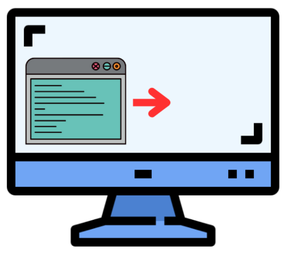
Adaptive Posture Regulation with Visual Stimuli

Imane Jennane, Oussama Jaffal, Simon Gmür

Introduction

In today's digital age, prolonged periods of sitting and working at a computer are common, leading to an increase in posture-related health issues. Poor posture can result in various musculoskeletal problems, including back pain, neck strain, and repetitive stress injuries. Addressing these concerns requires innovative solutions that promote better posture habits in an engaging and effective manner. This project aims to tackle this challenge by leveraging real-time posture monitoring and dynamic visual feedback to encourage users to maintain healthy sitting postures.



The core concept of the project involves using a sensor and posture estimation models to continuously observe a user's posture as they work on a screen.

The interactive element aiming to induce better posture: the window moves around the screen based on the user's current posture. The window serves as a visual stimulus designed to subtly prompt the user to correct their posture. For instance, if the user is slouching, the window might move in a direction that encourages them to sit up straighter to keep their focus on the screen. Conversely, if the user leans too far forward or backward, the window's movement might prompt them to adjust to a more neutral position.

The adaptive nature of this feedback mechanism provides ongoing, personalized guidance. This interactive feedback loop not only makes users more aware of their posture but also encourages the development of better habits over time.

The ultimate goal is to create a system that integrates seamlessly into a user's daily routine, promoting long-term health benefits without causing significant disruption to their work.

Project definition

Our contribution to the broader project involves a comprehensive approach to problem modeling, algorithm development and performance evaluation within a reinforcement learning framework. Following the provided guidelines, we began by modeling the problem of adaptive posture regulation and implementing multiple distinct versions of the control learning algorithms for comparison. The algorithms were designed to learn optimal policies for moving the visual stimulus (the window on the screen) based on real-time posture data.

We designed and implemented a robust simulation environment that mimics real-world scenarios where a user's posture is continuously monitored.

To realistically simulate the diverse behaviors of users interacting with the system, we had to generate simulated users with varying characteristics. Specifically, we created multiple artificial users with different probabilities of readjusting their posture in response to the visual stimulus and with different fatigue ratios that affect how quickly they become less responsive over time. These variations ensured that our simulation environment could accurately reflect the nonstationary and dynamic nature of real-world user interactions.

It is important to highlight that all the user data was entirely simulated, which, while perhaps seeming absurd, is necessary to rigorously test and develop the adaptive posture regulation algorithms in the absence of real-world data. This project can be understood as a preparatory step to the real-life experiences.

Design choices

Stationary vs Nonstationary

We chose a nonstationary model because human posture is not constant over time; it varies based on numerous factors such as fatigue, attention, and individual habits. As a result, the probability of changing posture will change over time, which reflects real human interactions with the environment. The fatigue is modeled as follows

$F(t) = 1 - e^{-\lambda t}$, where λ is the fatigue parameter, varying between different humans. This parameter is one of the characteristics that we will randomly generate, when simulating multiple different users.

This measure of fatigue is taken from the paper 'Incorporating human fatigue and recovery into the learning–forgetting process' [4].

Episodic vs Continuing

We opted for a continuing problem model, as we believe it imitates real-world scenarios where posture regulation is an ongoing process without a defined endpoint.

Stateful vs Stateless

As both approaches can be justified in our context, we decided to design both, stateful and stateless models, to be able to compare and conclude later on.

Continuous vs Event-triggered iterations

We defined an iteration as a continuous process, with each iteration lasting two minutes. This allows the system to regularly update and adjust based on user posture changes. In this context, we had to consider the effects of the action (reward) and induced delays.

Number of iterations (for learning convergence)

Regarding the number of iterations needed for convergence, we had to hypothesise about realistic conditions for the real-world application. We decided that in the real-world experiment, the agent will have to adapt to an individual user in 2-3 days, assuming two one hour sessions per day.

Noise

We simulated noise by adding a range of 5% to the probability each user has for changing posture, to account for errors in the sensors capturing the posture of the user.

Actions

Besides moving the window either up, down, left or right, we decided to add the no-action of maintaining the same position. Since each iteration is constantly a time-frame of two minutes and the users posture is not static in this time-frame, this no-action accounts for situations where no adjustment can be beneficial for the goal of changing the posture.

Control Learning approaches

The **Five-Armed Bandit Environment** simulates a user model with five possible actions: up, down, left, right, and stay. The reward structure is influenced by the user's posture, which changes with a specified probability that decreases over time due to fatigue. This environment models real-world scenarios where user behavior and response times change due to fatigue.

UCB (Upper Confidence Bound)

We implemented a **UCB (Upper Confidence Bound) Agent** to solve the bandit problem. UCB is a well-known algorithm that balances between exploration and exploitation approaches. The explore rate in UCB helps in efficiently exploring less-tried actions to avoid premature convergence to sub-optimal actions.

SARSA

SARSA is an on-policy algorithm that updates the action-value function, Q(s,a), based on the actions taken by the current policy. This method ensures that the learned policy is directly influenced by the exploration strategy, making it more sensitive to the policy's behavior.

Q-Learning

Q-Learning is an off-policy algorithm that learns the optimal action-value function, Q(s,a), by updating its Q-values based on the maximum estimated future rewards. This approach helps the agent to efficiently learn the best actions to maximize cumulative rewards.

Evaluation, Performance measures

UCB

The plots display the performance of UCB algorithm across multiple users. The average reward over time shows that the algorithm effectively learns to increase rewards, with most users achieving positive rewards after several iterations. The RMSE (Root Mean Square Error) plot indicates that the error stabilizes as the iterations progress, reflecting the algorithm's ability to improve its predictions and make better decisions over time.

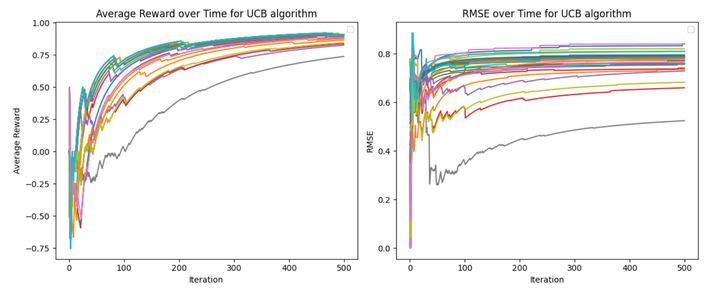


Figure 1. Average reward and RMSE of UCB algorithm

SARSA

The plots display the performance of the SARSA algorithm across multiple users. It is measured by analyzing the average reward and RMSE for different user profiles with varying probabilities and fatigue rates. The results in the plots show that the average reward fluctuates a lot over time, but the RMSE steadily decreases, indicating that the algorithm gets better at making accurate predictions despite the random changes in the environment.

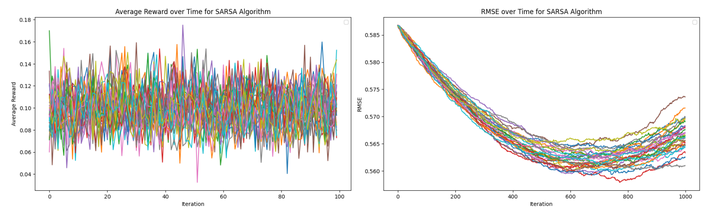


Figure 2. Average reward and RMSE of SARSA algorithm

Q-Learning

The evaluation of the Q-Learning algorithm is conducted through two performance metrics: average rewards and regret metrics over multiple iterations. In the given plot, the average rewards for all users tend to stabilize around zero after a certain number of iterations, indicating that the agent is able to balance exploration and exploitation effectively. The regret on the other hand varies for each user and doesn't stabilize, reflecting the stochastic nature of the learning process and the differences in the environments each user experiences.

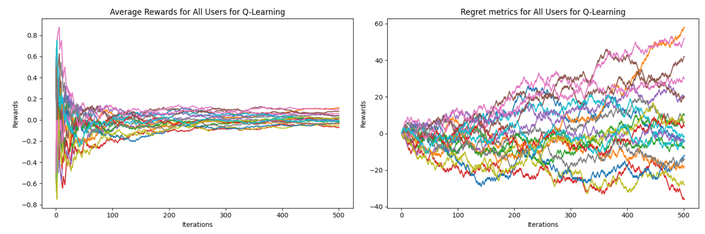


Figure 3. Average reward and regret metric of Q-Learning algorithm

Conclusion

In this Adaptive Posture Regulation with Visual Stimuli project, we faced challenges in implementation as we generated our own data to produce results for a concept not yet tested with real people. Despite these challenges, our work sets the stage for future use with actual participants.

References

- Tobias Baur Hannes Ritschel and Elisabeth Andre. Adapting a robot's linguistic style based on socially-aware reinforcement learning. *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2017.
- Michael Widrich Thomas Unterthiner Johannes Brandstetter Sepp Hochreiter Jose A. Arjona-Medina, Michael Gillhofer. Rudder: Return decomposition for delayed rewards. *LIT AI Lab, Institute for Machine Learning Johannes Kepler University Linz, Austria, Institute of Advanced Research in Artificial Intelligence (IARAI)*, 2019.
- Ilhan Aslan Florian Lingenfelser Elisabeth André Klaus Weber, Hannes Ritschel. How to shape the humor of a robot - social behavior adaptation based on reinforcement learning. *ICMI'18, October 16-20, 2018, Boulder, CO, USA*, 2018.
- W.P. Neumann M.Y. Jaber, Z.S. Givi. Incorporating human fatigue and recovery into the learning–forgetting process. *Department of Mechanical and Industrial Engineering, Ryerson University, Toronto, ON, Canada*, 2013.
- B. Bruno P. Dillenbourg Wang Chenyang, D. Carnieto Tozadore. Writeupright: Regulating children's handwriting body posture by unobstrusively error amplification via slow visual stimuli on tablets. *CHI Conference on Human Factors in Computing Systems Proceedings, Honolulu, HI, USA, May 11–16, 2024*, 2024.