

Robustness

Jesse Perla, Thomas J. Sargent and John Stachurski

September 8, 2020

1 Contents

- Overview [2](#)
- The Model [3](#)
- Constructing More Robust Policies [4](#)
- Robustness as Outcome of a Two-Person Zero-Sum Game [5](#)
- The Stochastic Case [6](#)
- Implementation [7](#)
- Application [8](#)
- Appendix [9](#)

2 Overview

This lecture modifies a Bellman equation to express a decision maker's doubts about transition dynamics.

His specification doubts make the decision maker want a *robust* decision rule.

Robust means insensitive to misspecification of transition dynamics.

The decision maker has a single *approximating model*.

He calls it *approximating* to acknowledge that he doesn't completely trust it.

He fears that outcomes will actually be determined by another model that he cannot describe explicitly.

All that he knows is that the actual data-generating model is in some (uncountable) set of models that surrounds his approximating model.

He quantifies the discrepancy between his approximating model and the genuine data-generating model by using a quantity called *entropy*.

(We'll explain what entropy means below)

He wants a decision rule that will work well enough no matter which of those other models actually governs outcomes.

This is what it means for his decision rule to be "robust to misspecification of an approximating model".

This may sound like too much to ask for, but ...

... a *secret weapon* is available to design robust decision rules.

The secret weapon is max-min control theory.

A value-maximizing decision maker enlists the aid of an (imaginary) value-minimizing model chooser to construct *bounds* on the value attained by a given decision rule under different models of the transition dynamics.

The original decision maker uses those bounds to construct a decision rule with an assured performance level, no matter which model actually governs outcomes.

Note

In reading this lecture, please don't think that our decision maker is paranoid when he conducts a worst-case analysis. By designing a rule that works well against a worst-case, his intention is to construct a rule that will work well across a *set* of models.

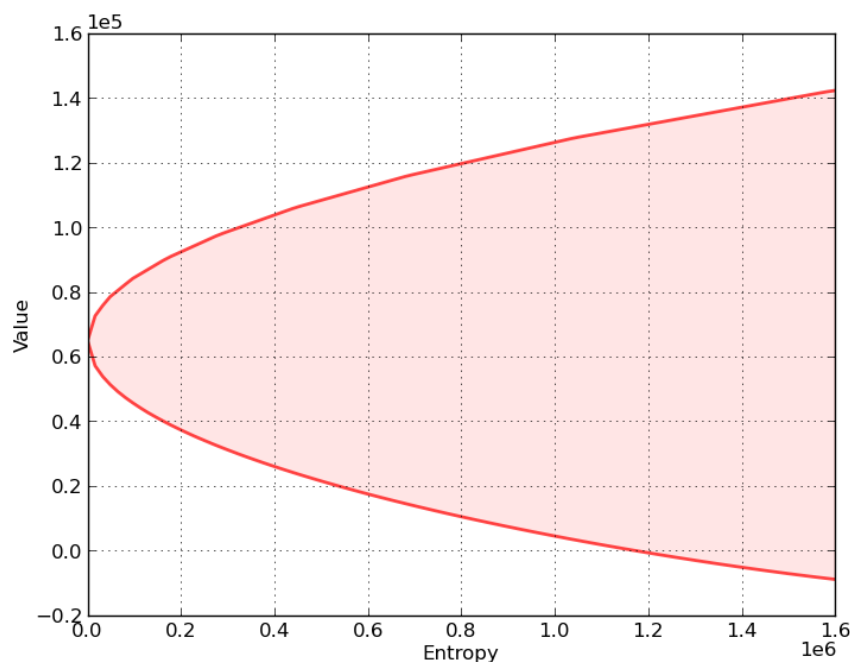
2.1 Sets of Models Imply Sets Of Values

Our "robust" decision maker wants to know how well a given rule will work when he does not *know* a single transition law ...

... he wants to know *sets* of values that will be attained by a given decision rule F under a *set* of transition laws.

Ultimately, he wants to design a decision rule F that shapes these *sets* of values in ways that he prefers.

With this in mind, consider the following graph, which relates to a particular decision problem to be explained below



The figure shows a *value-entropy correspondence* for a particular decision rule F .

The shaded set is the graph of the correspondence, which maps entropy to a set of values associated with a set of models that surround the decision maker's approximating model.

Here

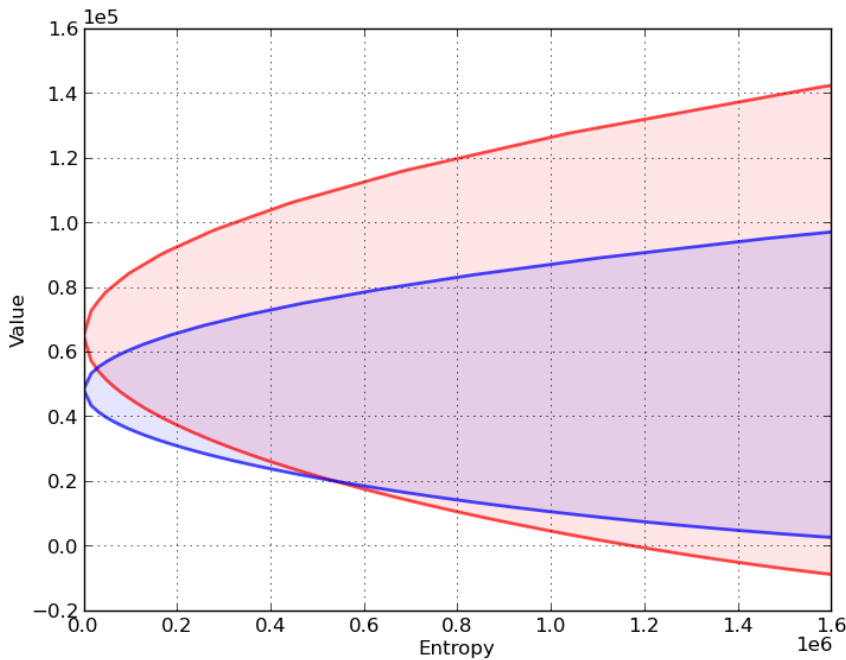
- *Value* refers to a sum of discounted rewards obtained by applying the decision rule F when the state starts at some fixed initial state x_0 .
- *Entropy* is a nonnegative number that measures the size of a set of models surrounding the decision maker's approximating model.
 - Entropy is zero when the set includes only the approximating model, indicating that the decision maker completely trusts the approximating model.
 - Entropy is bigger, and the set of surrounding models is bigger, the less the decision maker trusts the approximating model.

The shaded region indicates that for **all** models having entropy less than or equal to the number on the horizontal axis, the value obtained will be somewhere within the indicated set of values.

Now let's compare sets of values associated with two different decision rules, F_r and F_b .

In the next figure,

- The red set shows the value-entropy correspondence for decision rule F_r .
- The blue set shows the value-entropy correspondence for decision rule F_b .



The blue correspondence is skinnier than the red correspondence.

This conveys the sense in which the decision rule F_b is *more robust* than the decision rule F_r .

- *more robust* means that the set of values is less sensitive to *increasing misspecification* as measured by entropy.

Notice that the less robust rule F_r promises higher values for small misspecifications (small entropy).

(But it is more fragile in the sense that it is more sensitive to perturbations of the approximating model)

Below we'll explain in detail how to construct these sets of values for a given F , but for now ...

Here is a hint about the *secret weapons* we'll use to construct these sets

- We'll use some min problems to construct the lower bounds.
- We'll use some max problems to construct the upper bounds.

We will also describe how to choose F to shape the sets of values.

This will involve crafting a *skinnier* set at the cost of a lower *level* (at least for low values of entropy).

2.2 Inspiring Video

If you want to understand more about why one serious quantitative researcher is interested in this approach, we recommend [Lars Peter Hansen's Nobel lecture](#).

2.3 Other References

Our discussion in this lecture is based on

- [2]
- [1]

2.4 Setup

```
In [1]: using InstantiateFromURL
        # optionally add arguments to force installation: instantiate = true,
        ↪precompile = true
        github_project("QuantEcon/quantecon-notebooks-julia", version = "0.8.0")
```

```
In [2]: using LinearAlgebra, Statistics
```

3 The Model

For simplicity, we present ideas in the context of a class of problems with linear transition laws and quadratic objective functions.

To fit in with [our earlier lecture on LQ control](#), we will treat loss minimization rather than value maximization.

To begin, recall the [infinite horizon LQ problem](#), where an agent chooses a sequence of controls $\{u_t\}$ to minimize

$$\sum_{t=0}^{\infty} \beta^t \{x_t' R x_t + u_t' Q u_t\} \quad (1)$$

subject to the linear law of motion

$$x_{t+1} = Ax_t + Bu_t + Cw_{t+1}, \quad t = 0, 1, 2, \dots \quad (2)$$

As before,

- x_t is $n \times 1$, A is $n \times n$
- u_t is $k \times 1$, B is $n \times k$
- w_t is $j \times 1$, C is $n \times j$
- R is $n \times n$ and Q is $k \times k$

Here x_t is the state, u_t is the control, and w_t is a shock vector.

For now we take $\{w_t\} := \{w_t\}_{t=1}^{\infty}$ to be deterministic — a single fixed sequence.

We also allow for *model uncertainty* on the part of the agent solving this optimization problem.

In particular, the agent takes $w_t = 0$ for all $t \geq 0$ as a benchmark model, but admits the possibility that this model might be wrong.

As a consequence, she also considers a set of alternative models expressed in terms of sequences $\{w_t\}$ that are “close” to the zero sequence.

She seeks a policy that will do well enough for a set of alternative models whose members are pinned down by sequences $\{w_t\}$.

Soon we’ll quantify the quality of a model specification in terms of the maximal size of the expression $\sum_{t=0}^{\infty} \beta^{t+1} w_{t+1}' w_{t+1}$.

4 Constructing More Robust Policies

If our agent takes $\{w_t\}$ as a given deterministic sequence, then, drawing on intuition from earlier lectures on dynamic programming, we can anticipate Bellman equations such as

$$J_{t-1}(x) = \min_u \{x' R x + u' Q u + \beta J_t(Ax + Bu + Cw_t)\}$$

(Here J depends on t because the sequence $\{w_t\}$ is not recursive)

Our tool for studying robustness is to construct a rule that works well even if an adverse sequence $\{w_t\}$ occurs.

In our framework, “adverse” means “loss increasing”.

As we’ll see, this will eventually lead us to construct the Bellman equation.

$$J(x) = \min_u \max_w \{x' R x + u' Q u + \beta [J(Ax + Bu + Cw) - \theta w' w]\} \quad (3)$$

Notice that we’ve added the penalty term $-\theta w' w$.

Since $w'w = \|w\|^2$, this term becomes influential when w moves away from the origin.

The penalty parameter θ controls how much we penalize the maximizing agent for “harming” the minimizing agent.

By raising θ more and more, we more and more limit the ability of maximizing agent to distort outcomes relative to the approximating model.

So bigger θ is implicitly associated with smaller distortion sequences $\{w_t\}$.

4.1 Analyzing the Bellman equation

So what does J in (3) look like?

As with the [ordinary LQ control model](#), J takes the form $J(x) = x'Px$ for some symmetric positive definite matrix P .

One of our main tasks will be to analyze and compute the matrix P .

Related tasks will be to study associated feedback rules for u_t and w_{t+1} .

First, using [matrix calculus](#), you will be able to verify that

$$\begin{aligned} \max_w \{ (Ax + Bu + Cw)'P(Ax + Bu + Cw) - \theta w'w \} \\ = (Ax + Bu)' \mathcal{D}(P)(Ax + Bu) \end{aligned} \quad (4)$$

where

$$\mathcal{D}(P) := P + PC(\theta I - C'PC)^{-1}C'P \quad (5)$$

and I is a $j \times j$ identity matrix. Substituting this expression for the maximum into (3) yields

$$x'Px = \min_u \{ x'Rx + u'Qu + \beta (Ax + Bu)' \mathcal{D}(P)(Ax + Bu) \} \quad (6)$$

Using similar mathematics, the solution to this minimization problem is $u = -Fx$ where $F := (Q + \beta B' \mathcal{D}(P)B)^{-1} \beta B' \mathcal{D}(P)A$.

Substituting this minimizer back into (6) and working through the algebra gives $x'Px = x' \mathcal{B}(\mathcal{D}(P))x$ for all x , or, equivalently,

$$P = \mathcal{B}(\mathcal{D}(P))$$

where \mathcal{D} is the operator defined in (5) and

$$\mathcal{B}(P) := R - \beta^2 A'PB(Q + \beta B'PB)^{-1}B'PA + \beta A'PA$$

The operator \mathcal{B} is the standard (i.e., non-robust) LQ Bellman operator, and $P = \mathcal{B}(P)$ is the standard matrix Riccati equation coming from the Bellman equation — see [this discussion](#).

Under some regularity conditions (see [1]), the operator $\mathcal{B} \circ \mathcal{D}$ has a unique positive definite fixed point, which we denote below by \hat{P} .

A robust policy, indexed by θ , is $u = -\hat{F}x$ where

$$\hat{F} := (Q + \beta B' \mathcal{D}(\hat{P})B)^{-1} \beta B' \mathcal{D}(\hat{P})A \quad (7)$$

We also define

$$\hat{K} := (\theta I - C' \hat{P}C)^{-1} C' \hat{P}(A - B\hat{F}) \quad (8)$$

The interpretation of \hat{K} is that $w_{t+1} = \hat{K}x_t$ on the worst-case path of $\{x_t\}$, in the sense that this vector is the maximizer of (4) evaluated at the fixed rule $u = -\hat{F}x$.

Note that $\hat{P}, \hat{F}, \hat{K}$ are all determined by the primitives and θ .

Note also that if θ is very large, then \mathcal{D} is approximately equal to the identity mapping.

Hence, when θ is large, \hat{P} and \hat{F} are approximately equal to their standard LQ values.

Furthermore, when θ is large, \hat{K} is approximately equal to zero.

Conversely, smaller θ is associated with greater fear of model misspecification, and greater concern for robustness.

5 Robustness as Outcome of a Two-Person Zero-Sum Game

What we have done above can be interpreted in terms of a two-person zero-sum game in which \hat{F}, \hat{K} are Nash equilibrium objects.

Agent 1 is our original agent, who seeks to minimize loss in the LQ program while admitting the possibility of misspecification.

Agent 2 is an imaginary malevolent player.

Agent 2's malevolence helps the original agent to compute bounds on his value function across a set of models.

We begin with agent 2's problem.

5.1 Agent 2's Problem

Agent 2

1. knows a fixed policy F specifying the behavior of agent 1, in the sense that $u_t = -Fx_t$ for all t
2. responds by choosing a shock sequence $\{w_t\}$ from a set of paths sufficiently close to the benchmark sequence $\{0, 0, 0, \dots\}$

A natural way to say ‘‘sufficiently close to the zero sequence’’ is to restrict the summed inner product $\sum_{t=1}^{\infty} w_t' w_t$ to be small.

However, to obtain a time-invariant recursive formulation, it turns out to be convenient to restrict a discounted inner product

$$\sum_{t=1}^{\infty} \beta^t w_t' w_t \leq \eta \quad (9)$$

Now let F be a fixed policy, and let $J_F(x_0, \mathbf{w})$ be the present-value cost of that policy given sequence $\mathbf{w} := \{w_t\}$ and initial condition $x_0 \in \mathbb{R}^n$.

Substituting $-Fx_t$ for u_t in (1), this value can be written as

$$J_F(x_0, \mathbf{w}) := \sum_{t=0}^{\infty} \beta^t x'_t (R + F'QF)x_t \quad (10)$$

where

$$x_{t+1} = (A - BF)x_t + Cw_{t+1} \quad (11)$$

and the initial condition x_0 is as specified in the left side of (10).

Agent 2 chooses \mathbf{w} to maximize agent 1's loss $J_F(x_0, \mathbf{w})$ subject to (9).

Using a Lagrangian formulation, we can express this problem as

$$\max_{\mathbf{w}} \sum_{t=0}^{\infty} \beta^t \{x'_t (R + F'QF)x_t - \beta\theta(w'_{t+1}w_{t+1} - \eta)\}$$

where $\{x_t\}$ satisfied (11) and θ is a Lagrange multiplier on constraint (9).

For the moment, let's take θ as fixed, allowing us to drop the constant $\beta\theta\eta$ term in the objective function, and hence write the problem as

$$\max_{\mathbf{w}} \sum_{t=0}^{\infty} \beta^t \{x'_t (R + F'QF)x_t - \beta\theta w'_{t+1}w_{t+1}\}$$

or, equivalently,

$$\min_{\mathbf{w}} \sum_{t=0}^{\infty} \beta^t \{-x'_t (R + F'QF)x_t + \beta\theta w'_{t+1}w_{t+1}\} \quad (12)$$

subject to (11).

What's striking about this optimization problem is that it is once again an LQ discounted dynamic programming problem, with $\mathbf{w} = \{w_t\}$ as the sequence of controls.

The expression for the optimal policy can be found by applying the usual LQ formula ([see here](#)).

We denote it by $K(F, \theta)$, with the interpretation $w_{t+1} = K(F, \theta)x_t$.

The remaining step for agent 2's problem is to set θ to enforce the constraint (9), which can be done by choosing $\theta = \theta_\eta$ such that

$$\beta \sum_{t=0}^{\infty} \beta^t x'_t K(F, \theta_\eta)' K(F, \theta_\eta) x_t = \eta \quad (13)$$

Here x_t is given by (11) — which in this case becomes $x_{t+1} = (A - BF + CK(F, \theta))x_t$.

5.2 Using Agent 2's Problem to Construct Bounds on the Value Sets

5.2.1 The Lower Bound

Define the minimized object on the right side of problem (12) as $R_\theta(x_0, F)$.

Because “minimizers minimize” we have

$$R_\theta(x_0, F) \leq \sum_{t=0}^{\infty} \beta^t \{-x'_t(R + F'QF)x_t\} + \beta\theta \sum_{t=0}^{\infty} \beta^t w'_{t+1} w_{t+1},$$

where $x_{t+1} = (A - BF + CK(F, \theta))x_t$ and x_0 is a given initial condition.

This inequality in turn implies the inequality

$$R_\theta(x_0, F) - \theta \text{ ent} \leq \sum_{t=0}^{\infty} \beta^t \{-x'_t(R + F'QF)x_t\} \quad (14)$$

where

$$\text{ent} := \beta \sum_{t=0}^{\infty} \beta^t w'_{t+1} w_{t+1}$$

The left side of inequality (14) is a straight line with slope $-\theta$.

Technically, it is a “separating hyperplane”.

At a particular value of entropy, the line is tangent to the lower bound of values as a function of entropy.

In particular, the lower bound on the left side of (14) is attained when

$$\text{ent} = \beta \sum_{t=0}^{\infty} \beta^t x'_t K(F, \theta)' K(F, \theta) x_t \quad (15)$$

To construct the *lower bound* on the set of values associated with all perturbations \mathbf{w} satisfying the entropy constraint (9) at a given entropy level, we proceed as follows:

- For a given θ , solve the minimization problem (12).
- Compute the minimizer $R_\theta(x_0, F)$ and the associated entropy using (15).
- Compute the lower bound on the value function $R_\theta(x_0, F) - \theta \text{ ent}$ and plot it against ent .
- Repeat the preceding three steps for a range of values of θ to trace out the lower bound.

Note

This procedure sweeps out a set of separating hyperplanes indexed by different values for the Lagrange multiplier θ .

5.2.2 The Upper Bound

To construct an *upper bound* we use a very similar procedure.

We simply replace the *minimization* problem (12) with the *maximization* problem.

$$V_{\tilde{\theta}}(x_0, F) = \max_{\mathbf{w}} \sum_{t=0}^{\infty} \beta^t \left\{ -x'_t (R + F' Q F) x_t - \beta \tilde{\theta} w'_{t+1} w_{t+1} \right\} \quad (16)$$

where now $\tilde{\theta} > 0$ penalizes the choice of \mathbf{w} with larger entropy.

(Notice that $\tilde{\theta} = -\theta$ in problem (12))

Because “maximizers maximize” we have

$$V_{\tilde{\theta}}(x_0, F) \geq \sum_{t=0}^{\infty} \beta^t \left\{ -x'_t (R + F' Q F) x_t \right\} - \beta \tilde{\theta} \sum_{t=0}^{\infty} \beta^t w'_{t+1} w_{t+1}$$

which in turn implies the inequality

$$V_{\tilde{\theta}}(x_0, F) + \tilde{\theta} \text{ent} \geq \sum_{t=0}^{\infty} \beta^t \left\{ -x'_t (R + F' Q F) x_t \right\} \quad (17)$$

where

$$\text{ent} \equiv \beta \sum_{t=0}^{\infty} \beta^t w'_{t+1} w_{t+1}$$

The left side of inequality (17) is a straight line with slope $\tilde{\theta}$.

The upper bound on the left side of (17) is attained when

$$\text{ent} = \beta \sum_{t=0}^{\infty} \beta^t x'_t K(F, \tilde{\theta})' K(F, \tilde{\theta}) x_t \quad (18)$$

To construct the *upper bound* on the set of values associated all perturbations \mathbf{w} with a given entropy we proceed much as we did for the lower bound.

- For a given $\tilde{\theta}$, solve the maximization problem (16).
- Compute the maximizer $V_{\tilde{\theta}}(x_0, F)$ and the associated entropy using (18).
- Compute the upper bound on the value function $V_{\tilde{\theta}}(x_0, F) + \tilde{\theta} \text{ent}$ and plot it against ent .
- Repeat the preceding three steps for a range of values of $\tilde{\theta}$ to trace out the upper bound.

5.2.3 Reshaping the set of values

Now in the interest of *reshaping* these sets of values by choosing F , we turn to agent 1’s problem.

5.3 Agent 1's Problem

Now we turn to agent 1, who solves

$$\min_{\{u_t\}} \sum_{t=0}^{\infty} \beta^t \{x_t' R x_t + u_t' Q u_t - \beta \theta w_{t+1}' w_{t+1}\} \quad (19)$$

where $\{w_{t+1}\}$ satisfies $w_{t+1} = K x_t$.

In other words, agent 1 minimizes

$$\sum_{t=0}^{\infty} \beta^t \{x_t' (R - \beta \theta K' K) x_t + u_t' Q u_t\} \quad (20)$$

subject to

$$x_{t+1} = (A + CK)x_t + Bu_t \quad (21)$$

Once again, the expression for the optimal policy can be found [here](#) — we denote it by \tilde{F} .

5.4 Nash Equilibrium

Clearly the \tilde{F} we have obtained depends on K , which, in agent 2's problem, depended on an initial policy F .

Holding all other parameters fixed, we can represent this relationship as a mapping Φ , where

$$\tilde{F} = \Phi(K(F, \theta))$$

The map $F \mapsto \Phi(K(F, \theta))$ corresponds to a situation in which

1. agent 1 uses an arbitrary initial policy F
2. agent 2 best responds to agent 1 by choosing $K(F, \theta)$
3. agent 1 best responds to agent 2 by choosing $\tilde{F} = \Phi(K(F, \theta))$

As you may have already guessed, the robust policy \hat{F} defined in (7) is a fixed point of the mapping Φ .

In particular, for any given θ ,

1. $K(\hat{F}, \theta) = \hat{K}$, where \hat{K} is as given in (8)
2. $\Phi(\hat{K}) = \hat{F}$

A sketch of the proof is given in [the appendix](#).

6 The Stochastic Case

Now we turn to the stochastic case, where the sequence $\{w_t\}$ is treated as an iid sequence of random vectors.

In this setting, we suppose that our agent is uncertain about the *conditional probability distribution* of w_{t+1} .

The agent takes the standard normal distribution $N(0, I)$ as the baseline conditional distribution, while admitting the possibility that other “nearby” distributions prevail.

These alternative conditional distributions of w_{t+1} might depend nonlinearly on the history $x_s, s \leq t$.

To implement this idea, we need a notion of what it means for one distribution to be near another one.

Here we adopt a very useful measure of closeness for distributions known as the *relative entropy*, or **Kullback-Leibler divergence**.

For densities p, q , the Kullback-Leibler divergence of q from p is defined as

$$D_{KL}(p, q) := \int \ln \left[\frac{p(x)}{q(x)} \right] p(x) dx$$

Using this notation, we replace (3) with the stochastic analogue

$$J(x) = \min_u \max_{\psi \in \mathcal{P}} \left\{ x'Rx + u'Qu + \beta \left[\int J(Ax + Bu + Cw) \psi(dw) - \theta D_{KL}(\psi, \phi) \right] \right\} \quad (22)$$

Here \mathcal{P} represents the set of all densities on \mathbb{R}^n and ϕ is the benchmark distribution $N(0, I)$.

The distribution ϕ is chosen as the least desirable conditional distribution in terms of next period outcomes, while taking into account the penalty term $\theta D_{KL}(\psi, \phi)$.

This penalty term plays a role analogous to the one played by the deterministic penalty $\theta w'w$ in (3), since it discourages large deviations from the benchmark.

6.1 Solving the Model

The maximization problem in (22) appears highly nontrivial — after all, we are maximizing over an infinite dimensional space consisting of the entire set of densities.

However, it turns out that the solution is tractable, and in fact also falls within the class of normal distributions.

First, we note that J has the form $J(x) = x'Px + d$ for some positive definite matrix P and constant real number d .

Moreover, it turns out that if $(I - \theta^{-1}C'PC)^{-1}$ is nonsingular, then

$$\begin{aligned} \max_{\psi \in \mathcal{P}} \left\{ \int (Ax + Bu + Cw)'P(Ax + Bu + Cw) \psi(dw) - \theta D_{KL}(\psi, \phi) \right\} \\ = (Ax + Bu)' \mathcal{D}(P)(Ax + Bu) + \kappa(\theta, P) \end{aligned} \quad (23)$$

where

$$\kappa(\theta, P) := \theta \ln[\det(I - \theta^{-1}C'PC)^{-1}]$$

and the maximizer is the Gaussian distribution

$$\psi = N((\theta I - C'PC)^{-1}C'P(Ax + Bu), (I - \theta^{-1}C'PC)^{-1}) \quad (24)$$

Substituting the expression for the maximum into Bellman equation (22) and using $J(x) = x'Px + d$ gives

$$x'Px + d = \min_u \{x'Rx + u'Qu + \beta(Ax + Bu)'D(P)(Ax + Bu) + \beta[d + \kappa(\theta, P)]\} \quad (25)$$

Since constant terms do not affect minimizers, the solution is the same as (6), leading to

$$x'Px + d = x'\mathcal{B}(\mathcal{D}(P))x + \beta[d + \kappa(\theta, P)]$$

To solve this Bellman equation, we take \hat{P} to be the positive definite fixed point of $\mathcal{B} \circ \mathcal{D}$.

In addition, we take \hat{d} as the real number solving $d = \beta[d + \kappa(\theta, P)]$, which is

$$\hat{d} := \frac{\beta}{1 - \beta} \kappa(\theta, P) \quad (26)$$

The robust policy in this stochastic case is the minimizer in (25), which is once again $u = -\hat{F}x$ for \hat{F} given by (7).

Substituting the robust policy into (24) we obtain the worst case shock distribution:

$$w_{t+1} \sim N(\hat{K}x_t, (I - \theta^{-1}C'\hat{P}C)^{-1})$$

where \hat{K} is given by (8).

Note that the mean of the worst-case shock distribution is equal to the same worst-case w_{t+1} as in the earlier deterministic setting.

6.2 Computing Other Quantities

Before turning to implementation, we briefly outline how to compute several other quantities of interest.

6.2.1 Worst-Case Value of a Policy

One thing we will be interested in doing is holding a policy fixed and computing the discounted loss associated with that policy.

So let F be a given policy and let $J_F(x)$ be the associated loss, which, by analogy with (22), satisfies

$$J_F(x) = \max_{\psi \in \mathcal{P}} \left\{ x'(R + F'QF)x + \beta \left[\int J_F((A - BF)x + Cw) \psi(dw) - \theta D_{KL}(\psi, \phi) \right] \right\}$$

Writing $J_F(x) = x'P_Fx + d_F$ and applying the same argument used to derive (23) we get

$$x'P_Fx + d_F = x'(R + F'QF)x + \beta [x'(A - BF)'\mathcal{D}(P_F)(A - BF)x + d_F + \kappa(\theta, P_F)]$$

To solve this we take P_F to be the fixed point

$$P_F = R + F'QF + \beta(A - BF)'\mathcal{D}(P_F)(A - BF)$$

and

$$d_F := \frac{\beta}{1 - \beta} \kappa(\theta, P_F) = \frac{\beta}{1 - \beta} \theta \ln[\det(I - \theta^{-1}C'P_FC)^{-1}] \quad (27)$$

If you skip ahead to [the appendix](#), you will be able to verify that $-P_F$ is the solution to the Bellman equation in agent 2's problem [discussed above](#) — we use this in our computations.

7 Implementation

The [QuantEcon.jl](#) package provides a type called **RBLQ** for implementation of robust LQ optimal control.

The code can be found [on GitHub](#).

Here is a brief description of the methods of the type

- `d_operator()` and `b_operator()` implement \mathcal{D} and \mathcal{B} respectively
- `robust_rule()` and `robust_rule_simple()` both solve for the triple $\hat{F}, \hat{K}, \hat{P}$, as described in equations (7) – (8) and the surrounding discussion
 - `robust_rule()` is more efficient
 - `robust_rule_simple()` is more transparent and easier to follow
- `K_to_F()` and `F_to_K()` solve the decision problems of [agent 1](#) and [agent 2](#) respectively
- `compute_deterministic_entropy()` computes the left-hand side of (13)
- `evaluate_F()` computes the loss and entropy associated with a given policy — see [this discussion](#)

8 Application

Let us consider a monopolist similar to [this one](#), but now facing model uncertainty.

The inverse demand function is $p_t = a_0 - a_1y_t + d_t$

where

$$d_{t+1} = \rho d_t + \sigma_d w_{t+1}, \quad \{w_t\} \stackrel{\text{iid}}{\sim} N(0, 1)$$

and all parameters are strictly positive.

The period return function for the monopolist is

$$r_t = p_t y_t - \gamma \frac{(y_{t+1} - y_t)^2}{2} - c y_t$$

Its objective is to maximize expected discounted profits, or, equivalently, to minimize $\mathbb{E} \sum_{t=0}^{\infty} \beta^t (-r_t)$.

To form a linear regulator problem, we take the state and control to be

$$x_t = \begin{bmatrix} 1 \\ y_t \\ d_t \end{bmatrix} \quad \text{and} \quad u_t = y_{t+1} - y_t$$

Setting $b := (a_0 - c)/2$ we define

$$R = - \begin{bmatrix} 0 & b & 0 \\ b & -a_1 & 1/2 \\ 0 & 1/2 & 0 \end{bmatrix} \quad \text{and} \quad Q = \gamma/2$$

For the transition matrices we set

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \rho \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} 0 \\ 0 \\ \sigma_d \end{bmatrix}$$

Our aim is to compute the value-entropy correspondences [shown above](#).

The parameters are

$$a_0 = 100, a_1 = 0.5, \rho = 0.9, \sigma_d = 0.05, \beta = 0.95, c = 2, \gamma = 50.0$$

The standard normal distribution for w_t is understood as the agent's baseline, with uncertainty parameterized by θ .

We compute value-entropy correspondences for two policies.

1. The no concern for robustness policy F_0 , which is the ordinary LQ loss minimizer.
2. A “moderate” concern for robustness policy F_b , with $\theta = 0.02$.

The code for producing the graph shown above, with blue being for the robust policy, is as follows

```
In [3]: using QuantEcon, Plots, LinearAlgebra, Interpolations
        gr(fmt = :png);

        # model parameters
        a_0 = 100
        a_1 = 0.5
        ρ = 0.9
        σ_d = 0.05
```

```

β = 0.95
c = 2
γ = 50.0
θ = 0.002
ac = (a_0 - c) / 2.0

# Define LQ matrices
R = [ 0   ac   0;
      ac -a_1 0.5;
      0.  0.5  0]
R = -R # For minimization
Q = Matrix([γ / 2.0]')
A = [1. 0. 0.;
      0. 1. 0.;
      0. 0. ρ]
B = [0. 1. 0.]'
C = [0. 0. σ_d]'

## Functions

function evaluate_policy(θ, F)
    rlq = RBLQ(Q, R, A, B, C, β, θ)
    K_F, P_F, d_F, O_F, o_F = evaluate_F(rlq, F)
    x0 = [1.0 0.0 0.0]'
    value = - x0' * P_F * x0 .- d_F
    entropy = x0' * O_F * x0 .+ o_F
    return value[1], entropy[1] # return scalars
end

function value_and_entropy(emax, F, bw, grid_size = 1000)
    if lowercase(bw) == "worst"
        θs = 1 ./ range(1e-8, 1000, length = grid_size)
    else
        θs = -1 ./ range(1e-8, 1000, length = grid_size)
    end

    data = zeros(grid_size, 2)

    for (i, θ) in enumerate(θs)
        data[i, :] = collect(evaluate_policy(θ, F))
        if data[i, 2] ≥ emax # stop at this entropy level
            data = data[1:i, :]
            break
        end
    end
    return data
end

## Main

# compute optimal rule
optimal_lq = QuantEcon.LQ(Q, R, A, B, C, zero(B'A), bet=β)
Po, Fo, Do = stationary_values(optimal_lq)

# compute robust rule for our θ
baseline_robust = RBLQ(Q, R, A, B, C, β, θ)
Fb, Kb, Pb = robust_rule(baseline_robust)

```



```

# Check the positive definiteness of worst-case covariance matrix to
# ensure that  $\theta$  exceeds the breakdown point
test_matrix = I - (C' * Pb * C ./  $\theta$ )[1]
eigenvals, eigenvecs = eigen(test_matrix)
@assert all(x -> x  $\geq$   $\theta$ , eigenvals)

emax = 1.6e6

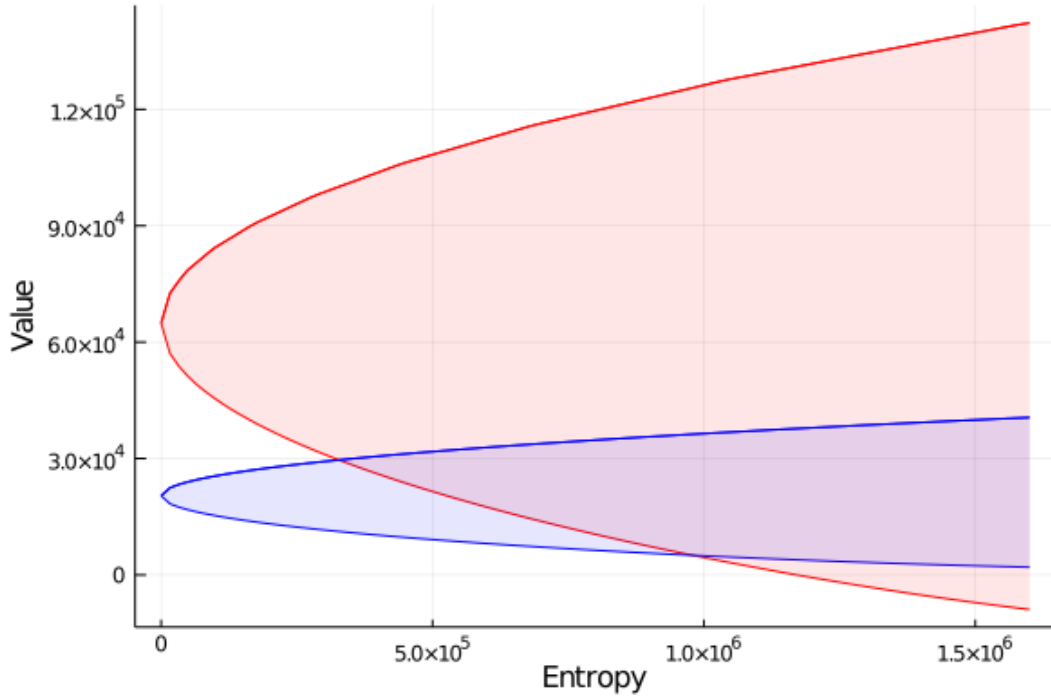
# compute values and entropies
optimal_best_case = value_and_entropy(emax, Fo, "best")
robust_best_case = value_and_entropy(emax, Fb, "best")
optimal_worst_case = value_and_entropy(emax, Fo, "worst")
robust_worst_case = value_and_entropy(emax, Fb, "worst")

# we reverse order of "worst_case"s so values are ascending
data_pairs = ((optimal_best_case, optimal_worst_case),
              (robust_best_case, robust_worst_case))

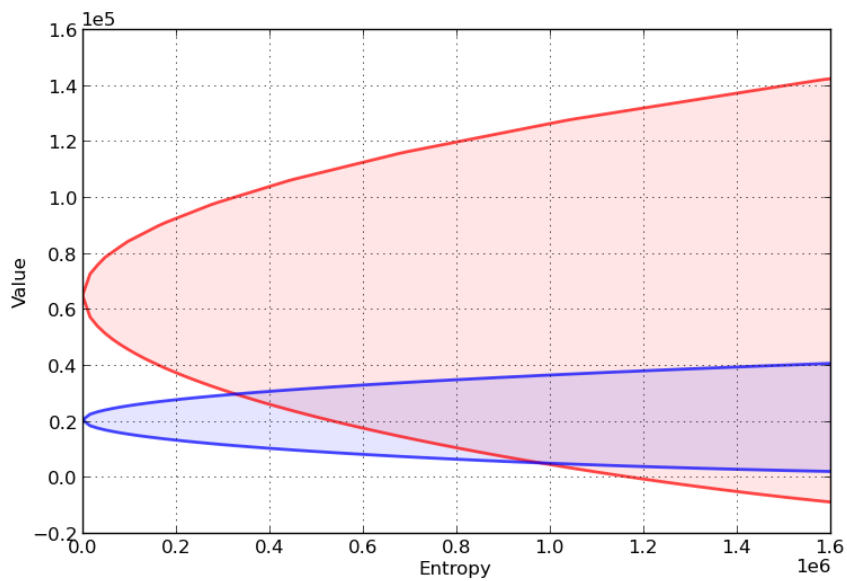
egrid = range( $\theta$ , emax, length = 100)
egrid_data = []
for data_pair in data_pairs
    for data in data_pair
        x, y = data[:, 2], data[:, 1]
        curve = LinearInterpolation(x, y, extrapolation_bc = Line())
        push!(egrid_data, curve.(egrid))
    end
end
plot(egrid, egrid_data, color=[:red :red :blue :blue])
plot!(egrid, egrid_data[1], fillrange=egrid_data[2],
      fillcolor=:red, fillalpha=0.1, color=:red, legend=:none)
plot!(egrid, egrid_data[3], fillrange=egrid_data[4],
      fillcolor=:blue, fillalpha=0.1, color=:blue, legend=:none)
plot!(xlabel="Entropy", ylabel="Value")

```

Out[3]:



Here's another such figure, with $\theta = 0.002$ instead of 0.02



Can you explain the different shape of the value-entropy correspondence for the robust policy?

9 Appendix

We sketch the proof only of the first claim in [this section](#), which is that, for any given θ , $K(\hat{F}, \theta) = \hat{K}$, where \hat{K} is as given in (8).

This is the content of the next lemma.

Lemma. If \hat{P} is the fixed point of the map $\mathcal{B} \circ \mathcal{D}$ and \hat{F} is the robust policy as given in (7), then

$$K(\hat{F}, \theta) = (\theta I - C' \hat{P} C)^{-1} C' \hat{P} (A - B \hat{F}) \quad (28)$$

Proof: As a first step, observe that when $F = \hat{F}$, the Bellman equation associated with the LQ problem (11) – (12) is

$$\tilde{P} = -R - \hat{F}' Q \hat{F} - \beta^2 (A - B \hat{F})' \tilde{P} C (\beta \theta I + \beta C' \tilde{P} C)^{-1} C' \tilde{P} (A - B \hat{F}) + \beta (A - B \hat{F})' \tilde{P} (A - B \hat{F}) \quad (29)$$

(revisit [this discussion](#) if you don't know where (29) comes from) and the optimal policy is

$$w_{t+1} = -\beta (\beta \theta I + \beta C' \tilde{P} C)^{-1} C' \tilde{P} (A - B \hat{F}) x_t$$

Suppose for a moment that $-\hat{P}$ solves the Bellman equation (29).

In this case the policy becomes

$$w_{t+1} = (\theta I - C' \hat{P} C)^{-1} C' \hat{P} (A - B \hat{F}) x_t$$

which is exactly the claim in (28).

Hence it remains only to show that $-\hat{P}$ solves (29), or, in other words,

$$\hat{P} = R + \hat{F}' Q \hat{F} + \beta (A - B \hat{F})' \hat{P} C (\theta I - C' \hat{P} C)^{-1} C' \hat{P} (A - B \hat{F}) + \beta (A - B \hat{F})' \hat{P} (A - B \hat{F})$$

Using the definition of \mathcal{D} , we can rewrite the right-hand side more simply as

$$R + \hat{F}' Q \hat{F} + \beta (A - B \hat{F})' \mathcal{D}(\hat{P})(A - B \hat{F})$$

Although it involves a substantial amount of algebra, it can be shown that the latter is just \hat{P} .

(Hint: Use the fact that $\hat{P} = \mathcal{B}(\mathcal{D}(\hat{P}))$)

References

- [1] L P Hansen and T J Sargent. *Robustness*. Princeton University Press, 2008.
- [2] Lars Peter Hansen and Thomas J Sargent. Wanting robustness in macroeconomics. *Manuscript, Department of Economics, Stanford University.*, 4, 2000.