Decentralised and Privacy-Aware Learning of Traversal Time Models

Thanh Le Van, Aurélien Bellet, Jan Ramon

Decentralised and Privacy-Aware Learning of Traversal Time Models

Thanh Le Van, Aurélien Bellet, and Jan Ramon

INRIA

Abstract. Estimating traversal time is an essential problem in urban computing. Traditional methods learn a predictive model from user traces collected in a central server, which potentially threatens the privacy of the users, and which may be hard to realize in an online setting where communication with large amounts of cars is needed.
In this paper, we propose a new approach to solve these problems by proposing a a privacy-friendly algorithm requiring only local communication. First, we introduce a new optimisation-based formalisation, which can take into account user-specific driving styles and the homophily of the traffic in road networks. We then discuss how we can solve this problem in a decentralised setting, where each user stores his/her sensitive data locally (without uploading it to a central server) and only shares indirect information in a peer-to-peer manner. Finally, we discuss strategies to learn the model without revealing sensitive information such as locations and user identities.

Keywords: Traversal time estimation, decentralised learning, privacy.

1 Introduction

Traversal time estimation aims to build a predictive model to estimate the time a particular user needs to travel on a road segment at a particular time point. This problem is essential to study, for example, driving behaviour of users or traffic flows in cities. There has been several proposals for solving the traversal time estimation problem from traces of GPS data of cars, see for instance [1, 3]. In general, these methods require the collected data to be stored in (or at least be accessible from) a central server in order to learn a predictive model. This approach has several shortcomings. First, the central server has direct access to sensitive information about the users, such as user location, behavior and user identity. Second, especially in an online setting, it is expensive or even intractable to quickly transfer user data to the central server, for instance because of limited network bandwidth. Hence, the main research question that we ask in this paper is whether we can build a privacy-friendly online machine learning algorithm to predict the traversal time for users.

In this extended abstract, we propose to learn a traversal time prediction model in a decentralised fashion, where users store their data in their local devices and collaboratively learn the global model. We build upon recent work

which has studied decentralised protocols for machine learning [2] (without privacy constraints). Our contribution is to formalise the traversal time prediction problem as an optimization problem which can be solved in a decentralised manner, and to introduce strategies to preserve the privacy of the users participating in the system.

The rest of the paper is organised as followed. First, we introduce an optimisation problem to formally define our traversal time prediction problem. We then discuss how the problem can be solved in the decentralised setting. Finally, we discuss strategies to learn the model without revealing user identities.

2 Methodology

2.1 The traversal time estimation problem

We start by formally defining our problem in the setting where we have unlimited storage capacity and computing power to process the training data in a central server. Our formalisation is based on the following two modelling assumptions: 1) users might have different styles of driving, e.g, some users might drive faster/slower than the others; 2) at closeby points in time, the traffic on nearby roads is similar.

Let $G = (V, E)$ be the road network, the vertices V representing crossroads and the edges E representing road segments, and let $\mathcal{K}$ be the set of all road types, e.g., highway, residential area road, narrow road, ..., with $|\mathcal{K}| = K$. The type of each road segment $e \in E$ is represented by a Boolean vector $z^e = (z_1^e, \ldots, z_K^e)^\top$, $\sum_i z_i^e = 1$. The type of the road segments can be determined from auxiliary public information such as the speed limits and the number of lanes. Let $\mathbb{T}$ be the space of all time points and $\mathcal{T}$ be a partitioning of $\mathbb{T}$ into time slots, $|\mathcal{T}| = T$. Given time $t \in \mathbb{T}$, the function $\tau(t)$ gives the time slot t belongs to, e.g., $\tau(\text{“18/05/2017,08:09”}) = \text{weekday\_morning}$. We also denote with $\overline{d}_{e,t}$ the observed average traversal time on road $e \in E$ during timeslot $t \in \mathcal{T}$. The driving style of each user $u \in \mathcal{U}$ is represented by vector $w^u = (w_1^u, \ldots, w_K^u)^\top \in \mathbb{R}^{1\times K}$. Each weight $w_i^u, i = 1 \ldots K$, indicates how fast/slowly on average the user u drives in roads of type i. Let $\mathcal{X}$ be the set of training examples, $\mathcal{X} = \{x_i\}_{i=1}^N$. Each $x_i = (x_i^u, x_i^e, x_i^t, x_i^d)$ is a tuple specifying the user, the road segment, the time, and the observed duration the user spent on travelling the road segment at that time. Let $c_i \in \mathbb{R}, i = 1 \ldots N$, be the delay added by congestion for the corresponding training example x_i.

The machine learning task we consider is to find the user weights $\mathbf{w} = (w^1, \ldots, w^{|\mathcal{U}|})$ and the congestion values $\mathbf{c} = (c_1, \ldots, c_N)$ which together min-

imise the following loss function:

$$\mathcal{L}(\mathbf{w}, \mathbf{c}) = \sum_{i=1}^{N} (x_i^d - (z^{x_i^e})^\top w^{x_i^u} \overline{d}_{x_i^e, \tau(x_i^t)} - c_i)^2 + \lambda_1 \sum_{i=1}^{N} \sum_{j=1}^{N} (c_i - c_j)^2 \exp(-\lambda_2 (d(x_i^e, x_j^e)^2 + (x_i^t - x_j^t)^2)) + \lambda_3 \sum_{i=1}^{N} c_i^2 \quad (1)$$

where $d(,)$ returns the distance between two road segments, and $\lambda_1, \lambda_2, \lambda_3 \geq 0$ are the hyperparameters of the method.

The first term in (1) implies that the user weights $\mathbf{w}$ should be able to explain how fast/slowly the users drive. The difference between the observed traversal time and the weight value of the user for the given road type and time-of-day is assumed to be caused by (additional) congestion, which is represented by c_i. The second term favours congestion values that are correlated over time and space. Finally, the last term is an L2 regularisation to prevent overfitting (we prefer to explain the data assuming a smaller amount of congestion). Note that the problem (1) is jointly convex in the variables $\mathbf{w}$ and $\mathbf{c}$.

When data is stored in a central server, the optimisation problem (1), which is a quadratic program, can be solved by standard optimization algorithms such as (stochastic) gradient descent. We can then use the obtained solution to predict future traversal time for the users. In particular, we can estimate the congestion situation from the data up to the current point in time, we assume that the congestion situation evolves only slowly (next to the modeled dependency of the time-of-day and day-of-week), and we can combine it with the user weights to predict traversal times in the near future.

2.2 Decentralised Estimation

The main question that we ask in this section is whether we can still solve the machine learning problem described in Section 2.1 in the decentralised setting. In this context, users keep their information (essentially their observed traversal times, weights w^u and trajectories) on their local devices and only exchange the necessary information in a peer-to-peer fashion to collaboratively learn the model. Being able to solve this problem in the decentralised setting is an important step towards a privacy-preserving solution.

We assume that each user can broadcast information to its *neighbours* (fellow users who are currently in the same local region in the map of the road network). Then, building upon the work of [2], we propose to solve the traversal time estimation problem using the following decentralised algorithm:

- At time step t_0 (initialisation): each user u has a local training set $\mathcal{X}^u(t_0) = \{(x_i^e, x_i^t, x_t^d)\}_{i=1}^{N_u(t_0)}$ of recently observed traversal times and performs the following steps:

  - $w^u(t_0) = (w_1^u, \ldots, w_K^u)$ is estimated by averaging the observed traversal time over the road segment types (if no training examples are present yet for a road segment type, some default value is used, e.g., the speed limit));
  - $c^u(t_0) = (c_1, \ldots, c_{N_u(t_0)})$ is estimated by taking the subtraction of the observed traversal time and the corresponding weight;
  - Broadcast $c^u(t_0)$ to their neighbors $\mathcal{N}(u)$.
- At time step $t > t_0$: each user u performs:
  - Given the local dataset $\mathcal{X}^u(t)$ and the congestion variables $c^{u'}(t-1), u' \in \mathcal{N}(u)$ obtained from neighbors,[1] compute the updated values $c^u(t)$ and $w^u(t_0)$ which minimise (1);
  - Broadcast $c^u(t)$.

When the local dataset and spatial location of each user are fixed during learning, this decentralised algorithm will converge to the solution of (1) due to the local nature of the dependence between congestion values, as seen in the second term of (1). In the on-line setting (users move around and collect additional data), we expect the solution given by the algorithm at any time step to be close to the optimal solution as the new data only influences the congestion values of the recent past, see again (1). In other words, while the "target" (the current traffic situation) is moving, we can show that if users exchange information at a sufficient rate their estimation of the congestion situation in the different areas will always be close to the actual situation.

2.3 Privacy

In our algorithm, users only communicate the computed congestion values $c^u(t)$ to nearby users. From a privacy perspective, this is much better than uploading all traversal data to the central server. However, the users may still reveal their identities when broadcasting the information. The question that we would like to study is how a particular user can anonymously broadcast information while he/she is still able to convince the others that the message comes from a trusted source (a valid user of the system).

We propose an answer to the above question based on the concept of *digital signatures*. In the field of cryptography, many different approaches are studied, a simple strategy could be the following. There is a trusted party, e.g., a government, who can verify identities. This identity authority has a public/private key pair. Each driver generates a set of public/private key pairs, and registers them with the identity authority, obtaining for each public key a certificate that that public key is owned by a registered driver. While driving, the car broadcasts its messages and signs them using one of his registered key pairs, for a trajectory which is sufficiently short so that one can't identify the driver from the trajectory. Then, the car switches to another registered key pair. Receivers

[1] The local dataset $\mathcal{X}^u(t)$ at time t contains $\mathcal{X}^u(t-1)$ and possibly additional training examples generated by the user since the previous update.

of messages see (i) the message, (ii) the public key, (iii) the ability of the sender to decrypt messages encrypted using that public key and (iv) the certificate of the identity authority that the presented public key is sound and connected to a registered person. Together, the provided information doesn't reveal the identity of the sender (unless the identity authority would collude) but still proves the message is trustworthy (and in case of proven fraud, one could trace back the sender with the help of the identity authority).

3 Conclusion

Our approach is still preliminary and we aim at further improving its two technical components. First, we intend to optimize the decentralized learning algorithm, of which we may be able to increase the convergence quality. Second, we want to improve the cryptographic elements guaranteeing privacy, to maximize security and minimize communication overhead.

More generally, the problem of traversal time estimation studied here is only one of the many tasks which connected cars may want to perform collaboratively and in a privacy-friendly way. We hope that our method can be generalized to other problems, such as parking space search, map making/updating or ride sharing.

References

1. Idé, T., Kato, S.: Travel-time prediction using Gaussian process regression: A trajectory-based approach. In: Proceedings of SIAM International Conference on Data Mining (SDM-09). pp. 1185–1196 (2009)
2. Vanhaesebrouck, P., Bellet, A., Tommasi, M.: Decentralized Collaborative Learning of Personalized Models over Networks. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS-17). pp. 509–517 (2017)
3. Wang, Y., Zheng, Y., Xue, Y.: Travel time estimation of a path using sparse trajectories. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-14). pp. 25–34 (2014)