Improving Online Learning Through Knowledge Guided Exploration

D. Singh  A. Song  L. Padgham

School of Computer Science & Information Technology
RMIT University

Under review by the
International Conference on Machine Learning, 2009
Outline

1 Motivation
   ■ Performance Issues in Online Learning
   ■ Leveraging Knowledge

2 ProximityQ I
   ■ ProximityQ Algorithm
   ■ Experimentation
   ■ Results

3 ProximityQ II
   ■ ProximityQ with Dynamic Benefit
   ■ Experimentation
   ■ Results
Outline

1 Motivation
   - Performance Issues in Online Learning
   - Leveraging Knowledge

2 ProximityQ I
   - ProximityQ Algorithm
   - Experimentation
   - Results

3 ProximityQ II
   - ProximityQ with Dynamic Benefit
   - Experimentation
   - Results
The Learners Dilemma
Exploration vs. Exploitation

In a given environment state should the agent...

- **Exploit** current knowledge to select the action that gives the best return, or
- **Explore** a new action that could potentially give a higher return
$\epsilon$-greedy Performance
In Room Heating Scenario with Multiple Learners

Strategy is to **explore randomly** with probability $\epsilon$ else **exploit greedily** using the current knowledge.
Favourable case ($\textit{Heaters} = 2, |Q| = 36000, |E| = 50$): Solution is learnt but after a large number of samples.
Unfavourable case ($\text{Heaters} = 10, |Q| = 36000, |E| = 50)$: Solution is too hard for $\epsilon$-greedy.

\begin{center}
\begin{tikzpicture}
\begin{axis}[
    title={ProximityQ Experimentation Results},
    xlabel={Episodes},
    ylabel={Success},
    xtick={0,100,200,300,400,500},
    ytick={0,0.2,0.4,0.6,0.8,1},
    legend entries={Heaters:02, Heaters:10},
    legend pos=north east
]
\addplot [color=green, dashed]
    coordinates {
        (0,0) (500,0.95)
    };
\addplot [color=red]
    coordinates {
        (0,0) (500,0.95)
    };
\end{axis}
\end{tikzpicture}
\end{center}
Motivation
In Room Heating Scenario with Multiple Learners

Improve learning performance in our multi-agent setting by leveraging knowledge in some way.

- Focus on online settings where experimentation can be costly.
- Improve early stages of learning where performance is particularly poor.
- Use prior knowledge (in this case prior learning) instead of learning from scratch.
Motivation
In Room Heating Scenario with Multiple Learners

Improve learning performance in our multi-agent setting by leveraging knowledge in some way.

- Focus on online settings where experimentation can be costly.
- Improve early stages of learning where performance is particularly poor.
- Use prior knowledge (in this case prior learning) instead of learning from scratch.
Motivation
In Room Heating Scenario with Multiple Learners

Improve learning performance in our multi-agent setting by leveraging knowledge in some way.

- Focus on online settings where experimentation can be costly.
- Improve early stages of learning where performance is particularly poor.
- Use prior knowledge (in this case prior learning) instead of learning from scratch.
Motivation

In Room Heating Scenario with Multiple Learners

Improve learning performance in our multi-agent setting by leveraging knowledge in some way.

- Focus on online settings where experimentation can be costly.
- Improve early stages of learning where performance is particularly poor.
- Use prior knowledge (in this case prior learning) instead of learning from scratch.
Learn by imitation: E.g. using knowledge obtained from observing a mentor [Price & Boutilier, JAIR 2003].

Learn from advise: E.g. action preferences using if-then type rules [Maclin et al., NCAI 2005].

Learn from past experiences: E.g. Exploit past policies for action selection [Fernández & Veloso, AAMAS 2006].

Transfer learning: E.g. From a simpler related source task to the target task [Taylor et al., JMLR 2007].
Learn by imitation: E.g. using knowledge obtained from observing a mentor [Price & Boutilier, JAIR 2003].

Learn from advise: E.g. action preferences using if-then type rules [Maclin et al., NCAI 2005].

Learn from past experiences: E.g. Exploit past policies for action selection [Fernández & Veloso, AAMAS 2006].

Transfer learning: E.g. From a simpler related source task to the target task [Taylor et al., JMLR 2007].
Learn by imitation: E.g. using knowledge obtained from observing a mentor [Price & Boutilier, JAIR 2003].

Learn from advise: E.g. action preferences using if-then type rules [Maclin et al., NCAI 2005].

Learn from past experiences: E.g. Exploit past policies for action selection [Fernández & Veloso, AAMAS 2006].

Transfer learning: E.g. From a simpler related source task to the target task [Taylor et al., JMLR 2007].
Related Work
In Knowledge Reuse

- Learn by imitation: E.g. using knowledge obtained from observing a mentor [Price & Boutilier, JAIR 2003].
- Learn from advise: E.g. action preferences using if-then type rules [Maclin et al., NCAI 2005].
- Learn from past experiences: E.g. Exploit past policies for action selection [Fernández & Veloso, AAMAS 2006].
- Transfer learning: E.g. From a simpler related source task to the target task [Taylor et al., JMLR 2007].
We propose to improve learning performance by making the exploration effort more productive.

- Use **directed randomness** i.e. apply random exploration selectively to interesting areas of a large search space.
- Use **prior knowledge** to determine interesting areas for directed exploration.
We propose to improve learning performance by making the exploration effort more productive.

- Use directed randomness i.e. apply random exploration selectively to interesting areas of a large search space.
- Use prior knowledge to determine interesting areas for directed exploration.
We propose to improve learning performance by making the exploration effort more productive.

- Use **directed randomness** i.e. apply random exploration selectively to interesting areas of a large search space.
- Use **prior knowledge** to determine interesting areas for directed exploration.
1 Motivation
- Performance Issues in Online Learning
- Leveraging Knowledge

2 ProximityQ I
- ProximityQ Algorithm
- Experimentation
- Results

3 ProximityQ II
- ProximityQ with Dynamic Benefit
- Experimentation
- Results
Q-Learning
The Action-Value Function using Reinforcement Learning

\[ Q^h(x, u) \]

<table>
<thead>
<tr>
<th>( u_{g-1} )</th>
<th>( u_g )</th>
<th>( u_{g+1} )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( x_{i-1} )</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( x_i )</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( x_{i+1} )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Action-value function \( Q^h(x, u) \) defines for each state \( x \in X \) and action \( u \in U \) the expected reward for performing action \( u \) in state \( x \) using policy \( h \).

\[ u_g = h_{\text{greedy}}(x_i) \]
ProximityQ
A Knowledge-Guided Exploration Strategy

$Q^h(x, u)$

<table>
<thead>
<tr>
<th>$u_{g-1}$</th>
<th>$u_g$</th>
<th>$u_{g+1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_{i-1}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_i$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_{i+1}$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The ProximityQ idea is that if action $u_g$ worked well in state $x_i$ in the past, then it may be beneficial to explore a new action $u'$ that has some likeness to $u_g$ when revisiting the same state $x_i$ in the new task.

$u_g = h_{\text{greedy}}(x_i)$
ProximityQ
A Knowledge-Guided Exploration Strategy

The ProximityQ idea is that if action $u_g$ worked well in state $x_i$ in the past, then it may be beneficial to explore a new action $u'$ that has some likeness to $u_g$ when revisiting the same state $x_i$ in the new task.

Applicability limited to domains where actions can be ordered based on similarity.
ProximityQ replaces random exploration in $\epsilon$-greedy with a knowledge guided heuristic $h_{\text{proximityQ}}$ where $Q_{\text{past}}$ is the past policy used to bias the exploration and $d$ is the distance that defines the exploration space.
**ProximityQ**

**A Knowledge-Guided Exploration Strategy**

\[
Q^h(x, u) \begin{array}{|c|c|c|} 
\hline
u_{g-1} & u_g & u_{g+1} \\
\hline
x_{i-1} & & \\
\hline
x_i & & \\
\hline
x_{i+1} & & \\
\hline
\end{array}
\]

- **d** defines the exploration space about \(u_g\).

\[d(\epsilon) = (1 - \epsilon^2) \frac{|U|}{2}\]

- Strong bias early on when exploration is frequent (\(\epsilon \rightarrow 1\)).
- Increased random exploration later on as exploration becomes infrequent (\(\epsilon \rightarrow 0\)).
- Distance d updated every episode.
ProximityQ
A Knowledge-Guided Exploration Strategy

\[ Q^h(x, u) \]

- \( u_{g-1} \)
- \( u_g \)
- \( u_{g+1} \)

\[ d(\epsilon) = (1 - \epsilon^2) \frac{|U|}{2} \]

- \( d \) defines the exploration space about \( u_g \).

- **Strong bias early on when exploration is frequent** (\( \epsilon \to 1 \)).

- **Increased random exploration later on as exploration becomes infrequent** (\( \epsilon \to 0 \)).

- Distance \( d \) updated every episode.
ProximityQ
A Knowledge-Guided Exploration Strategy

\[ Q^h(x, u) \]

\[ u_{g-1} \quad u_g \quad u_{g+1} \]

\[ x_{i-1} \]

\[ x_i \]

\[ x_{i+1} \]

\[ u_g = h_{\text{greedy}}(x_i) \]

- **d** defines the exploration space about \( u_g \).

\[ d(\epsilon) = (1 - \epsilon^2) \frac{|U|}{2} \]

- Strong bias early on when exploration is frequent (\( \epsilon \to 1 \)).

- Increased random exploration later on as exploration becomes infrequent (\( \epsilon \to 0 \)).

- Distance \( d \) updated every episode.
Knowledge Guided Learning
Singh, Song, Padgham

Motivation
Performance Issues in Online Learning
Leveraging Knowledge

ProximityQ I
ProximityQ Algorithm
Experimentation
Results

ProximityQ II
ProximityQ with Dynamic Benefit
Experimentation
Results

Summary

ProximityQ
A Knowledge-Guided Exploration Strategy

\[ Q^h(x, u) \]

- \( u_{g-1} \)
- \( u_g \)
- \( u_{g+1} \)

- \( x_{i-1} \)
- \( x_i \)
- \( x_{i+1} \)

\[ u_g = h_{greedy}(x_i) \]

\[ d(\epsilon) = (1 - \epsilon^2) \frac{|U|}{2} \]

- \( d \) defines the exploration space about \( u_g \).

- **Strong bias early on when exploration is frequent** (\( \epsilon \rightarrow 1 \)).

- **Increased random exploration** later on as exploration becomes infrequent (\( \epsilon \rightarrow 0 \)).

- Distance \( d \) updated every episode.
ProximityQ
A Knowledge-Guided Exploration Strategy

\[ h_{\text{proximityQ}}(Q_{\text{past}}, Q_{\text{new}}, x, d) \]

1. \( u_g \leftarrow \text{new}(x) ? h_{\text{greedy}}(Q_{\text{past}}, x) : h_{\text{greedy}}(Q_{\text{new}}, x) \)
2. \( i_u \leftarrow \text{actionIndex}(U, u_g) \)
3. \( i_{\text{upper}} \leftarrow \min(|U|, \text{floor}(i_{u_g} + d)) \)
4. \( i_{\text{lower}} \leftarrow \max(1, \text{ceiling}(i_{u_g} - d)) \)
5. \( i_{u'} \leftarrow i_{\text{lower}} + \text{round}(\text{random}(i_{\text{upper}} - i_{\text{lower}})) \)
6. \( u' \leftarrow \text{action}(U, i_{u'}) \)
7. \text{return } u'
ProximityQ
A Knowledge-Guided Exploration Strategy

$h_{proximityQ}(Q_{past}, Q_{new}, x, d)$

1. $u_g \leftarrow \text{new}(x) ? h_{greedy}(Q_{past}, x) : h_{greedy}(Q_{new}, x)$
2. $i_{ug} \leftarrow \text{actionIndex}(U, u_g)$
3. $i_{upper} \leftarrow \min(|U|, \text{floor}(i_{ug} + d))$
4. $i_{lower} \leftarrow \max(1, \text{ceil}(i_{ug} - d))$
5. $i_u' \leftarrow i_{lower} + \text{round}(\text{random}(i_{upper} - i_{lower}))$
6. $u' \leftarrow \text{action}(U, i_u')$
7. return $u'$
ProximityQ
A Knowledge-Guided Exploration Strategy

\[ h_{\text{proximityQ}}(Q_{\text{past}}, Q_{\text{new}}, x, d) \]

\begin{align*}
1 & \quad u_g \leftarrow \text{new}(x) ? h_{\text{greedy}}(Q_{\text{past}}, x) : h_{\text{greedy}}(Q_{\text{new}}, x) \\
2 & \quad i_{ug} \leftarrow \text{actionIndex}(U, u_g) \\
3 & \quad i_{\text{upper}} \leftarrow \min(|U|, \text{floor}(i_{ug} + d)) \\
4 & \quad i_{\text{lower}} \leftarrow \max(1, \text{ceiling}(i_{ug} - d)) \\
5 & \quad i_{u'} \leftarrow i_{\text{lower}} + \text{round}(\text{random}(i_{\text{upper}} - i_{\text{lower}})) \\
6 & \quad u' \leftarrow \text{action}(U, i_{u'}) \\
7 & \quad \text{return } u' 
\end{align*}
ProximityQ
A Knowledge-Guided Exploration Strategy

\[ h_{\text{proximityQ}}(Q_{\text{past}}, Q_{\text{new}}, x, d) \]

1. \( u_g \leftarrow \text{new}(x) ? h_{\text{greedy}}(Q_{\text{past}}, x) : h_{\text{greedy}}(Q_{\text{new}}, x) \)
2. \( i_{ug} \leftarrow \text{actionIndex}(U, u_g) \)
3. \( i_{\text{upper}} \leftarrow \min(|U|, \text{floor}(i_{ug} + d)) \)
4. \( i_{\text{lower}} \leftarrow \max(1, \text{ceiling}(i_{ug} - d)) \)
5. \( i_u' \leftarrow i_{\text{lower}} + \text{round}(\text{random}(i_{\text{upper}} - i_{\text{lower}})) \)
6. \( u' \leftarrow \text{action}(U, i_u') \)
7. return \( u' \)
**ProximityQ**

A Knowledge-Guided Exploration Strategy

\[
h_{\text{proximityQ}}(Q_{\text{past}}, Q_{\text{new}}, x, d)
\]

1. \(u_g \leftarrow \text{new}(x) \ ? h_{\text{greedy}}(Q_{\text{past}}, x) : h_{\text{greedy}}(Q_{\text{new}}, x)\)
2. \(i_u \leftarrow \text{actionIndex}(U, u_g)\)
3. \(i_{\text{upper}} \leftarrow \min(|U|, \text{floor}(i_u + d))\)
4. \(i_{\text{lower}} \leftarrow \max(1, \text{ceiling}(i_u - d))\)
5. \(i_{u'} \leftarrow i_{\text{lower}} + \text{round}(\text{random}(i_{\text{upper}} - i_{\text{lower}}))\)
6. \(u' \leftarrow \text{action}(U, i_{u'})\)
7. \text{return } u'
ProximityQ
A Knowledge-Guided Exploration Strategy

\[ h_{proximityQ}(Q_{past}, Q_{new}, x, d) \]

1. \( u_g \leftarrow \text{new}(x) \ ? \ h_{greedy}(Q_{past}, x) : h_{greedy}(Q_{new}, x) \)
2. \( i_{ug} \leftarrow \text{actionIndex}(U, u_g) \)
3. \( i_{upper} \leftarrow \min(|U|, \text{floor}(i_{ug} + d)) \)
4. \( i_{lower} \leftarrow \max(1, \text{ceiling}(i_{ug} - d)) \)
5. \( i_{u'} \leftarrow i_{lower} + \text{round}(\text{random}(i_{upper} - i_{lower})) \)
6. \( u' \leftarrow \text{action}(U, i_{u'}) \)
7. \( \text{return } u' \)
**ProximityQ**
A Knowledge-Guided Exploration Strategy

\[ h_{\text{proximityQ}}(Q_{\text{past}}, Q_{\text{new}}, x, d) \]

```
1  \( u_g \leftarrow \text{new}(x) \oplus h_{\text{greedy}}(Q_{\text{past}}, x) : h_{\text{greedy}}(Q_{\text{new}}, x) \)
2  \( i_{u_g} \leftarrow \text{actionIndex}(U, u_g) \)
3  \( i_{\text{upper}} \leftarrow \min(|U|, \text{floor}(i_{u_g} + d)) \)
4  \( i_{\text{lower}} \leftarrow \max(1, \text{ceiling}(i_{u_g} - d)) \)
5  \( i_{u'} \leftarrow i_{\text{lower}} + \text{round}(\text{random}(i_{\text{upper}} - i_{\text{lower}})) \)
6  \( u' \leftarrow \text{action}(U, i_{u'}) \)
7  \text{return } u' 
```
Room with multiple heaters acting concurrently so that each impacts the learning of others.

Each heater $a_i$ has knowledge $Q_{past}^i$ from prior learning in the room alone.

Task is to each learn $Q_{new}^i$ such that the joint policies achieve the desired temperature range.

Actions represented as $\pm t$ where $\pm$ indicates on/off state and $t$ is time.

No communication between agents.
Experimentation
Room Heating Scenario

- Room with multiple heaters acting concurrently so that each impacts the learning of others.
- Each heater $a_i$ has knowledge $Q^i_{past}$ from prior learning in the room alone.
- Task is to each learn $Q^i_{new}$ such that the joint policies achieve the desired temperature range.
- Actions represented as $\pm t$ where $\pm$ indicates on/off state and $t$ is time.
- No communication between agents.
Room with multiple heaters acting concurrently so that each impacts the learning of others.

Each heater $a_i$ has knowledge $Q_{past}^i$ from prior learning in the room alone.

Task is to each learn $Q_{new}^i$ such that the joint policies achieve the desired temperature range.

Actions represented as $\pm t$ where $\pm$ indicates on/off state and $t$ is time.

No communication between agents.
Experimentation
Room Heating Scenario

- Room with multiple heaters acting concurrently so that each impacts the learning of others.
- Each heater $a_i$ has knowledge $Q_{past}^i$ from prior learning in the room alone.
- Task is to each learn $Q_{new}^i$ such that the joint policies achieve the desired temperature range.
- Actions represented as $\pm t$ where $\pm$ indicates on/off state and $t$ is time.
- No communication between agents.
Room with multiple heaters acting concurrently so that each impacts the learning of others.

Each heater $a_i$ has knowledge $Q^i_{past}$ from prior learning in the room alone.

Task is to each learn $Q^i_{new}$ such that the joint policies achieve the desired temperature range.

Actions represented as $\pm t$ where $\pm$ indicates on/off state and $t$ is time.

No communication between agents.
Conducted experiments highlighting two extreme cases for knowledge reuse.

Experiments repeated 10 times to eliminate randomisation effects.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Heater</th>
<th>Knowledge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Favourable</td>
<td>02</td>
<td>Strong</td>
</tr>
<tr>
<td>Unfavourable</td>
<td>10</td>
<td>Poor</td>
</tr>
</tbody>
</table>

[\[T_{hi} \ldots T_{lo}\]]
Conducted experiments highlighting two extreme cases for knowledge reuse.

Experiments repeated 10 times to eliminate randomisation effects.
Favourable case (2 heaters + strong knowledge): Improvement is $+55.89\%$ @ 50, $+5.56\%$ @ 500 episodes.
Unfavourable case (10 heaters + poor knowledge):
Improvement is +91.90% @ 50, +94.67% @ 500 episodes.
Results Summary
ProximityQ vs. $\epsilon$-greedy

- Comparison in two experiments (repeated 10 times), favourable and unfavourable to knowledge reuse. In both cases ProximityQ outperformed $\epsilon$-greedy.
  - Favourable case (2 heaters + strong knowledge): ProximityQ outperforms but shows performance loss in early episodes.
  - Unfavourable case (10 heaters + poor knowledge): ProximityQ finds a solution even with poor knowledge where $\epsilon$-greedy fails altogether.
Comparison in two experiments (repeated 10 times), favourable and unfavourable to knowledge reuse. In both cases ProximityQ outperformed $\epsilon$-greedy.

Favourable case (2 heaters + strong knowledge): ProximityQ outperforms but shows performance loss in early episodes.

Unfavourable case (10 heaters + poor knowledge): ProximityQ finds a solution even with poor knowledge where $\epsilon$-greedy fails altogether.
Comparison in two experiments (repeated 10 times), favourable and unfavourable to knowledge reuse. In both cases ProximityQ outperformed $\epsilon$-greedy.

Favourable case (2 heaters + strong knowledge): ProximityQ outperforms but shows performance loss in early episodes.

Unfavourable case (10 heaters + poor knowledge): ProximityQ finds a solution even with poor knowledge where $\epsilon$-greedy fails altogether.
Aim is to overcome the **over-exploration** seen in early episodes due to a rigid exploration strategy.
Idea is to dynamically adjust exploration based on the effective benefit of applied bias.

This benefit $s$ is taken to be the measured success at each episode such that $s \rightarrow 1$ implies effective bias while $s \rightarrow 0$ implies otherwise.

The fixed strategy $d(\epsilon)$ is then replaced by a dynamic exploration strategy $d(s, \epsilon)$ given benefit $s$.

$$d(s, \epsilon) = \left[ (1 - \epsilon^{1-s} \cdot p) \cdot s \cdot q \right] \frac{|U|}{2}$$

If benefit $\rightarrow 1$ then exploration $\rightarrow$ greedy

If benefit $\rightarrow 0$ then exploration $\rightarrow$ random
Idea is to dynamically adjust exploration based on the effective benefit of applied bias.

This benefit $s$ is taken to be the measured success at each episode such that $s \rightarrow 1$ implies effective bias while $s \rightarrow 0$ implies otherwise.

The fixed strategy $d(\epsilon)$ is then replaced by a dynamic exploration strategy $d(s, \epsilon)$ given benefit $s$.

$$d(s, \epsilon) = \left[ (1 - \epsilon(1-s)p)q \right] \frac{|U|}{2}$$

If benefit $\rightarrow 1$ then exploration $\rightarrow$ greedy
If benefit $\rightarrow 0$ then exploration $\rightarrow$ random
Idea is to dynamically adjust exploration based on the effective benefit of applied bias.

This benefit $s$ is taken to be the measured success at each episode such that $s \to 1$ implies effective bias while $s \to 0$ implies otherwise.

The fixed strategy $d(\epsilon)$ is then replaced by a dynamic exploration strategy $d(s, \epsilon)$ given benefit $s$.

$$d(s, \epsilon) = \left[ (1 - \epsilon(1-s)p)q \right] \frac{|U|}{2}$$

If benefit $\to 1$ then exploration $\to$ greedy
If benefit $\to 0$ then exploration $\to$ random
Idea is to dynamically adjust exploration based on the effective benefit of applied bias.

This benefit $s$ is taken to be the measured success at each episode such that $s \rightarrow 1$ implies effective bias while $s \rightarrow 0$ implies otherwise.

The fixed strategy $d(\epsilon)$ is then replaced by a dynamic exploration strategy $d(s, \epsilon)$ given benefit $s$.

$$d(s, \epsilon) = \left[ (1 - \epsilon^{1-s}) \cdot p \right] \cdot q \cdot \frac{|U|}{2}$$

If benefit $\rightarrow 1$ then exploration $\rightarrow$ greedy
If benefit $\rightarrow 0$ then exploration $\rightarrow$ random
**Experimentation**

**Room Heating Scenario**

- **Scenario**
  - Favourable: 02, Strong
  - Unfavourable: 10, Poor
  - Typical: 10, Strong

- **Equation**:
  \[
  [T_{hi} \ldots T_{lo}]
  \]

- **Diagram**:
  - $Q_{past}$
  - $Q_{new}$
  - $u_k$
  - $u_n$
  - $u_g$

- **Text**:
  - **Same setup as before** where task is to each learn $Q_{new}^i$ such that the joint policies achieve the desired temperature range.
  - **Added a new typical case** with strong prior knowledge in a loosely related task.
Experimentation
Room Heating Scenario

- Same setup as before where task is to each learn $Q^i_{new}$ such that the joint policies achieve the desired temperature range.
- Added a new typical case with strong prior knowledge in a loosely related task.
Favourable case: Improvement is +71.95% @ 50 (was +55.89%), +24.97% @ 500 (was +5.56%) episodes.
Results
ProximityQ with Dynamic Benefit vs. $\epsilon$-greedy

**Unfavourable case:** Improvement is $+87.65\%$ @ 50 (was $+91.90\%$), $+97.56\%$ @ 500 (was $+94.67\%$) episodes.

![Graph showing comparison between $\epsilon$-greedy and ProximityQ over episodes](image)
Results
ProximityQ with Dynamic Benefit vs. PRQ-Learning

Favourable case (2 heaters + strong knowledge): Improvement is +12.37% @ 50, +23.30% @ 500 episodes.
Unfavourable case (10 heaters + poor knowledge): Improvement is +58.64% @ 50, +36.03% @ 500 episodes.
**Results**
ProximityQ with Dynamic Benefit vs. PRQ-Learning

**Typical case (10 heaters + strong knowledge):**
Improvement is $+41.03\% @ 50$, $+25.19\% @ 500$ episodes.
Results Summary

ProximityQ with Dynamic Benefit

- Conducted three experiments (repeated 10 times), favourable, unfavourable and typical for knowledge reuse.

- In all cases ProximityQ outperforms $\epsilon$-greedy (as before) and further improves performance by eliminating the over-exploration seen earlier.

- ProximityQ also outperforms PRQ-Learning, a state-of-the-art knowledge-reuse algorithm in all experiments.
Conducted three experiments (repeated 10 times), favourable, unfavourable and typical for knowledge reuse.

In all cases ProximityQ outperforms $\epsilon$-greedy (as before) and further improves performance by eliminating the over-exploration seen earlier.

ProximityQ also outperforms PRQ-Learning, a state-of-the-art knowledge-reuse algorithm in all experiments.
Conducted three experiments (repeated 10 times), favourable, unfavourable and typical for knowledge reuse.

In all cases ProximityQ outperforms $\epsilon$-greedy (as before) and further improves performance by eliminating the over-exploration seen earlier.

ProximityQ also outperforms PRQ-Learning, a state-of-the-art knowledge-reuse algorithm in all experiments.
Knowledge-guided exploration is a beneficial strategy for improving online performance in Reinforcement Learning.

ProximityQ provides superior performance to state-of-the-art knowledge-reuse methods (like PRQ-Learning) in domains where actions can be ordered.
References

J. Li, G. Poulton, G. James
Agent-Based Distributed Energy Management

F. Fernández, M. Veloso
Probabilistic policy reuse in a reinforcement learning agent
The Fifth International Joint Conference on Autonomous Agents and Multiagent Systems, 720–727, 2006. ACM.