Toward Crowdsourced Transportation Mode Identification: A Semisupervised Federated Learning Approach

Reporter: Qihan Huang
Supervisor: Jing Zhang

November 16, 2022

Outline

1. Introduction
2. Preliminaries
3. Methodology
4. Experiments

Outline

1. Introduction
2. Preliminaries
3. Methodology
4. Experiments

Background

Privacy-preserving transportation mode identification (TMI) is among the key challenges toward future intelligent transportation systems. With recent developments in federated learning (FL), crowdsourcing has emerged as a promising costeffective data source for training powerful TMI classifiers without compromising users data privacy.

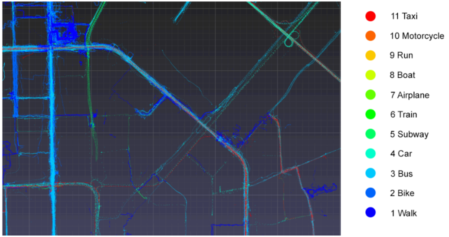

Figure 1: Transportation Mode Identification

Problem of TMI

However, existing TMI approaches have relied heavily on the availability of transportation mode labels, which is often limited in real-world applications.

- Training highly accurate deep learning models primarily requires immense amounts of data.
- Existing TMI research typically employs machine or deep learning models in fully supervised settings, thereby assuming the availability of sufficient GPS trajectories labeled by transportation mode at training time.

Problem of TMI

Data crowdsourcing seeks to alleviate this issue by allowing multiple users to collectively generate large data sets in a distributed manner. A centralized server is always set to receive the crowdsourced data and train them. These data are usually transmitted to the centralized server in unprocessed forms. However, information implied by the data may have a strong connection to the users privacy, which raises a privacy concern.

Contributions

The main contributions of this article are summarized as follows.

- They propose MTSSFL, a new FL-based semi-supervised learning scheme that incorporates an ensemble-learningbased TMI model.
- They introduce consistency updating, a novel approach for training local models that only possess unlabeled data guided by the centralized model trained on few labeled data.
- They design an exponential moving average (EMA)-based secure parameter aggregation mechanism termed meanteacher-averaging to improve the global model without additional training.
- They conduct extensive case studies to assess the performance of MTSSFL on IID and non-IID data.

Outline

1 Introduction

2 Preliminaries

3 Methodology

4 Experiments

GPS-Based Transportation Mode Identification

Definition (GPS Segment)

Let $\mathcal{F}_i$ denote the calculated features of the ith GPS point in a segment. We can then reform GPS segment k of length T as $\mathrm{GPS}_k = \{\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_T, \mathit{label}_k\}$, where label_k is the corresponding transportation mode of the segment. Our aim is to train a deep learning classifier to classify $\mathrm{GPS}_k, \forall k$, by transportation mode. This is formulated as

$$\mathrm{GPS}_k\left[\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_T\right] \xrightarrow{f(\cdot)} \mathrm{GPS}_k\left[\text{ label }_k\right]$$

where $f(\cdot)$ denotes the classification function learned by the classifier model.

In this work, we propose an FL framework for crowdsourced TMI. We use the term publisher to refer to the initiator of the crowdsourcing task, who also owns a central server. We refer to the local (distributed) entities as workers.

Statement

Let q denote the total number of workers; we then have the set of workers $\mathcal{C} = \{C_1, C_2, \ldots, C_q\}$. Each worker C_i uses their respective database D_i to store their sensed GPS trajectories, resulting in the database set $\mathcal{D} = \{D_1, D_2, \ldots, D_q\}$. In FL, worker C_i uses their locally stored data to train their local model $M_i \in \mathcal{M} = \{M_1, M_2, \ldots, M_q\}$, where the learned parameters of M_i are denoted by ϕ_i. [a]

[a] This work assumes that both the publisher and the workers are reliable and low-latency communicators.

Each worker is likely to have more data corresponding to some transportation modes than others. This will inevitably lead to a skewed local class distribution, i.e., nonindependent and identically distributed (non-IID) data.

Definition (The metric R to measure the level of non-IID)

Specifically, the class distribution of database D_i belonging to worker C_i is defined as $P_i = [p_1, p_2, \ldots, p_c] \in \mathbb{R}^c$, where p_j is the fraction of the j th class in D_i and c denotes the total number of classes. R can be defined as

$$R = \frac{1}{2} \sum_{1 \leq i < o \leq q} \|P_i - P_o\|_1 \frac{1}{q(q-1)/2}$$

where $\|\cdot\|_1$ denotes the L_1 norm, $\|P_i - P_o\|_1$ is the variation distance, and $q(q-1)/2$ corresponds to the total number of worker pairs.

Outline

1 Introduction

2 Preliminaries

3 Methodology

4 Experiments

This segmentation method is based on the intuition that the transportation mode separating any other two has to be walk.

As such, this method first classifies each GPS point as walk or nonwalk according to velocity and acceleration thresholds before forming single-transportation-mode segments by aggregating adjacent points with the same predicted label.

Using the same thresholds defined by [1], we split all trajectories into a total of T^* single-transportation-mode segments $\{\text{GPS}_1, \text{GPS}_2 \ldots, \text{GPS}_{T^*}\}$

[1] TY. Zheng, L. Liu, L. Wang, and X. Xie, Learning transportation mode from raw GPS data for geographic applications on the Web, in Proc. 17th Int. Conf. World Wide Web, 2008, pp. 247256.

We first calculate the relative distance between every two consecutive GPS records using the Vincenty Formula, which can be denoted as

$$d_i = \text{ Vincenty } \left(\text{lat}_i, \text{lng}_i; \text{lat}_{i+1}, \text{lng}_{i+1}\right)$$

Based on d_i, we can then estimate speed s_i, acceleration a_i, and jerk j_i for the i th GPS point as follows:

$$\begin{aligned} s_i &= \frac{d_i}{\Delta t_i}, \quad 1 \leq i \leq T, \quad s_T = s_{T-1} \\ a_i &= \frac{s_{i+1} - s_i}{\Delta t_i}, \quad 1 \leq i \leq T, \quad a_T = 0 \\ j_i &= \frac{a_{i+1} - a_i}{\Delta t_i}, \quad 1 \leq i \leq T, \quad j_T = 0 \end{aligned}$$

where T is the number of GPS records in a GPS segment. In this way, a motion feature vector $x_i \equiv \mathcal{F}_i = (d_i, s_i, a_i, j_i)$ can be extracted for the ith GPS point.

TMI Model

Overview

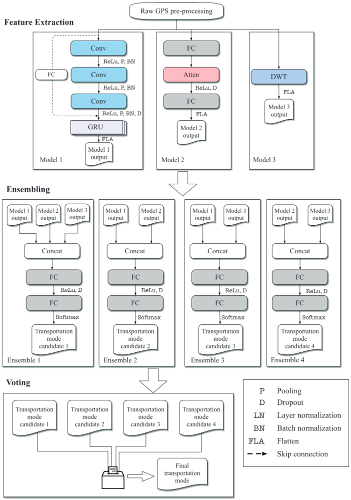

Figure 2: Framework of TMI model.

Model 1

This submodel first employs three convolution layers with skip connections to capture the spatial features within the input motion features $\mathbf{X}$. Next, we stack eight layers of gated recurrent units (GRUs) with hidden size 16 to exploit the temporal dependencies within the input data.

Gated Recurrent Units (GRU)

$$r_t = \sigma\left(W^{(r)}\mathbf{X}_t + U^{(r)}h_{t-1}\right)$$
$$z_t = \sigma\left(W^{(z)}\mathbf{X}_t + U^{(z)}h_{t-1}\right)$$
$$h'_t = \tanh\left(W\mathbf{X}_t + r_t \odot Uh_{t-1}\right)$$
$$h_t = z^t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t$$

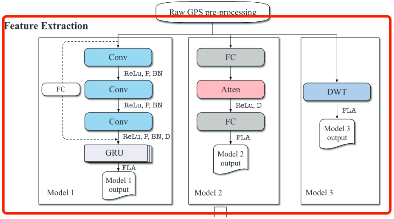

Figure 3: Framework of TMI model.

Model 2

This submodel incorporates an attention module between two linear layers. The orchestra of multihead enables the model to collectively involve knowledge learned from multiple representation subspaces. The processes can be formulated as

$$\mathbf{X}^A = \operatorname{concat}\left(hd_1, hd_2, \ldots, hd_{n^*}\right) W^O$$

where

$$hd_i = \operatorname{softmax}\left(\frac{QW_i^Q \left(KW_i^K\right)^T}{\sqrt{d_k}}\right) VW_i^V$$

where Q, K, and V are the query, key, and value, and the mechanism requires that they are all $\mathbf{X}$; $d_k = (hidd_a/n^*)$ denotes the dimension of keys where hidd_a denotes the hidden size of the attention module, which is set to 128 ; hd_i denotes each attentional head and n^* is the number of heads; concat $(\cdot)$ represents the concatenation operation; W^O, W^Q, W^K, and W^V are the corresponding weight matrices.

Model 3

To better analyze the data characteristics, a wavelet representation-based feature extracting approach named discrete wavelet transform (DWT) is adopted in Model 3 to further exploit the hidden time-domain feature from the feature vectors.

Specifically, DWT employs discrete wavelets $\psi_{a,b}(t)$ to convolve the input, which can be defined as

$$\psi_{a,b}(t) = \frac{1}{2^a}\psi\left(\frac{t}{2^a} - b\right), a, b \in \mathbb{Z}$$

where $\psi(t)$ is predefined mother wavelet, a denotes the oscillatory level, and b denotes the shifted position of DWT.

TMI Model

Ensembling

Multiview ensemble learning (MEL) is a type of semisupervised learning aiming to train different learning models with different views of the original data. Following the concept of MEL, we create four ensembles by concatenating the outputs of the three submodels as:

$$\mathbf{X}^{E1} = \text{concat}\left(\mathbf{X}^{M1}, \mathbf{X}^{M2}, \mathbf{X}^{M3}\right)$$

$$\mathbf{X}^{E2} = \text{concat}\left(\mathbf{X}^{M1}, \mathbf{X}^{M2}\right)$$

$$\mathbf{X}^{E3} = \text{concat}\left(\mathbf{X}^{M1}, \mathbf{X}^{M3}\right)$$

$$\mathbf{X}^{E4} = \text{concat}\left(\mathbf{X}^{M2}, \mathbf{X}^{M3}\right)$$

where $\mathbf{X}^{M1}, \mathbf{X}^{M2}$, and $\mathbf{X}^{M3}$ are the outputs of the three submodels; $\mathbf{X}^{E1}, \mathbf{X}^{E2}, \mathbf{X}$ and $\mathbf{X}^{E4}$ denote the four ensemble tensors for corresponding ensembles.

Finally, we use a voting strategy to decide the final transportation mode classification from the predictions of the four ensembles.

Specifically, we regard the inferred transportation mode of each ensemble as a candidate. A hard voting procedure is adopted, in which we select the final class as the one with the largest sum of votes among the classes predicted by the four candidates. In the case of a tie, we consider the transportation mode classification provided by Ensemble 1 as the final classification, since Ensemble 1 is trained on more inputs and is thus expected to be more reliable than the others.

Mean Teacher Semisupervised Federated Learning

Architecture, Participants, and Communication Protocol

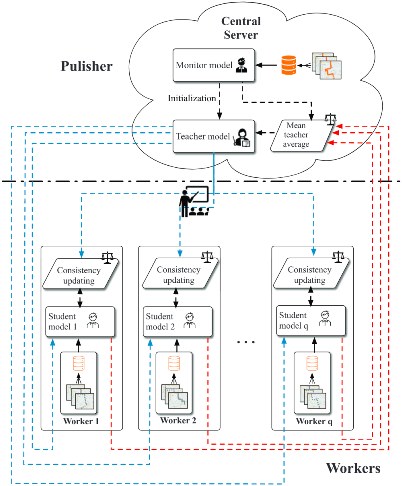

Figure 4: Framework overview of the proposed MTSSFL.

Pseudolabeling Approach

The pseudolabeling approach considers the training balance between labeled data and unlabeled data, which defines a loss function as

$$J = \frac{1}{B} \sum_{m=1}^{B} \sum_{i=1}^{c} J\left(y_i^m, \hat{y}_i^m\right) + \alpha(t) \frac{1}{B'} \sum_{m=1}^{B'} \sum_{i=1}^{c} J\left(y_i'^m, \hat{y}_i'^m\right) \tag{1}$$

where B and B' denote the size of minibatch in labeled and unlabeled data, respectively; $\hat{y}_i^m$ and $\hat{y}_i'^m$ are the output of m samples in labeled and unlabeled data, respectively; y_i^m and $y_i'^m$ are the labels of m samples in labeled and unlabeled data, respectively; $\alpha(t)$ is a balancing weight.

A consistency cost J_{con} is introduced to measure the distance between the prediction of student model and teacher model on unlabeled data using L2 loss, which can be formulated as

$$J_{\text{con}} = \frac{1}{c} \sum_{i=1}^{c} \left(\dot{f}\left(x_i'^m\right) - f\left(x_i'^m\right) \right)^2 \tag{2}$$

where $\dot{f}$ and f denote the teacher and student TMI model, respectively; $x_i'^m$ is the unlabeled data. Furthermore, we can define a final loss function J_s for the student model's update by combining (1) and the second half of the (2), which can be formulated as

$$J_s = \frac{1}{B'} \sum_{m=1}^{B'} J_{\text{con}} + \frac{1}{B'} \sum_{m=1}^{B'} \sum_{i=1}^{c} J\left(y_i'^m, f\left(x_i'^m\right)\right) \tag{3}$$

In this work, we proposed a mean-teacher-averaging as a secure parameter aggregation mechanism for the TMI models based on the approach proposed in [2]. Tarvainen and Valpola proposed an EMA-based approach to update the model parameter of the teacher model at communication round t, which can be defined as

$$\dot{\phi}_t = \delta\dot{\phi}_{t-1} + (1-\delta)\phi_t \tag{4}$$

where $\dot{\phi}$ and ϕ are the parameters of teacher model and student model, respectively; and δ is a smoothing coefficient.

[2] A. Tarvainen and H. Valpola, Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results, in Advances in Neural Information Processing Systems. Red Hook, NY, USA: Curran, 2017, pp. 11951204.

We extend this method into a model parameter aggregation method applicable to FL. Specifically, we first use the naive FedAvg [56] to aggregate model updates of student and monitor models, which can be formulated as

$$\phi_{\Sigma,t} = \frac{1}{1+q}\left(\sum_{i=1}^{q}\phi_{i,t} + \ddot{\phi}_t\right) \tag{5}$$

where q denotes the number of student models (i.e., workers); $\ddot{\phi}$ is the model parameter of monitor model; and ϕ_Σ represents the aggregated parameters from student and monitor models by FedAvg. Subsequently, we incorporate ϕ_Σ by (4) into (5), and finally, obtain the aggregation mechanism for MTSSFL as

$$\dot{\phi}_t = \delta\dot{\phi}_{t-1} + (1-\delta)\phi_{\Sigma,t}. \tag{6}$$

Outline

1. Introduction
2. Preliminaries
3. Methodology
4. Experiments

Geolife

An open data set of real-world GPS trajectories by Microsoft Research Asia. It contains a total of 17 621 trajectories collected by 182 users over 2090 days, with a total traveled distance of 1 292 951 km. Among these users, only 69 have labeled parts of their trajectories by transportation mode.

Our simulations were conducted on a server equipped with eight NVIDIA GeForce RTX 2080 GPUs and an Intel Xeon E5-2620 v4 CPU. All neural networks were developed using PyTorch v1.6.

Table 1: Comparision of TMI Methods

Method	Accuracy	Semi-supervised	Secure crowdsourcing
MLP	33.1%	–	–
SVM	47.0%	–	–
KNN	54.9%	–	–
CNN	83.6%	–	–
LSTM	81.7%	–	–
SPL	72.5%	✓	–
SGAN	83.1%	✓	–
SECA	73.2%	✓	–
STS	59.1%	✓	–
ELSTM	90.3%	✓	–
MTSSFL	89.2%	✓	✓

Table 2: Sensitivity of TMI Accuracy to Percentage of Unlabeled Data

Method	Accuracy (%)					
	$\gamma = 0.99$	$\gamma = 0.95$	$\gamma = 0.90$	$\gamma = 0.80$	$\gamma = 0.50$	$\gamma = 0$ (Supervised)
SPL	50.9	56.0	61.8	68.6	72.5	75.4
SGAN	68.4	77.7	80.5	82.1	83.1	83.8
SECA	52.0	56.1	62.9	69.3	73.2	76.8
STS	50.7	53	50.6	54.4	57.7	59.1
ELSTM	84.8	86.5	89.0	90.0	90.8	91.5
MTSSFL	82.4	83.1	85.7	87.3	89.2	91.4
MTSSFL-non-IID	82.3	82.5	85.7	86.9	89.0	91.3

Identification Accuracy

Table 3: Accuracy and Training Time of MTSSFL for Different Numbers of Local Epochs

E_l	Accuracy (%)	Training time (s)
1	89.1	750 ± 30
2	88.9	1480 ± 30
3	88.8	1920 ± 45
4	88.3	2620 ± 70
5	89.2	3110 ± 80

Thank you!