p0samples)
118     p0 = p0samples(i);
119     q0 = 1-p0;
120     dpSo1.prob = [p0/4 p0/4 p0/4 p0/4 q0/2 q0/2]';
121     dpSo2.prob = [p0/4 p0/4 p0/4 p0/4 q0/2 q0/2]';
122     dpSu1.prob = [p0/4 p0/4 p0/4 p0/4 q0/2 q0/2]';
123     drSo1 = dmatrix(dpSo1);
124     drSo2 = dmatrix(dpSo2);
125     drSu1 = dmatrix(dpSu1);
126     %% drSu2 has been calculated in the line 103.
127     VSo1 = scaling(drSo1.dvalues);
128     VSo2 = scaling(drSo2.dvalues);
129     VSo = 0.5*VSo1 + 0.5*VSo2;
```

```
130        VSo = scaling(VSo);
131        VSu1 = scaling(drSu1.dvalues);
132        VSu = 0.5*VSu1 + 0.5*VSu2;
133        VSu = scaling(VSu);
134        for j=1:length(wsamples)
135            w = wsamples(j);
136            V = w*VSo + (1-w)*VSu;
137            [roskaa, z(j,i)] = max(V);
138        end
139 end
140
141 %% Draw the picture. See the m-file example_so.m for comments.
142 contourf(p0samples, wsamples, z)
143 set(gca, "fontsize", 24);
144 xlabel("p_0")
145 ylabel("w")
146 red     = [1 0 0];
147 yellow  = [1 1 0];
148 green   = [0 1 0];
149 colormap([red; yellow; green])
150 text(0.70,0.50, "a_1", "fontsize",24)
151 text(0.08,0.08, "a_2", "fontsize",24)
152 text(0.92,0.92, "a_3", "fontsize",24)
153 %% Uncommet the following lines to draw the picture in a landcape paper in
154 %% PDF format to the file sosu_picture.pdf. Your system may not support the
155 %% PDF format format. In that case you can try, e.g., the JPG format. This
156 %% means naming the print file as so_picture.jpg. You may also want to leave
157 %% the orient line commented.
158 %orient("landscape")
159 %print("sosu_picture.pdf")
```

https://www.uwasa.fi/~tsottine/orms2020/en/example_sosu.m

Chapter 6

Catherine Cato's Scientific Report

This chapter is an example for you second assignment "Ministry Report" given in Chapter 8.

Problem Description

The Caligula Institute is a libertarian think tank. Catherine Cato works for the Caligula Institute. She is assigned to write a scientific report on what kind of policy should Finland adopt in the pending Koala Ebola pandemic. Recall that so far the Koala Ebola has only been encountered in Outdonesia. Unfortunately, Outdonesia is a popular tourist attraction and the Koala Ebola has penetrated the entire land. Outdonesia has not enforced any kind of quarantine yet, so it may be only a matter of time before we may have a global pandemic at our hand.

In this chapter we use the decision matrix technique in less than honest way. We have decided what we want and then we cook the numbers to show the results we wanted. How this is done is revealed in the Appendix B: Secret Supplementary Protocol.

Catherine decides to consider the following four alternatives for the general economic policy for Finland:

a_1 = Extreme laissez-faire policy.
a_2 = Laissez-faire policy.
a_3 = Current policy.
a_4 = Regulated policy.
a_5 = Extreme regulated policy.

We do not go into details what these policies entail.

The decision matrix will be the GDP development (change in per cents). There are two essential uncertainties involved:

- Is Koala Ebola pandemic on or not?
- Is Finland's export thriving or not?

From these two uncertainties we obtain the following four scenarios:

s_1 = Koala Ebola is contained, and export is thriving.
s_2 = Koala Ebola is contained, but export is not thriving.
s_3 = Koala Ebola is pandemic, but export is thriving.

s_4 = Koala Ebola is pandemic, and export is not thriving.

By using the Harrison–Stetson method we estimate the scenario probabilities to be

$$
\begin{aligned}
p_1 &= 0.75p_0, \\
p_2 &= 0.25p_0, \\
p_3 &= 0.25q_0, \\
p_4 &= 0.75q_0,
\end{aligned}
$$

where

$$
q_0 = 1 - p_0
$$

an the sensitivity parameter p_0 for Koala Ebola containment is unknown. We treat p_0 as a sensitivity parameter.

Decision Matrix and Rules

According to best economists we have the following decision matrix telling the change (in per cent) of Finland's GDP under different economic policies and under different scenarios. This is the starting point of Catherine Cato's analysis.

	GDP Change in Per Cents			
Choice	$0.75p_0$	$0.25p_0$	$0.25q_0$	$0.75q_0$
Extreme laissez-faire policy	8	2	3	–5
Laissez-faire policy	6	2	2	–3
Current policy	2	1	1	–2
Regulated polity	0	–1	0	–2
Extreme regulated policy	–2	–1	–1	–1

The numbers are based on the following observations

- Extreme laissez-fair policy is good, if Koala Ebola is contained and export is thriving. Koala Ebola risk for this policy is considerable.
- Laissez-fair policy has less Koala Ebola risk than extreme laissez-fair policy, but the risk is still considerable.
- Current policy has little Koala Ebola risk, but does not exploit thricing exports very well.
- Regulated policy can handle Koala Ebola risk very well, but can not exploit the thriving export very well.
- Extreme regulatex policy can handle Koala Ebola risk very well, but is the worst in exploting thriving export.

In macro level decision making such as Catherine's problem, the expected value rule is the most sensible choice. The expected value rule is, however, sensitive to the uncertain probabilities. This is why we temper the expected value rule with the Hurwicz rule. This gives us in principle two additional sensitivity parameters: the compromise weight for expected value

rule and the Hurwicz rule, and the degree of optimism in the Hurwicz rule. To have only two sensitivity parameters, we use equal weight in combining the decision rules. This leads to the value function

$$V(p_0, w) \;=\; 0.5 \cdot V^{\mathrm{ev}}(p_0) \;+\; 0.5 \cdot V^{\mathrm{Hur}}(w),$$

where the summands $V^{\mathrm{ev}}(p_0)$ and $V^{\mathrm{Hur}}(w)$ are assumed to be affinely scaled.

Sensitivity Analysis

We obtain the following sensitivity picture (see Appendix A: Calculations, m-file `example_cc.m`):
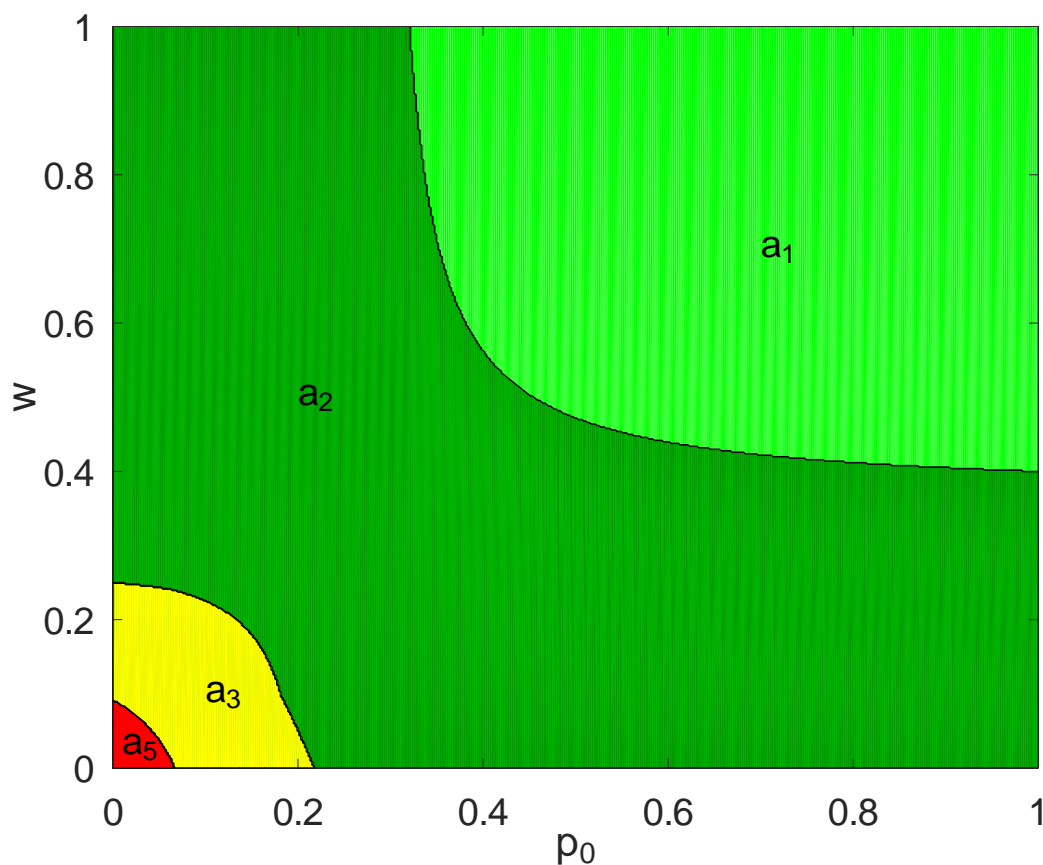
Recall that

a_1 = Extreme laissez-faire policy.
a_2 = Laissez-faire policy.
a_3 = Current policy.
a_4 = Regulated policy.
a_5 = Extreme regulated policy.

The picture says it all! Laissez-faire and extreme laissez-faire are good policies and the regulated policies are good only if we are very pessimistic and the Koala Ebola pandemic probability is very small.

Appendix A: Calculations

Catherine's calculations are in the m-file `example_cc.m` that is listed below.

```
1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %%
3  %% FILE: example_cc.m
4  %%
5  %% Catherine Cato's scientific report.
6  %%
7  %% Alternatives:
8  %%
9  %% a1 = Extreme laissez-faire policy.
10 %% a2 = Laissez-faire policy.
11 %% a3 = Current policy.
12 %% a4 = Regulated policy.
13 %% a5 = Extreme regulated policy.
14 %%
15 %% Scenarios:
16 %%
17 %% s1 = Koala Ebola is contained, and export is thriving.
18 %% s2 = Koala Ebola is contained, but export is not thriving.
19 %% s3 = Koala Ebola is pandemic, but export is thriving.
20 %% s4 = Koala Ebola is pandemic, and export is not thriving.
21 %%
22 %% Scenario probabilities:
23 %%
24 %% p1 = 0.75*p0,
25 %% p2 = 0.25*p0,
26 %% p3 = 0.25*q0,
27 %% p4 = 0.75*q0,
28 %%
29 %% where
30 %%
31 %% q0 = 1-p0
32 %%
33 %% Here p0 is a sensitivity parameter.
34 %%
35 %% Catherine uses the expected value rule and the Hurwicz rule.  They are
36 %% combined with equal weights.  The Hurwicz degree of optimism, w, is a
37 %% sensitivity parameter.
38 %%
39 %% This file uses the functions dmatrix and scaling.
40 %%
41 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
42
43 %% Decision matrix.
44
45 R = [
46   8    2    3   -5 ;
47   6    2    2   -3 ;
48   2    1    1   -2 ;
49   0   -1    0   -2 ;
50  -2   -1   -1   -1
51 ];
52
53 %% Fixed fields for decision problems.
54
```

```
55  dp_Hur.dmat = R;
56  dp_Hur.drule = "Hur";
57
58  dp_ev.dmat = R;
59  dp_ev.drule = "ev";
60
61  %% Calculations for changing p0 and w.
62
63  %% Sampling frequency.  For N=500 the calculations take 30 seconds.
64  N = 500;
65
66  p0samples = linspace(0,1,N);
67  wsamples = linspace(0,1,N);
68
69  for i=1:length(p0samples)
70      p0 = p0samples(i);
71      q0 = 1-p0;
72      dp_ev.prob = [0.75*p0  0.25*p0  0.25*q0  0.75*q0]';
73      dr_ev = dmatrix(dp_ev);
74      V_ev = scaling(dr_ev.dvalues);
75      for j=1:length(wsamples)
76          w = wsamples(j);
77          dp_Hur.odegree = w;
78          dr_Hur = dmatrix(dp_Hur);
79          V_Hur = scaling(dr_Hur.dvalues);
80          V = 0.5*(V_ev + V_Hur);
81          [tmp, z(j,i)] = max(V);
82      end
83  end
84
85  %% Draw the picture.
86  contourf(p0samples, wsamples, z)
87  set(gca, "fontsize", 24);
88  xlabel("p_0")
89  ylabel("w")
90  red    = [1 0 0];
91  green  = [0 1 0];
92  yellow = [1 1 0];
93  colormap([green; 0.6*green; yellow; 0.6*red; red])
94  text(0.70,0.70, "a_1", "fontsize",24)
95  text(0.20,0.50, "a_2", "fontsize",24)
96  text(0.10,0.10, "a_3", "fontsize",24)
97  text(0.01,0.03, "a_5", "fontsize",24)
98  %orient("landscape");
99  %print("cc_picture.pdf");
```

https://www.uwasa.fi/~tsottine/orms2020/en/example_cc.m

Appendix B: Secret Supplementary Protocol

The problem treatment was honest except for a one minor point: the decision matrix came from the "best economists': this is to say me. I cooked the numbers until I finally got what I wanted. This illustrates that the numbers in the decision matrix R are the most important ingredient in the decision matrix technique. Indeed, if the matrix numbers are good then the analysis is good, and if they are bullshit then the analysis is bullshit.

Part IV

Assignments

Chapter 7

Assignment 1: Your Fall

Problem Description

The assignment is to plan your next fall by using the decision matrix technique. You can use the backdrop of pending Koala Ebola pandemic: the Koala Ebola has been encountered in Outdonesia. Unfortunately, Outdonesia is a popular tourist attraction and the Koala Ebola has penetrated the entire land. Outdonesia has not enforced any kind of quarantine yet, so it may be only a matter of time before we may have a global pandemic at our hand. There is no vaccine for Koala Ebola yet, but there is a test for it.

Now, it is your task now to decide what to do next fall. You may consider **for example** the alternatives

- study full-time,
- some work with some studies,
- work full-time,
- take the semester off and find yourself by meditating in a Distantistanian monastery.

If you consider the pending Koala Ebola pandemic relevant to your alternatives, then the following scenarios **can be** relevant to you

- Is the society (and borders) closed due to Koala Ebola pandemic.
- Is the university in distant teaching.
- Is the world economy in recess.

You can use a completely different set of scenarios, if you want.

Then, consider one or two objectives. This means that you build one or two decision matrices. Choose opposite objectives to make the problem interesting. The decision matrices can be **for example**

- income,
- earned study credit points,
- personal (subjective) comfort.

After this, choose one or two decision rules.

Finally, choose two sensitivity parameters, plot your optimal decisions, and analyze the result.

For full three points you also have to consider a compromise problem. This can be done from the perspective of your spouse/mother/cat/dog, or whatever. It is enough to consider one decision matrix and one decision rule for your spouse/mother/cat/dog, whatever. Choose two sensitivity parameters for the compromise problem, plot your results, and analyze them.

Remark Chapter 5 is a guide on how to do this assignment. You should use its structure in your assignment, unless you find a different structure more suitable.

Guidelines

Below is a detailed list, or guidelines, on how to do your assignment, and what should be in it. These guidelines follow the Sophie Sophomore's example of Chapter 5. If you decide to be imaginative, and not imitate Sophie's example, you can take a relaxed attitude towards these guidelines.

(i) Introduce yourself and the problem at least in the level done in Chapter 5.
(ii) Describe the timeline of you problem.
(iii) Choose approximately 3–8 alternatives.
(iv) Choose approximately 3–8 scenarios.
(v) Estimate the probabilities for the scenarios. You may want to have the probabilities to depend on a sensitivity parameter p_0. If you use only non-stochastic decision rules, you can omit this point.
(vi) Choose approximately 1–2 objectives or decision matrices. You may want to pick opposite objectives to make your analysis more interesting.
(vii) Build the decision matrices. This is where you should spend most of your time. If the numbers in the decision matrices are reasonable, then your analysis is reasonable, and if not, then not.
(viii) In this point you may want to go through the parts (iii)–(vii) and see if they make a good setting as a whole. Revise if necessary
(ix) Choose approximately 1–3 decision rules
(x) Choose 2 sensitivity parameters and calculate the decision picture. Analyze the picture.
(xi) Investigate the decision problem from the point fo view of your spouse/mother/cat/dog, whatever.
(xii) Build the compromise decision picture with two sensitivity parameters and analyze the result. You may want to use the compromise combination weight w as a sensitivity parameter, or you can set $w = 0.5$ for equal weight.
(xiii) Return the completed assignment to the teacher via email as a PDF attachment. If you intend to also do the Assignment 2, return both assignments at the same time. Remember to include your student number when returning the assignments.

Grading

This Assignment 1 (also Assignment 2) is graded on the scale 0–3 grade points. 1 point means that you will pass the course with grade 1 (if you get no points from Assignment 2). Similarly, with 3 points you get the grade 3 (if you get no points from Assignment 2).

There are both qualitative and quantitative criteria in the grading. To get the points, you must meet both the qualitative and the quantitative criteria.

Qualitative criteria in the grading are:

3 p The assignment is written so that any reasonable human being can understand it. The appearance and the language is polished.

2 p The assignment is written so that anyone who has passed this course can understand it. The appearance and the language is reasonably good.

1 p The assignment is written so that the teacher can understand it. The appearance and the language does not have serious flaws.

Fo the quantitative criteria we assume that you imitate the example of Chapter 5. If you decide to be imaginative and not follow Sophie's example, then you can take a more relaxed attitude towards the quantitative criteria listed below, i.e., the teacher will reward imaginative solutions.

Quantitative criteria in grading are:

3 p All the points in (i)–(xiii) in the list of the previous section have been completed.

2 p In the list of the previous section, the "compromise points" (xi)–(xii) are not done, but all the other points are done.

1 p In the list of the previous section, the "compromise points" (xi)–(xii) are not done, and in the point (vi) and in the point (ix) only one decision matrix and one decision rule is implemented, but all the other points are done.

Minimal Example

The m-file `example_ll.m` listed below, is a short solved Assignment 1. It explains Laila Laihian's fall. If the contents and the picture of `example_ll.m` are explained in a report, then it would be a grade 1 solution to Assignment 1.

```
1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %%
3  %% FILE: example_ll.m
4  %%
5  %% Laila Laihian's fall: a minimal m-file for Assignment "Your Fall".
6  %% Adding report to these kind of calculations will give you an accepted
7  %% assignment with grade 1.
8  %%
9  %% Laila Laihian lives with her mother in Laihia and contemplates what to do
10 %% in the next fall.
11 %%
12 %% Laila considers the following alternatives:
13 %%
14 %% a1 = Move to Vaasa into a VOAS flat.
15 %% a2 = Commute daily between Laihia and Vaasa.
16 %% a3 = Distant studies from Laihia.
17 %%
18 %% The scenarios are:
19 %%
20 %% s1 = Koala Ebola is contained.
21 %% s2 = Koala Ebola is pandemic, Campus is open, teaching is partly distant.
```

```
22 %% s3 = Koala Ebola is pandemic, Campus is closed, teaching is distant.
23 %%
24 %% The scenario probabilities are
25 %%
26 %% p0 = Koala Ebola pandemic probability(p0 = p2+p3)
27 %% q0 = 1-p0
28 %%
29 %% p1 = q0
30 %% p2 = p0/2
31 %% p3 = p0/2
32 %%
33 %% Laila has two decision matrices and she uses the expected value rule.
34 %%
35 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
36
37 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
38 %% Decition matrices (multi-objective decision problem)
39 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
40
41 %% Monthly expenses
42 R1 = [
43 100+40    100+50   100+100;    %% VOAS flat   (rent+food)
44    120   0.5*120      0*120;   %% Driving to Vaasa (100%-50%-0% on site teaching)
45      0         0          0    %% Studying from Laihia   (baseline)
46 ];
47
48 R1 = -R1;      %% Maximun form!
49
50
51 %% Study credit points
52 R2 = [
53 30   25   20;   %% VOAS flat
54 25   25   20;   %% Driving to Vaasa
55 10   15   20    %% Studying from Laihia
56 ];
57
58 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
59 %% Sensitivity calculations and visuatization
60 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
61
62 %% Samplin frequency.  For N=500 the calculations take 3 seconds.
63 N = 500;
64 p0samples = linspace(0,1,N);
65 wsamples = linspace(0,1,N);
66
67 %% Decision problem's constant fields.
68 dp1.drule = "ev";
69 dp1.dmat = R1;
70 dp2.drule = "ev";
71 dp2.dmat = R2;
72
73 %% Calculating the sensitivity matrix z.
74 for i=1:N
75     p0 = p0samples(i);
76     q0 = 1-p0;
77     dp1.prob = [q0  p0/2  p0/2]';
78     dp2.prob = [q0  p0/2  p0/2]';
79     dr1 = dmatrix(dp1);
```

```
80       dr2 = dmatrix(dp2);
81       V1 = dr1.dvalues;
82       V1 = scaling(V1);
83       V2 = dr2.dvalues;
84       V2 = scaling(V2);
85       for j=1:N
86           w = wsamples(j);
87           V = w*V1 + (1-w)*V2;
88           [tmp, z(j,i)] = max(V);
89       end
90 end
91
92 %% Drawing the sensitivity matrix z.
93 contourf(p0samples, wsamples, z);
94 %% Uncomment the following lines for PDF file output.
95 %orient("landscape")
96 %print("ll_picture.pdf")
```

Below is the sensitivity picture the m-file example_ll.m produces:
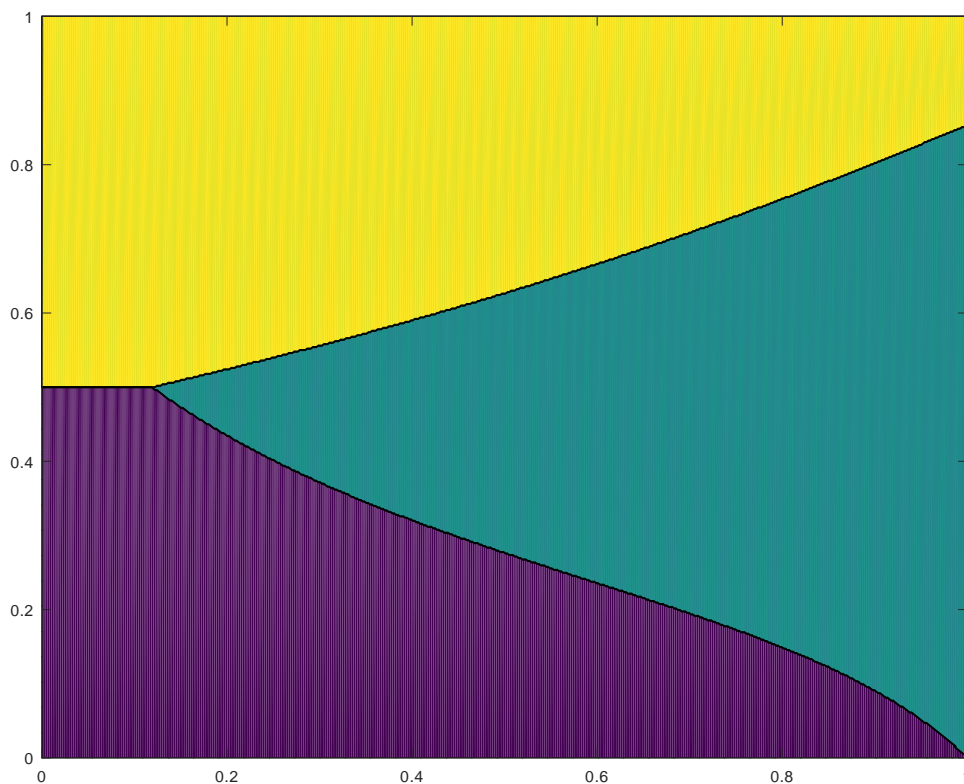
Chapter 8

Assignment 2: Ministry Report

Problem Description

You are a junior specialist in the Ministry of Health and Wellbeing. You are tasked to give recommendations for the government of Finland in the pending Koala Ebola pandemic. Recall that the Koala Ebola has been encountered in Outdonesia. Unfortunately, Outdonesia is a popular tourist attraction and the Koala Ebola has penetrated the entire land. Outdonesia has not enforced any kind of quarantine yet, so it may be only a matter of time before we may have a global pandemic at our hand. There is no vaccine for Koala Ebola yet, but there is a test for it.

You may consider, **for example**, the following actions

- travel restrictions to Outdonesa,
- quarantine to those who have traveled to Outdonesia,
- closing borders,
- quarantining Finland,
- wait for the vaccine and then (forcefully) vaccinate everybody.

Your objectives can be, **for example**,

- lost healthy living years,
- GDP change,
- "civic liberties".

The decision rules can be, **for example**,

- expected value rule,
- Hurwicz rule,
- CARA rule.

Build your decision matrix or matrices and perform sensitivity analysis by using two sensitivity parameters.

Remark The example of Chapter 6 may help a little in doing this assignment. Also note that unlike in the Assignment 1, in this Assignment 2 there is no need to do compromise analysis. If you wish, you can do the compromise analysis. You may choose for example another ministry, say the Treasury, and its report.

Guidelines

Unlike Assignment 1, this Assignment 2 requires some independent thinking: it is not a straightforward version of the example of Chapter 5 as Assignment 1 is. Therefore the assignment itself is narrower is scope. Indeed, below are the guidelines for Assignment 2:

   (i) Choose approximately 3–8 alternatives.
  (ii) Choose approximately 3–8 scenarios
 (iii) Estimate the probabilities for the scenarios. You may want to have the Koala Ebola pandemic probability as a sensitivity parameter p_0. IF you use only non-stochastic decision rules, you can mot this point.
  (iv) Choose approximately 1–2 objectives or decision matrices. You may want to pick opposite objectives to make your analysis more interesting
   (v) Build the decision matrices. This is where you should send most of your time. If the numbers in the decision matrices are reasonable, then your analysis is reasonable, and if not, then not.
  (vi) In this point you may want to go through the parts (i)–(v) and see if they make a good setting as a whole. Revise if necessary.
 (vii) Choose approximately 1–3 decision rules.
(viii) Choose 2 sensitivity parameters and calculate the decision picture. Analyze the picture.
  (ix) State the final recommendation of the government of Finland.
   (x) Return the completed assignment to the teacher via email as a PDF attachment. If you intend to also do the Assignment 1, return both assignments at the same time. Remember to include your student number when returning the assignments.

Grading

This Assignment 2 is graded almost in the same way as Assignment 1.

   Qualitative criteria in the grading are:

3 p The assignment is written so that any reasonable human being can understand it. The appearance and the language is polished.
2 p The assignment is written so that anyone who has passed this course can understand it. The appearance and the language is reasonably good.
1 p The assignment is written so that the teacher can understand it. The appearance and the language does not have serious flaws.

   Quantitative criteria in grading are:

3 p All the points (i)–(x) in the list of the previous section have been completed. The numbers in the decision matrices as well as the scenario probabilities are well argued.
2 p All the points (i)–(x) in the list of the previous section have been completed. The numbers in the decision matrices or the scenario probabilities are not well argued.
1 p All the points (i)–(x) in the list of the previous section have been done. The numbers in the decision matrices and the scenario probabilities are not well argued. The sensitivity analysis is lacking.