Empirical Evaluation for Theorem 1 in "Eliciting User Preferences for Personalized Multi-Objective Decision Making through Comparative Feedback"

Anonymous Author(s)
Affiliation
Address
email

Abstract

This project tests the empirical performance of the algorithm proposed by Shao et al. [2023] in order to find optimal policy for users with unkown preferences through comparative feedback in a Multi-Objective Reinforcement Learning setting. The number of comparative feedback required in practice for user preference estimation meets the theoretical prediction. The optimality gap to the best personalized value confirms the theoretical statement. However, for a simple planning problem, the provided theoretical upper bound for the absolute performance gap is too conservative. The relative performance gap is a better performance evaluation metric. In the worst case scenario, the relative performance optimality gap reaches 80%. This poor performance might be caused by the poor preference estimation. Therefore, it is necessary to provide worst case analysis for the relative performance gap and improve user preference estimation.

1 Introduction

Many real world problems require people to make sequential decisions that balance multiple but sometimes conflicting objectives. For example, in the area of autonomous driving, safety, speed, and comfort are all desired objectives, while speed could negatively impact safety. In classical reinforcement learning, the reward is a scalar that combines several objectives in an arbitrary way. However, different users might have different preferences. Thus, it is important to extend the scalar reward into a vector and design an optimal personalized policy for a given user using as few comparative feedback from them as possible. This paper Shao et al. [2023] provides a provably efficient algorithm to estimate user's personalized policy in a tabular setting. As stated in *Theorem 1* Shao et al. [2023], given a planning problem, we are able to estimate the personalized optimal value by user's pairwise feedback, with accuracy $O((\sqrt{K}+1)^{d+\frac{14}{3}}\epsilon^{\frac{1}{3}})$ and $O(Klog(\frac{K}{\epsilon}))$ number of queries, where K is the number of objective, d is the rank of state value matrix, and ϵ is user's comparison distinguishablity. However, there is no simulation study in the paper to evaluate the empirical performance of the algorithm. This might cast doubt on the the algorithm's practicality. Therefore, this project aims to bridge the theory-practice gap by adding reinforcement learning experiments to test the algorithm's empirical performance.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

2 Methodology

2.1 Problem Setup

If the decision problem has K objectives to consider, then a user is characterized by its preference vector $\omega^* \in \mathcal{R}^K$, $\|\omega^*\|_1 = 1$. The policy $\pi : |S||A| \rightarrow \Delta_K$ is a mapping from the state action pair to a discrete distribution vector with size K. The state value function of policy π starting from initial state s_0 is $V^\pi(s_0) \in R^K$. $V^\pi = E_{s_0 \sim \rho}[V^\pi(s_0)] \in \mathcal{R}^K$, ρ is the initial state distribution. The personalized value of policy π is $< \omega^*, V^\pi > \in \mathcal{R}^+$. The optimal policy $\pi^* := \arg\max_{\pi \in \Pi} < \omega^*, V^\pi >$, and its corresponding optimal personalized value $\nu^* :=< \omega^*, V^{\pi^*} >$.

2.2 Preference Estimation Algorithm

To estimate ω^*, we need three steps, identification of basis policy (see Algorithm 1), computation of basis ratios (see Algorithm 2), and solving a linear system $\hat{A}\hat{\omega} = e_1$ (see Equation 1). Then the estimated optimal policy is $\pi^{\hat{\omega}} = \arg\max_{\pi \in \Pi} < \hat{\omega}, V^\pi >$.

Initialize $\pi^{e^*} \leftarrow \pi^{e_1}$
for $j = 2, \ldots, k$ **do**
  compare π^{e^*} and π^{e_j}
  if $\pi^{e_j} > \pi^{e^*}$ then $\pi^{e^*} \leftarrow \pi^{e_j}$
end
$\pi_1 \leftarrow \pi^{e^*}$ and $u_1 \leftarrow \frac{V^{\pi^{e^*}}}{\|V^{\pi^{e^*}}\|_2}$
for $i = 2, \ldots, k$ **do**
  Arbitrarily pick an orthonormal basis $\rho_1, \ldots, \rho_{k+1-i}$ of span$(V^{\pi_1}, \ldots, V^{\pi_{i-1}})^\perp$
  $j_{\max} \leftarrow \arg\max_{j \in [k+1-i]} \max(|\nu^{\rho_j}|, |\nu^{-\rho_j}|)$
  if $\max(|\nu^{\rho_{j\max}}|, |\nu^{-\rho_{j\max}}|)$ **then**
    $\pi_i \leftarrow \pi^{\rho_{j\max}}$ if $|\nu^{\rho_{j\max}}| > |\nu^{-\rho_{j\max}}|$ else $\pi_i \leftarrow \pi^{-\rho_{j\max}}$. $u_i \leftarrow \rho_{j\max}$
  end
  else output $(\pi_1, \pi_2, \ldots), (u_1, u_2, \ldots)$ and **stop**
end

Algorithm 1: Identification of Basis Policies

$$\hat{A} = \begin{bmatrix} V^{\pi_1 T} \\ (\hat{\alpha}_1 V^{\pi_1} - V^{\pi_2})^T \\ \cdots \\ (\hat{\alpha}_{d-1} V^{\pi_1} - V^{\pi_d})^T \end{bmatrix} \tag{1}$$

2.3 Multi-Objective Reinforcement Learning (MORL) Environment

I extend the Gridworld environment used in HW4 into a MOLR environment. Firstly, I remove the red-colored self-absorbing states with rewards -10. Secondly, each time, for user with K objectives, I randomly sample K cells from the grids (corresponds to $[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 15, 17, 18, 19, 20, 21, 22, 23, 24]$) except the obstacle states. For each sampled K cell, I set the reward as 1, to satisfy the bounded reward assumption, $r_t \in [0, 1]$. I make it as a termination state and adjust the transition matrix T accordingly. I make each cell correspond to user's objective. The reward function in MORL setting is a $K \times |S| \times |S| \times |A|$ matrix. In addition, one key assumption in the paper Shao et al. [2023] is the existence of "doing nothing" policy π_0 and its corresponding value function $V^0 = 0$. To be consistent with this assumption, I add "STAY" action in addition to the previous four actions, "Move North", "Move South", "Move East", and "Move West". Therefore, $|A| = 5$. Taking "STAY" action will always receive reward 0. The noise structure in state transition is the same as the setting in HW4.

Input: $(V^{\pi_1}, \dots V^{\pi_d})$ and $C_\alpha = 2K$ (see Lemma 1 Shao et al. [2023]), $\hat{\alpha_i} \in [0, C_\alpha], \forall i$
for $i = 1, \dots d-1$ **do**
  let $l = 0, h = 2C_\alpha, \hat{\alpha_i} = C_\alpha$
  while ***True*** **do**
    if $\hat{\alpha_i} > 1$ **then**
      compare π_1 and $\frac{1}{\hat{\alpha_i}}\pi_{i+1} + (1 - \frac{1}{\hat{\alpha_i}})\pi_0$. **If** $\pi_1 > \frac{1}{\hat{\alpha_i}}\pi_{i+1} + (1 - \frac{1}{\hat{\alpha_i}})\pi_0$, **then** $h \leftarrow \hat{\alpha_i}, \hat{\alpha_i} \leftarrow \frac{l+h}{2}$. **If** $\pi_1 < \frac{1}{\hat{\alpha_i}}\pi_{i+1} + (1 - \frac{1}{\hat{\alpha_i}})\pi_0$, **then** $l \leftarrow \hat{\alpha_i}, \hat{\alpha_i} \leftarrow \frac{l+h}{2}$
    end
    else compare π_{i+1} and $\hat{\alpha_1}\pi_1 + (1 - \hat{\alpha_i})\pi_0$. **If** $\hat{\alpha_1}\pi_1 + (1 - \hat{\alpha_i})\pi_0 > \pi_{i+1}$, **then** $h \leftarrow \hat{\alpha_i}, \hat{\alpha_i} \leftarrow \frac{l+h}{2}$. **If** $\hat{\alpha_1}\pi_1 + (1 - \hat{\alpha_i})\pi_0 < \pi_{i+1}$, then $l \leftarrow \hat{\alpha_i}, \hat{\alpha_i} \leftarrow \frac{l+h}{2}$.
    if *Indistinguishable* **then**
      break
    end
  end
end
Output $\hat{\alpha_1}, \dots \hat{\alpha_{d-1}}$

Algorithm 2: Computations of Basis Ratios

2.4 Policy Evaluation & Value Iteration

Implementing the algorithm requires the knowledge of reinforcement learning. To compare two policies, for example, π_1 and $\frac{1}{\hat{\alpha_i}}\pi_{i+1} + (1 - \frac{1}{\hat{\alpha_i}})\pi_0$, firstly we need to estimate V^{π_1} and $V^{\frac{1}{\hat{\alpha_i}}\pi_{i+1} + (1 - \frac{1}{\hat{\alpha_i}})\pi_0}$. To compute V^π, I use *Policy Evaluation* (see Algorithm 3) and assume initial state following a uniform distribution. Secondly, based on the estimated state value function, user returns "indistinguishable " if $| < \omega^*, V^{\pi_1} > - < \omega^*, V^{\frac{1}{\hat{\alpha_i}}\pi_{i+1} + (1 - \frac{1}{\hat{\alpha_i}})\pi_0} > | < \epsilon$. Otherwise, user will choose the policy with higher personalized value. $\pi_1, \dots \pi_d$ are greedy policies with respect to $V^{\pi_1}, \dots, V^{\pi_d}$ accordingly. They are jointly estimated by *Value Iteration* (see Algorithm 4).

Input: $\theta > 0$ tolerance parameter, γ discount factor, π, policy to evaluation
Initialize $V(s)$ arbitrarily, with $V(\text{terminal}) = 0$
Repeat:
$\Delta \leftarrow 0$
for $s \in S$ **do**
  $v \leftarrow V(s)$
  $V(s) \leftarrow \sum_{a \in A(s)} \pi(a|s) \sum_{s',r} p(s', r|s, a)(r + \gamma V(s'))$
  $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
end
Until $\Delta < \theta$
Output: $V^\pi(s), \forall s$

Algorithm 3: Policy Evaluation

Input: $\theta > 0$ tolerance parameter, γ discount factor
Initialize $V(s)$ arbitrarily, with $V(\text{terminal}) = 0$
Repeat:
$\Delta \leftarrow 0$
for $s \in S$ **do**
  $v \leftarrow V(s)$
  $V(s) \leftarrow \max_{a \in A(s)} \sum_{s',r} p(s', r|s, a)(r + \gamma V(s'))$
  $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
end
Until $\Delta < \theta$
Output: $V(s), \forall s$ and π: greedy policy w.r.t. $V(s)$

Algorithm 4: Value Iteration

Parameter	θ	γ	noise	ϵ	K	rep
Description	tolerance threshold	discounting factor	-	indistinguishablility	no.of objectives	-
Value	0.01	0.99	0.1	0.01	$[3, 10]$	5

Table 1: Experiment Parameters

2.4.1 Experiment Setup

3 Result & Discussion

3.1 Sanity Check

In Figure 1, we set $K = 3$. The user has three objectives, going to the green cell, going to the blue cell, and going to the yellow cell. The left column shows state value estimates ($V^{\pi T}(s)\omega, \forall s \in S$) and corresponding optimal actions in the MORL setting when we input the preference vector $[1, 0, 0], [0, 1, 0], [0, 0, 1]$. This corresponds to prioritizing going to the green cell, the blue cell, and the yellow cell. The right column shows state value estimates and associated optimal actions in the GridWorld setting with single objective and scalar reward. We can see two settings having the same estimates and optimal actions. This confirms the correctness of *Value Iteration* implementation in the MORL setting. The last subplot in Figure 1 shows the state value estimates of "doing nothing" policy. In the implementation, $\pi_0 \in \mathcal{R}^{|S| \times |A|}, \pi_0[:, \text{STAY}] = 1$. This policy is supposed to have $V_0 = 0$ and this is what we get. This confirms the correctness of *Policy Evaluation* implementation in the MORL setting.

3.2 Performance Evaluation

We want to evaluate the empirical performance of Theorem 1. The full statement Shao et al. [2023] is

Theorem 1 *Consider the algorithm of computation $\hat{A}$ defined in Eq(1) and any solution $\hat{\omega}$ to $\hat{A}x = e_1$ and outputting the policy $\pi^{\hat{\omega}} = \arg\max_{\pi \in \Pi} < \hat{\omega}, V^{\pi} >$, which is the optimal personalized policy for preference vector $\hat{\omega}$. Then the output policy $\pi^{\hat{\omega}}$ satisfying that $\nu^* - < \omega^*, V^{\pi^{\hat{\omega}}} > \leq O((\sqrt{K}+1)^{d+\frac{14}{3}}\epsilon^{\frac{1}{3}})$ by using $O(K log(\frac{K}{\epsilon}))$ comparison queries.*

In Figure 2, the red dotted line plots the theoretical upper bound $O(K \log(\frac{K}{\epsilon}))$ of the number of comparison queries required to estimate ω^*. The black dotted line shows the average number of comparison queries required in 5 repeated trials for each $K \in [3, 10]$ simulated under different seeds. The shaded blue region indicates the minimum and maximum number of comparison queries for each K. None of the required number of comparison queries exceeds the theoretical upper bound. Practice meets theory. Since the maximum number of required queries hits the upper bound when $K = 5$, this suggests that the theoretical upper bound is tight.

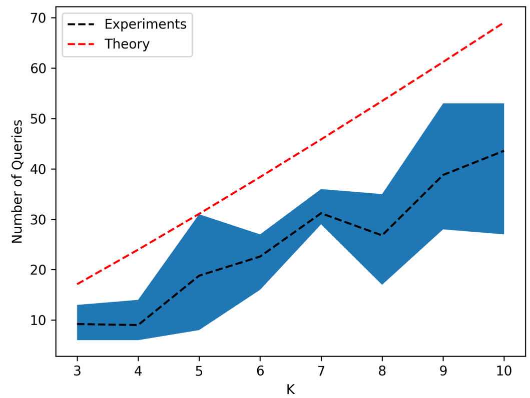

Figure 2: Simulation Results of Number of Comparison Queries Required in Practice to Estimate ω^*

Figure 3 shows the empirical performance of the absolute performance gap defined as $\nu^* - <\omega^*, V^{\pi^{\hat{\omega}}}>$. Similar to Figure 2, for each K, the black dotted line shows the average performance over 5 trials and the shaded blue region denotes the minimum and maximum. When $K = 3, d = 1$, the theory (see Theorem 1) predicts the upper bound as $(\sqrt{K} + 1)^{d+\frac{14}{3}} \epsilon^{\frac{1}{3}} = 64 \geq 1$. It is increasing in terms of K and d. Therefore, the absolute performance gap in practice automatically meets the theoretical statement for all K. However, we do not observe an obvious increase in the absolute performance gap in terms of K and d, as opposed to the theoretical prediction.

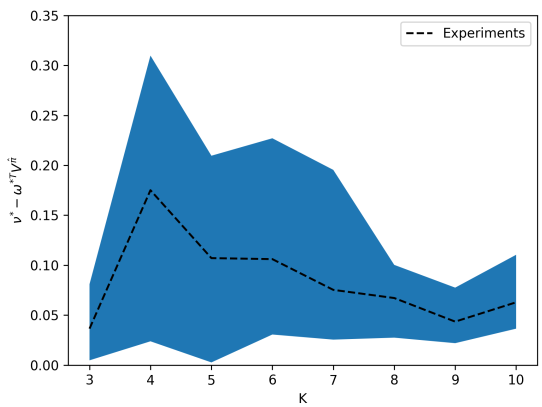

Figure 3: Simulation Results of Absolute Performance Gap in Practice

Since the implemented MORL is very simple, the theoretical upper bound for the absolute performance gap is too loose. In this case, the relative performance gap, $\frac{\nu^* - <\omega^*, V^{\pi^{\hat{\omega}}}>}{\nu^*}$ should be a better metric to evaluate the actual performance of the proposed algorithm. As Figure 4 suggests, even though in the best case, the optimality gap is around 10% to 20%, in the worst case, however, the relative performance gap could be as huge as 80%. This signifies the importance of understanding and bounding the relative performance gap in the worst case scenario, especially when we have simple planning problems.

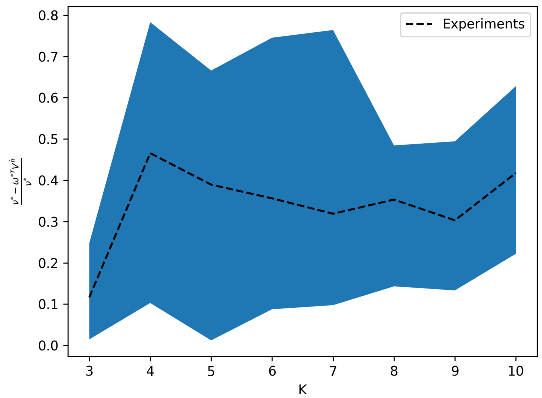

Figure 4: Simulation Results of Relative Performance Gap in Practice

3.3 Preference estimation

To understand which factor is associated with the relative performance gap, I plot the randomly generated true preference and the estimated preference for $K = 3$ (see Figure 5). Due to the space limit and human interpretability constraint, I omit showing plots for $K > 3$. These plots can be reproduced here. For trial number 0, the true preference is $[0.33, 0.33, 0.34]$, while the estimated preference is $[0, 0, 1]$. These two preference vectors look very different but they have very similar personalized values. This is because when $\omega > 0$, the three objectives going to green, blue, or yellow cells, are exchangeable. Situations get more complicated, however, when ω includes negative values, i.e., users prefer going to the yellow cell while avoiding going to the blue cell. Intuitively, we expect good sign alignment in each coordinate of ω^* and $\hat{\omega}$. Unfortunately, Table 2 shows that this conjecture is not always true. The worst case could happen when only 1 out of 5 signs aligns between the true preference and the estimated preference. Even though the personalized values between the true preference and the estimated preference could be similar, it is counter intuitive for a preference estimation with many "prefer / hate" flips to be a good estimation and has low optimality gap.

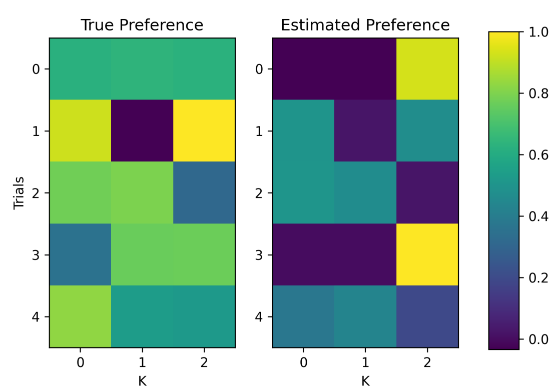

Figure 5: Compare True Preference and Estimated Preference

K	3	4	5	6	7	8	9	10
Mean	2.2	3.8	3.8	5.2	4.6	7.2	6.6	7.4
Min	1	3	1	4	3	5	4	7
Max	3	4	5	6	7	8	9	9

Table 2: Counts of Aligned Sign between True Preference & Estimated Preference

4 Conclusion

In conclusion, empirical assessment does not show any violation of *Theorem 1*. For the implemented simple MORL, the theoretical bound of the absolute performance gap is too loose. Instead, the relative performance gap is a better performance evaluation metric. In the worst case scenario, the relative performance optimality gap could be as large as 80%. This poor performance might be caused by the poor preference estimation assessed by the proportion of flipped signs. Based on these observations, I suggest Shao et al. [2023] to consider two additional analysis. Firstly, provides worst case analysis for the relative performance gap. Secondly, bounds the difference between $\hat{\omega}$ and ω^* by putting necessary restrictions on $\hat{\omega}$. If $d << K$, the solution space of $\hat{A}x = e_1$ might be very too large, which makes $\hat{\omega}$ too flexible to be true.

5 Limitations

This project has several limitations. First, the implemented MORL environment is almost the simplest tabular MDP. Given its simple structure, it might not be able to test the robustness of the $O((\sqrt{K}+1)^{d+\frac{14}{3}}\epsilon^{\frac{1}{3}})$ in the worse case, especially when H is large. In addition, randomly sampling K cells from the grid and setting their rewards as 1 is not a good way to model K different objectives. For the implemented environment, K objectives are exchangeable when $\omega > 0$. In reality, however, K objectives can conflict with each other and in general, they are not exchangeable.

References

H. Shao, L. Cohen, A. Blum, Y. Mansour, A. Saha, and M. R. Walter. Eliciting user preferences for personalized multi-objective decision making through comparative feedback, 2023.

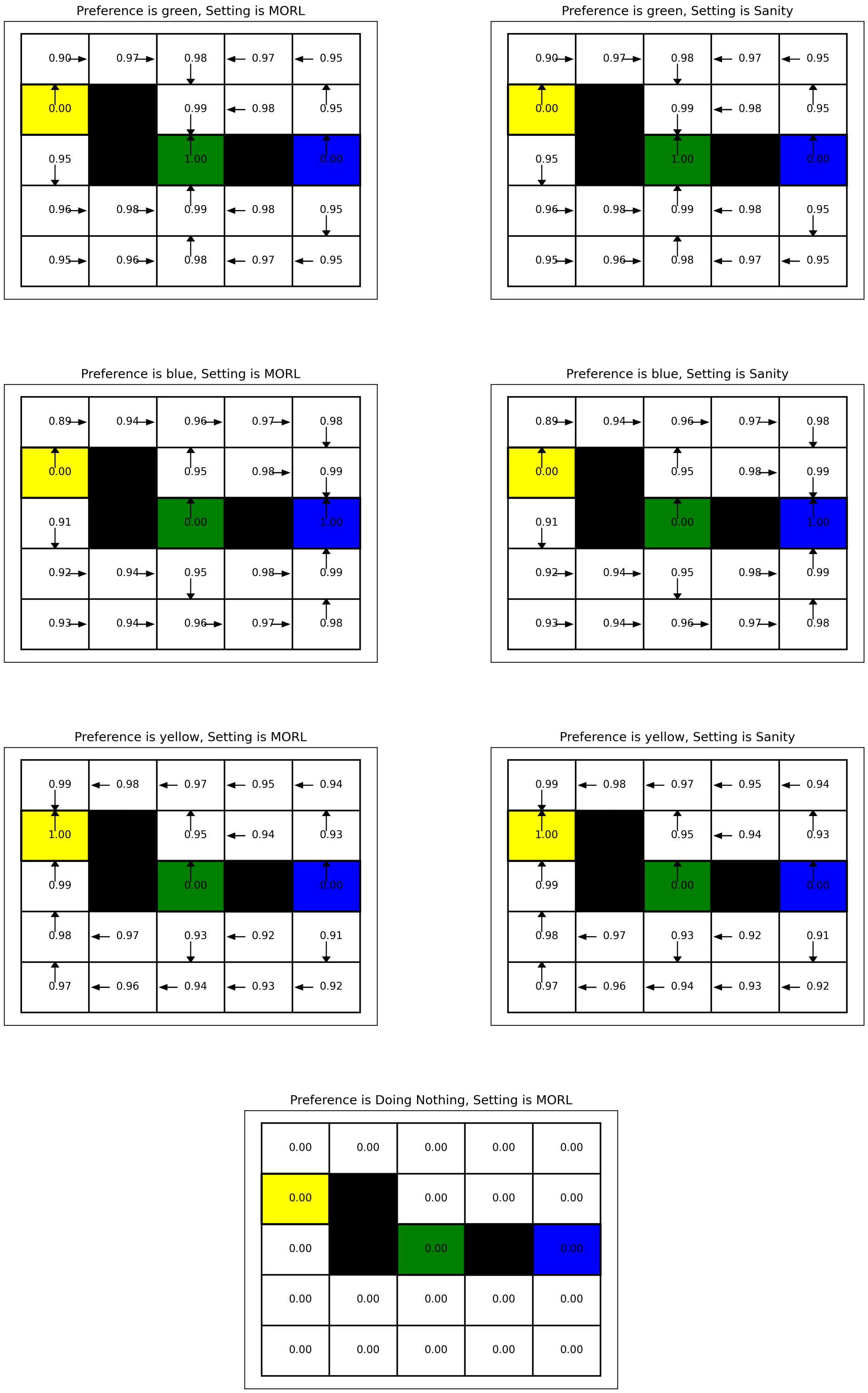

Figure 1: Sanity Check