Short-term traffic flow forecasting using a distributed spatial-temporal model

A A Agafonov[1] and A S Yumaganov[1]

[1]Samara National Research University, Moskovskoye shosse 34, Samara, Russia, 443086

Abstract. In this paper, we consider the problem of short-term traffic flow prediction. We propose a distributed model for short-term traffic flow forecasting based on the k nearest neighbors method, that takes into account spatial and temporal traffic flow distribution. To consider spatial-temporal correlations, we partition a transportation graph in clusters by an area and describe traffic flow by a feature vector defined for each cluster. The proposed model is implemented as a MapReduce based algorithm on an Apache Spark framework. The proposed traffic flow prediction model is tested using the actual average traffic speed data over a road network in Samara, Russia.

CEUR-WS.org/Vol-2210/paper51.pdf

1. Introduction

Issues related to the traffic flow management are common in every major city around the world. Traffic congestion leads to economic, environmental and social problems, which emphasizes the importance of the transport planning and logistics. To solve these problems, it is important to obtain accurate and timely traffic flow information. Due to this fact, road traffic forecasting has been a subject of active research for more than 40 years.

Efforts devoted to mitigate the traffic congestion problem are usually classified in three directions: modification of the transport infrastructure, improving the operational quality of the public transport and managing traffic flows. The first and the second directions are often limited by economic or social factors, while the traffic flow management has been continuously improving due to the development of traffic data collecting and processing technologies.

Recently, much attention has been paid to the data-driven programming paradigm. This interest is explained by the development of new technologies, methods and techniques for massive data processing within the Big Data concept, the availability of multiple data sources for predicting traffic flows, and the "open data" idea, that some data should be freely available to everyone to use, without restrictions from copyright, patents or other mechanisms of control.

Short-term traffic flow forecasting considers the traffic prediction problem on the basis of current and archived information about the traffic flows state. A review of the latest achievements in the road traffic forecasting field, as well as the main unresolved technical challenges, can be found in [1]. Most research on this topic has focused on developing methods for modeling the characteristics of traffic flows (for example, density or speed). An overview of the methods of short-term traffic forecasting presented in the [2]. These methods can be classified into three categories:

1) Parametric methods [2, 3], including time series models [4], state space models, etc.

2) Non-parametric methods [5], including models of artificial neural networks [6], k-nearest neighbor (kNN) [7], support vector regression (SVR) [8].
3) Hybrid methods that combine parametric and non-parametric methods [9, 10].

However, these methods have both advantages and disadvantages when working under different conditions using different datasets, so it is hard to conclude that one method significantly superior others.

In this paper, we propose an approach based on the k nearest neighbor algorithm - one of the main non-parametric techniques for short-term traffic flow prediction. Results presented in [5, 11, 12] showed that kNN outperformed other modern comparable models, including ANN, SARIMA, random forest, and Naïve Bayes. However, if the sample data size is too large, kNN may not be suitable for real-time prediction due to the computational costs. Despite this issue, a relatively small number of works are devoted to the short-term traffic flow forecasting with a focus on processing big traffic data using the distributed computations, in particular, using the MapReduce framework [12, 13].

In this article, we consider a problem of short-term traffic flow forecasting for 10 minutes ahead. We focus on developing a distributed forecasting model based on the weighted kNN algorithm, taking into account the spatial and temporal characteristics of the transport flows in the spatially compact area of a transport network. For distributed data processing, we use MapReduce processing model implemented in the open source cluster-computing framework Apache Spark. Experimental analysis on real-world traffic data sets allows us to conclude that the proposed model has a high prediction accuracy and reasonable execution time, sufficient for real-time prediction.

The paper is organized as follows. Section 2 contains the formulation of the problem. The proposed model and its distributed implementation are described in detail, respectively, in Sections 3 and 4. In section 5, we provide experimental results of the proposed model and verify the accuracy of the proposed approach. Finally, we conclude the paper, and then present possible directions for further research.

2. Problem formulation

A road network is considered as a directed graph $G = (V, E))$, with nodes $V, N_V = |V|$ representing the road intersections and edges $E, N_E = |E|$ denoting road segments.

Let V_t^j denotes an observed traffic flow characteristic on an edge $j \in E$ at time interval t. As a traffic flow characteristic can be used travel time, average speed, density or flow.

In this work as a predicted traffic flow characteristic for the experimental study, we use the average traffic speed.

The short-term traffic flow forecasting problem can be formulated as follows: given a graph $G(V, E)$ and sequence $\{V_t^j\}, j \in E, t = 1, 2, \ldots T$ of observed traffic flow data, predict the traffic flow characteristic $\hat{V}_{t+\Delta}^j, j \in E$ at time interval $(t + \Delta)$ for a predefined prediction horizon Δ.

3. Proposed model

In this paper, we propose a short-term traffic flow forecasting model based on non-parametric regression k-nearest neighbors algorithm. To apply the kNN method to the traffic flow prediction problem, it is necessary to solve the following tasks:

1. Define a feature vector to describe traffic flow.
2. Define a suitable distance metric to determine the proximity between a feature vector describing current traffic flow characteristics and feature vectors describing historical traffic flow observations.

3. Define a prediction function to forecast a traffic flow characteristic by selected nearest neighbors.

These challenges are described in the next subsections.

3.1. Feature vector

The choice of a feature vector in the kNN method depends on the particular application of the method in practice. To solve the traffic flow prediction problem, it is reasonable to use a feature vector that takes into account spatial and temporal correlations of the traffic flow characteristics.

In the paper [12] as a feature vector authors used traffic flow of targeted road segment j, downstream road segment $j-1$ and upstream road segment $j+1$ for T time intervals:

$$(V_{t-T}^{j}, \ldots, V_{t-1}^{j}, V_{t}^{j}, V_{t-T}^{j-1}, \ldots, V_{t-1}^{j-1}, V_{t}^{j-1}, V_{t-T}^{j+1}, \ldots, V_{t-1}^{j+1}, V_{t}^{j+1}) \tag{1}$$

However, such feature vector does not consider traffic flow on adjacent segments. In addition, in some cases, the upstream / downstream road segment cannot be uniquely determined. Therefore, to describe traffic flow, it is proposed to use a feature vector that taking into account the traffic flow characteristics in the spatially-compact cluster of the transport network graph.

In this paper, we define the feature vector as follows:

1. The transportation network graph is partitioned into several spatially compact clusters $\{G_i\}$. In each cluster i the feature vector is defined as follows:

$$\{V_t^j\}^i, j \in G_i, t = t_{cur} - T, \ldots, t_{cur} \tag{2}$$

2. For the defined feature vector $\{V\}^i$ in the cluster i dimension reduction is performed using principal component analysis procedure. Result of this procedure is a new feature vector $\{X_n\}^i, n = 1, \ldots, N$.
3. Proposed feature vector for each road segment $j \in E$ is defined from the initial feature vector of the targeted road segment j and the feature vector of the cluster i such that $j \in G_i$:

$$S_j = (\{V_t^j\}, \{X_n\}^i), \quad j \in G_i; \quad t = t_{cur} - T, \ldots, t_{cur}; \quad n = 1, \ldots, N. \tag{3}$$

Graph partitioning algorithm is described in the next subsection.

3.2. Graph partitioning

Let each edge $i \in E$ corresponding to the road segment e_i with two terminal points $x_{start}^i = \left(x_{start}^0, x_{start}^1\right)^i$ and $x_{end}^i = \left(x_{end}^0, x_{end}^1\right)^i$.

Then graph partitioning by an area can be described as follows:

1. Choose the numbers of clusters M_0, M_1.
2. The cluster G_m with index $m = m_0 M_1 + m_1, \left(m_0 = \overline{0, M_0 - 1}; m_1 = \overline{0, M_1 - 1}\right)$ contains the edges $i \in E$, for which coordinates of at least one of the corresponding terminal points are inside the corresponding rectangular area Π_{m_0,m_1}:

$$G_{m_0 M_1 + m_1} \equiv \left\{ i \in E : x_{start}^i \in \Pi_{m_0,m_1} \vee x_{end}^i \in \Pi_{m_0,m_1} \right\}, \tag{4}$$

where

$$\Pi_{m_0,m_1} \equiv \left[x_{min}^0 + \frac{m_0}{M_0}\left(x_{max}^0 - x_{min}^0\right), x_{min}^0 + \frac{m_0+1}{M_0}\left(x_{max}^0 - x_{min}^0\right) \right] \times \left[x_{min}^1 + \frac{m_0}{M_0}\left(x_{max}^1 - x_{min}^1\right), x_{min}^1 + \frac{m_0+1}{M_0}\left(x_{max}^1 - x_{min}^1\right) \right],$$

$$x^s_{min} = \min_{\substack{v=\{start,end\}\\ i\in E}} x^{s,i}_v, \quad x^s_{max} = \max_{\substack{v=\{start,end\}\\ i\in E}} x^{s,i}_v, \quad s = 0, 1.$$

The number of clusters along the vertical and horizontal axis M_0, M_1 is chosen empirically. We assume, that each edge of the graph can get into only one cluster.

3.3. Proximity measure

To define the proximity between the feature vectors, it is necessary to determine a suitable distance metric. Different distance functions between feature vectors are available in the literature, including Euclidean, Mahalanobis, Hamming distance.

In this paper, we use a weighted Euclidean distance, modified to use the feature vector describing transportation network clusters. The distance is considered separately for parts of the feature vector describing traffic flows on the current segment $\{V\}$ and in the corresponding cluster $\{X\}$.

$$d(S, \bar{S}^i) = \sqrt{\sum_{t=1}^{T} \beta^{T-t+1} \left(V_t - \bar{V}^i_t\right)^2} + \alpha \sqrt{\sum_{n=1}^{N} \left(X_n - \bar{X}^i_n\right)^2}. \tag{5}$$

where $0 < \alpha \leq 1$,
T denotes the total number of time intervals in the feature vector,
N denotes the total number of elements in the feature vector describing the graph cluster,
S is the feature vector describing current traffic flow,
$\bar{S}^i$ is the feature vector describing ith historical traffic flow,
$V_t, \bar{V}^i_t$ are the feature vectors values representing respectively current and historical traffic flows on the selected road segment at time interval t,
$X_n, \bar{X}^i_n$ are the nth feature vectors values representing respectively current and historical traffic flows in the graph cluster.

3.4. Prediction function

The traditional approach for estimating the value in k-NN regression is to choose the average or the weighted average of the values of its k nearest neighbors [5].

A prediction function by the average has the following form:

$$\hat{X}_{T+1} = \frac{1}{k} \sum_{k=1}^{K} X^k_{T+1} \tag{6}$$

where $\hat{X}_{T+1}$ is the predicted traffic flow value at the next time interval $T+1$, X^k_{T+1} is the traffic flow value of the kth nearest neighbor at the time interval $T+1$, K is the total number of the neighbors.

A prediction function by the weighted average has the following form:

$$\hat{X}_{T+1} = \sum_{k=1}^{K} \frac{d_k^{-1}}{\sum_{k=1}^{K} d_k^{-1}} X^k_{T+1} \tag{7}$$

where d_k denotes the distance between the feature vector describing the current traffic data and the kth nearest neighbors.

We use the prediction function by the weighted average.

4. MapReduce implementation

The proposed model of traffic flow prediction uses a large amount of current and historical traffic flow data. To improve the efficiency of the proposed model, we implement it on the basis of MapReduce model [14] for distributed computing using Apache Spark engine [15].

MapReduce provides parallel processing of big amount of data in computing clusters. MapReduce model usually consists of three main steps: Map, Shuffle and Reduce. Figure 1 illustrates a computation flowchart of the proposed model based on MapReduce engine.

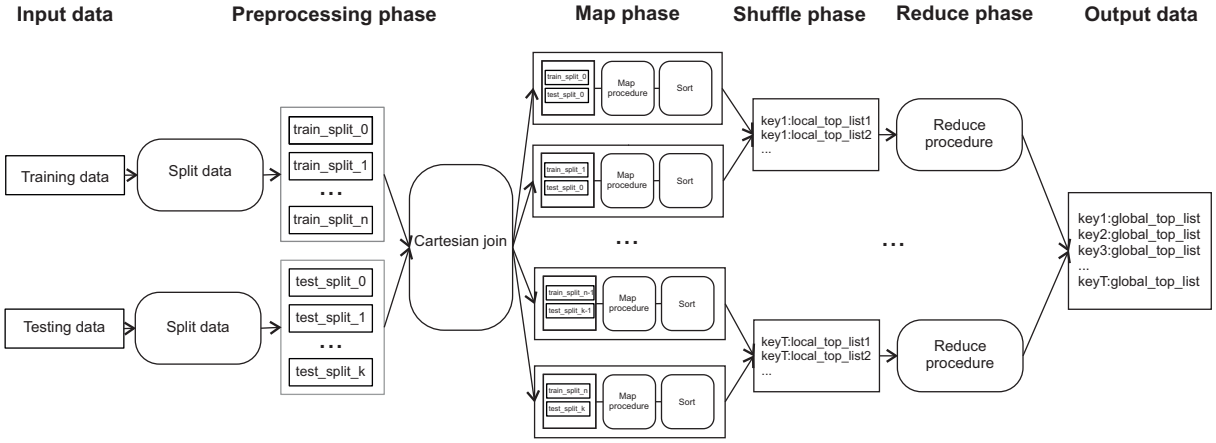

Figure 1. Data flow of MapReduce in the proposed model.

As illustrated in Figure 1, the first step is a preparation of input data for Map phase. At first, the historical and test data are divided into partitions. The optimal number of such partitions depends on the amount of processed data and the number of computing nodes. As mentioned in official Apache Spark documentation, the recommended value of partitions is 2-3 partition per CPU core in the cluster. Then, ordered pairs of historical and test data partitions are formed using the Cartesian product. Next, in the Map phase, a map function is applied to each pair of partitions. This function returns an intermediate set of key / value pairs - the test element / local list of k nearest neighbors. At the Shuffle phase, the key-value pairs are grouped and transferred to the reduce functions. At the final Reduce step for each test data element, the set of local k nearest neighbors lists is converted to the resulting (global) list of k nearest neighbors. The resulting lists of k nearest neighbors are subsequently used to find the predicted value traffic flow.

The results of evaluating the efficiency of the proposed model based on the MapReduce concept are presented in Section 5.

5. Experiments

In this work, in the experimental study, we predict average traffic speed in the city of Samara, Russia for short-term prediction horizon 10 minutes. The dataset contains records for 34 days. We compare the proposed model with the model described in [12]. This model uses feature vector in form (1), where the feature vector considers time domain and upstream / downstream road segments (denoted below as "TDUD"). Our model we denote as "Clusters" because the feature vector considers spatial-temporal correlations in graph clusters.

During testing, these models are performed on each day (test set) and the remaining days considered as a historical dataset (training set). Then the average performance across the full data set is calculated.

We conduct the experiments on an Apache Spark cluster. The traffic flow was predicted for a small area contained 698 road segments (Figure 2). Each road segment is considered as two edges with different directions. The total size of the dataset was 3.5 GB.

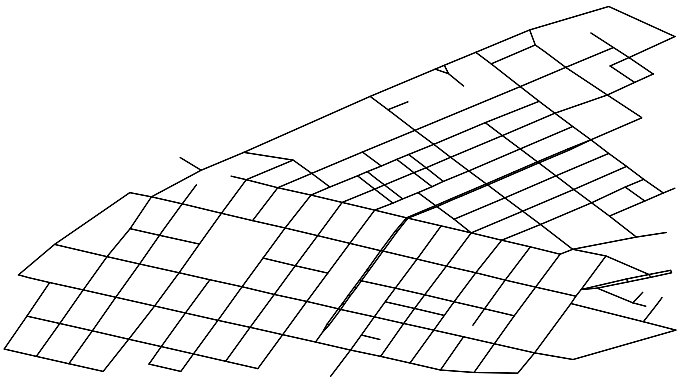

Figure 2. Samara city area.

To compare the performance of the proposed model, we use two standard metrics: mean absolute error (MAE) and mean absolute percentage error (MAPE) that can be formulated as:

$$\text{MAPE} = \frac{1}{n}\sum_{t=1}^{n}|V_t - \hat{V}_t| \tag{8}$$

$$\text{MAE} = \frac{1}{n}\sum_{t=1}^{n}\frac{|V_t - \hat{V}_t|}{V_t} \tag{9}$$

where V_t is the actual value of traffic flow at time interval t, $\hat{V}_t$ is the predicted value for the same time interval t, n is the total number of traffic flow observations.

Table 1. Algorithms comparison

	MAE	MAPE
TDUD	0.157	2.92
Clusters	0.149	2.82

Figure 3 and Figure 4 show the prediction result by the MAE and MAPE metrics for different days, respectively.

Based on the results above, we can conclude that considering the spatial-temporal correlation of the traffic flow in graph clusters allows improving the accuracy of the k nearest neighbor method.

6. Conclusion

The paper presents the distributed spatial-temporal model of short-term traffic forecasting based on the method of non-parametric regression k nearest neighbors. In the model, spatial and temporal characteristics of the transport flow in a compact cluster of the transport network are taken into account for the feature space description.

For distributed Big Data processing, we use MapReduce processing model implemented in the open source cluster-computing framework Apache Spark. Experimental analysis on real-world traffic data sets allows us to conclude that the proposed model has a high prediction accuracy and reasonable execution time, sufficient for real-time prediction.

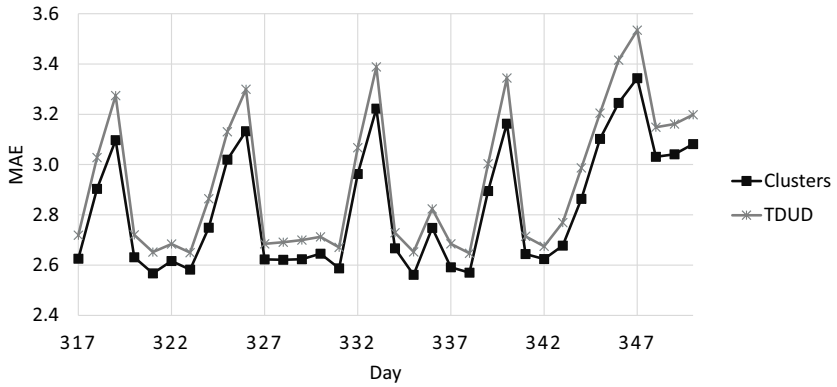

Figure 3. Mean absolute error.

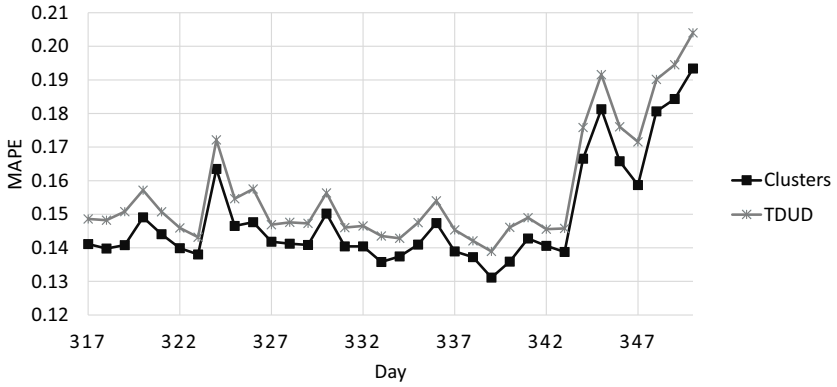

Figure 4. Mean absolute percentage error.

The possible direction of further research including dataset filtering for weekday / weekends traffic data and development of graph partitioning algorithms based on the traffic flow characteristics during a specific time period.

7. References

[1] Lana I, Del Ser J, Velez M and Vlahogianni E 2018 Road traffic forecasting: Recent advances and new challenges *IEEE Intelligent Transportation Systems Magazine* **10** 93-109
[2] Vlahogianni E, Golias J and Karlaftis M 2004 Short-term traffic forecasting: Overview of objectives and methods *Transport Reviews* **24** 533-557
[3] Karlaftis M and Vlahogianni E 2011 Statistical methods versus neural networks in transportation research: Differences, similarities and some insights *Transportation Research Part C: Emerging Technologies* **19** 387-399
[4] Shekhar S and Williams B 2007 Adaptive seasonal time series models for forecasting short-term traffic flow *Transportation Research Record* 116-125
[5] Smith B, Williams B and Keith Oswald R 2002 Comparison of parametric and nonparametric models for traffic flow forecasting *Transportation Research Part C: Emerging Technologies* **10** 303-321
[6] Yin H, Wong S, Xu J and Wong C 2002 Urban traffic flow prediction using a fuzzy-neural approach *Transportation Research Part C: Emerging Technologies* **10** 85-98

[7] Zheng Z and Su D 2014 Short-term traffic volume forecasting: A k-nearest neighbor approach enhanced by constrained linearly sewing principle component algorithm *Transportation Research Part C: Emerging Technologies* **43** 143-157
[8] Wu C H, Ho J M and Lee D 2004 Travel-time prediction with support vector regression *IEEE Transactions on Intelligent Transportation Systems* **5** 276-281
[9] Sun S and Zhang C 2007 The selective random subspace predictor for traffic flow forecasting *IEEE Transactions on Intelligent Transportation Systems* **8** 367-373
[10] Agafonov A and Myasnikov V 2015 Traffic flow forecasting algorithm based on combination of adaptive elementary predictors *Communications in Computer and Information Science* **542** 163-174
[11] Smith B and Demetsky M 1997 Traffic flow forecasting: Comparison of modeling approaches *Journal of Transportation Engineering* **123** 261-266
[12] Xia D, Wang B, Li H, Li Y and Zhang Z 2016 A distributed spatial-temporal weighted model on mapreduce for short-term traffic flow forecasting *Neurocomputing* **179** 246-261
[13] Lv Y, Duan Y, Kang W, Li Z and Wang F Y 2015 Traffic flow prediction with big data: A deep learning approach *IEEE Transactions on Intelligent Transportation Systems* **16** 865-873
[14] Dean J and Ghemawat S 2008 Mapreduce: Simplified data processing on large clusters *Communications of the ACM* **51** 107–113
[15] ApacheSpark 2018 (Access mode: `https://spark.apache.org/`)

Acknowledgments
This work was supported by the Russian Foundation for Basic Research (RFBR) grant 18-07-00605, grant 18-29-03135.