Ad-click bidding

Roomsage.com

ADVERTISING AND MARKETING
MADE SIMPLE FOR ALL LODGING ESTABLISHMENTS
Revenue management

- Earning vs. learning
  - You want to maximize revenues, but you do not know how demand responds to price.

You earn the most with prices near the middle, but you do not learn anything.

You learn the most by sampling endpoints, but then you do not earn anything.
Learning problems

- Health sciences
  - Sequential design of experiments for drug discovery
  - Drug delivery – Optimizing the design of protective membranes to control drug release
  - Medical decision making – Optimal learning for medical treatments.
Drug discovery

Designing molecules

\[ Y = \theta_0 + \sum_{\text{sites } i} \sum_{\text{substituents } j} \theta_{ij} X_{ij} \]

» X and Y are sites where we can hang substituents to change the behavior of the molecule. We approximate the performance using a linear belief model:

» How to sequence experiments to learn the best molecule as quickly as possible?
Ride sharing

- Uber/Lyft
  - Provides real-time, on-demand transportation.
  - Drivers are encouraged to enter or leave the system using pricing signals and informational guidance.

- Decisions:
  - How to price to get the right balance of drivers relative to customers.
  - Real-time management of drivers.
  - Policies (rules for managing drivers, customers, …)
Matching buyers with sellers

- Now we have a logistic curve for each origin-destination pair \((i,j)\)
  
  \[
P^Y(p, a \mid \theta) = \frac{e^{\theta_{ij}^0 + \theta_{ij} p + \theta_{ij}^a a}}{1 + e^{\theta_{ij}^0 + \theta_{ij} p + \theta_{ij}^a a}}
  \]

- Number of offers for each \((i,j)\) pair is relatively small.

- Need to generalize the learning across hundreds to thousands of markets.
Emergency storm response

Hurricane Sandy
» Once in 100 years?
» Rare convergence of events
» But, meteorologists did an amazing job of forecasting the storm.

The power grid
» Loss of power creates cascading failures (lack of fuel, inability to pump water)
» How to plan?
» How to react?
Meeting variability with *portfolios* of generation with mixtures of *dispatchability*
Storage applications

How much energy to store in a battery to handle the volatility of wind and spot prices to meet demands?
**Modeling**

- Before we can *solve* complex problems, we have to know how to *think* about them.

![Image of wind turbines]

- The biggest challenge when making decisions under uncertainty is *modeling.*
Modeling

For deterministic problems, we speak the language of mathematical programming

» Linear programming:

\[ \min_x cx \]
\[ Ax = b \]
\[ x \geq 0 \]

» For time-staged problems

\[ \min_{x_0,...,x_T} \sum_{t=0}^{T} c_t x_t \]
\[ A_tx_t - B_{t-1}x_{t-1} = b_t \]
\[ D_t x_t \leq u_t \]
\[ x_t \geq 0 \]

Arguably Dantzig’s biggest contribution, more so than the simplex algorithm, was his articulation of optimization problems in a standard format, which has given algorithmic researchers a common language.
Outline

- Elements of a dynamic model
- Modeling uncertainty
- Designing policies
- The four classes of policies
- From deterministic to stochastic optimization
Outline

- Elements of a dynamic model
- Modeling uncertainty
- Designing policies
- The four classes of policies
- From deterministic to stochastic optimization
Modeling dynamic systems

All sequential decision problems can be modeled using five core components:

» State variables
  • What do we need to know at time t?

» Decision variables
  • What are our decisions?

» Exogenous information
  • What do we learn for the first time between t and t+1?

» Transition function
  • How do the state variables evolve over time?

» Objective function
  • What are our performance metrics?
Modeling dynamic problems

The state variable:

- Controls community
  \[ x_t = "\text{Information state}\" \]
- Operations research/MDP/Computer science
  \[ S_t = (R_t, I_t, B_t) = \text{System state}, \quad \text{where:} \]
  \[ R_t = \text{Resource state (physical state)} \]
  \[ \quad \text{Location/status of truck/train/plane} \]
  \[ \quad \text{Energy in storage} \]
  \[ I_t = \text{Information state} \]
  \[ \quad \text{Prices} \]
  \[ \quad \text{Weather} \]
  \[ B_t = \text{Belief state ("state of knowledge")} \]
  \[ \quad \text{Belief about performance of a drug or catalyst} \]
  \[ \quad \text{Belief about the status of equipment} \]
The state variable

- My definition of a state variable:

Definition 9.3.1  A state variable is:

a) Policy-dependent version  A function of history that, combined with the exogenous information (and a policy), is necessary and sufficient to compute the cost/contribution function, the decision function (the policy), and any information required to model the evolution of information needed in the cost/contribution and decision functions.

b) Optimization version  A function of history that is necessary and sufficient to compute the cost/contribution function, the constraints, and any information required to model the evolution of information needed in the cost/contribution function and the constraints.

- The first depends on a policy. The second depends only on the problem (and includes the constraints).

- Using either definition, all properly modeled problems are Markovian!
Modeling dynamic problems

- Decisions:
  - Markov decision processes/Computer science
  - \( a_t = \) Discrete action
  - Control theory
  - \( u_t = \) Low-dimensional continuous vector
  - Operations research
  - \( x_t = \) Usually a discrete or continuous but high-dimensional vector of decisions.

At this point, we do not specify how to make a decision. Instead, we define the function \( X^\pi(s) \) (or \( A^\pi(s) \) or \( U^\pi(s) \)), where \( \pi \) specifies the type of policy. "\( \pi \)" carries information about the type of function \( f \), and any tunable parameters \( \theta \in \Theta^f \).
The decision variables

- **Styles of decisions**
  - **Binary**
    \[ x \in X = \{0,1\} \]
  - **Finite**
    \[ x \in X = \{1,2,...,M\} \]
  - **Continuous scalar**
    \[ x \in X = [a,b] \]
  - **Continuous vector**
    \[ x = (x_1,...,x_K), \quad x_k \in \mathbb{R} \]
  - **Discrete vector**
    \[ x = (x_1,...,x_K), \quad x_k \in \mathbb{Z} \]
  - **Categorical**
    \[ x = (a_1,...,a_I), \quad a_i \text{ is a category (e.g. red/green/blue)} \]
Modeling dynamic problems

**Exogenous information:**

\[ W_t = \text{New information that first became known at time } t \]
\[ = (\hat{R}_t, \hat{D}_t, \hat{p}_t, \hat{E}_t) \]
\[ \hat{R}_t = \text{Equipment failures, delays, new arrivals} \]
\[ \text{New drivers being hired to the network} \]
\[ \hat{D}_t = \text{New customer demands} \]
\[ \hat{p}_t = \text{Changes in prices} \]
\[ \hat{E}_t = \text{Information about the environment (temperature, ...)} \]

*Note: Any variable indexed by } t \text{ is known at time } t. \text{ This convention, which is not standard in control theory, dramatically simplifies the modeling of information.}*

Below, we let \( \omega \) represent a sequence of actual observations \( W_1, W_2, .... \)
\( W_t(\omega) \) refers to a sample realization of the random variable \( W_t \).
Modeling dynamic problems

· The transition function

\[ S_{t+1} = S^M (S_t, x_t, W_{t+1}) \]
\[ R_{t+1} = R_t + x_t + \hat{R}_{t+1} \quad \text{Inventories} \]
\[ p_{t+1} = p_t + \hat{p}_{t+1} \quad \text{Spot prices} \]
\[ D_{t+1} = D_t + \hat{D}_{t+1} \quad \text{Market demands} \]

Also known as the:

· “System model”
· “State transition model”
· “Plant model”
· “Plant equation”
· “Transition law”

For many applications, these equations are unknown. This is known as “model-free” dynamic programming.
Modeling stochastic, dynamic problems

- Objective functions
  - Cumulative reward (“online learning”)
    \[
    \max_\pi \mathbb{E} \left\{ \sum_{t=0}^{T} C_t \left( S_t, X_t^\pi (S_t), W_{t+1} \right) \mid S_0 \right\}
    \]
    - Policies have to work well over time.
  - Final reward (“offline learning”)
    \[
    \max_\pi \mathbb{E} \left\{ F(x^{\pi,N}, \hat{W}) \mid S_0 \right\}
    \]
    - We only care about how well the final decision \( x^{\pi,N} \) works.
  - Risk
    \[
    \max_\pi \rho \left\{ C(S_0, X_0^\pi (S_0)), C(S_1, X_1^\pi (S_1)), \ldots, C(S_T, X_T^\pi (S_T)) \mid S_0 \right\}
    \]
Modeling stochastic, dynamic problems

- The complete model:
  - Objective function
    - Cumulative reward ("online learning")
      $$\max_{\pi} \mathbb{E}\left\{ \sum_{t=0}^{T} C_t \left( S_t, X_t^\pi (S_t), W_{t+1} \right) | S_0 \right\}$$
    - Final reward ("offline learning")
      $$\max_{\pi} \mathbb{E}\left\{ F(x^\pi, N, \hat{W}) | S_0 \right\}$$
    - Risk:
      $$\max_{\pi} \rho \left\{ C(S_0, X_0^\pi (S_0)), C(S_1, X_1^\pi (S_1)), ..., C(S_T, X_T^\pi (S_T)) | S_0 \right\}$$
  - Transition function:
    $$S_{t+1} = S^M \left( S_t, x_t, W_{t+1} (\omega) \right)$$
  - Exogenous information:
    $$(S_0, W_1, W_2, ..., W_T)$$
The modeling process

Modeling real applications

» I conduct a conversation with a domain expert to fill in the elements of a problem:

- State variables
- Decision variables
- New information
- Transition function
- Objective function

What we need to know (and only what we need)

What we control

What we didn’t know when we made our decision

How the state variables evolve

Performance metrics
Outline

- Elements of a dynamic model
- Modeling uncertainty
- Designing policies
- The four classes of policies
- From deterministic to stochastic optimization
Modeling uncertainty

Classes of uncertainty

- Observational uncertainty
- Prognostic uncertainty (forecasting)
- Experimental noise/variability
- Transitional uncertainty
- Inferential uncertainty
- Model uncertainty
- Systematic exogenous uncertainty
- Control/implementation uncertainty
- Algorithmic noise
- Goal uncertainty

Modeling uncertainty in the context of stochastic optimization is a relatively untapped area of research.
Outline

- Elements of a dynamic model
- Modeling uncertainty
- Designing policies
- The four classes of policies
- From deterministic to stochastic optimization
Designing policies

- We have to start by describing what we mean by a policy.
  » Definition:

  A policy is a mapping from a state to an action.

  ... any mapping.

- How do we search over an arbitrary space of policies?
# Designing policies

## “Policies” and the English language

<table>
<thead>
<tr>
<th>Behavior</th>
<th>Habit</th>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Belief</td>
<td>Laws/bylaws</td>
<td>Process</td>
</tr>
<tr>
<td>Bias</td>
<td>Manner</td>
<td>Protocols</td>
</tr>
<tr>
<td>Commandment</td>
<td>Method</td>
<td>Recipe</td>
</tr>
<tr>
<td>Conduct</td>
<td>Mode</td>
<td>Ritual</td>
</tr>
<tr>
<td>Convention</td>
<td>Mores</td>
<td>Rule</td>
</tr>
<tr>
<td>Culture</td>
<td>Patterns</td>
<td>Style</td>
</tr>
<tr>
<td>Customs</td>
<td>Plans</td>
<td>Technique</td>
</tr>
<tr>
<td>Dogma</td>
<td>Policies</td>
<td>Tenet</td>
</tr>
<tr>
<td>Etiquette</td>
<td>Practice</td>
<td>Tradition</td>
</tr>
<tr>
<td>Fashion</td>
<td>Prejudice</td>
<td>Way of life</td>
</tr>
<tr>
<td>Formula</td>
<td>Principle</td>
<td></td>
</tr>
</tbody>
</table>


Designing policies

Two fundamental strategies for finding policies:

1) Policy search – Search over a class of functions for making decisions to optimize some metric.

\[
\max_{\pi=(f \in F, \theta^f \in \Theta^f)} E \left\{ \sum_{t=0}^{T} C_t \left( S_t, X^\pi_t (S_t | \theta) \right) \mid S_0 \right\}
\]

2) Lookahead approximations – Approximate the impact of a decision now on the future.

\[
X^*_t (S_t) = \arg \max_{x_t} \left[ C(S_t, x_t) + E \left\{ \max_{\pi \in \Pi} \left\{ E \sum_{t'=t+1}^{T} C_{t'} (S_{t'}, X^\pi_{t'} (S_{t'})) \mid S_{t+1} \right\} \mid S_t, x_t \right\} \right]
\]
Designing policies

Policy search:

1a) Policy function approximations (PFAs) $x_t = X^{PFA} (S_t | \theta)$
   • Lookup tables
     – “when in this state, take this action”
   • Parametric functions
     – Order-up-to policies: if inventory is less than $s$, order up to $S$.
     – Affine policies - $x_t = X^{PFA} (S_t | \theta) = \sum_{f \in F} \theta_f \phi_f (S_t)$
     – Neural networks
   • Locally/semi/non parametric
     – Requires optimizing over local regions

1b) Cost function approximations (CFAs)
   • Optimizing a deterministic model modified to handle uncertainty
     (buffer stocks, schedule slack)
     \[
     X^{CFA} (S_t | \theta) = \arg \max_{x_t \in \mathcal{X}_t} \pi (S_t, x_t | \theta)
     \]
Designing policies

- Lookahead approximations – Approximate the impact of a decision now on the future:
  
  2a) Approximating the value of being in a state (VFA):

  \[ X_t^* (S_t) = \arg \max_{x_t} \left[ C(S_t, x_t) + E \left\{ V_{t+1}(S_{t+1}) \mid S_t, x_t \right\} \right] \]

  \[ X_t^{VFA} (S_t) = \arg \max_{x_t} \left[ C(S_t, x_t) + E \left\{ \bar{V}_{t+1}(S_{t+1}) \mid S_t, x_t \right\} \right] \]

  \[ = \arg \max_{x_t} \left[ C(S_t, x_t) + \bar{V}_t^x (S_t^x) \right] \]

  2b) Direct lookahead (DLA)

  Optimal policy:

  \[ X_t^* (S_t) = \arg \max_{x_t} \left[ C(S_t, x_t) + E \left\{ \max_{\pi \in \Pi} \left\{ E \sum_{t' = t+1}^T C(S_{t'}, X_t^{\pi} (S_t')) \mid S_{t+1} \right\} \mid S_t, x_t \right\} \right] \]

  Approximate policy – solve an approximate lookahead model:

  \[ X_t^{DLA} (S_t) = \arg \max_{x_t} \left[ C(S_t, x_t) + \tilde{E} \left\{ \max_{\tilde{\pi} \in \tilde{\Pi}} \left\{ \tilde{E} \sum_{t' = t+1}^{t+H} C(\tilde{S}_{t'}, \tilde{X}_t^{\tilde{\pi} (\tilde{S}_{t'})}) \mid \tilde{S}_{t,t+1} \right\} \mid \tilde{S}_t, x_t \right\} \right] \]
Four (meta)classes of policies

1) Policy function approximations (PFAs)
   - Lookup tables, rules, parametric/nonparametric functions

2) Cost function approximation (CFAs)
   \[ X^{CFA} (S_t | \theta) = \arg \max_{\pi \in \Pi_t} \bar{C}^{\pi} (S_t, x_t | \theta) \]

3) Policies based on value function approximations (VFAs)
   \[ X^{VFA} (S_t) = \arg \max_{x_t} \left( C(S_t, x_t) + \bar{V}_t^x \left( S_t^x(S_t, x_t) \right) \right) \]

4) Direct lookahead policies (DLAs)
   - Deterministic lookahead/rolling horizon proc./model predictive control
     \[ X^{LA-D} (S_t) = \arg \max_{\tilde{x}_t, \tilde{x}_{t+1}, \ldots, \tilde{x}_{t+H}} C(\tilde{S}_t, \tilde{x}_t) + \sum_{t' = t+1}^{T} C(\tilde{S}_{t'}, \tilde{x}_{t'}) \]
   - Chance constrained programming
     \[ P[A_t x_t \leq f(W)] \leq 1 - \delta \]
   - Stochastic lookahead/stochastic prog/Monte Carlo tree search
     \[ X^{LA-S} (S_t) = \arg \max_{\tilde{x}_t, \tilde{x}_{t+1}, \ldots, \tilde{x}_{t+T}} C(\tilde{S}_t, \tilde{x}_t) + \sum_{\tilde{\omega} \in \Omega_t} p(\tilde{\omega}) \sum_{t' = t+1}^{T} C(\tilde{S}_{t'}, \tilde{x}_{t'}(\tilde{\omega})) \]
   - “Robust optimization”
     \[ X^{LA-RO} (S_t) = \arg \max_{\tilde{x}_t, \ldots, \tilde{x}_{t+H}} \min_{w \in W_t(\theta)} C(\tilde{S}_t, \tilde{x}_t) + \sum_{t' = t+1}^{T} C(\tilde{S}_{t'}, (w), \tilde{x}_{t'}(w)) \]
Four (meta)classes of policies

1) **Policy function approximations (PFAs)**
   » Lookup tables, rules, parametric/nonparametric functions

2) **Cost function approximation (CFAs)**
   » \( X^{\text{CFA}}(S_t \mid \theta) = \arg \max_{x_t \in \bar{X}_t^\pi(\theta)} \bar{C}^\pi(S_t, x_t \mid \theta) \)

3) **Policies based on value function approximations (VFAs)**
   » \( X^{\text{VFA}}(S_t) = \arg \max_{x_t} \left( C(S_t, x_t) + \bar{V}_t^x(S_t^x(S_t, x_t)) \right) \)

4) **Direct lookahead policies (DLAs)**
   » *Deterministic lookahead/rolling horizon proc./model predictive control*
     \[ X^{\text{LA-D}}_t(S_t) = \arg \max_{\tilde{x}_t, \ldots, \tilde{x}_{t+H}} C(\tilde{S}_t, \tilde{x}_t) + \sum_{t'=t+1}^{T} C(\tilde{S}_{t'}, \tilde{x}_{t'}) \]
   » *Chance constrained programming*
     \[ P[A_t x_t \leq f(W)] \leq 1 - \delta \]
   » *Stochastic lookahead /stochastic prog/Monte Carlo tree search*
     \[ X^{\text{LA-S}}_t(S_t) = \arg \max_{\tilde{x}_t, \ldots, \tilde{x}_{t+T}} C(\tilde{S}_t, \tilde{x}_t) + \sum_{\tilde{\omega} \in \Omega_t} p(\tilde{\omega}) \sum_{t'=t+1}^{T} C(\tilde{S}_{t'}, \tilde{x}_{t'}(\tilde{\omega})) \]
   » *“Robust optimization”*
     \[ X^{\text{LA-RO}}_t(S_t) = \arg \max_{\tilde{x}_t, \ldots, \tilde{x}_{t+H}} \min_{w \in W_t(\theta)} C(\tilde{S}_t, \tilde{x}_t) + \sum_{t'=t+1}^{T} C(\tilde{S}_{t'}(w), \tilde{x}_{t'}(w)) \]
Four (meta)classes of policies

1) Policy function approximations (PFAs)
   » Lookup tables, rules, parametric/nonparametric functions

2) Cost function approximation (CFAs)
   » \( X^{CFA}(S_t | \theta) = \arg \max_{x_t \in X^i_t(\theta)} C^\pi(S_t, x_t | \theta) \)

3) Policies based on value function approximations (VFAs)
   » \( X^{VFA}(S_t) = \arg \max_{x_t} \left( C(S_t, x_t) + V^x_t(S^x_t(S_t, x_t)) \right) \)

4) Direct lookahead policies (DLAs)
   » Deterministic lookahead/rolling horizon proc./model predictive control
     \( X^{LA-D}(S_t) = \arg \max_{\tilde{x}_t, \ldots, \tilde{x}_{t+H}} C(\tilde{S}_t, \tilde{x}_t) + \sum_{t'=t+1} C(\tilde{S}_{t'}, \tilde{x}_{t'}) \)
   » Chance constrained programming
     \( P[A_t x_t \leq f(W)] \leq 1 - \delta \)
   » Stochastic lookahead/stochastic prog/Monte Carlo tree search
     \( X^{LA-S}(S_t) = \arg \max_{\tilde{x}_t, \ldots, \tilde{x}_{t+T}} C(\tilde{S}_t, \tilde{x}_t) + \sum_{t'=t+1} \sum_{\tilde{\omega} \in \Omega_t} p(\tilde{\omega}) C(\tilde{S}_{t'}, (\tilde{\omega}), \tilde{x}_{t'}(\tilde{\omega})) \)
   » “Robust optimization”
     \( X^{LA-RO}(S_t) = \arg \max_{\tilde{x}_n , \ldots, \tilde{x}_{t+H}} \min_{w \in W_t(\theta)} C(\tilde{S}_t, \tilde{x}_t) + \sum_{t'=t+1} C(\tilde{S}_{t'}, (w), \tilde{x}_{t'}(w)) \)
Learning problems

Classes of learning problems in stochastic optimization

1) Approximating the objective

\[ \tilde{F}(x|\theta) \approx \mathbb{E}F(x,W). \]

2) Designing a policy \( X^\pi(S|\theta) \).

3) A value function approximation

\[ \tilde{V}_t(S_t|\theta) \approx V_t(S_t). \]

4) Designing a cost function approximation:
   - The objective function \( \tilde{C}^\pi(S_t,x_t|\theta) \).
   - The constraints \( X^\pi(S_t|\theta) \)

5) Approximating the transition function

\[ \tilde{S}^M(S_t,x_t,W_{t+1}|\theta) \approx S^M(S_t,x_t,W_{t+1}) \]
Approximation strategies

- Lookup tables
  - Independent beliefs
  - Correlated beliefs

- Linear parametric models
  - Linear models
  - Sparse-linear
  - Tree regression

- Nonlinear parametric models
  - Logistic regression
  - Neural networks

- Nonparametric models
  - Gaussian process regression
  - Kernel regression
  - Support vector machines
  - Deep neural networks
Learning challenges

The learning challenge

From big (batch) data… … to recursive learning
Learning challenges

- Variable-dimensional learning
Outline

- Elements of a dynamic model
- Modeling uncertainty
- Designing policies
- The four classes of policies
- From deterministic to stochastic optimization
Outline

The four classes of policies

» Policy function approximations (PFAs)
» Cost function approximations (CFAs)
» Value function approximations (VFAs)
» Direct lookahead policies (DLAs)
» A hybrid lookahead/CFA
Outline

The four classes of policies

» Policy function approximations (PFAs)
» Cost function approximations (CFAs)
» Value function approximations (VFAs)
» Direct lookahead policies (DLAs)
» A hybrid lookahead/CFA
Policy function approximations

Battery arbitrage – When to charge, when to discharge, given volatile LMPs
Grid operators require that batteries bid charge and discharge prices, an hour in advance.

We have to search for the best values for the policy parameters $\theta^\text{Charge}$ and $\theta^\text{Discharge}$.
Policy function approximations

- Our policy function might be the parametric model (this is nonlinear in the parameters):

\[
X^\pi (S_t \mid \theta) = \begin{cases} 
+1 & \text{if } p_t < \theta^{\text{charge}} \\
0 & \theta^{\text{charge}} < p_t < \theta^{\text{discharge}} \\
-1 & \text{if } p_t > \theta^{\text{charge}} 
\end{cases}
\]

Energy in storage:

Price of electricity:
Policy function approximations

Finding the best policy

» We need to maximize

$$\max_{\theta} F(\theta) = \mathbb{E} \sum_{t=0}^{T} \gamma^t C\left(S_t, X_t^{\pi}(S_t \mid \theta)\right)$$

» We cannot compute the expectation, so we run simulations:
Outline

The four classes of policies

» Policy function approximations (PFAs)
» Cost function approximations (CFAs)
» Value function approximations (VFAs)
» Direct lookahead policies (DLAs)
» A hybrid lookahead/CFA
Cost function approximations

- **Lookup table**
  - We can organize potential catalysts into groups.
  - Scientists using domain knowledge can estimate correlations in experiments between similar catalysts.

<table>
<thead>
<tr>
<th></th>
<th>1.4 nm Fe</th>
<th>1 nm Fe</th>
<th>2 nm Fe</th>
<th>10 nm ALD Al2O3+1.2 nm IBS Fe</th>
<th>2 nm Ni</th>
<th>Ni 0.6 nm</th>
<th>10 nm ALD Al2O3+1 nm Ni</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.4 nm Fe</td>
<td>1</td>
<td>0.7</td>
<td>0.7</td>
<td>0.6</td>
<td>0.4</td>
<td>0.4</td>
<td>0.2</td>
</tr>
<tr>
<td>1 nm Fe</td>
<td>0.7</td>
<td>1</td>
<td>0.7</td>
<td>0.6</td>
<td>0.4</td>
<td>0.4</td>
<td>0.2</td>
</tr>
<tr>
<td>2 nm Fe</td>
<td>0.7</td>
<td>0.7</td>
<td>1</td>
<td>0.6</td>
<td>0.4</td>
<td>0.4</td>
<td>0.2</td>
</tr>
<tr>
<td>10 nm ALD Al2O3+1.2 nm IBS Fe</td>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
<td>1</td>
<td>1</td>
<td>0.3</td>
<td>0</td>
</tr>
<tr>
<td>2 nm Ni</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>1</td>
<td>1</td>
<td>0.7</td>
<td>0.6</td>
</tr>
<tr>
<td>Ni 0.6 nm</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.3</td>
<td>0.7</td>
<td>1</td>
<td>0.6</td>
</tr>
<tr>
<td>10 nm ALD Al2O3+1 nm Ni</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0</td>
<td>0.6</td>
<td>0.6</td>
<td>1</td>
</tr>
</tbody>
</table>
Cost function approximations

- Correlated beliefs: Testing one material teaches us about other materials
Cost function approximations

Cost function approximations (CFA)

» Upper confidence bounding

\[ X^{UCB} (S^n | \theta^{UCB}) = \arg \max_x \left( \bar{\mu}_x^n + \theta^{UCB} \sqrt{\frac{\log n}{N_x^n}} \right) \]

» Interval estimation

\[ X^{IE} (S^n | \theta^{IE}) = \arg \max_x \left( \bar{\mu}_x^n + \theta^{IE} \bar{\sigma}_x^n \right) \]

» Boltzmann exploration ("soft max")

- Choose \( x \) with probability:

\[ P^n_x (\theta) = \frac{e^{\theta \bar{\mu}_x^n}}{\sum_{x'} e^{\theta \bar{\mu}_{x'}^n}} \]

\[ X^{Boltz} (S^n|\theta) = \arg \max_{x} \{ x| P^n_x (\theta) \leq U \} \]
Cost function approximations

- Picking $\theta^{IE} = 0$ means we are evaluating each choice at the mean.
Cost function approximations

Picking $\theta^{IE} = 2$ means we are evaluating each choice at the 95th percentile.
Cost function approximations

- Optimizing the policy
  - We optimize $\theta^{IE}$ to maximize:

    $$\max_{\theta^{IE}} F(\theta^{IE}) = EF(x^{\pi,N}, W)$$

    where

    $$x^n = X^{IE}(S^n | \theta^{IE}) = \arg\max_x \left( \mu^n_x + \theta^{IE} \sigma^n_x \right) \quad S^n = (\mu^n_x, \sigma^n_x)$$

- Notes:
  - This can handle any belief model, including correlated beliefs, nonlinear belief models.
  - All we require is that we be able to simulate a policy.
Cost function approximations

Other applications

» Airlines optimizing schedules with schedule slack to handle weather uncertainty.

» Manufacturers using buffer stocks to hedge against production delays and quality problems.

» Grid operators scheduling extra generation capacity in case of outages.

» Adding time to a trip planned by Google maps to account for uncertain congestion.
Outline

The four classes of policies

» Policy function approximations (PFAs)
» Cost function approximations (CFAs)
» Value function approximations (VFAs)
» Direct lookahead policies (DLAs)
» A hybrid lookahead/CFA
Value function approximations

- Q-learning (for discrete actions)

\[
\hat{q}^n(s^n, a^n) = r(s^n, a^n) + g \max_{a'} \overline{Q}^{n-1}(s', a')
\]

\[
\overline{Q}^n(s^n, a^n) = (1 - a_{n-1})\overline{Q}^{n-1}(s^n, a^n) + a_{n-1}\hat{q}^n(s^n, a^n)
\]

» But what if the action \( a \) is a vector?
Blood management

- Managing blood inventories

![Blood Typing Diagram](image-url)
Blood management

Managing blood inventories over time

Week 0

\[ S_0 \]

\[ x_0 \]

\[ \hat{R}_1, \hat{D}_1 \]

\[ t=0 \]

Week 1

\[ S_1 \]

\[ x_1 \]

\[ \hat{R}_2, \hat{D}_2 \]

\[ t=1 \]

Week 2

\[ S_2 \]

\[ x_2 \]

\[ \hat{R}_3, \hat{D}_3 \]

\[ t=2 \]

Week 3

\[ S_3 \]

\[ x_3 \]

\[ \hat{R}_3, \hat{D}_3 \]

\[ t=3 \]
\[ S_t = \begin{pmatrix} R_t \\ \hat{D}_t \end{pmatrix}, \quad R_t^x \]

\[ R_{t,(AB+,0)} \rightarrow AB+,0 \]

\[ R_{t,(AB+,1)} \rightarrow AB+,1 \]

\[ R_{t,(AB+,2)} \rightarrow AB+,2 \]

\[ R_{t,(O-,0)} \rightarrow O-,0 \]

\[ R_{t,(O-,1)} \rightarrow O-,1 \]

\[ R_{t,(O-,2)} \rightarrow O-,2 \]

\[ \vdots \]

\[ \vdots \]

\[ \hat{D}_{t,AB+} \]

\[ \hat{D}_{t,AB-} \]

\[ \hat{D}_{t,A+} \]

\[ \hat{D}_{t,A-} \]

\[ \hat{D}_{t,B+} \]

\[ \hat{D}_{t,B-} \]

\[ \hat{D}_{t,O+} \]

\[ \hat{D}_{t,O-} \]

\[ \text{Satisfy a demand} \]

\[ \text{Hold} \]
Solve this as a linear program.
Dual variables give value additional unit of blood.
Updating the value function approximation

- Estimate the gradient at $R^n_t$
Updating the value function approximation

- Update the value function at $R_{t-1}^{x,n}$
Updating the value function approximation

- Update the value function at $R_{t-1}^{x,n}$
Updating the value function approximation

- Update the value function at $R_{t-1}^{x,n}$
Exploiting concavity

- Derivatives are used to estimate a piecewise linear approximation
Iterative learning
Iterative learning
Iterative learning
Iterative learning
Approximate dynamic programming

... a typical performance graph.
Outline

The four classes of policies

- Policy function approximations (PFAs)
- Cost function approximations (CFAs)
- Value function approximations (VFAs)
- Direct lookahead policies (DLAs)
- A hybrid lookahead/CFA
Lookahead policies

- Planning your next chess move:

» You put your finger on the piece while you think about moves into the future. This is a lookahead policy, illustrated for a problem with discrete actions.
Lookahead policies

- Decision trees:
Lookahead policies

Modeling lookahead policies

» Lookahead policies solve a lookahead model, which is an approximation of the future.

» It is important to understand the difference between the:

  • Base model – this is the model we are trying to solve by finding the best policy. This is usually some form of simulator.

  • The lookahead model, which is our approximation of the future to help us make better decisions now.

» The base model is typically a simulator, or it might be the real world.
Emergency storm response
Emergency storm response
Emergency storm response
Lookahead policies

- Monte Carlo tree search:

- Action selection
- Sampling

- Expansion

- Simulation

- Backpropagation

Outline

The four classes of policies

» Policy function approximations (PFAs)
» Cost function approximations (CFAs)
» Value function approximations (VFAs)
» Direct lookahead policies (DLAs)
» A hybrid lookahead/CFA
Parametric cost function approximation

- An energy storage problem:

The state of the system can be represented by the following five-dimensional vector,

$$S_t = (R_t, E_t, P_t, D_t, G_t)$$

where

- $R_t \in [0, R_{\text{max}}]$ is the level of energy in storage at time $t$
- $E_t$ is the amount of energy available from wind
- $P_t$ is the spot price of electricity
- $D_t$ is the power demand
- $G_t$ is the energy available from the grid
Parametric cost function approximation

- Forecasts evolve over time as new information arrives:

Rolling forecasts, updated each hour.

Forecast made at midnight:

Actual
Parametric cost function approximation

Benchmark policy – Deterministic lookahead

$$X_{t}^{D-LA}(S_{t}) = \arg\min_{x_{t},(\tilde{x}_{tt'}, t'=t+1, \ldots, t+H)} \left( C(S_{t}, x_{t}) + \sum_{t'=t+1}^{t+H} \tilde{c}_{tt'} \tilde{x}_{tt'} \right)$$

$$\tilde{x}_{tt'}^{wd} + \beta \tilde{x}_{tt'}^{rd} + \tilde{x}_{tt'}^{gd} \leq f_{tt'}^{D}$$

$$\tilde{x}_{tt'}^{rd} + \tilde{x}_{tt'}^{rg} \leq \tilde{R}_{tt'}$$

$$\tilde{x}_{tt'}^{wr} + \tilde{x}_{tt'}^{gr} \leq R_{max} - \tilde{R}_{tt'}$$

$$\tilde{x}_{tt'}^{wr} + \tilde{x}_{tt'}^{wd} \leq f_{tt'}^{E}$$

$$\tilde{x}_{tt'}^{wr} + \tilde{x}_{tt'}^{gr} \leq \gamma^{charge}$$

$$\tilde{x}_{tt'}^{rd} + \tilde{x}_{tt'}^{rg} \leq \gamma^{discharge}$$
Parametric cost function approximation

Parametric cost function approximations

» Replace the constraint

\[ \tilde{x}_{tt'}^{wr} + \tilde{x}_{tt'}^{wd} \leq f_{tt'}^E \]

with:

» Lookup table modified forecasts (one adjustment term for each time \( \tau = t' - t \) in the future):

\[ x_{tt'}^{wr} + x_{tt'}^{wd} \leq \theta_{t'-t} f_{tt'}^E \]

» Exponential function for adjustments (just two parameters)

\[ x_{tt'}^{wr} + x_{tt'}^{wd} \leq \theta_1 e^{\theta_2 (t' - t)} f_{tt'}^E \]

» Constant adjustment (one parameter)

\[ x_{tt'}^{wr} + x_{tt'}^{wd} \leq \theta_3 f_{tt'}^E \]
Parametric cost function approximation

- Improvement over deterministic benchmark:
  - Lookup table
  - Exponential
  - Constant
Consider a basic energy storage problem:

We are going to show that with minor variations in the characteristics of this problem, we can make each class of policy work best.
An energy storage problem

We can create distinct flavors of this problem:

» Problem class 1 – Best for PFAs
  • Highly stochastic (heavy tailed) electricity prices
  • Stationary data

» Problem class 2 – Best for CFAs
  • Stochastic prices and wind (but not heavy tailed)
  • Stationary data

» Problem class 3 - Best for VFAs
  • Stochastic wind and prices (but not too random)
  • Time varying loads, but inaccurate wind forecasts

» Problem class 4 – Best for deterministic lookaheads
  • Relatively low noise problem with accurate forecasts

» Problem class 5 – A hybrid policy worked best here
  • Stochastic prices and wind, nonstationary data, noisy forecasts.
An energy storage problem

The policies

- The PFA:
  - Charge battery when price is below $p_1$
  - Discharge when price is above $p_2$

- The CFA
  - Optimize over a horizon $H$; maintain upper and lower bounds ($u$, $l$) for every time period except the first (note that this is a hybrid with a lookahead).

- The VFA
  - Piecewise linear, concave value function in terms of energy, indexed by time.

- The lookahead (deterministic)
  - Optimize over a horizon $H$ (only tunable parameter) using forecasts of demand, prices and wind energy

- The lookahead CFA
  - Use a lookahead policy (deterministic), but with a tunable parameter that improves robustness.
An energy storage problem

- Each policy is best on certain problems
  - Results are percent of posterior optimal solution

<table>
<thead>
<tr>
<th>Problem:</th>
<th>Problem description</th>
<th>PFA</th>
<th>CFA Error correction</th>
<th>VFA</th>
<th>Determ. Lookahead</th>
<th>CFA Lookahead</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A stationary problem with heavy-tailed prices, relatively low noise, moderately accurate forecasts.</td>
<td>0.959</td>
<td>0.839</td>
<td>0.936</td>
<td>0.887</td>
<td>0.887</td>
</tr>
<tr>
<td>B</td>
<td>A time-dependent problem with daily load patterns, no seasonalities in energy and price, relatively low noise, less accurate forecasts.</td>
<td>0.714</td>
<td>0.752</td>
<td>0.712</td>
<td>0.746</td>
<td>0.746</td>
</tr>
<tr>
<td>C</td>
<td>A time-dependent problem with daily load, energy and price patterns, relatively high noise, forecast errors increase over horizon.</td>
<td>0.865</td>
<td>0.590</td>
<td>0.914</td>
<td>0.886</td>
<td>0.886</td>
</tr>
<tr>
<td>D</td>
<td>A time-dependent problem, relatively low noise, very accurate forecasts.</td>
<td>0.962</td>
<td>0.749</td>
<td>0.971</td>
<td>0.997</td>
<td>0.997</td>
</tr>
<tr>
<td>E</td>
<td>Same as (C), but the forecast errors are stationary over the planning horizon.</td>
<td>0.865</td>
<td>0.590</td>
<td>0.914</td>
<td>0.922</td>
<td>0.934</td>
</tr>
</tbody>
</table>

- ... any policy might be best depending on the data.

Joint research with Prof. Stephan Meisel, University of Muenster, Germany.
Outline

- Elements of a dynamic model
- Modeling uncertainty
- Designing policies
- The four classes of policies
- From deterministic to stochastic optimization
From deterministic to stochastic

Imagine that you would like to solve the time-dependent linear program:

\[
\min_{x_0, \ldots, x_T} \sum_{t=0}^{T} c_t x_t
\]

subject to

\[
A_0 x_0 = b_0
\]
\[
A_t x_t - B_{t-1} x_{t-1} = b_t, \quad t \geq 1.
\]

We can convert this to a proper stochastic model by replacing \( x_t \) with \( X_t^\pi (S_t) \) and taking an expectation:

\[
\min_{\pi} \mathbb{E} \sum_{t=0}^{T} c_t X_t^\pi (S_t)
\]

The policy \( X_t^\pi (S_t) \) has to satisfy \( A_t x_t = R_t \) with transition function:

\[
S_{t+1} = S^M (S_t, x_t, W_{t+1})
\]
Modeling

- Deterministic
  - Objective function
    \[ \min_{x_0,\ldots,x_T} \sum_{t=0}^{T} c_t x_t \]
  - Decision variables:
    \( (x_0,\ldots,x_T) \)
  - Constraints:
    \[ \begin{align*}
    \text{at time } t & \quad A_t x_t = R_t \bigg\} X_t \\
    x_t & \geq 0
    \end{align*} \]
    - Transition function
      \[ R_{t+1} = b_{t+1} + B_t x_t \]

- Stochastic
  - Objective function
    \[ \max_{\pi} E^\pi \left\{ \sum_{t=0}^{T} C_t \left( S_t, X_t^\pi(S_t), W_{t+1} \right) \bigg| S_0 \right\} \]
  - Policy
    \[ X^\pi : S \mapsto X \]
  - Constraints at time \( t \)
    \[ x_t = X_t^\pi(S_t) \in X_t \]
  - Transition function
    \[ S_{t+1} = S^M \left( S_t, x_t, W_{t+1} \right) \]
  - Exogenous information
    \( (S_0, W_1, W_2,\ldots,W_T) \)
From deterministic to stochastic

**Stochastic problems**
- Modeling is the most important, and hardest, aspect of stochastic optimization.
- Searching for policies is important, but less critical.
- Modeling uncertainty is often overlooked, but is of central importance.
- Evaluating a policy is important, and difficult. In a simulator? In the field?

**Deterministic problems**
- Modeling is important, but not central.
- Algorithms are the most important, and hardest part.
- Huh?
- Just add up the costs!!
Modeling stochastic, dynamic problems

- The universal objective function

\[
\max_{\pi} E^\pi \left\{ \sum_{t=0}^{T} C_t \left( S_t, X_t^\pi (S_t), W_{t+1} \right) \mid S_0 \right\}
\]

with \( S_{t+1} = S^M \left( S_t, x_t, W_{t+1}(\omega) \right) \)

- You next need to develop a stochastic model:
  - Model uncertainty about parameters in \( S_0 \)
  - Model the stochastic process \( W_1, W_2, ..., W_N \) (for training)
  - Model the random variable \( \widehat{W} \) (for testing, if necessary)

- Then search for policies:
  - Policy search:
    - PFAs, CFAs
  - Lookahead policies:
    - VFAs, DLAs
Thank you!

For more information, go to

http://www.castlelab.princeton.edu/jungle/

Scroll to “Educational materials”
SEQUENTIAL DECISION ANALYTICS AND MODELING:
Modeling exercises with python

Warren B. Powell

September 10, 2018
1 Modeling sequential decision problems 1
   1.1 The modeling process 1
   1.2 Modeling time vs. iterations 3
   1.3 The mathematical modeling framework 3

2 An asset selling problem 9
   2.1 Narrative 10
   2.2 Basic model 10
   2.3 Modeling uncertainty 13
   2.4 Designing policies 15
   2.5 Policy evaluation 15
   2.6 Extensions 17
      2.6.1 Time series price processes 17
      2.6.2 Basket of assets 18

3 Adaptive market planning 19
   3.1 Narrative 19
   3.2 Basic model 21
   3.3 Designing policies 23
   3.4 Policy evaluation 24
4 Learning the best diabetes medication
  4.1 Narrative
  4.2 Basic model
  4.3 Modeling uncertainty
  4.4 Designing policies
  4.5 Policy evaluation
  4.6 Extensions
      Problems

5 Stochastic shortest path problems - Static
  5.1 Narrative
  5.2 Basic model
  5.3 Modeling uncertainty
  5.4 Designing policies
  5.5 Policy evaluation
  5.6 Extensions
      5.6.1 Stochastic shortest path - Adaptive

6 Stochastic shortest path problems - Dynamic
  6.1 Narrative
  6.2 Basic model
  6.3 Modeling uncertainty
  6.4 Designing policies
      Problems

7 Ad-click optimization
  7.1 Narrative
  7.2 Basic model
  7.3 Modeling uncertainty
  7.4 Designing policies
      7.4.1 Pure exploitation
      7.4.2 An excitation policy
      7.4.3 A value of information policy
Theory

Computation

Modeling

Applications