An Alternative Softmax Operator for Reinforcement Learning

Kavosh Asadi  Michael L. Littman

Department of Computer Science
Brown University

ICML, 2017
Outline

1. our motivation
2. non-expansion property and convergence
3. decision making with the operator
4. experiments
5. related and future work
Outline

1. our motivation
2. non-expansion property and convergence
3. decision making with the operator
4. experiments
5. related and future work
arguably one of the most important problems in RL (Thrun 1992)
arguably one of the most important problems in RL (Thrun 1992)

principled approaches such as PAC RL (Kearns and Singh 2002; Strehl et al. 2006) and Bayesian RL (Dearden et al. 1998) can address it
the exploration-exploitation problem

- arguably one of the most important problems in RL (Thrun 1992)
- principled approaches such as PAC RL (Kearns and Singh 2002; Strehl et al. 2006) and Bayesian RL (Dearden et al. 1998) can address it
- however, not always easy to scale up these approaches
the exploration-exploitation problem

- arguably one of the most important problems in RL (Thrun 1992)
- principled approaches such as PAC RL (Kearns and Singh 2002; Strehl et al. 2006) and Bayesian RL (Dearden et al. 1998) can address it
- however, not always easy to scale up these approaches
- an easy alternative: use a softmax
  - operator: $\mathbb{R}^n \mapsto \mathbb{R}$
  - policy: $\mathcal{A} \mapsto \mathcal{P}(\mathcal{A})$
softmax operators

- a softmax operator takes a number of (state-action) values and outputs a number to summarize the state’s utility, $\otimes : \mathbb{R}^n \mapsto \mathbb{R}$
softmax operators

- A softmax operator takes a number of (state-action) values and outputs a number to summarize the state’s utility, $\otimes: \mathbb{R}^n \rightarrow \mathbb{R}$.
- Widely used for value-function optimization.

---

Kavosh Asadi, Michael L. Littman
mellowmax
ICML, 2017 5 / 18
softmax operators

- a softmax operator takes a number of (state-action) values and outputs a number to summarize the state’s utility, $\otimes: \mathbb{R}^n \mapsto \mathbb{R}$
- widely used for value-function optimization
- in Q-learning (Rummery and Niranjan 1994) and expected SARSA (Van Seijen et al. 2009):

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left( r + \gamma \otimes_{a'} Q(s', a') - Q(s, a) \right)$$
softmax operators

- A softmax operator takes a number of (state-action) values and outputs a number to summarize the state’s utility, \( \otimes : \mathbb{R}^n \rightarrow \mathbb{R} \).
- Widely used for value-function optimization.
- In Q-learning (Rummery and Niranjan 1994) and expected SARSA (Van Seijen et al. 2009):

\[
Q(s, a) \leftarrow Q(s, a) + \alpha \left( r + \gamma \otimes a' Q(s', a') - Q(s, a) \right)
\]

- And the expectation of the update performed by SARSA! (Singh and Sutton 1996)
softmax operators

- a softmax operator takes a number of (state-action) values and outputs a number to summarize the state’s utility, \( \otimes : \mathbb{R}^n \rightarrow \mathbb{R} \)
- widely used for value-function optimization
- in Q-learning (Rummery and Niranjan 1994) and expected SARSA (Van Seijen et al. 2009):

\[
Q(s, a) \leftarrow Q(s, a) + \alpha \left( r + \gamma \otimes a' Q(s', a') - Q(s, a) \right)
\]

- and the expectation of the update performed by SARSA! (Singh and Sutton 1996)
- desired properties:
softmax operators

- A softmax operator takes a number of (state-action) values and outputs a number to summarize the state's utility, \( \otimes : \mathbb{R}^n \mapsto \mathbb{R} \).
- Widely used for value-function optimization.
- In Q-learning (Rummery and Niranjan 1994) and expected SARSA (Van Seijen et al. 2009):

\[
Q(s, a) \leftarrow Q(s, a) + \alpha \left( r + \gamma \otimes a' Q(s', a') - Q(s, a) \right)
\]

- And the expectation of the update performed by SARSA! (Singh and Sutton 1996).
- Desired properties:
  - Differentiable
softmax operators

- A softmax operator takes a number of (state-action) values and outputs a number to summarize the state’s utility, $\otimes : \mathbb{R}^n \mapsto \mathbb{R}$.
- Widely used for value-function optimization.
- In Q-learning (Rummery and Niranjan 1994) and expected SARSA (Van Seijen et al. 2009):

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left( r + \gamma \otimes a' Q(s', a') - Q(s, a) \right)$$

- And the expectation of the update performed by SARSA! (Singh and Sutton 1996).
- Desired properties:
  - Differentiable
  - Approximate max, min, and mean using a knob (like $\epsilon$ in $\epsilon$-greedy).
softmax operators

- A softmax operator takes a number of (state-action) values and outputs a number to summarize the state’s utility, \( \otimes : \mathbb{R}^n \rightarrow \mathbb{R} \).
- Widely used for value-function optimization.
- In Q-learning (Rummery and Niranjan 1994) and expected SARSA (Van Seijen et al. 2009):

\[
Q(s, a) \leftarrow Q(s, a) + \alpha \left( r + \gamma \otimes a' Q(s', a') - Q(s, a) \right)
\]

- And the expectation of the update performed by SARSA! (Singh and Sutton 1996).
- Desired properties:
  - Differentiable
  - Approximate max, min, and mean using a knob (like \( \epsilon \) in \( \epsilon \)-greedy).
  - No action starvation.
softmax operators

- A softmax operator takes a number of (state-action) values and outputs a number to summarize the state’s utility, $\otimes : \mathbb{R}^n \mapsto \mathbb{R}$
- Widely used for value-function optimization
- In Q-learning (Rummery and Niranjan 1994) and expected SARSA (Van Seijen et al. 2009):

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left( r + \gamma \otimes a'Q(s', a') - Q(s, a) \right)$$

- And the expectation of the update performed by SARSA! (Singh and Sutton 1996)
- Desired properties:
  - Differentiable
  - Approximate max, min, and mean using a knob (like $\epsilon$ in $\epsilon$-greedy)
  - No action starvation
  - Convergent with bootstrapping (non-expansion)
<table>
<thead>
<tr>
<th>name</th>
<th>property</th>
<th>≈ mean</th>
<th>≈ max</th>
<th>no starvation</th>
<th>differentiable</th>
<th>non-expansion</th>
</tr>
</thead>
</table>

An alternative softmax operator:

$$\omega_{\text{mellowmax}}(X) = \log\left( \frac{1}{n} \sum_{i=1}^{n} e^{\omega x_i} \right)$$

ωmellowmax has all the properties above.
common operators and their properties

<table>
<thead>
<tr>
<th>name</th>
<th>≈ mean</th>
<th>≈ max</th>
<th>no starvation</th>
<th>differentiable</th>
<th>non-expansion</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
common operators and their properties

<table>
<thead>
<tr>
<th>name</th>
<th>≈ mean</th>
<th>≈ max</th>
<th>no starvation</th>
<th>differentiable</th>
<th>non-expansion</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>max</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
</tbody>
</table>

mellowmax has all the properties above

An alternative softmax operator:

\[
\omega \cdot \text{mellowmax}(X) = \log\left( \frac{1}{n} \sum_{i=1}^{n} e^{\omega \cdot x_i} \right)
\]

\(\omega\) mellowmax is a smooth approximation of max in optimization literature.
# Common Operators and Their Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Property</th>
<th>$\approx$ Mean</th>
<th>$\approx$ Max</th>
<th>No Starvation</th>
<th>Differentiable</th>
<th>Non-Expansion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Max</td>
<td>×</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>$\epsilon$-Greedy</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>✓</td>
</tr>
</tbody>
</table>

**Mellowmax** has all the properties above, which is a smooth approximation of max in optimization literature.
## common operators and their properties

<table>
<thead>
<tr>
<th>name</th>
<th>property</th>
<th>≈ mean</th>
<th>≈ max</th>
<th>no starvation</th>
<th>differentiable</th>
<th>non-expansion</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td></td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>max</td>
<td></td>
<td>×</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>ε-greedy</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>Boltzmann</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
</tr>
</tbody>
</table>

An alternative softmax operator: $mm(\omega(X)) = \log\left(\frac{1}{n} \sum_{i=1}^{n} e^{\omega x_i}\right)$

$mellowmax$ has all the properties above.

Smooth approximation of max in optimization literature.
common operators and their properties

<table>
<thead>
<tr>
<th>name</th>
<th>≈ mean</th>
<th>≈ max</th>
<th>no starvation</th>
<th>differentiable</th>
<th>non-expansion</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>max</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>$\epsilon$-greedy</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Boltzmann</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
</tbody>
</table>

- an alternative softmax operator:

$$\text{mm}_\omega(X) = \log\left(\frac{1}{n} \sum_{i=1}^{n} e^{\omega x_i}\right)$$
### common operators and their properties

<table>
<thead>
<tr>
<th>name</th>
<th>property</th>
<th>(\approx) mean</th>
<th>(\approx) max</th>
<th>no starvation</th>
<th>differentiable</th>
<th>non-expansion</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>max</td>
<td>×</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>(\epsilon)-greedy</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Boltzmann</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
</tr>
</tbody>
</table>

- an alternative softmax operator:

\[
mm_\omega(x) = \log\left(\frac{1}{n} \sum_{i=1}^{n} e^{\omega x_i}\right)
\]

- *mellowmax* has all the properties above
common operators and their properties

<table>
<thead>
<tr>
<th>name</th>
<th>property</th>
<th>≈ mean</th>
<th>≈ max</th>
<th>no starvation</th>
<th>differentiable</th>
<th>non-expansion</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>max</td>
<td>×</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>ε-greedy</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Boltzmann</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
</tr>
</tbody>
</table>

an alternative softmax operator:

\[ \text{mm}_\omega(X) = \log \left( \frac{1}{n} \sum_{i=1}^{n} e^{\omega x_i} \right) \]

- *mellowmax* has all the properties above
- smooth approximation of max in optimization literature
Outline

1. our motivation
2. non-expansion property and convergence
3. decision making with the operator
4. experiments
5. related and future work
on the importance of non-expansion

- non-expansion under $\infty$-norm:

$$\left| \bigotimes_a Q_1(s, a) - \bigotimes_a Q_2(s, a) \right| \leq \max_a \left| Q_1(s, a) - Q_2(s, a) \right|$$
on the importance of non-expansion

- non-expansion under $\infty$-norm:

\[
\| \bigotimes_a Q_1(s, a) - \bigotimes_a Q_2(s, a) \| \leq \max_a \| Q_1(s, a) - Q_2(s, a) \|
\]

- non-expansion guarantees convergence of “Q-learning like” algorithms to a unique fixed point (Littman and Szepesvári 1996)
on the importance of non-expansion

- non-expansion under $\infty$-norm:

\[
\left| \bigotimes_a Q_1(s, a) - \bigotimes_a Q_2(s, a) \right| \leq \max_a \left| Q_1(s, a) - Q_2(s, a) \right|
\]

- non-expansion guarantees convergence of “Q-learning like” algorithms to a unique fixed point (Littman and Szepesvári 1996)

- max operator is a non-expansion

\[
\left| \max_a Q_1(s, a) - \max_a Q_2(s, a) \right| \leq \max_a \left| Q_1(s, a) - Q_2(s, a) \right|
\]
on the importance of non-expansion

- non-expansion under $\infty$-norm:

  \[ \left| \bigoplus_a Q_1(s, a) - \bigoplus_a Q_2(s, a) \right| \leq \max_a \left| Q_1(s, a) - Q_2(s, a) \right| \]

- non-expansion guarantees convergence of “Q-learning like” algorithms to a unique fixed point (Littman and Szepesvári 1996)

- max operator is a non-expansion

  \[ \left| \max_a Q_1(s, a) - \max_a Q_2(s, a) \right| \leq \max_a \left| Q_1(s, a) - Q_2(s, a) \right| \]

- basis of proof for convergence of Value Iteration and Q-learning
mellowmax is a non-expansion, but Boltzmann expands

- mellowmax is a non-expansion along with the other desired properties
mellowmax is a non-expansion, but Boltzmann expands

- mellowmax is a non-expansion along with the other desired properties
- Boltzmann is not, misbehavior may happen (multiple fixed points, no convergence, ...) even in tabular setting
mellowmax is a non-expansion, but Boltzmann expands

- mellowmax is a non-expansion along with the other desired properties
- Boltzmann is not, misbehavior may happen (multiple fixed points, no convergence, ...) even in tabular setting
- an example with multiple fixed points is shown in the paper
mellowmax is a non-expansion, but Boltzmann expands

- mellowmax is a non-expansion along with the other desired properties
- Boltzmann is not, misbehavior may happen (multiple fixed points, no convergence, ...) even in tabular setting
- an example with multiple fixed points is shown in the paper
- Value Iteration’s misbehavior on randomly generated MDPs:

<table>
<thead>
<tr>
<th></th>
<th>MDPs, no terminate</th>
<th>MDPs, &gt; 1 fixed points</th>
<th>average iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>boltz_β</td>
<td>8 of 200</td>
<td>3 of 200</td>
<td>231.65</td>
</tr>
<tr>
<td>mm_ω</td>
<td>0</td>
<td>0</td>
<td>201.32</td>
</tr>
</tbody>
</table>
Outline

1. our motivation
2. non-expansion property and convergence
3. decision making with the operator
4. experiments
5. related and future work
maximum entropy mellowmax policy
solution of the following optimization problem

\[
\pi_{mm} = \arg\min_{\pi} \sum_{a \in A} \pi(a|s) \log \left( \pi(a|s) \right)
\]

subject to \( \sum_{a \in A} \pi(a|s)Q(s,a) = mm\omega(Q(s, .)) \)
\( \pi(a|s) \geq 0 \)
\( \sum_{a \in A} \pi(a|s) = 1 \)
solution of the following optimization problem

\[ \pi_{mm} = \arg\min_{\pi} \sum_{a \in A} \pi(a|s) \log(\pi(a|s)) \]

subject to

\[ \sum_{a \in A} \pi(a|s) Q(s, a) = mm_\omega(Q(s, .)) \]
\[ \pi(a|s) \geq 0 \]
\[ \sum_{a \in A} \pi(a|s) = 1 \]

solved by finding the root of a polynomial
maximum entropy mellowmax policy

- solution of the following optimization problem

\[
\pi_{mm} = \arg\min_\pi \sum_{a \in A} \pi(a|s) \log (\pi(a|s))
\]

subject to \[
\sum_{a \in A} \pi(a|s) Q(s,a) = mm \omega(Q(s, .))
\]

- solved by finding the root of a polynomial
- SARSA with a softmax policy and a state-dependent temperature parameter. Also utilized for neural net activation functions
maximum entropy mellowmax policy

- solution of the following optimization problem

\[
\pi_{mm} = \arg\min_{\pi} \sum_{a \in A} \pi(a|s) \log(\pi(a|s))
\]

subject to \[
\sum_{a \in A} \pi(a|s) Q(s, a) = mm \omega(Q(s, .))
\]

\[
\sum_{a \in A} \pi(a|s) \geq 0
\]

\[
\sum_{a \in A} \pi(a|s) = 1
\]

- solved by finding the root of a polynomial
- SARSA with a softmax policy and a state-dependent temperature parameter. Also utilized for neural net activation functions
- similar ideas presented by prior work (Todorov 2007; Peters et al. 2010; Ziebart 2010)
Maximum entropy mellowmax policy

- solution of the following optimization problem

$$\pi_{mm} = \arg\min_{\pi} \sum_{a \in A} \pi(a|s) \log (\pi(a|s))$$

subject to:

$$\sum_{a \in A} \pi(a|s) Q(s, a) = mm\omega(Q(s,.))$$

$$\pi(a|s) \geq 0$$

$$\sum_{a \in A} \pi(a|s) = 1$$

- solved by finding the root of a polynomial
- SARSA with a softmax policy and a state-dependent temperature parameter. Also utilized for neural net activation functions
- similar ideas presented by prior work (Todorov 2007; Peters et al. 2010; Ziebart 2010)
- and also by more recent and independent studies (Fox et al. 2016; Nachum et al. 2017; Neu et al. 2017)
Outline

1. our motivation
2. non-expansion property and convergence
3. decision making with the operator
4. experiments
5. related and future work
we evaluated max entropy mellowmax policy on multi-passenger taxi domain with SARSA
we evaluated max entropy mellowmax policy on multi-passenger taxi domain with SARSA

SARSA under Boltzmann performs well here, outperforming Bayesian RL algorithms (Dearden et al. 1998)
experiments on taxi

- we evaluated max entropy mellowmax policy on multi-passenger taxi domain with SARSA

- SARSA under Boltzmann performs well here, outperforming Bayesian RL algorithms (Dearden et al. 1998)

- we show mean episode return after a fixed number of episodes with respect to knobs (averaged over 300 independent runs)
experiments on taxi

- we evaluated max entropy mellowmax policy on multi-passenger taxi domain with SARSA
- SARSA under Boltzmann performs well here, outperforming Bayesian RL algorithms (Dearden et al. 1998)
- we show mean episode return after a fixed number of episodes with respect to knobs (averaged over 300 independent runs)
we evaluated maximum entropy mellowmax policy on lunar lander, and as activation function of a neural net
we evaluated maximum entropy mellowmax policy on lunar lander, and as activation function of a neural net.

networks are trained using REINFORCE (Williams 1992)
we evaluated maximum entropy mellowmax policy on lunar lander, and as activation function of a neural net

networks are trained using REINFORCE (Williams 1992)

we show mean episode return after a fixed number of episodes (averaged over 400 independent runs)
we evaluated maximum entropy mellowmax policy on lunar lander, and as activation function of a neural net networks are trained using REINFORCE (Williams 1992)
we show mean episode return after a fixed number of episodes (averaged over 400 independent runs)
Outline

1. our motivation
2. non-expansion property and convergence
3. decision making with the operator
4. experiments
5. related and future work
related work:

- the operator emerges from the definition of the value function (Todorov 2007)
- minimizing for softmax temporal consistency (Nachum et al. 2017)
- soft Q-learning using mellowmax (Fox et al. 2016)
- a general framework for entropy regularizing MDPs (Neu et al. 2017)
- non-expansion and function approximation (Gordon 2001)
- more discussions on the properties of Boltzmann (Gao and Pavel 2017)

future work:

- an investigation of the bias in Q-learning under the operator
related work:

- the operator emerges from the definition of the value function (Todorov 2007)
- minimizing for softmax temporal consistency (Nachum et al. 2017)
- soft Q-learning using mellowmax (Fox et al. 2016)
- a general framework for entropy regularizing MDPs (Neu et al. 2017)
- non-expansion and function approximation (Gordon 2001)
- more discussions on the properties of Boltzmann (Gao and Pavel 2017)

future work:

- an investigation of the bias in Q-learning under the operator
related work:

- the operator emerges from the definition of the value function (Todorov 2007)
- minimizing for softmax temporal consistency (Nachum et al. 2017)
- soft Q-learning using mellowmax (Fox et al. 2016)
- a general framework for entropy regularizing MDPs (Neu et al. 2017)
- non-expansion and function approximation (Gordon 2001)
- more discussions on the properties of Boltzmann (Gao and Pavel 2017)

future work:

- an investigation of the bias in Q-learning under the operator
- more theory and experiments in function approximation setting
related work:

- the operator emerges from the definition of the value function (Todorov 2007)
- minimizing for softmax temporal consistency (Nachum et al. 2017)
- soft Q-learning using mellowmax (Fox et al. 2016)
- a general framework for entropy regularizing MDPs (Neu et al. 2017)
- non-expansion and function approximation (Gordon 2001)
- more discussions on the properties of Boltzmann (Gao and Pavel 2017)

future work:

- an investigation of the bias in Q-learning under the operator
- more theory and experiments in function approximation setting
- inverse RL by exploiting differentiability, convexity, and non-expansion
acknowledgements

- we thank George Konidaris

- and anonymous ICML reviewers for their outstanding feedback
The End