Predicting Requests in Large-Scale Online P2P Ridesharing

Filippo Bistaffa
IIIA-CSIC
Barcelona, Spain
filippo.bistaffa@iiia.csic.es

Juan A. Rodríguez-Aguilar
IIIA-CSIC
Barcelona, Spain
jar@iiia.csic.es

Jesús Cerquides
IIIA-CSIC
Barcelona, Spain
cerquide@iiia.csic.es

ABSTRACT

Peer-to-peer ridesharing (P2P-RS) enables people to arrange one-time rides with their own private cars, without the involvement of professional drivers. It is a prominent *collective intelligence* application producing significant benefits both for individuals (reduced costs) and for the entire community (reduced pollution and traffic), as we showed in a recent publication where we proposed an online approximate solution algorithm for large-scale P2P-RS.

In this paper we tackle the fundamental question of assessing the benefit of predicting ridesharing requests in the context of P2P-RS optimisation. Results on a public real-world show that, by employing a *perfect* predictor, the total reward can be improved by 5.27% with a forecast horizon of 1 minute. On the other hand, a vanilla *long short-term memory* neural network cannot improve upon a baseline predictor that simply replicates the previous day's requests, whilst achieving an almost-double accuracy.

KEYWORDS

Online P2P ridesharing; optimisation; prediction; deep-learning.

1 INTRODUCTION

With the growing popularity of the sharing economy, ridesharing services are called to transform urban mobility. Shared mobility is expected to have major environmental and economic impacts by reducing pollution, traffic congestion, and energy consumption, which could be further enhanced with the advent of electrical vehicles. Moreover, ridesharing is set to become even more attractive in a near-future world of self-driving cars, spurring a transition from solo driving to mass transit. This major endeavour addresses key topics in Sustainable Development Goals 11 ("sustainable cities and communities") and 7 ("affordable and clean energy").

The concept of ridesharing has recently received increasing attention, both in the transportation industry (e.g., UberPool, Lyft, and Maramoja) and in academia. However, *ridesharing* may have different interpretations that result in significant differences in the computational models that can be used to represent and address the problem. This problem has been addressed from a computational view point only very recently using various algorithms [1, 3, and references therein]. Most of these approaches (e.g., [1]) focus on maximising the benefits of the passengers considering mainly the concept of quality of service (i.e., minimising the delay experienced by the users). In a recent work [3] we proposed a different perspective on the *peer-to-peer ridesharing* (P2P-RS) problem by considering not only the quality of service, but also the environmental benefits resulting from ridesharing (e.g., CO_2 and traffic congestion reduction). This perspective address the trade-off between those two objectives when forming shared rides, looking at ridesharing as a method to foster sustainable mobility for policy-makers and not only as a profitable alternative for commuters.

On the one hand, despite tackling an inherently *online* problem (i.e., ridesharing requests are not known in advance), the majority of the works in the ridesharing optimisation literature do not employ *predictions* of forthcoming requests to improve the quality of their online solutions.[1] On the other hand, even though there exist some works aiming at predicting *mobility on demand* requests (see, e.g., [9]), such works only look at the *accuracy* of the predictions, whereas their impact on the optimisation process remains unclear.

In this paper we tackle the fundamental question of assessing the benefit of predictions in the context of P2P-RS optimisation. Specifically, we present our preliminary study that aims at including predictions in our P2P-RS solution technique [3], with the objective of improving the total reward of the computed solution. Results on a public real-world dataset [7] show that, by employing a *perfect* predictor, the total reward can be improved by 5.27% (6.31% during weekend days) with a forecast horizon of 1 minute. On the other hand, a *long short-term memory* (LSTM) neural network [6] cannot improve upon a predictor that simply replicates the previous day's requests, whilst achieving an almost-double accuracy.

2 THE P2P RIDESHARING PROBLEM

We consider the P2P-RS problem as defined in [3], i.e., as an *online stochastic scheduling* problem [8] over a discrete time horizon $H = [1, h]$. Our problem takes place in an area divided in n zones, i.e., $Z = \{1, \ldots, n\}$. At each step $t \in H$, the system receives a (possibly empty) set of requests R_t. Each request $r \in R_t$ is a tuple $\langle i, j, d, \delta \rangle$ characterised by a starting zone $i \in Z$ and a destination zone $j \in Z$, a Boolean value $d \in \mathbb{B}$ indicating whether the corresponding commuter has a car or not (i.e., whether it is a driver or not), and the maximum time $\delta \in H$ the commuter is willing to wait to be assigned to a car. Formally, $r \in \mathcal{R} = Z \times Z \times \mathbb{B} \times H$. Hence, $\langle R_1, \ldots, R_t, \ldots, R_h \rangle$ represents the input of the problem. We measure the performance of a given set S of requests (i.e., a car) both in terms of *environmental benefits* and *quality of service* (QoS). Formally, we define $V : 2^{\mathcal{R}} \rightarrow \mathbb{R}$ as the total reward (i.e., considering environmental benefits and the quality of service) associated to a given car S as

$$V(S) = \rho_{CO_2} \cdot E_{CO_2}(S) + \rho_{\text{noise}} \cdot E_{\text{noise}}(S) + \rho_{\text{traffic}} \cdot E_{\text{traffic}}(S) + \rho_{\text{QoS}} \cdot Q(S), \quad (1)$$

Proc.of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 2020, Auckland, New Zealand

https://doi.org/doi

[1] In [3] we propose a *look-ahead* mechanism that filters out immediately profitable cars whose formation could hinder the formation of even better cars in the near future.

where ρ_{CO_2}, ρ_{noise}, and ρ_{traffic} measure the importance of the corresponding environmental benefit and ρ_{QoS} that of the QoS. We refer the interested reader to [3] for further technical details.

In our previous work [3], we defined $Q : 2^{\mathcal{R}} \rightarrow \mathbb{R}_{\leq 0}$, namely the *quality of service* associated to a given car S, as $Q(S) = -\sum_{r \in S} \frac{t_r - t_r^*}{t_r}$, where t_r and t_r^* are the travel times associated to request r with and without ridesharing, respectively. Since t_r^* is the optimal travel time associated to $r = \langle i, j, d, \delta \rangle$ (i.e., obtained by driving through the shortest path from zone i to zone j), then $0 \leq \frac{t_r - t_r^*}{t_r} \leq 1$.

In this paper we extend such a definition by considering a more realistic QoS measure that also takes into account the *to-be-assigned* (TBA) delay, i.e., the delay from the time t_r of arrival of a request r and the time of its assignment to a car S. Notice that, given a car S, the assignment time of any request $r \in S$ to S corresponds to the time when S is actually formed, i.e., when the the latest request $r \in S$ has entered the system. Formally,

$$Q(S) = -\sum_{r \in S} \frac{t_r - t_r^*}{t_r} - \sum_{r \in S} \frac{\max_{r' \in S} t_{r'} - t_r}{\delta_r}. \tag{2}$$

Notice that, in order to be consistent with our previous definition of $Q(\cdot)$, we normalise each TBA delay by dividing it by the maximum assignment time, hence ensuring that $0 \leq \frac{\max_{r' \in S} t_{r'} - t_r}{\delta_r} \leq 1$.

Intuitively, the P2P-RS problem aims at arranging, at each time step t, a (possibly empty) set $\mathcal{S}_t$ of non-overlapping cars among the current set of active requests, with the objective of maximising the sum of the associated rewards over the entire time horizon H. The formal definition of the P2P-RS problem can be found in [3].

2.1 Solving the P2P-RS Problem

We tackle the P2P-RS problem by solving a *sequence* of offline problems by means of our approximate offline approach [3]. Our approximate offline approach employs a well-known technique for approximately solving time-constrained large-scale combinatorial optimisation problems modelled as *integer linear programs* (ILP), which consists in (1) removing those variables from the model that, a priori, do not seem to help for the generation of good solutions, and (2) passing the reduced ILP to a solver in order to get the best solution possible in the available computation time.

In the specific case of our approximate offline approach, we iteratively apply a probabilistic greedy heuristic to generate feasible cars that represent "good candidates" for the final optimisation solution. We then formulate and solve an ILP (specifically, a *weighted set packing* problem) whose decision variables correspond only to the cars in the above-mentioned set of good candidates, rather than to the set of all possible cars (whose enumeration would be infeasible in the time budget we consider for the P2P-RS scenario, i.e., 1 minute for solving the offline problems corresponding to each time step). Finally, the set of cars resulting from the solution of the above-discussed reduced ILP is filtered by means of a *look-ahead* mechanism that discards immediately profitable cars whose formation could hinder the formation of even better cars in the near future. Notice that, even though such a technique allows us to put each offline solution in the context of the overall online problem, it does not represent a *predictive* technique, since it does not take into account the history of requests, and, more importantly, it is applied *after* the optimisation, and not *before*.

3 PREDICTIONS FOR P2P RIDESHARING

As previously discussed, none of the recent works on ridesharing optimisation [1, 3, and references therein] employs prediction to improve the quality of the solution in terms of total reward. Indeed, rather than employing a *predictive* approach, the majority of the works embrace *reactive* techniques that aim at taking into account the requests in the optimisation process once they already materialised inside the system (including our own *look-ahead* method [3]). As a possible notable exception, Bicocchi *et al.* [2] proposed a systems for recommending ridesharing matches based on the historical data provided by a large Italian telecom operator.

To the best of our knowledge, the majority of the works focusing on predicting *mobility on demand* requests are usually interested in measuring and maximising the accuracy of predicting the number of requests (e.g., [9]). On the other hand, the actual impact of such predictions on the optimisation process remains unclear.

Here we tackle the following fundamental questions: are predictions beneficial *wrt* to reactive techniques in the context of P2P-RS optimisation? If so, which is the best *forecast horizon* f? Which is the maximum improvement in terms of total reward that one can expect by employing predictions? How do predictions impact on the solution quality of our approximate offline approach, which receives a larger amount of input requests in the same time budget?

To address the above questions, we incorporate 3 different types of predictors in our P2P-RS online solution algorithm: (1) a *perfect* predictor with 100% accuracy, (2) a predictor that replicates the requests of the previous day (as also considered in [9]), and (3) a vanilla LSTM neural network [6], comprised of a LSTM layer with a hidden vector h of size $Z \times Z$ (i.e., the number of different possible requests), a fully connected layer, and a ReLU layer.

4 EXPERIMENTAL EVALUATION

4.1 Experimental Methodology

Our experimental evaluation has the following objectives:

(1) determine the maximum improvement in terms of total reward when employing a *perfect* predictor,
(2) determine the best forecast horizon f for predictions,
(3) compare a vanilla LSTM predictor with a *perfect* predictor, a *yesterday* predictor, and our *look-ahead* approach [3].

If not otherwise specified, our experimental evaluation follows the same methodology adopted in our previous work [3]. We employ the same publicly available real-world dataset of taxi trips in Manhattan, New York City [7], considering the most recent data available at the time of writing (i.e., June 2019). As previously mentioned, in all our experiments we consider a more realistic measure for QoS *wrt* [3], i.e., one that also takes into account the *to-be-assigned* delay (Equation 2). In all our experiments we consider a *maximum assignment time* $\delta = 5$ and a time horizon $h = 1$ day.

Following [9], we employ the *symmetric mean absolute percentage error* (SMAPE) [5] to measure the accuracy of all predictors:

$$\text{SMAPE} = \frac{1}{n} \sum_{t=1}^{n} \frac{|P_t - G_t|}{(P_t + G_t)/2},$$

where P_t and G_t are the prediction and the ground truth at time t.

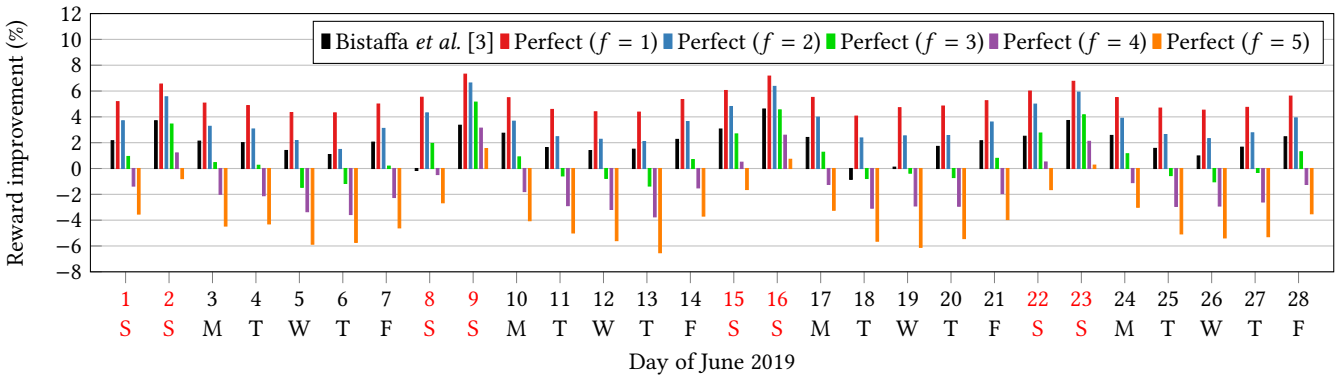

Figure 1: Total reward percentage improvement *wrt* no predictions (best viewed in colour).

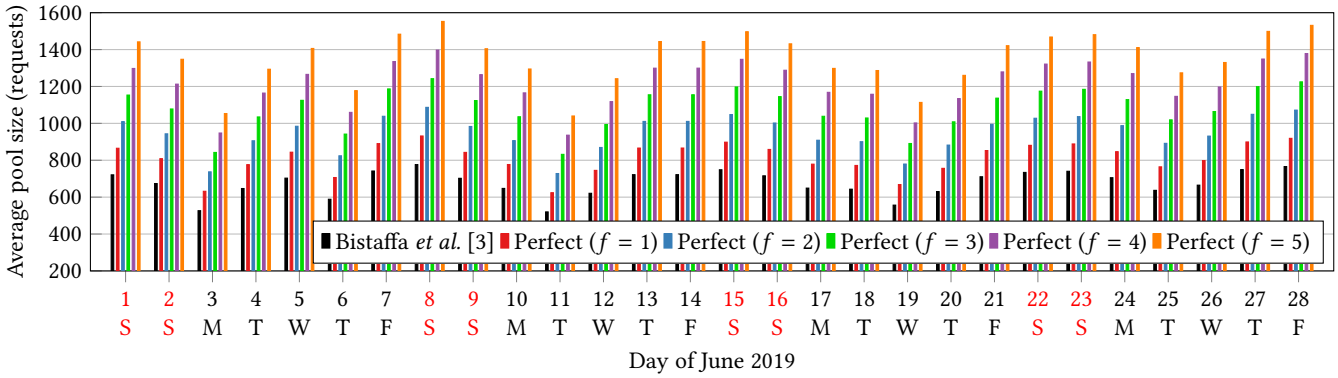

Figure 2: Average number of requests (including predictions) in the pool (best viewed in colour).

Our LSTM predictor has been trained on a machine with a 12-cores 3.70GHz CPU, 32GB of RAM, and two GeForce RTX 2080 Ti GPUs. All optimisation experiments have been run on a cluster whose computing nodes have two 4-cores 2.26GHz CPUs and 32GB of RAM, using CPLEX 12.7 as ILP solver.[2]

4.2 Perfect Predictor

First, we aim at determining the maximum improvement in terms of total reward by employing a *perfect* predictor when considering a forecast horizon f up to the considered maximum assignment time $\delta = 5$, i.e., $f \in \{1, 2, 3, 4, 5\}$. Notice that, since the behaviour and the performance of our online optimisation algorithm depends on the parameters γ, d_{rate}, and l_{size} (namely, the solution time budget, the determinism rate, and the candidate list size), following [3] we determined the values of such parameters for each f using the `irace` package. Figure 1 shows the total reward percentage improvement when using a perfect predictor as well as our *look-ahead* approach [3], compared with the case with no prediction (i.e., $f = 0$). Results clearly show two interesting behaviours.

First, for every considered prediction technique, the improvement obtained by employing predictions follows a pattern that clearly repeats every week, reaching the maximum improvement during the weekend (days marked in red). Second, results clearly show that the reward improvement decreases when increasing the forecast horizon f, providing a tangible benefit only up to $f = 2$. Although the explanation of these results is not straightforward, by looking at the average number of requests in the pool in the corresponding days (Figure 2) we get some interesting insights.

Indeed, we notice a correlation between the behaviour in Figure 1 and 2, suggesting that a slightly larger amount of requests during the weekend allows for a larger reward improvement due to the employment of predictions. These observations seem related to our previous findings, as in [3] we noticed that the quality of our online approach *wrt* the offline approach progressively increased as the problem size increased. Nonetheless, it seems that once the number of requests in the pool grows excessively due to the longer forecast horizon (the average pool size for $f = 5$ is more than double *wrt* the case without predictions, i.e., when using the *look-ahead* approach), our approximate offline approach cannot produce a solution of the same quality in the same time budget (i.e., 1 minute). We will further investigate these aspects in our future work.

Our main conclusion is that our online optimisation algorithm clearly benefits from perfect predictions only when using a forecast horizon $f \in \{1, 2\}$, reaching an average improvement of 5.27% (6.31% during weekend days) with $f = 1$. In comparison, our *look-ahead* approach achieves an average improvement of 2%.

[2] Source code: https://github.com/filippobistaffa/P2P-RS/tree/OptLearnMAS.

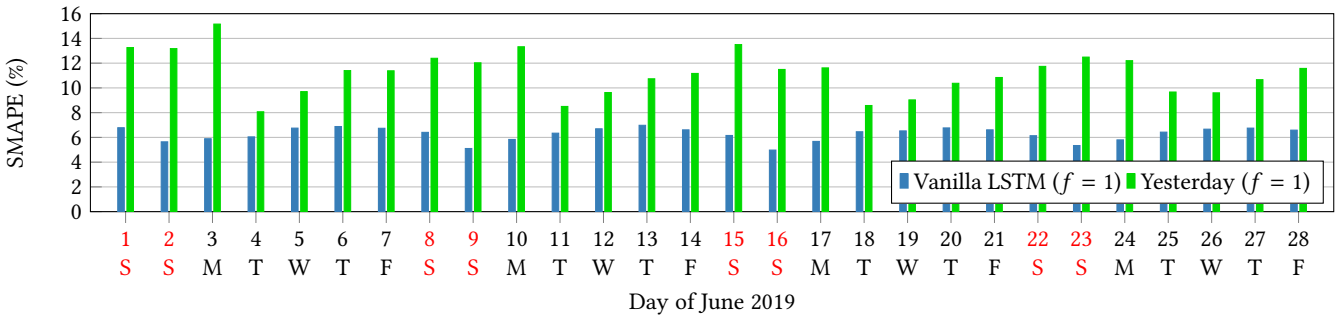

Figure 3: SMAPE for LSTM and *yesterday* predictors (best viewed in colour).

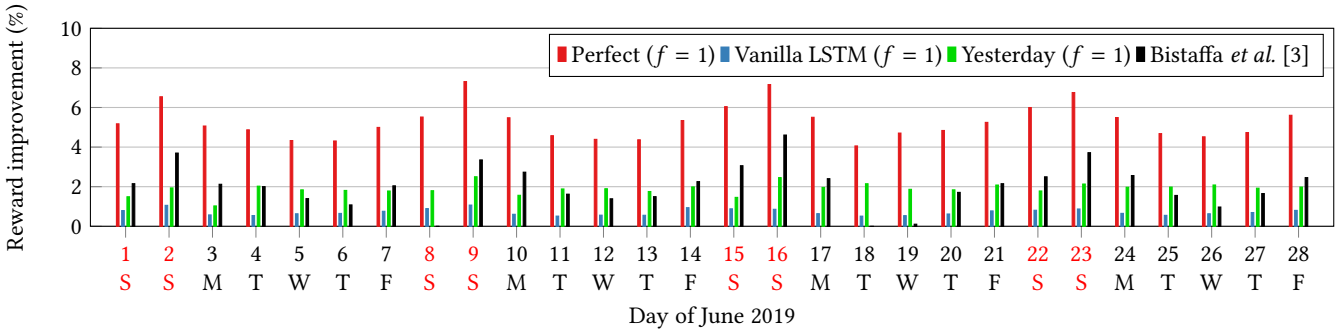

Figure 4: Total reward percentage improvement *wrt* no predictions (best viewed in colour).

4.3 LSTM and *Yesterday* Predictors

In our second set of experiments we aim at evaluating two baseline predictors, a vanilla LSTM predictor and a *yesterday* predictor, which simply replicates the request of the previous day. We measure the performance both in terms of accuracy (SMAPE) and total reward improvement. Given the above-discussed results, we consider a forecast horizon of 1 time step for both these predictors.

Figure 3 shows that the vanilla LSTM significantly outperforms the *yesterday* predictor in terms of accuracy, achieving an average SMAPE of 6.27 *vs* 11.18. Nonetheless, these results are not reflected when measuring the improvement in terms of total reward (Figure 4), as the vanilla LSTM is actually performing *worse* than the counterpart (+0.71% *vs* +1.89%). Note that our previous *look-ahead* approach (+2%) slightly outperforms both baseline predictors.

5 CONCLUSIONS

In this preliminary study concerning the integration of predictions in our online approximate algorithm for large-scale P2P-RS optimisation, we showed that, in our experiments, predicting future requests is beneficial, *in the best case*, only for up to 2 time steps of forecast horizon, and that a vanilla LSTM predictor *cannot* improve upon a *yesterday* predictor in terms of total reward improvement, whilst achieving an almost-double accuracy. Thus, we decisively conclude that accuracy alone cannot be taken as a sole performance measure for predictions, especially in the context of solving an online optimisation process. Nonetheless, given our sometimes counter-intuitive findings, we will further investigate the impact of predictions on the optimisation process. Furthermore, we aim at evaluating different and more advanced predictors, e.g., CNN-LSTM [4], which are specifically designed for prediction problems on inputs with spatial dependencies, like the P2P-RS problem.

ACKNOWLEDGEMENT

This project has received funding from the EU H2020 programme under grant agreement #769142.

REFERENCES

[1] J. Alonso-Mora, S. Samaranayake, A. Wallar, E. Frazzoli, and D. Rus. 2017. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences* 114, 3 (2017), 462–467.
[2] N. Bicocchi, M. Mamei, A. Sassi, and F. Zambonelli. 2017. On recommending opportunistic rides. *IEEE Transactions on Intelligent Transportation Systems* 18, 12 (2017), 3328–3338.
[3] F. Bistaffa, C. Blum, J. Cerquides, A. Farinelli, and J. A. Rodríguez-Aguilar. 2020. A computational approach to quantify the benefits of ridesharing for policy makers and travellers. *IEEE Transactions on Intelligent Transportation Systems* (2020).
[4] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2625–2634.
[5] B. Flores. 1986. A pragmatic view of accuracy measurement in forecasting. *Omega* 14, 2 (1986), 93–98.
[6] S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.
[7] NYC Taxi and Limousine Commission. 2019. Trip record data. (2019). Online at https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page.
[8] P. Van Hentenryck and R. Bent. 2009. *Online Stochastic Combinatorial Optimization*. The MIT Press.
[9] L. Zhu and N. Laptev. 2017. Deep and confident prediction for time series at Uber. In *IEEE International Conference on Data Mining Workshops*. 103–110.