Computing the Feedback Capacity of Finite State Channels using Reinforcement Learning

Ziv Aharoni, Oron Sabag, Haim H. Permuter

Ben-Gurion University of the Negev

International Symposium on Information Theory
July 9th, 2019
Communication with Feedback

- Unifilar finite state channel (FSC):
  \[
  p(y_t|x_t, s_{t-1}) \\
  s_t = f(x_t, y_t, s_{t-1})
  \]

- **The goal**: compute the capacity and coding scheme
The Capacity

Theorem (Permuter-Cuff-Van Roy-Weissman’08)

The feedback capacity of unifilar FSC

\[ C_{fb} = \lim_{n \to \infty} \max_{\{p(x_i|s_{i-1},y^{i-1})\}_{i=1}^n} \frac{1}{n} I(X^n \rightarrow Y^n) \]

- The directed information (Massey 1990)

\[ I(X^n \rightarrow Y^n) = \sum_{i=1}^{n} I(X_i; Y_i|Y^{i-1}) \]

- This is a multi-letter expression
Markov Decision Process (MDP) Formulation

Theorem (Permuter-Cuff-Van Roy-Weissman’08)

The feedback capacity of a unifilar FSC can be formulated as a Markov decision process.
The feedback capacity of a unifilar FSC can be formulated as a Markov decision process

- MDP’s state: $p(s_t|y^t)$
Theorem (Permuter-Cuff-Van Roy-Weissman’08)

The feedback capacity of a unifilar FSC can be formulated as a Markov decision process

- MDP’s state: \( p(s_t|y^t) \)

**Solution methods:**

- Dynamic programming - value iteration algorithm
  - Effective only for binary alphabet
Markov Decision Process (MDP) Formulation

**Theorem (Permuter-Cuff-Van Roy-Weissman’08)**

The feedback capacity of a unifilar FSC can be formulated as a Markov decision process

- MDP’s state: $p(s_t|y^t)$

**Solution methods:**

- Dynamic programming - value iteration algorithm
  Effective only for binary alphabet

- Reinforcement learning (RL)
  Effective for large alphabets
Reinforcement Learning

- $Z_{t-1}$ - current state
- $U_t$ - action
- $R_t$ - reward
- $Z_t$ - next state
The goal: maximize the expected average reward

\[
\mathbb{E}_\pi [G] = \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_\pi [R_t]
\]
The goal: maximize the expected average reward

\[ \mathbb{E}_\pi [G] = \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_\pi [R_t] \]

The state-action value function

\[ Q_\pi (z, u) = \mathbb{E}_\pi [G | Z_1 = z, U_1 = u] \]
Q-Learning Approach

Agent

Environment

Experience Buffer

Actor

Critic

$Z_{t-1}$

$Z_t, R_t$

$\pi_\mu(z)$

$Q_\omega(z, u)$

$\Delta$

$U_t$

$\Delta$

$Z_t, R_t$

Train

Improve

Improve

Feedforward Capacity of FSC using RL

Ziv Aharoni, Oron Sabag, Haim H. Permuter

Feedback Capacity of FSC using RL

7 / 18
The DDPG Algorithm

Deep Deterministic Policy Gradient, (Lillicrap et al.’16)

- Draw $N$ interactions from experience $(z_i, u_i, r_i, z'_i)$
- Train critic: minimize by $\omega$

$$\frac{1}{N} \sum_{i=1}^{N} [Q_\omega(z_i, u_i) - [r_i - \rho_\mu + Q_\omega(z'_i, \pi_\mu(z'_i))]^2$$

- Improve actor: maximize by $\mu$

$$\frac{1}{N} \sum_{i=1}^{N} \nabla_u Q_\omega(z_i, u) \big|_{u=\pi_\mu(z_i)} \nabla_\mu \pi_\mu(z_i)$$
The Ising Channel

- Defined by Berger and Bonomi (1990):

\[ Y_i = \begin{cases} 
   S_{i-1} & \text{w.p. } 0.5 \\
   X_i & \text{w.p. } 0.5 
\end{cases} \]

\[ S_{i-1} = X_{i-1} \]

- Models channel with ISI, magnetic recording
The Ising Channel

Defined by Berger and Bonomi (1990):

\[ Y_i = \begin{cases} 
S_{i-1}, & \text{w.p. } 0.5 \\
X_i, & \text{w.p. } 0.5
\end{cases}, \quad S_{i-1} = X_{i-1} \]

Models channel with ISI, magnetic recording

Solved the binary case (Elischo-Permuter’14, Sharov-Roth’16)
The Ising Channel

- Defined by Berger and Bonomi (1990):
  \[
  Y_i = \begin{cases} 
  S_{i-1}, & \text{w.p. 0.5} \\
  X_i, & \text{w.p. 0.5}
  \end{cases}, \quad S_{i-1} = X_{i-1}
  \]

- Models channel with ISI, magnetic recording

- Solved the binary case (Elischo-Permuter'14, Sharov-Roth'16)

- **The goal:** apply RL methodology to larger alphabets
The goal: maximize **average reward**
The goal: maximize **achievable rate**
Numerical results - Achievable Rate

Reveal the **structure** of the optimal solution
Properties of the Estimated Solution

State histogram of estimated transmitter

\[ p(s_t = 1 | y^{t-1}) \]

\[ p(s_t = 0 | y^{t-1}) \]
Properties of the Estimated Solution

State histogram of estimated transmitter

\[ p(s_t = 1 | y^{t-1}) \]
\[ p(s_t = 0 | y^{t-1}) \]

- Optimal input distribution structure
Properties of the Estimated Solution

- Optimal input distribution structure
- Transitions between states as function of channel's output
Transitions of states by a Q-graph

\[ Y = 0 \]
\[ Y = 1 \]
\[ Y = 2 \]
Transitions of states by a Q-graph

Design coding scheme
Transitions of states by a Q-graph

Design coding scheme
Prove upper-bound
Code scheme

**Pre-transmission:** generate an information sequence s.t.

\[ x_i = \begin{cases} 
    x_{i-1}, & \text{w.p } p \\ 
    \text{Unif}[\mathcal{X} \setminus x_{i-1}], & \text{w.p } 1 - p 
\end{cases} \]
Coding scheme

- **Pre-transmission**: generate an information sequence s.t.

\[ x_i = \begin{cases} 
  x_{i-1}, & \text{w.p } p \\
  \text{Unif}[\mathcal{X} \setminus x_{i-1}], & \text{w.p } 1 - p
\end{cases} \]

- At the beginning: assume decoder knows \( s \)
Coding scheme

- **Pre-transmission:** generate an information sequence s.t.
  \[ x_i = \begin{cases} 
  x_{i-1}, & \text{w.p } p \\
  \text{Unif}[\mathcal{X} \setminus x_{i-1}], & \text{w.p } 1 - p
  \end{cases} \]

- At the beginning: assume decoder knows \( s \)

- **Encoder:**
  Transmit \( x \), if \( y = s \), repeat \( x \)
**Coding scheme**

- **Pre-transmission:** generate an information sequence s.t.

  \[ x_i = \begin{cases} 
  x_{i-1} & \text{w.p } p \\
  \text{Unif}[\mathcal{X} \setminus x_{i-1}] & \text{w.p } 1 - p 
  \end{cases} \]

- At the beginning: assume decoder knows \( s \)

- **Encoder:**
  Transmit \( x \), if \( y = s \), repeat \( x \)

- **Decoder:**
  Receive \( y \), if \( y \neq s \) store \( y \), else store next output
Coding scheme

- **Pre-transmission:** generate an information sequence s.t.
  \[ x_i = \begin{cases} 
  x_{i-1} & \text{, w.p } p \\
  \text{Unif}[\mathcal{X}\setminus x_{i-1}] & \text{, w.p } 1 - p 
  \end{cases} \]

- At the beginning: assume decoder knows \( s \)

- **Encoder:**
  Transmit \( x \), if \( y = s \), repeat \( x \)

- **Decoder:**
  Receive \( y \), if \( y \neq s \) store \( y \), else store next output

- **The rate of the scheme:**
  \[
  R(\mathcal{X}) = \max_{p \in [0,1]} 2 \frac{H_2(p) + (1 - p) \log (|\mathcal{X}|-1)}{p + 3}
  \]
Theorem (Sabag-Permuter-Pfister’17)

For any choice of Q-graph

\[ C_{fb} \leq \max_{p(x|s,q) \in \mathcal{P}_{\pi}} I(X, S; Y|Q) \]
Upper-bound

**Theorem (Sabag-Permuter-Pfister’17)**

*For any choice of Q-graph*

\[ C_{fb} \leq \max_{p(x|s,q)\in \mathcal{P}_\pi} I(X, S; Y|Q) \]

**Theorem (Duality bound)**

*For any FSC channel and $T_{Y|Q}$*

\[ C_{fb} \leq \lim_{n \to \infty} \max f(x^n||y^{n-1}) \frac{1}{n} \sum_{i=1}^{n} \mathbb{E} \left[ D \left( P_{Y|X=x_i, X^{-}=x_{i-1}} \parallel T_{Y|Q=Q_{i-1}} \right) \right] \]
The feedback capacity

**Theorem**

For all $|\mathcal{X}| \leq 8$, the feedback capacity of the Ising channel is given by

$$C_{fb} (\mathcal{X}) = \max_{p \in [0,1]} 2 \frac{H_2(p) + (1 - p) \log (|\mathcal{X}| - 1)}{p + 3}$$

- What happens for $|\mathcal{X}| \geq 9$?
- Asymptotic better rate

$$R (\mathcal{X}) = \frac{3}{4} \log \frac{|\mathcal{X}|}{2}$$
Improving RL: DDPG with planning

Alphabet size 3

Achievable rate vs. episodes

Alphabet size 6

Achievable rate vs. episodes

Alphabet size 9

Achievable rate vs. episodes
Improving RL: DDPG with planning

Alphabet size 3

Alphabet size 6

Alphabet size 9

achievable rate

episodes

vanilla
planning

capacity

achievable rate

episodes

vanilla
planning

capacity

achievable rate

episodes

vanilla
planning

capacity
Conclusions

- RL methodology for analytic feedback capacity of unifilar FSC
- Analytic solution to improve RL
Conclusions

- RL methodology for analytic feedback capacity of unifilar FSC
- Analytic solution to improve RL

Future Work:
- What is the solution for $|\mathcal{X}| \geq 9$?
Conclusions

- RL methodology for analytic feedback capacity of unifilar FSC
- Analytic solution to improve RL

Future Work:

- What is the solution for $|\mathcal{X}| \geq 9$?
- Solve more channels to constitute an IT benchmark for RL algorithms
Conclusions

- RL methodology for analytic feedback capacity of unifilar FSC
- Analytic solution to improve RL

Future Work:
- What is the solution for $|\mathcal{X}| \geq 9$?
- Solve more channels to constitute an IT benchmark for RL algorithms

Thank You!
Transitions of states by a directed graph
The DDPG Algorithm

Deep Deterministic Policy Gradient, (Lillicrap et al.’16)

- Draw $N$ interactions from experience $(z_i, u_i, r_i, z'_i)$
- Train critic: minimize by $\omega$
  \[
  \frac{1}{N} \sum_{i=1}^{N} \left[ Q_\omega(z_i, u_i) - \left( r_i - \rho_\mu + \sum_{z'} p(z'|z_i, u_i) Q_\omega(z', \pi_\mu(z')) \right) \right]^2
  \]
- Improve actor: maximize by $\mu$
  \[
  \frac{1}{N} \sum_{i=1}^{N} \nabla_u Q_\omega(z_i, u) \big|_{u=\pi_\mu(z_i)} \nabla_\mu \pi_\mu(z_i)
  \]