Spatio-temporal data mining: A practical application in taxi fare prediction

Jorge Renato Torres-Garcia[1]

[1] Faculdade de Economia, U.Porto, 4200-465 Porto, Portugal

Abstract. This paper explains the first place solution of the Kaggle competition organized by the 18th EPIA Conference celebrated in 2017. The challenge consisted in predicting the fare (with price as a categorical variable) of one month of taxi trips in the city of Thessaloniki, Greece. A deep learning approach using a clustering technique combined with a multi-layer perceptron network and an embedding space for the categorical variables is used. As the score of this competition was the quadratic weighted kappa, which made this not a common classification problem, the problem is treated as a regression one with mean squared error as loss function for latter decoding the outputs.

Keywords: EPIA 2017, Kaggle, Taxi fare prediction, Thessaloniki, Quadratic Weighted Kappa, Deep learning, MLP, Python, Keras, Theano

1 Introduction

As availability of information grows, stakeholders in different areas are required to incorporate this information in their decision-making processes. In the transportation field, specifically in the taxi sector, the stakeholders are eager to know, e.g., the demand in order to cover it with their fleets of taxis in each unique area. In particular, for addressing the profitability of a taxi fleet company, the revenue of each taxi trip plays an important role and justifies the interest in its prediction.

This paper presents a solution for the Discovery Challenge organized by the EPIA Conference celebrated in 2017. The participants were asked to predict the fare of taxi trips with the spatio-temporal data of its starting point. A deep learning model approach combined with a clustering technique is used. The software used is Python version 2.7. Keras [1] library is used for the deep learning models. Theano [2] library is used as backend.

The paper is structured as follows: In the next section, the challenge, the challenges face to solve the problem, and the approach that is taken to solve it are described in detail. Section 3 will show the results and a discussion about them, and section 4 will present the conclusions.

2 Description of the data and the model (Data and procedures)

2.1 EPIA Discovery Challenge 2017

In the Discovery Challenge of the 18th EPIA Conference on Artificial Intelligence, the challenge was to predict the taxi service fare type. The data consists of the initial starting time and location of a taxi trip for 1148 taxis in the city of Thessaloniki, Greece. As the training set is given the data collected between January and March of 2015, it is asked to make the predictions for April 2015. The dependent variable can take five values: low, normal, medium, high and very high.

The following independent variables are given: taxi_id, timestamp, starting_latitude, starting_longitude. These relate to a spatio-temporal dataset with different individuals: the timestamp variable represents the exact time of the start of the taxi trip, the latitude and longitude to the exact starting point and the taxi id allows for the identification of different taxis.

The submissions are scored in this challenge by using the quadratic weight kappa. This metric measures the agreement between two ratings, and the maximum score that can be achieved is one, which indicates complete agreement between different raters. Later in this section, some particular characteristics of this score that make this an atypical classification problem are mentioned.

2.2 Challenges

From the beginning, it is apparent that the training set is very unbalanced. The "very high" class (class 5) only accounts for only 1.5% of the training set. There are techniques to handle this, such as inputting weights to penalize the loss function when an incorrect prediction of a small class is made.

It may be confusing but this is not a classic classification problem, mainly because the score is assigned by the quadratic weighted kappa (QWK) metric. For example, if the true value of a taxi trip is 5, predicting 4 is less worse than predicting 1 for the QWK score. In a regular classification problem, the loss function (e.g. categorical cross-entropy) penalizes any wrong prediction equally. Therefore, a loss function that suits our problem should be found.

As already mentioned this is a spatio-temporal problem with different agents (taxis) involved. From the current available deep learning methods, a convolutional neural network may be suitable to represent the spatial data correctly. Moreover, a Long-Short term memory (LSTM) can be used for the temporal data. However, neither of these approaches are used as detailed below.

2.3 Deep learning approach

In the literature, one can find the winning solution of the 2015 ECML PKDD competition [3]. The challenge was to predict the taxi trip destination. It is known that the fare is related to the time and distance traveled for a given trip, so it is a suitable

model to use as a baseline. They used a Multi-layer percepton with ReLU as the activation function for the neurons. They also cluster the spatial data in order to help the network prediction to not be too sparse. Inspired by this, the latitude and longitude data are clustered to create a new variable that refers to which cluster each point belongs.

The mean-shift algorithm [4] is also used, as it is found computationally efficient when implemented in the scikit-learn package [5]. 1748 different clusters are obtained by applying this algorithm.

It is known that neural networks work best with continuous data. The embedding technique of mapping a discrete variable in a continuous space [6] is used to take advantage of this. This technique is well known in the natural language processing field where words, which are discrete, have to be mapped in a continuous space before inputting them in a neural network model.

Some variables are created in order to take into account the temporal dimension of our data. These variables are the following: quarter hour of each day and day of the week. A new variable is also created to identify holidays. This variable also takes into account if a day is before or after a holiday.

For inputting the time of the month in the network, embedding the date of the month would seem to be the best choice. However, it may lose information as not every month finishes with the same date in our dataset. So the time of the month is directly coded in two continuous variables using Fourier transformations and functions sine and cosine.

Three features are engineered. First, a variable that identifies the number of taxi trips that occurred in the last 15 minutes is created. This will help identify if there is an abnormal pattern in the demand of taxis. Next, two more variables are created:

- The Haversine distance between the starting point of a trip and the starting point of the next trip (for the same taxi).
- Time difference between the starting time of a trip and the starting time of the next trip (for the same taxi).

A neural network fit with only these two last variables get a 0.50 score in the quadratic weighted kappa metric on our validation set. A submission created with this model would be located in the fourth place in the private leaderboard on Kaggle.

One may also find in current literature that the least-square regression is related with the kappa score regression [7]. Moreover, in the Kaggle community, first, fitting a regression model, and then, decoding the outputs to the categories given when the score of the competition is related with the kappa score is a well-known method [8]. A regression model, also, helps deal with an unbalanced dataset. The output of the regression is decoded by, first, ranking the outputs of the training set in ascending order. Then, with the cumulative density function (CDF) of the classes of the training set, the offsets are obtained. For example, if the CDF for the first class is 20%, we will take the first 20% value as our offset. All values that are below this value will be assigned as class 1 and the same logic is used for the other classes.

In order to test the models, 10% of the training set is selected as a validation set. A stratified shuffle split is used to create this set as this maintains the percentage of samples for each class. The scikit learn package is also used for this purpose.

In Fig. 1, the final model is shown. The variables in green denote use of an embedding space before inputting them in the neural network. The choice of the

dimension of the embedding space is the same as the past winner's models except that it has an upper bond where it cannot be more than the unique categories in each variable [6]. The raw output refers to the output of the network with the mean squared error (MSE) loss. Then this output is decoded in one of the five categories as previously mentioned.

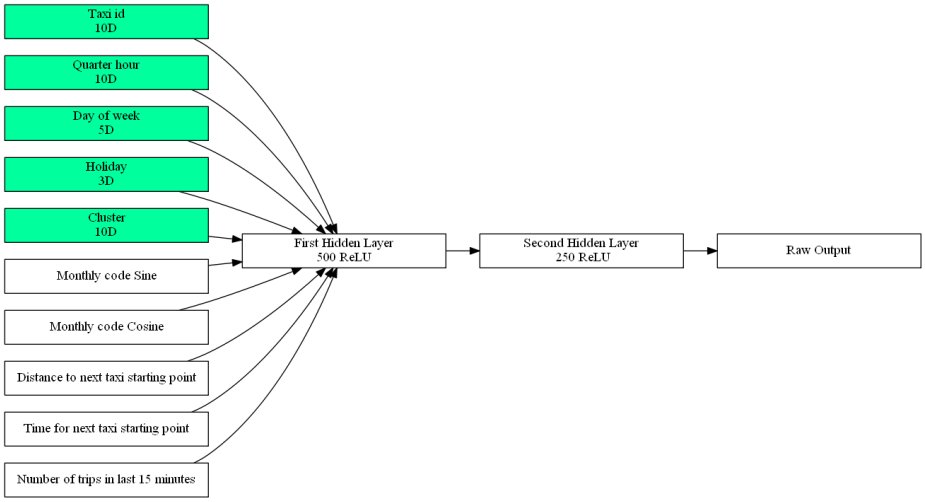

Fig. 1. Neural network structure. The variables that use an embedding space before inputting them in the neural network are in green background. The number of dimensions of this space for each variable is below its name.

3 Results and discussion

For training the network, 10 epochs with a batch size of 200 are used. The results are shown in Table 1. From the results, it can be inferred that the validation set used is a good approximation of the score in the public and private leaderboards on Kaggle. With this result, 1st place in both the public and private leaderboard is obtained with a 0.04 difference with the second place.

Table 1. Quadratic Weight Kappa and Mean Squared Error scores of the proposed model.

	Score (QWK)	Score (MSE)
Training	0.59548	0.5391
Validation	0.57040	0.5563
Public LB	0.57219	-
Private LB	0.57364	-

For the last trips of each taxi, a feed-forward neural network is trained without the last two engineered values (distance and time to next taxi starting point) as these

variables are unknown. This neural network consists of only one hidden layer of 200 ReLu neurons with a dropout of 20% before and after the hidden layer. It is trained for 20 epochs and uses a batch size of 200. As the number of forecasts with this model is negligible compared with the whole test set (less than 0.2%), more details about it are not given.

In this section, it is also relevant to show how the clustering technique and the embedding are combined in order to identify the similarities and differences between the different locations. This is essential for making a robust model. In Fig. 2, the heat-map of the starting trip locations is shown. Only for plotting this figure, the 5% data that deviates more from the mean is cut. An idea of the layout of the city of Thessaloniki can be made, with a central part that is easily recognizable.

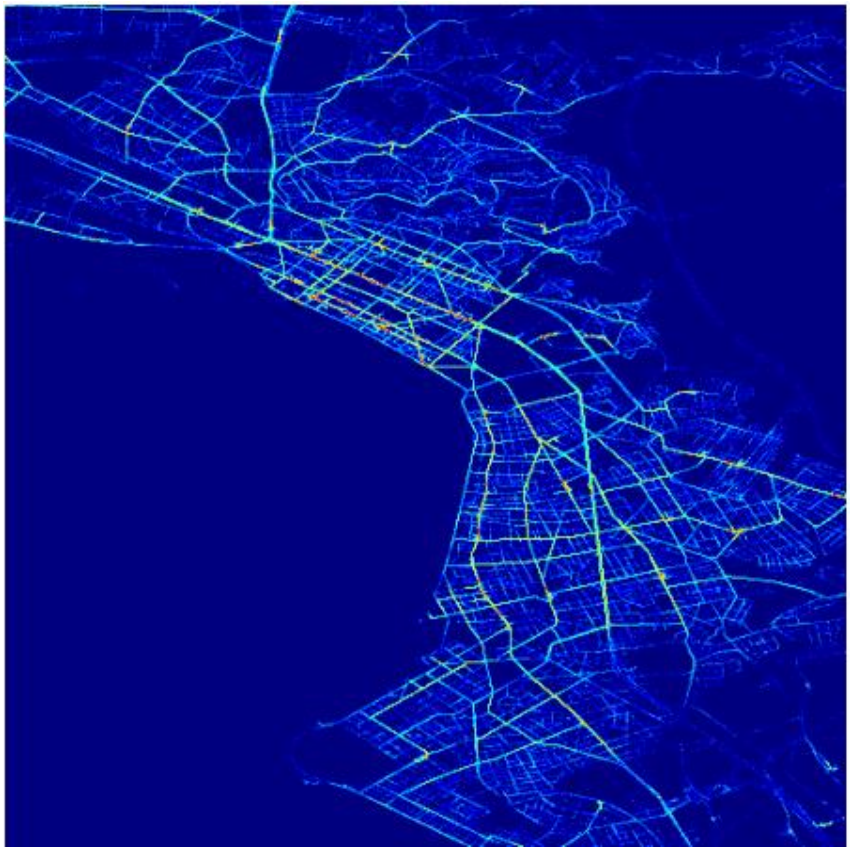

Fig. 2. Heat-map of the starting location of the each taxi trips.

In Fig. 3, the center of each cluster created by the mean-shift algorithm is plotted. The first clusters created are in blue and represent the densest zone in the map. By applying the categorical embedding, it is expected that the embedding technique will map the clusters in blue next to each other; the same can also be said about the red clusters that are mostly on the outside. Moreover, it is expected to obtain a

distinguishable separation between red and blue clusters in the embedding space. For showing this, a dimensionality reduction by the use of the t-SNE method [9] is done in order to have the embedding space in two dimensions, which can be easily plotted.

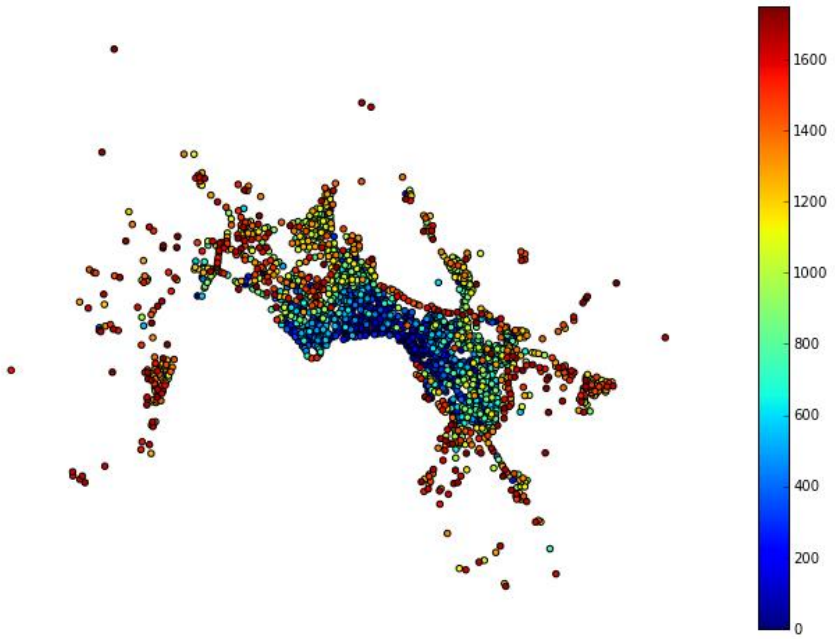

Fig. 3. Center of clusters map result of applying the mean-shift clustering algorithm to the starting location of taxi trips. There are 1748 clusters in total.

As expected, the blue clusters are close to each other. This also applies to the red clusters. The embedding space correctly learns to find the similarities between the clusters even though it does not know nothing about the location of each cluster. In addition, it can be seen that it separates the blue from the red cluster, showing that embedding space correctly distinguishes them.

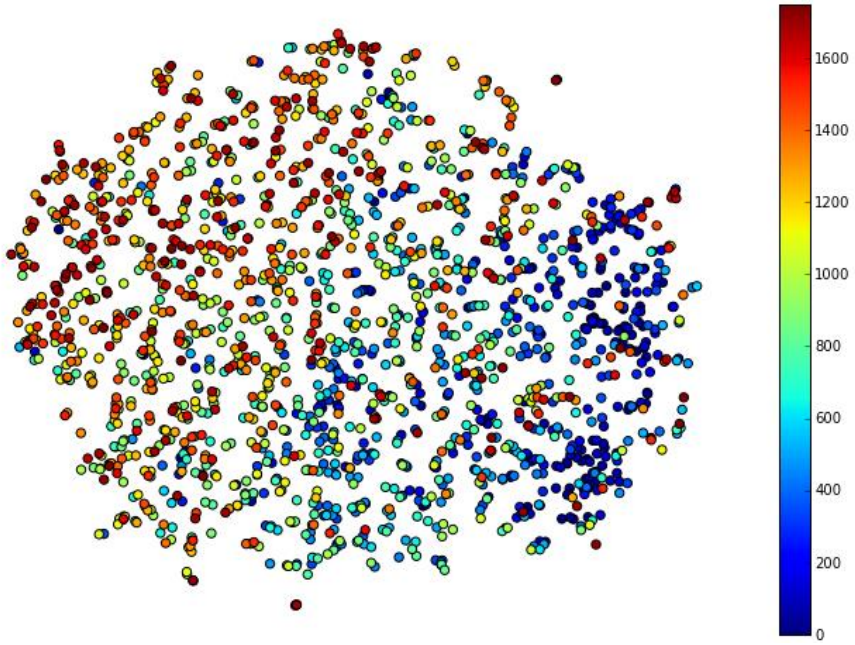

Fig. 4. 2-D reduction using t-SNE algorithm of the embedding space of each cluster.

As an additional example for showing how embedding works, in Fig. 5 the plot in two dimensions of the embedding space for the variable quarter hour is shown. As expected, the values that are consecutive in time are next to each other correctly identifying their similarities.

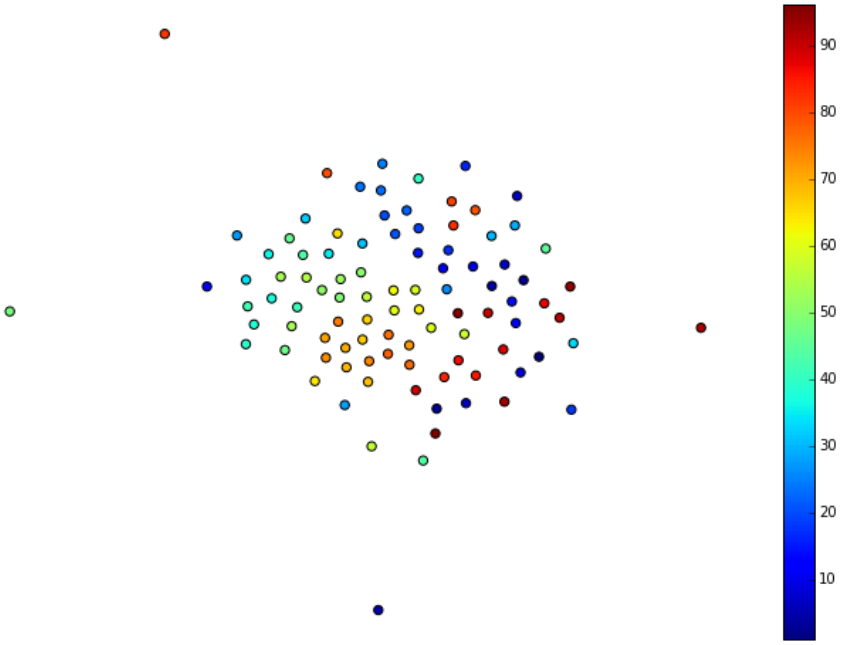

Fig. 5. 2-D reduction using t-SNE algorithm of the embedding space of each quarter hour.

4 Conclusions

For predicting the fares of taxi trips given the initial time and location, first the locations are clustered using the mean-shift algorithm. Thanks to the embedding technique that is used for projecting a discrete variable in a continuous space, any clustering algorithm could serve as a solution given that it obtains a good granularity.

For incorporating the time dimension in the problem, it is decoded in different variables that try for the best characterization: quarter hours in a day, time of the month, day of the week and holidays. By using a simple technique for coding the time of month and embedding the rest of the time variables, the time dimension is taken into account in the model.

The combination of the clustering and the embedding technique helps in learning similarities and differences in the spatial dimension. The embedding technique by itself is very powerful as it assigns the weights in order to be the "best representation of the variables" for the model. The drawback is that it can easily overfit a model by allowing too many dimensions for the embedding space.

As the relation between the Kappa score and the least-squares regression have already been proofed theoretically and empirically, it was only necessary to search for a decoding method for the results to fit the categories. A decoding method that already won a previous Kaggle competition with a slight modification was used.

For this particular problem, it can be argued that a more adequate deep learning model can be used. A convolutional LSTM network [10] theoretically will be the most suitable for this specific problem. However, the computer power needed for this kind of model is relatively higher.

Acknowledgments. The author wants to thank the Erasmus Mundus programme for financial support for taking the MADSAD (Master in Data Analytics) programme in the Universidade do Porto.

References

1. Chollet, F. Keras (2015). URL http://keras. io.
2. Al-Rfou, R., Alain, G., Almahairi, A., Angermueller, C., Bahdanau, D., Ballas, N., ... & Bengio, Y. (2016). Theano: A Python framework for fast computation of mathematical expressions. arXiv preprint arXiv:1605.02688.
3. de Brébisson, A., Simon, É., Auvolat, A., Vincent, P., & Bengio, Y. (2015). Artificial neural networks applied to taxi destination prediction. arXiv preprint arXiv:1508.00021.
4. Comaniciu, D., & Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. IEEE Transactions on pattern analysis and machine intelligence, 24(5), 603-619.
5. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825-2830.
6. Guo, C., & Berkhahn, F. (2016). Entity Embeddings of Categorical Variables. arXiv preprint arXiv:1604.06737.
7. Vaughn, D., & Justice, D. (2015). On the direct maximization of quadratic weighted kappa. arXiv preprint arXiv:1509.07107.
8. Chen, C. (2015). Solution for the Search Results Relevance Challenge. Retrieved from https://github.com/ChenglongChen/Kaggle_CrowdFlower/blob/master/Doc/Kaggle_CrowdFlower_ChenglongChen.pdf
9. Maaten, L. V. D., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of Machine Learning Research, 9(Nov), 2579-2605.
10. Xingjian, S. H. I., Chen, Z., Wang, H., Yeung, D. Y., Wong, W. K., & Woo, W. C. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In Advances in Neural Information Processing Systems (pp. 802-810).