Learning closed-loop policies from batch data: Model-Selection in RL

Csaba Szepesvári
University of Alberta
Department of Computing Science

Joint work with:
Amir massoud Farahmand
Batch RL:
The Questions..

• How good policies can be learnt? Universal consistency? Does u.c. make sense?

• Is there any difference to supervised learning? What?

• What methods to use?

• Can we use model selection? How?

• Can we predict the performance of a learnt policy? How? What are the limits?
Today: Model Selection
Today: Model Selection

• The problem of adaptation
Today: Model Selection

- The problem of adaptation
- Methods of supervised learning
Today: Model Selection

- The problem of adaptation
- Methods of supervised learning
- Adaptation to RL: Value function estimation
Today: Model Selection

• The problem of adaptation
• Methods of supervised learning
• Adaptation to RL: Value function estimation
• Finding a good policy: A model based approach
Today: Model Selection

- The problem of adaptation
- Methods of supervised learning
- Adaptation to RL: Value function estimation
- Finding a good policy: A model based approach
- Discussion
Contents

• The problem of adaptation
• Methods of supervised learning
• Adaptation to RL: Value function estimation
• Finding a good policy: A model based approach
• Discussion
Does your Algorithm have parameters?
Does your Algorithm have parameters?
The Tuning Business
The Tuning Business
The Tuning Business

• Approach #1:
The Tuning Business

• Approach #1:
  • Find the parameters that maximize performance on a set of benchmarks
The Tuning Business

• **Approach #1:**
  • Find the parameters that maximize performance on a set of benchmarks
  • One size fits all?
The Tuning Business

• Approach #1:
  • Find the parameters that maximize performance on a set of benchmarks
  • One size fits all?

• Approach #2:
The Tuning Business

• Approach #1:
  • Find the parameters that maximize performance on a set of benchmarks
  • One size fits all?

• Approach #2:
  • Find the parameters that work the best on the actual problem
The Tuning Business

• **Approach #1:**
  • Find the parameters that maximize performance on a set of benchmarks
  • One size fits all?

• **Approach #2:**
  • Find the parameters that work the best on the actual problem
  • When should we be satisfied?
The Goal

\[
\limsup_{n \to \infty} \frac{L_n^A}{\inf_{k \geq 1} L_n^k} \leq C
\]
The Goal

\[ \limsup_{n \to \infty} \frac{L_n^A}{\inf_{k \geq 1} L_n^k} \leq C \]
Contents

• The problem of adaptation

• Methods of supervised learning

• Adaptation to RL: Value function estimation

• Finding a good policy: A model based approach

• Discussion
How to achieve adaptivity in regression?
How to achieve adaptivity in regression?

• “Parameter”: Controls the regressor
How to achieve adaptivity in regression?

- “Parameter”: Controls the regressor
- Method:
How to achieve adaptivity in regression?

- “Parameter”: Controls the regressor
- Method:
  - Generate
How to achieve adaptivity in regression?

• “Parameter”: Controls the regressor

• Method:
  • Generate
  • Test
How to achieve adaptivity in regression?

• “Parameter”: Controls the regressor

• Method:
  • Generate
  • Test
  • Select
How to achieve adaptivity in regression?

- “Parameter”: Controls the regressor
- Method:
  - Generate
  - Test
  - Select

Train  Hold-out
How to achieve adaptivity in regression?

- “Parameter”: Controls the regressor
- Method:
  - Generate
  - Test
  - Select

[Diagram showing Train and Hold-out sections]
Refined method

• “Parameter”: Controls the regressor

• Method:
  • Generate
  • Test
  • Bias
  • Select

Train Hold-out

Sunday, December 19, 2010
Key Ingredients
Key Ingredients

- Ability to estimate the performance of a predictor
Key Ingredients

• Ability to estimate the performance of a predictor

• i.i.d. data
Key Ingredients

• Ability to estimate the performance of a predictor

• i.i.d. data

• The hold-out data is independent of the training data => no bias
Firming up..
Firming up..

based on: Barron ’91, Lugosi and Wegkamp ’04
Firming up..

$0 < a < 1$

based on: Barron '91, Lugosi and Wegkamp '04
0 < a < 1

\[ P((1 - a)R_k < L_k - \varepsilon) \leq c_1 \exp(-c_2\varepsilon) \]

based on: Barron ’91, Lugosi and Wegkamp ’04
Firming up..

\[ 0 < a < 1 \]

\[ \mathbb{P}\left( (1 - a)R_k < L_k - \varepsilon \right) \leq c_1 \exp(-c_2 \varepsilon) \]

\[ \mathbb{P}\left( \frac{1}{1 + a} R_k > \mathbb{E}[R_k] + \varepsilon \right) \leq c_3 \exp(-c_4 \varepsilon) \]

based on: Barron ’91, Lugosi and Wegkamp ’04
Firming up..

\[ 0 < a < 1 \]

\[
P(\,(1 - a)R_k < L_k - \varepsilon) \leq c_1 \exp(-c_2 \varepsilon)\]

\[
P\left(\frac{1}{1 + a} R_k > \mathbb{E}[R_k] + \varepsilon\right) \leq c_3 \exp(-c_4 \varepsilon)\]

\[
\hat{k} = \arg\min_{k \geq 1} [R_k + C_k]
\]

based on: Barron ’91, Lugosi and Wegkamp ’04
Firming up..

\[ 0 < a < 1 \]

\[ \mathbb{P}((1 - a)R_k < L_k - \varepsilon) \leq c_1 \exp(-c_2 \varepsilon) \]

\[ \mathbb{P} \left( \frac{1}{1 + a} R_k > \mathbb{E}[R_k] + \varepsilon \right) \leq c_3 \exp(-c_4 \varepsilon) \]

\[ \hat{k} = \arg\min_{k \geq 1} [R_k + C_k] \]

\[ c_5 = \sum_{k \geq 1} \exp(-c_2 (1 - a)C_k) < \infty \]

based on: Barron ’91, Lugosi and Wegkamp ’04
0 < a < 1

\[ \mathbb{P}((1 - a)R_k < L_k - \varepsilon) \leq c_1 \exp(-c_2\varepsilon) \]

\[ \mathbb{P}\left(\frac{1}{1 + a} R_k > \mathbb{E}[R_k] + \varepsilon\right) \leq c_3 \exp(-c_4\varepsilon) \]

\[ \hat{k} = \arg\min_{k \geq 1} [R_k + C_k] \]

\[ c_5 = \sum_{k \geq 1} \exp(-c_2(1 - a)C_k) < \infty \]

\[ c_6 = \sum_{k \geq 1} \exp\left( -c_4 \frac{1 + 2a}{1 + a} C_k \right) < \infty \]

based on: Barron ’91, Lugosi and Wegkamp ’04
Firming up..

\[ \mathbb{P}\left((1 - a)R_k < L_k - \varepsilon\right) \leq c_1 \exp(-c_2 \varepsilon) \]

\[ \mathbb{P}\left(\frac{1}{1 + a}R_k > \mathbb{E}[R_k] + \varepsilon\right) \leq c_3 \exp(-c_4 \varepsilon) \]

\[ 0 < a < 1 \]

\[ \hat{k} = \arg\min_{k \geq 1}[R_k + C_k] \]

\[ c_5 = \sum_{k \geq 1} \exp(-c_2(1 - a)C_k) < \infty \]

\[ c_6 = \sum_{k \geq 1} \exp\left(-c_4 \frac{1 + 2a}{1 + a} C_k\right) < \infty \]
Firming up..

\[ P((1 - a)R_k < L_k - \epsilon) \leq c_1 \exp(-c_2\epsilon) \]
\[ P\left(\frac{1}{1 + a}R_k > \mathbb{E}[R_k] + \epsilon\right) \leq c_3 \exp(-c_4\epsilon) \]
0 < a < 1
\[ \hat{k} = \arg\min_{k \geq 1} [R_k + C_k] \]
\[ c_5 = \sum_{k \geq 1} \exp(-c_2(1 - a)C_k) < \infty \]
\[ c_6 = \sum_{k \geq 1} \exp\left(-c_4 \frac{1 + 2a}{1 + a} C_k\right) < \infty \]

Then, w.p. \(1 - \delta\),

\[ L_{\hat{k}} < (1 - a^2) \min_{k=1,2,...} \{\mathbb{E}[R_k] + 2C_k\} + c \ln(1/\delta) \]
Firming up..

Then, w.p. $1 - \delta$,

$$L_{\hat{k}} < (1 - a^2) \min_{k=1,2,...} \{\mathbb{E}[R_k] + 2C_k\} + c \ln(1/\delta)$$

$$B_\alpha = \left\{ k' : L_{k'} > (1 - a^2) \min_{k=1,2,...} \{\mathbb{E}[R_k] + 2C_k\} + \alpha \right\}$$

\[ \mathbb{P}((1 - a)R_k < L_k - \varepsilon) \leq c_1 \exp(-c_2\varepsilon) \]
\[ \mathbb{P}\left(\frac{1}{1 + a}R_k > \mathbb{E}[R_k] + \varepsilon\right) \leq c_3 \exp(-c_4\varepsilon) \]

$0 < a < 1$

$$\hat{k} = \arg\min_{k \geq 1} [R_k + C_k]$$

$$c_5 = \sum_{k \geq 1} \exp(-c_2(1 - a)C_k) < \infty$$

$$c_6 = \sum_{k \geq 1} \exp\left(-c_4\frac{1 + 2a}{1 + a}C_k\right) < \infty$$
Then, w.p. $1 - \delta$,  

$$L_{\hat{k}} < (1 - a^2) \min_{k=1,2,...} \{ \mathbb{E}[R_k] + 2C_k \} + c \ln(1/\delta)$$  

$$B_\alpha = \left\{ k' : L_{k'} > (1 - a^2) \min_{k=1,2,...} \{ \mathbb{E}[R_k] + 2C_k \} + \alpha \right\}$$  

$$\mathbb{P} \left( \hat{k} \in B_\alpha \right) \leq c' \exp \left( - \frac{c'' \alpha}{1 - a^2} \right)$$
Contents

• The problem of adaptation
• Methods of supervised learning
• Adaptation to RL: Value function estimation
• Finding a good policy: A model based approach
• Discussion
The Problem
The Problem

• Given a batch of data in the form of a set of transitions
The Problem

• Given a batch of data in the form of a set of transitions

• Goal:
The Problem

- Given a batch of data in the form of a set of transitions

- Goal:
  - Find an action-value function $Q$ with a small Bellman error, $T^*Q - Q$
The Problem

- Given a batch of data in the form of a set of transitions

- **Goal:**
  - Find an action-value function $Q$ with a small Bellman error, $T*Q-Q$
  
- Can we use the generate+test pattern to pick the best parameters?
Issues to deal with
Issues to deal with

• How to measure $T^*Q-Q$?
Issues to deal with

• How to measure $T^*Q-Q$?
• Dependence:
Issues to deal with

- How to measure $T^*Q - Q$?
- Dependence:
  - Data has some good forgetting properties
Issues to deal with

• How to measure $T^*Q-Q$?

• Dependence:
  • Data has some good forgetting properties
  • Assume that forgetting time is known
Issues to deal with

• How to measure $T^*Q-Q$?

• Dependence:
  • Data has some good forgetting properties
  • Assume that forgetting time is known
  • Samson (2000): Bernstein inequality
The Proposed Algorithm: **BerMin**

**Algorithm 1** **BERMIN**(\(\{Q_k\}_{k=1,2,..., D_n}, \text{REGRESS}(\cdot), \delta, a, b\))

1. Split \(D_n\) into two disjoint parts of the same size: \(D_n = D_1' \cup D_2''\).
2. for \(k = 1, 2, \ldots\) do
3. \((\tilde{Q}_k, \bar{b}_k) \leftarrow \text{REGRESS}(D_{n,k}, \frac{2}{\pi^2} \frac{\delta}{k^2})\)
4. \(e_k \leftarrow \frac{1}{|D_n''|} \sum_{(X,A) \in D_n''} (Q_k(X,A) - \tilde{Q}_k(X,A))^2\)
5. \(R_{k}^{RL} \leftarrow \frac{1}{(1-a)^2} e_k + \bar{b}_k\)
6. end for
7. \(\hat{k} \leftarrow \arg\min_{k=1,2,...} \left[ R_{k}^{RL} + b \frac{\ln(k)}{a(1-a)^2 n} \right]\)
8. return \(\hat{k}\)

\(D_{n,k}' = \left\{ (X_1, A_1), (\hat{T}^* Q_k)(X_1, A_1) \right\}, \ldots, \left\{ (X_n, A_n), (\hat{T}^* Q_k)(X_n, A_n) \right\} \)

\((\hat{T}^* Q)(X_i, A_i) \overset{\text{def}}{=} R_i + \gamma \max_{a'} Q(Y_i, a'), \quad i = 1, \ldots, n.\)

\[\|\tilde{Q}_k - T^* Q_k\|_2^2 \leq \bar{b}_k\]
How to get
the error bound?

\[ \| \tilde{Q}_k - T^* Q_k \|_v^2 \leq \tilde{b}_k \]
How to get the error bound?

\[ \| \tilde{Q}_k - T^* Q_k \|_{\nu}^2 \leq \tilde{b}_k \]

\[ \| \tilde{Q}_k - T^* Q_k \|_{\nu}^2 = \mathbb{E} \left[ \| \tilde{Q}_k - T^* Q_k \|_{D_n''}^2 | D_n' \right] \]

\[ = \mathbb{E} \left[ \| \tilde{Q}_k - \hat{T}^* Q_k \|_{D_n''}^2 | D_n' \right] - \mathbb{E} \left[ \| T^* Q_k - \hat{T}^* Q_k \|_{D_n''}^2 \right]. \]
How to get the error bound?

\[ \| \tilde{Q}_k - T^* Q_k \|_{\mathcal{L}}^2 \leq \bar{b}_k \]

\[ \| \tilde{Q}_k - T^* Q_k \|_{\mathcal{L}}^2 = \mathbb{E} \left[ \| \tilde{Q}_k - T^* Q_k \|_{\mathcal{D}_n''}^2 \mid \mathcal{D}_n' \right] \]

\[ = \mathbb{E} \left[ \| \tilde{Q}_k - \hat{T}^* Q_k \|_{\mathcal{D}_n''}^2 \mid \mathcal{D}_n' \right] - \mathbb{E} \left[ \| T^* Q_k - \hat{T}^* Q_k \|_{\mathcal{D}_n''}^2 \right]. \]

\[ L_k^* = \mathbb{E} \left[ \| T^* Q_k - \hat{T}^* Q_k \|_{\mathcal{D}_n''}^2 \right]. \]
How to get the error bound?

\[ \| \tilde{Q}_k - T^* Q_k \|_2^2 \leq \tilde{b}_k \]

\[ \| \tilde{Q}_k - T^* Q_k \|_2^2 = \mathbb{E} \left[ \| \tilde{Q}_k - T^* Q_k \|_{D_n''}^2 | D_n' \right] 
= \mathbb{E} \left[ \| \tilde{Q}_k - \hat{T}^* Q_k \|_{D_n''}^2 | D_n' \right] - \mathbb{E} \left[ \| T^* Q_k - \hat{T}^* Q_k \|_{D_n''}^2 \right]. \]

\[ L^*_k = \mathbb{E} \left[ \| T^* Q_k - \hat{T}^* Q_k \|_{D_n''}^2 \right] \]

??
How to get the error bound?

$$\|\tilde{Q}_k - T^* Q_k\|_\nu^2 \leq \tilde{b}_k$$

$$\|\tilde{Q}_k - T^* Q_k\|_\nu^2 = \mathbb{E} \left[ \|\tilde{Q}_k - T^* Q_k\|_{D_n''}^2 | D'_n \right]$$

$$= \mathbb{E} \left[ \|\tilde{Q}_k - \hat{T}^* Q_k\|_{D_n''}^2 | D'_n \right] - \mathbb{E} \left[ \|T^* Q_k - \hat{T}^* Q_k\|_{D_n''}^2 \right].$$

$$L^*_k = \mathbb{E} \left[ \|T^* Q_k - \hat{T}^* Q_k\|_{D_n''}^2 \right]$$

Devroye et al. (2003)

regression + plug-in
Main Result: Assumptions “C”

1. The time-homogeneous Markov chain $X_1, X_2, \ldots, X_n$ uniformly quickly forgets its past with a forgetting time of $\tau$;

2. The functions $Q_k, \tilde{Q}_k, T^*Q_k (k \geq 1)$ are bounded by a deterministic quantity $B > 0$;

3. The functions $Q_k (k \geq 1)$ are deterministic;

4. Simultaneously, for all $k \geq 1$, $\|\tilde{Q}_k - T^*Q_k\|_{L^2}^2 \leq \bar{b}_k$ holds with probability at least $1 - \delta/3$, where $\bar{b}_k$ is $\sigma(D'_n)$-measurable;

5. $\bar{b}_k \leq 4B^2$ holds a.s.;

6. $D''_n$ holds $n$ datapoints.
Main Result
Main Result

Theorem: Let $A$ be the base-RL algorithm, REGRESS:=A. Then, under “C” and some more technical conditions, BerMin is an adaptive algorithm.
Contents

• The problem of adaptation
• Methods of supervised learning
• Adaptation to RL: Value function estimation
• Finding a good policy: A model based approach
• Discussion
Approach #1: Estimate Value Functions

$$\pi_i \longrightarrow \hat{V}^{\pi_i}$$

$$\Delta_i = |\hat{V}_\rho^{\pi_i} - V_\rho^{\pi_i}| \leq \|\hat{V}_\rho^{\pi_i} - V^{\pi_k}\|_{L^1(\rho)}$$

$$\Rightarrow$$ To get an overall adaptive procedure, we need an adaptive procedure for estimating the value functions!

$$\Rightarrow$$ previous problem
Approach #2:
Estimate a model
Approach #2: Estimate a model

• Idea: Estimate a model and use it to evaluate the policies
Approach #2: Estimate a model

- Idea: Estimate a model and use it to evaluate the policies
- Model-learning is just supervised learning; adaptation there is easier
Approach #2: Estimate a model

• Idea: Estimate a model and use it to evaluate the policies

• Model-learning is just supervised learning; adaptation there is easier

• How will errors propagate??
Approach #2: Estimate a model

• Idea: Estimate a model and use it to evaluate the policies

• Model-learning is just supervised learning; adaptation there is easier

• How will errors propagate??

• Let $\hat{V}^{\pi_i}$ be the value function of $\pi_i$ in the approximate model $\hat{M}$
Error Propagation
Error Propagation

\[ \hat{k} = \arg\max_{k} \hat{V}_{\rho}^{\pi_{k}} \]
Error Propagation

\[ \hat{k} = \arg\max_k \hat{\mathcal{V}}^{\pi_k} \]

\[ \varepsilon(\hat{M}) = \frac{C'}{1 - \gamma} \left\{ \| \delta_r \|_{L^2(\mu)} + \gamma \| \delta_p \|_{L^2(\mu \times \mu)} \right\} \]
Error Propagation

\[ \hat{k} = \arg\max_k \hat{V}_\rho^{\pi_k} \]

\[ \varepsilon(\hat{M}) = \frac{C'}{1 - \gamma} \left\{ \| \delta_r \|_{L^2(\mu)} + \gamma \| \delta_p \|_{L^2(\mu \times \mu)} \right\} \]

\[ \delta_r(x) = \max_{a \in A} |\hat{r}(x, a) - r(x, a)| \]

\[ \delta_p(y|x) = \max_{a \in A} |\hat{p}(y|x, a) - p(y|x, a)| \]
Error Propagation

\[ \hat{k} = \arg \max_k \hat{V}_\rho^{\pi_k} \]

\[ \varepsilon(\hat{M}) = \frac{C'}{1 - \gamma} \left\{ \| \delta_r \|_{L^2(\mu)} + \gamma \| \delta_p \|_{L^2(\mu \times \mu)} \right\} \]

\[ \delta_r(x) = \max_{a \in A} |\hat{r}(x, a) - r(x, a)| \]

\[ \delta_p(y|x) = \max_{a \in A} |\hat{p}(y|x, a) - p(y|x, a)| \]

**Theorem:** The value of the policy selected will not be worse than the value of the best policy plus \( 2\varepsilon(\hat{M}) \).
The Key Lemma

Let $B$ be a Banach-space with norm $\|\cdot\|$. Pick some $V \in B$. Let

$$T_1, T_2 : B \to B,$$

$$U_s = T_2^s V.$$

Assume that $T_1, T_2$ satisfy the following:

1. $T_1$ is a $\gamma$-contraction in $\|\cdot\|$ with fixed point $V_1^*$;

2. $\lim_{s \to \infty} \|U_s - V_2^*\| \to 0$ and $V_2^*$ is a fixed point of $T_2$.

Then,

$$\|V_1^* - V_2^*\| \leq \frac{\sup_{s \geq 1} \|T_1 U_s - T_2 U_s\|}{1 - \gamma}.$$
Final Step
Final Step

The Lemma gives:

\[ \| \hat{V}^{\pi_k} - V^{\pi_k} \|_{L^1(\rho)} \leq \varepsilon(\hat{M}) \]
Final Step

The Lemma gives:

$$\| \hat{V}^{\pi_k} - V^{\pi_k} \|_{L^1(\rho)} \leq \varepsilon(\hat{M})$$

Finally,

$$V^{\pi_{\hat{k}}}_{\rho} \geq \hat{V}^{\pi_{\hat{k}}}_{\rho} - \varepsilon(\hat{M}) \geq \hat{V}^{\pi_{k^*}}_{\rho} - \varepsilon(\hat{M}) \geq V^{\pi_{k^*}}_{\rho} - 2\varepsilon(\hat{M}).$$
Should we Learn Models?

STRING THEORY SUMMARIZED:

I just had an awesome idea. Suppose all matter and energy is made of tiny, vibrating “strings.”

Okay. What would that imply?

I dunno.

(xkcd)
Contents

• The problem of adaptation
• Methods of supervised learning
• Adaptation to RL: Value function estimation
• Finding a good policy: A model based approach
• Discussion
Summary
Summary

• Adaptive selection/tuning
Summary

• Adaptive selection/tuning
• Batch RL
Summary

- Adaptive selection/tuning
- Batch RL
  - Finding the action value function with the minimum Bellman error
Summary

• Adaptive selection/tuning
• Batch RL
  • Finding the action value function with the minimum Bellman error
  • Finding the best policy given a list
Issues not touched
Issues not touched

• Computational complexity
Issues not touched

• Computational complexity
• How to structure the search?
Issues not touched

• Computational complexity
• How to structure the search?
  • Search the penalty factors
Issues not touched

• Computational complexity
• How to structure the search?
  • Search the penalty factors
• Method “Generate” should do the feature selection!
Back to our questions..

• How good policies can be learnt? Universal consistency? Does u.c. make sense?

• Is there any difference to supervised learning? What?

• What methods to use?

• Can we use model selection? How?

• Can we predict the performance of a learnt policy? How? What are the limits?