Journal of Machine Learning Research 1 (2020) TBD Submitted 06/20; Published TBD

Testing RL with Planning for Demand Response in an Energy Social Game

Lucas Spangher LUCAS_SPANGHER@BERKELEY.EDU
Adam Bouyamourn ADAM.BOUYAMOURN@BERKELEY.EDU
Manan Khattar MKHATTAR@BERKELEY.EDU
Akaash Tawade AKAASHT@BERKELEY.EDU
Akash Gokul AKASHGOKUL@BERKELEY.EDU
Alex Devonport ALEX_DEVONPORT@BERKELEY.EDU
Costas Spanos SPANOS@BERKELEY.EDU
Division of Computer Science and Electrical Engineering
University of California
Berkeley, CA 94720-1776, USA

Editor: TBD

Abstract

While reinforcement learning on humans has shown incredible promise, it often suffers from a scarcity of data and few steps in the real world. In instances like this, a planning model of human behavior may greatly help. We present an experimental setup for the development and testing of two different RL architectures with several different neural architectures for planning models. Our RL architectures are Batch Constrained Q-Learning and Soft Actor Critic. Our primary planning models are various autoML frameworks and neural architectures. We present an experiment and preliminary simulation results by which we hope to verify the learning of the RL agents and the efficacy of the planning model.

Keywords: Demand Response, Energy, Social Games, Reinforcement Learning, Planning models

1. Introduction

1.1 Motivation in Energy (Background)

The California grid (CAISO) of 2020 reported 33% of electricity was generated from renewable sources. While admirable, this comes with challenges. Specifically, at certain times in the year, system-wide generation of energy (specifically solar) was too large for electricity grid absorb, and so CAISO simply shut down transmission going to those lines. CAISO reported that 3-4% of its renewable energy was curtailed due to overproduction in 2019. Curtailment was spread unevenly throughout the year, with certain days in the spring reporting 20-30% curtailment. Without solutions, this problem is likely to grow non-linearly with respect to the amount of renewable energy on the grid.

1.2 Energy storage and demand response

One solution to curtailment is demand-response. Demand-Response (DR) entails the deferment of energy demand from when it is demanded to when it is most opportune for it to be filled. DR is basically costless, as it requires no infrastructure, so it is important as a quick and direct solution. Consumer facing DR is nearly exclusively implemented with a price signal, which is the focus of our research. DR can further help by: (1) alleviating curtailment during high generation months to stretch battery reserves farther, (2) shift seasonal loads (3) alleviate peak loads during the day, to shift demand to other generation sources.

1.3 Social Games in building energy consumptions

The share of buildings' energy demands makes up a significant component of US energy demand and is increasing. In residential and commercial buildings, plug loads represent 30% of total electricity use (Lanzisera et al. (2013), Srinivasan et al. (2011)).

To this end, notable work has been performed in creating and administering "Social Games" – for our purposes, defined as competitions around energy use. Social Games increase the extent to which an individual is motivated to and able to save energy in their daily functioning within their office or home (Konstantakopoulos et al. (2017), Ratliff et al. (2014), Papaioannou et al. (2018), Papaioannou and Stamoulis (2018)) Cowley et al. (2011), Ayres et al. (2012)).

We are unaware of a study that attempts to aim a reinforcement learner at a DR price signal within a Social Game framework. Therefore, we aim to reproduce a Social Game experiment and quantify the difference in demand response from a variety of RL and RL augmentation techniques.

1.4 Paper Outline

This paper is intended to highlight the experimental setup testing two scopes of work: the RL architecture and planning model search. We intend to give emphasis to the experimental setup and de-emphasize the explanation of the RL techniques and planning component, referring the reader to Spangher et al. (2020) where appropriate. In Section 2, we will briefly describe the two RL architectures that we adapted and fine-tuned. We will then describe the planning models. After this, we will detail the reward specification and simulation that the fist two items will be testing in. In Section 3, "Simulated Results", we will show results from the simulation that we use to argue success or failure for the experiment. In Section 4, we will describe the prospective experimental timeline, which will help the reader understand how we intend to actually the RL architectures "in the wild."

2. Methods

For a quick overview on the proposed experiment/simulation, please see figure 1. We will describe each of the boxes drawn in this flow chart in detail.

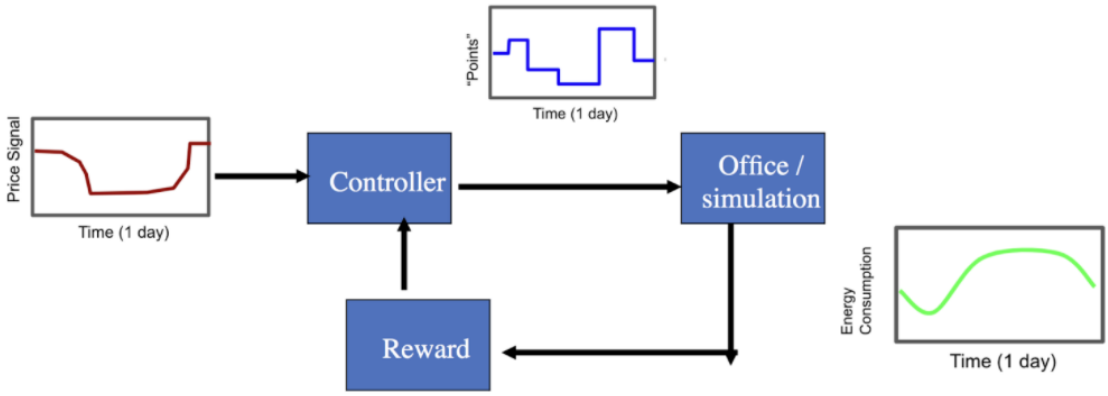

Figure 1: Diagram describing the flow of information in our experiment.

2.1 Reinforcement Learning techniques

Reinforcement Learning as a general technique came out of the physical simulators and computer gaming – arenas that are data rich in measurements and rewards. They might have access to many steps with millions of iterations to try various trajectories. We are dealing with a different scenario – so-called human-in-the-loop learning, where each step in the world is a day and data is very costly to attain.

As such, we need to consider off-policy RL: policies that do not rely solely on interactions with the environment, but with stored data that can be reused, with older data from other experiments, and with intelligent extrapolations. We choose two architectures: Batch Constrained Q Learning, and Soft Actor Critic V2. We skip a description of how the architectures are structured an function; please see Spangher et al. (2020) for a tutorial.

2.2 Planning models: improving RL agent's data efficiency

Although the algorithms that we describe above are important advances in RL for efficient data, we still are concerned that a lack of steps within the environment will hinder any algorithm from effectively learning. We propose a way to address this: planning.

Constructing a good model for human behavior in the office – and specifically response to points – could provide a way for the algorithm to explore more than it could by interfacing with just the world. Each night, the agent will have stepped once in the real world and the planning model will train with an extra observation, but after the planning model updates, the agent can then train thousands of times on the updated model, essentially exploring many more points values.

To this end, we propose different planning models. Each model is made and trained on a simulated dataset of 5 years worth of energy and points data. The training dataset we used and a full explanation of the planning models we tested are detailed in Spangher et al. (2020). Of various AutoML strategies, we attempt an out of the box GPyOpt LSTM search

(Knudde et al., 2017) and AutoKeras structured regressor (Jin et al., 2019); we compare these to two baseline models of a vanilla LSTM and OLS.

2.3 Reward

All agents use the same reward calculation. This reward is defined as the difference between the day's total cost of energy and the ideal cost of energy. The ideal cost of energy is obtained using a simple convex optimization. If $\vec{d}$ are the actual demand of energy computed for the day, $\vec{g}$ is the vector of the grid prices for the day, E is the total amount of energy, and d_{min}, d_{max} are 5% and 95% values of energy observed over the past year, then the ideal demands are calculated with the following optimization:

$$\begin{aligned} d^* = \min_{d} \quad & d^T g \\ \text{s.t.} \quad & \sum_{t=0}^{10} d = E \\ & d_{min} < d < d_{max} \end{aligned}$$

Then, the reward becomes:

$$R(d) = \frac{d^{*T} g - d^T g}{d^{*T} g}$$

I.e. taking the difference and scaling by the total ideal cost to normalize the outcome.

2.4 Simulation design

In order to test the aforementioned techniques, we will use a simulation that can provide some level of insight into the superiority of some methods over others. We define three deterministic responses to points, and populate our office with an equal mixture "people" who exhibit each response type. We will describe the linear response here and then refer the reader to Spangher et al. (2020) for definition of sinusoidal and threshold exponential response, as they are simple variations of the linear response.

For the *linear response*, we define a very simple person who decreases their energy consumption linearly below a baseline with respect to points given. Therefore, if b_t is the baseline energy consumption consumed at time t and p_t are the points given, the energy demand d is $d_t = b_t - p_t$, clipped at d_{min} and d_{max} as defined in Section 2.3.

3. Simulated Results

3.1 RL Architectures

In the RL architecture front, we report limited success with BCQL. Trained over a thirty day period, we see limited evidence that BCQL can learn, and it seems to exhibit little exploration; indeed, the reward seems to be largely constrained to a small bit of the range.

However, SAC V2 appears to be performing better. We report that it generally appears to learn across the three deterministic responses. The framework delivers a learning curve that exhibits some of the basic properties of a good learning curve: it spends time in the

Model	RMSE
GPyOpt LSTM	28
Vanilla LSTM	31
AutoKeras Structured Regressor	50
Regression	100

Table 1: Summary of results for the four planning models.

beginning exploring a large part of the domain, which then helps it improve. Interestingly, it tightens its search as it believes it has found an optimum; this turns out to be a local optimum and it breaks into newer parts of the domain, at which point it converges to 0, or near perfect match to the ideal demands.

3.2 Planning models

Here we present the results of each individual planning model. In summary, please see the RMSE values reported in Table 1.

The GPyOpt LSTM produced the best RMSE of any model that we evaluated. The Vanilla LSTM had a relatively high success; it trained in a small amount of time, and produced a relatively low RMSE of 30. The AutoKeras Structured Regressor was surprisingly ineffective given the success of the other neural methods. It returned an RMSE of 50. One possible reason for the lack of performance was the structure of the nets as being simple feed-forward networks. While the algorithm was able to push down the RMSE in successive iterations, they were not able to match the LSTMs for dealing with the time series. Finally, we have results from linear regressions to serve as a baseline above which to improve. Here, our best model appears to be $y_t \sim b_t + p_t$, and inclusion of an autoregressive term seems to strictly worsen the predictions. We believe that this is because predicting a longer horizon propogated uncertainty through the end of the prediction.

4. Proposed Experimental timeline

We observe two experimental units for a period of five months, from August to December. We am interested in estimating the causal effects of two distinct reinforcement learning architectures, RL_j, for $j \in \{1, 2\}$, in addition to the causal effect of combining reinforcement learning with behavioural feedback from the Social Cognitive Model.

We estimate seasonality effects at period t (δ_t) and improvements in learning due to the accumulation of observations (Ω_t), by taking the average difference between performance across all conditions at time t and t_{-1}. We also estimate the effect of incorporating parameters from the social cognitive model by comparing observations *within* a single RL architecture, controlling for seasonality. Finally, we causally identify the effect of each RL_j, which is the difference between the score for RL_j versus $RL_{\_j}$, and the difference in scores between RL_j and $RL_{\_j}$, *conditional on incorporating the social cognitive model*, controlling for seasonality in each case.

We then use later experimental periods to train the model on smaller subsets of our experimental subjects: smaller groups and then at the individual scale.

Month	Group 1	Group 2	Control
July	— System ID —		
August	RL_2	RL_1 + plan	
September	RL_1	RL_2 + plan	

Table 2: Experimental Timeline in which we compare two different RL architectures and the effect of a planning model.

5. Discussion

5.1 RL Architectures

Of the three architectures, BCQL seems far behind the others. We believe that an aspect of BCQL that we thought was a strength is actually a weakness: the batch that it fills to train off-policy. Because it needs to fill up the batch, it needs to explore the environment a bit before stepping or learning. Small sizes of the batch might prediscpose it to parts of the action space that are underexplored, and large sizes of the batch might mean that it is more data hungry than we thought it would be.

Our neural architecture search in the RL policies was relatively ad-hoc: larger hidden states and smaller batch sizes seems to improve the outcomes, and so we tended to gravitate towards them. However, we did not have the bandwidth to do a formal parameter search.

5.2 Planning model

The experiments in out of the box AutoML were mixed, with the GPyOpt returning a huge improvement over the vanilla models, and AutoKeras returning an score in the middle of the pack. We hypothesize that the recurrent models that GPyOpt was optimizing with were able to push the RMSE down far, whereas the feed-forward networks in AutoKeras had a threshold of performance that was difficult to push forward from. Ultimately, we do not know how similar to real data our synthetic data was, but we hope that this project lent at least some insights on models to carry forward.

6. Conclusion

We have presented the results of a simulation that tests some of the components of a system we hope to test. We also present the proposed experimental timeline for testing these components. We submit the experimental timeline and simulation for comments in this workshop, and hope to incorporate feedback as we go forward with the experiment.

References

Ian Ayres, Sophie Raseman, and Alice Shih. Evidence from Two Large Field Experiments that Peer Comparison Feedback Can Reduce Residential Energy Usage. *The Journal of Law, Economics, and Organization*, 29(5):992–1022, 08 2012. ISSN 8756-6222. doi: 10.1093/jleo/ews020. URL `https://doi.org/10.1093/jleo/ews020`.

Ben Cowley, Jose Luiz Moutinho, Chris Bateman, and Alvaro Oliveira. Learning principles and interaction design for "green my place": A massively multiplayer serious game. *Entertainment Computing*, 2(2):103 – 113, 2011. ISSN 1875-9521. doi: https://doi.org/10.1016/j.entcom.2011.01.001. URL http://www.sciencedirect.com/science/article/pii/S1875952111000024. Serious Games Development and Applications.

Haifeng Jin, Qingquan Song, and Xia Hu. Auto-keras: An efficient neural architecture search system. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1946–1956, 2019.

Nicolas Knudde, Joachim van der Herten, Tom Dhaene, and Ivo Couckuyt. Gpflowopt: A bayesian optimization library using tensorflow. *arXiv preprint arXiv:1711.03845*, 2017.

I. C. Konstantakopoulos, L. J. Ratliff, M. Jin, and C. J. Spanos. Leveraging correlations in utility learning. In *2017 American Control Conference (ACC)*, pages 5249–5256, May 2017. doi: 10.23919/ACC.2017.7963770.

S Lanzisera, S Dawson-Haggerty, H. Y. I. Cheung, J Taneja, D Culler, and R Brown. Methods for detailed energy data collection of miscellaneous and electronic loads in a commercial office building. *Building and Environment*, 65:170–177, 2013.

T.G. Papaioannou and G.D. Stamoulis. Teaming and competition for demand-side management in office buildings. volume 2018-January, pages 332–337, 2018. doi: 10.1109/SmartGridComm.2017.8340734. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-85041942675&doi=10.1109%2fSmartGridComm.2017.8340734&partnerID=40&md5=1566057fd32f9fc730fb497d9eee4c17.

T.G. Papaioannou, N. Dimitriou, K. Vasilakis, A. Schoofs, M. Nikiforakis, F. Pursche, N. Deliyski, A. Taha, D. Kotsopoulos, C. Bardaki, S. Kotsilitis, and A. Garbi. An iot-based gamified approach for reducing occupants' energy wastage in public buildings. *Sensors (Switzerland)*, 18(2), 2018. doi: 10.3390/s18020537. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-85041956116&doi=10.3390%2fs18020537&partnerID=40&md5=b8a4c19294b488934e08a6356c33065c.

L. J. Ratliff, M. Jin, I. C. Konstantakopoulos, C. Spanos, and S. S. Sastry. Social game for building energy efficiency: Incentive design. In *2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1011–1018, Sep. 2014. doi: 10.1109/ALLERTON.2014.7028565.

Lucas Spangher, Akash Gokul, Manan Khattar, Joseph Palakapilly, Akaash Tawade, Adam Bouyamourn, Alex Devonport, and Costas Spanos. Prospective experiment for reinforcement learning on demand response in a social game framework. In *Proceedings of the 2nd International Workshop on Applied Machine Learning for Intelligent Energy Systems (AMLIES) 2020*, 2020.

R. S. Srinivasan, J Lakshmanan, E Santosa, and D Srivastav. Plug load densities for energy analysis: K-12 schools,. *Energy and Buildings*, 43:3289 – 3294, 2011.