Adv Syst Sci Appl 2017; **3**:9–21
Published online at http://ijassa.ipu.ru/ojs/ijassa/article/view/501

Analysis of heterogeneity of transport flows

Igor O. Molybog[1,2*], Yury V. Chehovich[2]

[1]*Skolkovo Institute of Science and Technology, Moscow, Russia*
[2]*Dorodnitsyn Computing Center, Russian Academy of Sciences, Moscow, Russia*

Abstract: The situation when two independent transport flows are passing through the same edge of a transport network with different speeds is quite common for modern cities. The precision of mobile GPS devices, which are widely used for traffic monitoring, does not allow to determine the characteristics of transport flows in this situation because of their proximity. It causes difficulties in the analysis of the traffic situation, which may result in significant errors of transport routing systems. We propose a method to get the average speed of transport flows from common GPS traffic data using machine learning techniques. To do that we introduce a formal statement for the flow separation problem, which makes it possible to divide the problem into two sequential parts: statistical and optimizational. We analyze the possible approaches to the solution of both, construct features space for the statistical part and determine the complexity of the optimizational one. The developed techniques were implemented and tested to be working on real traffic data.

Keywords: traffic data analysis, classification, sequence separation

1. INTRODUCTION

Traffic data analysis plays important role in different areas of civil engineering such as infrastructure planning [1], transport flows control [2], and vehicles routing [3].

There is plenty of ways to collect the traffic data that can be used for different purposes. There are ways, based on counting flow through a section [4, 5], but in this work we focus on a way that based on following movements of individual particles in a system [6–10]. We consider GPS data from drivers' mobile devices, which are commonly used to estimate average speeds on different parts of a transport system. Commonly used ways to analyze the traffic data include time series analysis [11], and classification [12].

Modern routing systems consider average speed on a road to build optimal route for a vehicle. However, there is an effect called flow separation that may cause serious errors in average speed estimation. It is commonly observed before crossroads or exits from highways and consists in velocity heterogeneity across the flow. The problem that prevents us from considering the effect with standard engineering approaches is low precision of mobile GPS devices [13, 14]. It makes it impossible to determine the lane a vehicle holds on a road. Nevertheless, the accuracy becomes more and more important characteristic of routing systems and its improvement is of our interest.

Here we propose to apply machine learning techniques to get separate estimation of each transport flow in case of their separation. We use statistical approach to find the most likely value of the average speed of each flow and state optimizational problem to determine the distribution of the vehicles between flows.

*Corresponding author: igormolybog@gmail.com

While the flow separation effect arises in many applications [6, 15], the problem was neither stated, nor solved for the purposes we consider here.

2. INFORMAL DESCRIPTION

For simple explanation and intuition behind the problem take an example. Consider a two-lane road which is 8 meters wide and splits into two one-lane sleeves that go in opposite directions. Assume that cars can't change the lane along the way. Let's say the satellite can determine the position of a car with precision of 15 meters. This means that for any car that moves through the two-lane segment there is no way to detect immediately the lane it is moving through. When the car took its sleeve, we can conclude about the lane it was on and estimate the speed of the whole flow on that lane with the average speed of the car when it was on the two-lane segment. However, this information is out-of-date by the time when the car has left the two-lane segment and the speed of the flow could be changed during the time the car was changing to the sleeve.

One possible way to deal with the problem is to track cars that are following the current one and are still on the move through the two-lane segment when the current car left. We can pick those have similar behavior in the past and assume that they should be moving along the same lane as the current car used to. Those cars can help us to build a real-time estimation of the speed of the flow. The following section generalizes and formalizes the problem.

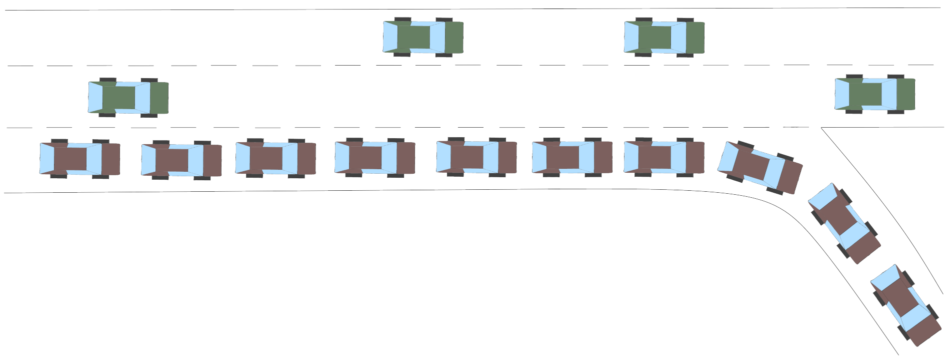

Fig. 2.1. Illustration for heterogeneous transport flow

3. FORMALIZATION

Consider a continuous (does not contain forks, exits or crossroads) segment s_0 of a multilane one-way road. Assume that the road has branching into W directions at some distance after the segment. Denote them as $\{\texttt{direction}_1, \dots \texttt{direction}_W\}$. Consider also s_{-1}, s_{-2}, $\dots$ s_{-b} — a continuous chain of straight segments of the road before s_0.

Every $\texttt{vehicle}$ on every segment has the following parameters: average speed $v(\texttt{vehicle})$, time of exit $\tau(\texttt{vehicle})$ from the segment and the further direction on the next branching $\delta(\texttt{vehicle}) \in \{\texttt{direction}_1, \dots \texttt{direction}_W\}$.

At every moment t for every vehicle the parameters can be known or unknown. Using this we can divide all of the vehicles into three groups: $G_1(t), G_2(t), G_3(t)$. Their description you can see in the Table 3.1.

Note that there is a time lag between the moment when a vehicle left the segment s_0 and the moment when we found out its further direction. This time lag depends on the average speed of the flow, changes with time and plays crucial role in division into the groups.

Copyright © 2017 ASSA.

Table 3.1. Groups of vehicles

	$G_1(t)$	$G_2(t)$	$G_3(t)$
speeds on $v_{s_{-b}}, \ldots, v_{s_0}$	known	known	unknown
further direction	known	unknown	unknown

Formally: $\exists$ lag(t) such that

$$G_1(t) = \{\texttt{vehicle} | \tau(\texttt{vehicle}) < t - \text{lag}(t)\},$$

$$G_2(t) = \{\texttt{vehicle} | t - \text{lag}(t) \leq \tau(\texttt{vehicle}) < t\},$$

$$G_3(t) = \{\texttt{vehicle} | t \leq \tau(\texttt{vehicle})\}.$$

Assume that the further direction ($\{\texttt{direction}_1, \ldots \texttt{direction}_W\}$) is determined by the lane the vehicle is passing through.

We calculate the average speed of a flow as Exponential Moving Average with parameter λ of speeds of the vehicles in it:

$$\text{AvSp}(t) = \lambda v(t) + (1 - \lambda)\text{AvSp}(t - 1), \lambda \in (0, 1)$$

The parameter λ is assumed to be given. It is usually tuned for a system of average speed estimation beforehand.

In respect to the introduced assumption, at the moment t average speeds on each lane

$$\text{AvSp}_1(t - \text{lag}(t)), \ldots, \text{AvSp}_W(t - \text{lag}(t))$$

can be calculated using information about vehicles in $G_1(t)$. At the same time,

$$\text{AvSp}_1(t), \ldots, \text{AvSp}_W(t)$$

are impossible to calculate because of lack of information about $G_2(t)$ vehicles. This may cause serious errors in speed estimations and eventually in routing results.

This paper deals with the problem of estimation of $\text{AvSp}_1(t), \ldots, \text{AvSp}_W(t)$. We are also interested in distribution of the vehicles between traffic flows. For practical purposes, it is especially important to have solutions of the problem for cases when the average speed differs significantly on lanes.

4. SEQUENCE SEPARATION

Let us consider a sequence of real numbers (speeds of vehicles as they are passing) $\{u_l\}_{l=1}^n$. It can be broken into W subsequences (according to the transport flows):

$$\{u_l\}_{l \in L_1}, \{u_l\}_{l \in L_2}, \ldots \{u_l\}_{l \in L_W},$$

where $\{L_w\}_{w=1}^W$ is an unknown splitting of $\{1, \ldots n\}$. Every element L of a splitting is a subsequence of $\{1, \ldots n\}$ and contains elements in ascending order.

Let's consider the statistics of Exponential Moving Average as a function of element of the splitting:

$$A(L_w) = \lambda \sum_{i=1}^{|L_w|} (1 - \lambda)^{i-1} u_{l_i}, l_i \in L_w.$$

The function represents the average speed of the flow that contains only vehicles with indexes from L_w.

Copyright © 2017 ASSA.

The problem is to find splitting $\{K_w\}_{w=1}^W$ such that the values of A on the elements of $\{K_w\}_{w=1}^W$ are as close as possible to the values of A on the true splitting $\{L_w\}_{w=1}^W$, which is unknown.

To formalize the sense of distance, introduce the loss function:

$$F = \sum_{w=1}^{W} \alpha_w (A(L_w) - A(K_w))^2, \alpha_w > 0.$$

4.1. *Vector representation*

Introduce $2W$ vectors of size n each:

Definition 4.1:
$j_w = j(L_w, \lambda) : j_w^k := \begin{cases} 0 & k \notin L_w \\ (1-\lambda)^{m-1} & k = l_m \in L_w \end{cases}$

where j_w^k is the slightly modified indicator that a vehicle k is actually in the flow w.

Definition 4.2:
$i_w = i(K_w, \lambda) : i_w^k := \begin{cases} 0 & k \notin K_w \\ (1-\lambda)^{m-1} & k = k_m \in K_w \end{cases}$

where j_w^k is the slightly modified indicator that a vehicle k is assumed by the model to be in the flow w.

In the vector j_w, the components with numbers in $\{l_m\}_{m=1}^{|L_w|}$ are equal to $(1-\lambda)^{m-1}$, all the other components are zeros. Similarly for the vector i_w.

Define the loss for the flow w:

Definition 4.3:
$F_w = \sum\limits_{m=1}^{|L_w|} (1-\lambda)^{m-1} u_{l_m} - \sum\limits_{m=1}^{|K_w|} (1-\lambda)^{m-1} u_{k_m}$

Form vector u out of the sequence $\{u_l\}_{l=1}^n$.

Then

$$F_w = i_w^T u - j_w^T u = (i_w^T - j_w^T) u$$

.

Note that the total loss now is $F = \lambda^2 \sum\limits_{w=1}^{W} \alpha_w F_w^2 = \lambda^2 \sum\limits_{w=1}^{W} \alpha_w (i_w^T - j_w^T) u (i_w^T - j_w^T) u$

4.2. *Risk minimization*

We introduce a class probability distribution on $1, \dots, n$, and assume that numbers belong to classes independently. A number belongs to class t if it lies in L_w.

Matrix P of size $n \times T$ contains probabilities that index i belongs to the subsequence L_w. Also p_w^i is the probability that the i-th component of j_w is not zero: $[j_w]_i \neq 0$.

Definition 4.4:
$\mathbb{P}(j_w)$ — *probability of the realization of j_w. It is the probability that the flow w really consists of vehicles with indexes that correspond to nonzero components of j_w.*

Introduce the expected loss on a flow of the prediction i_w:

Definition 4.5:
$\mathcal{E}_w(i) = \mathbb{E}_{j_w}\{F_w^2 | i_w = i\} = \sum\limits_{j_w} F_w^2 \mathbb{P}(j_w)\big|_{i_w = i}$

Copyright © 2017 ASSA.

Lemma 4.1:

$$\mathcal{E}_w(i_w) = i_w^T u u^T i_w - 2(\sum_{j_w} \mathbb{P}(j_w) j_w)^T u u^T i_w + u^T (\sum_{j_w} \mathbb{P}(j_w) j_w j_w^T) u$$

Proof
From definition 4.3 and the expressions followed:

$$\mathcal{E}_w(i) = \sum_{j_w} (i^T - j_w^T) u (i^T - j_w^T) u \mathbb{P}(j_w) = \sum_{j_w} u^T (i - j_w)(i^T - j_w^T) u \mathbb{P}(j_w) =$$

$$= \sum_{j_w} u^T (i i^T - j_w i^T - i j_w^T + j_w j_w^T) u \mathbb{P}(j_w) =$$

$$= \sum_{j_w} (u^T i i^T u \mathbb{P}(j_w) - u^T j_w i^T u \mathbb{P}(j_w) - u^T i j_w^T u \mathbb{P}(j_w) + u^T j_w j_w^T u \mathbb{P}(j_w))$$

Decompose the sum:

- $\sum\limits_{j_w} u^T i i^T u \mathbb{P}(j_w) = i^T u u^T i \sum\limits_{j_w} \mathbb{P}(j_w) = i^T u u^T i$
- $\sum\limits_{j_w} u^T j_w i^T u \mathbb{P}(j_w) = \sum\limits_{j_w} \mathbb{P}(j_w) j_w^T u u^T i = (\sum\limits_{j_w} \mathbb{P}(j_w) j_w^T) u u^T i$
- $\sum\limits_{j_w} u^T i j_w^T u \mathbb{P}(j_w) = \sum\limits_{j_w} i^T u u^T j_w \mathbb{P}(j_w) = i^T u u^T (\sum\limits_{j_w} \mathbb{P}(j_w) j_w) = (\sum\limits_{j_w} \mathbb{P}(j_w) j_w^T) u u^T i$
- $\sum\limits_{j_w} u^T j_w j_w^T u \mathbb{P}(j_w) = u^T (\sum\limits_{j_w} \mathbb{P}(j_w) j_w j_w^T) u$

Joining the sum, we get the statement of the lemma. □

Lemma 4.2:
k-th component of the vector $\pi(p_w, \lambda) = \sum\limits_{j_w} \mathbb{P}(j_w) j_w$ can be written as:

$$[\sum_{j_w} \mathbb{P}(j_w) j_w]_k = p_w^k \sum_{s=0}^{k-1} (1-\lambda)^s \sum_{M \subset \{1, \dots k-1\} : |M| = s} (\prod_{m \in M} p_w^m \prod_{m \notin M} (1 - p_w^m))$$

Proof
A component of expectation of a vector is equal to expectation of the component of the vector because of linearity of mathematical expectation.

$$[\sum_{j_w} \mathbb{P}(j_w) j_w]_k = \sum_{s=0}^{k-1} (1-\lambda)^s \mathbb{P}\{[j_w]_k = (1-\lambda)^s\}.$$

$[j_w]_k = (1-\lambda)^s \Leftrightarrow [j_w]_k \neq 0, \exists M \subset \{1, \dots k-1\}, |M| = s :$
$\forall m \in M : [j_w]_m \neq 0, \forall m \notin M : [j_w]_m = 0.$
$\mathbb{P}\{[j_w]_k \neq 0\} = p_w^k.$
As the event $\{[j_w]_k \neq 0\}$ is independent, we can get the formula. □

Similarly, for $\Pi(p_w, \lambda) = \sum\limits_{j_w} \mathbb{P}(j_w) j_w j_w^T$:

Lemma 4.3:

$$[\sum_{j_w} \mathbb{P}(j_w) j_w j_w^T]_k^l = \begin{cases} p_w^k p_w^l \sum_{s=0}^{l-1} (1-\lambda)^s \sum_{m=s+1}^{k-l+s} (1-\lambda)^m \sum_{A \subset \{1,\dots l-1\}:|A|=s} \sum_{B \subset \{l+1,\dots k-1\}:|B|=m-s-1} \\ \left(\prod_{a \in A \cup B} p_w^a \prod_{a \in \{1,\dots k-1\} \setminus (A \cup B \cup \{l\})} (1-p_w^a) \right), \quad k > l \\ p_w^k \sum_{s=0}^{k-1} (1-\lambda)^{2s} \sum_{M \subset \{1,\dots k-1\}:|M|=s} (\prod_{m \in M} p_w^m \prod_{m \notin M} (1-p_w^m)), \quad k = l \end{cases} \tag{4.1}$$

Proof
A component of expectation of a matrix is equal to expectation of the component of the matrix.

Case $k = l$ is very similar to previous lemma. Consider $k \neq l$. The matrix is symmetric, so prove for $k > l$.

$$[\sum_{j_w} \mathbb{P}(j_w) j_w j_w^T]_k^l = \sum_{s=0}^{l-1} (1-\lambda)^s \sum_{m=s+1}^{k-l+s} (1-\lambda)^m \mathbb{P}\{[j_w]_l = (1-\lambda)^s; [j_w]_k = (1-\lambda)^m\}.$$

$[j_w]_l = (1-\lambda)^s \Leftrightarrow [j_w]_l \neq 0, \exists A \subset \{1, \dots l-1\}, |A| = s : \forall a \in A [j_w]_a \neq 0, \forall a \notin A [j_w]_a = 0$

$[j_w]_k = (1-\lambda)^m \Leftrightarrow [j_w]_k \neq 0, \exists B \subset \{1, \dots k-1\}, |B| = m : \forall b \in B [j_w]_b \neq 0, \forall b \notin B [j_w]_b = 0$

Take into account that $l \in B$.

As the event $\{[j_w]_k \neq 0\}$ does not depend on $\{[j_w]_{k'} \neq 0\} \forall\ k' \neq k$, we can derive the formula from the statement of the lemma. $\square$

Both vector π and matrix Π depend only on parameter λ and column p_w of the matrix P. Therefore, $\mathcal{E}_w(i)$ is a quadratic function, and its minimization is a particular case of discrete quadratic programming problem.

If P consists of 0 and 1, $\mathcal{E}_w(i)$ has evident zero in $i = \pi(p_w, \lambda)$. In other words, if vector j_w is not stochastic, optimal value of i is $i_w = j_w$.

Total risk function, which is the cumulative expected loss for all flows:

$$\mathcal{E}(i_1, \dots, i_W) = \mathbb{E}_{j_1,\dots,j_W}\{F | i_1, \dots i_W\} = \lambda^2 \sum_{w=1}^{W} \alpha_w \mathcal{E}_w(i)$$

$$\mathcal{E}(i_1, \dots, i_W) = \lambda^2 \sum_{w=1}^{W} \alpha_w (i_w^T u u^T i_w - 2\pi(p_w, \lambda)^T u u^T i_w + u^T \Pi(p_w, \lambda) u)$$

5. FEASIBILITY SET

Describe the set of all possible vectors that i_w can be chosen from. It is denoted I and consists of indicator-like vectors for all possible subsequences $[k_1, \dots, k_d]$ of $1, \dots, n$. Formally:

$$I = \left\{ i \middle| \exists d \leq n, \exists [k_1, \dots, k_d] \text{ subsequent of } [1, 2, \dots, n] : i^k = \begin{cases} 0 & k \notin [k_1, \dots, k_d] \\ (1-\lambda)^{m-1} & k = k_m \end{cases} \right\}$$

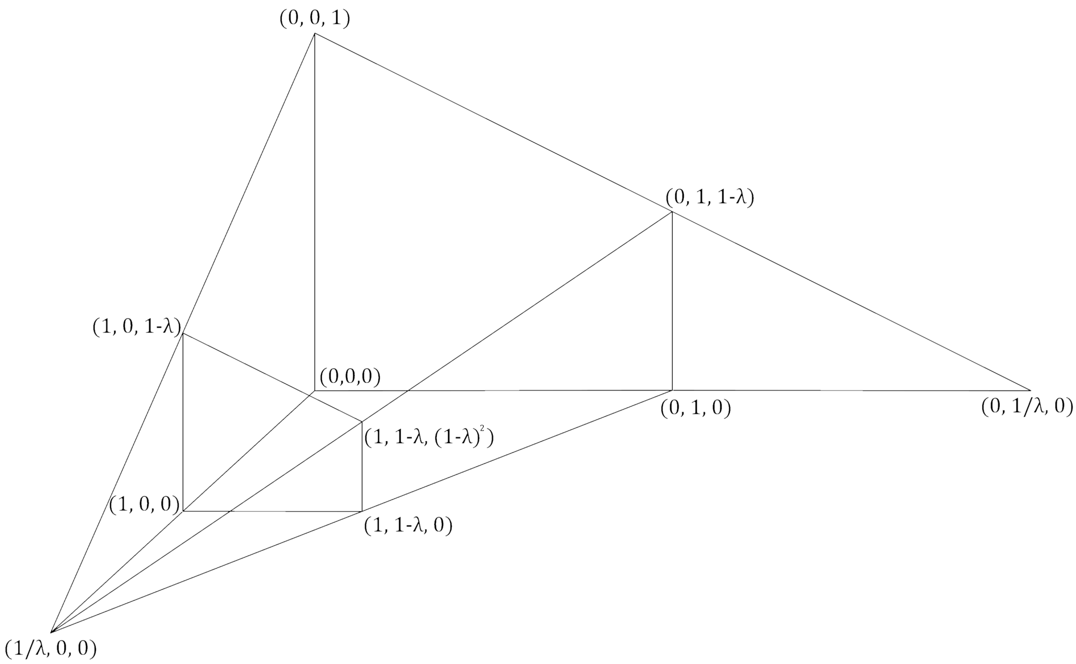

Fig. 5.2. Illustration for the lemma 5.1

Lemma 5.1:
$conv(I) = K \cap \{x \in \mathbb{R}^{n-1} : 0 \leq x_1 \leq 1\}$, where K is a multifaceted cone with $2n-2$ facets and $(\frac{1}{\lambda}, 0, \ldots, 0)^T$ as the apex.

Proof

- For each $l \neq 1$ consider set $\{x \in$I: $x_l = 0\}$. All of the points there lie in the $n-1$-dimensional hyperplane with $(\frac{1}{\lambda}, 0, \ldots, 0)^T$. There are $n-1$ such hyperplanes
- For each $l \neq 1$ consider set $\{x \in$I: $x_l \neq 1\}$. A vector from the set has the form $\begin{pmatrix} \vdots \\ (1-\lambda)^r \\ \vdots \end{pmatrix}$ if there are r non-zero coordinates among its coordinates number $1, \ldots, l-1$.

  Denote $n_l = \begin{pmatrix} 1 \\ \vdots \\ 1 \\ \frac{1}{\lambda} \\ 0 \\ \vdots \\ 0 \end{pmatrix}$, where $\frac{1}{\lambda}$ is on the position l.

Now, show that $n_l^T \begin{pmatrix} \vdots \\ (1-\lambda)^r \\ \vdots \end{pmatrix} = \frac{1}{\lambda} = n_l^T \begin{pmatrix} \frac{1}{\lambda} \\ 0 \\ \vdots \\ 0 \end{pmatrix}$.

To do that we can use $\frac{\lambda(1-\lambda)^{r-1}+(1-\lambda)^r}{\lambda} = \frac{1}{\lambda}(1-\lambda)^{r-1}$:

$$n_l^T \begin{pmatrix} \vdots \\ (1-\lambda)^d \\ \vdots \end{pmatrix} = \sum_{k=0}^{r-1}(1-\lambda)^k + \frac{1}{\lambda}(1-\lambda)^r = \frac{1}{\lambda}$$

- All of the vectors with $x_1 = 1$ share a hyperplane, just like all of the vectors with $x_1 = 0$.

In this way, all of the vectors from I with same state of a coordinate (zero or non-zero), share a $n-1$-dimensional hyperplane. $2(n-1)$ of them have intersection in $(\frac{1}{\lambda}, 0, \dots, 0)^T$.

□

Definition 5.1:
$$N = \begin{pmatrix} \frac{1}{\lambda} & 1 & \cdots & 1 \\ 0 & \frac{1}{\lambda} & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{\lambda} \end{pmatrix}$$

It is easy to check that from the proof above follows:

$$\text{conv}(I) = \{x \in \mathbb{R}^n \big| N^T x \leq \frac{1}{\lambda}\mathbb{1}; x \geq 0\}$$

For compact notation, make several formal definitions:

Definition 5.2:
$\mathbf{I} = [i_1, \dots, i_W]$

Mutual constraints on vectors $i_1, \dots, i_W$ consist in that only one element in every row of $\mathbf{I}$ is nonzero. It can be written in the form of sparsity constraints

$$\begin{cases} \|\mathbf{I}\|_0 = n \\ \mathbf{I}\mathbb{1} > \mathbb{0} \end{cases}$$

Definition 5.3:
$\mathcal{A} = diag(\sqrt{\alpha_1}, \dots, \sqrt{\alpha_W})$

Definition 5.4:
$\pi = [\pi(p_1, \lambda), \dots, \pi(p_W, \lambda)]$

Lemma 5.2:
$\mathcal{E}(i_1, \dots, i_W) = \mathcal{E}(\mathbf{I}) = \lambda^2(\mathcal{A}\mathbf{I}^T u)^2 + \lambda^2(\mathcal{A}\mathbf{I}^T u)(\mathcal{A}\pi^T u)^T + const$

Proof
Directly from the definition and lemma 4.1. □

6. FINAL PROBLEM FORMULATION

To estimate $\text{AvSp}_1(t), \dots, \text{AvSp}_W(t)$ we propose to use information about vehicles from $G_2(t)$. Let $\{\texttt{vehicle}_m\}_{m=1}^n$ sequence of vehicles from $G_2(t)$ in descending order of exits from s_0. Then $\{u_m\}_{m=1}^n$ is a corresponding sequence of speeds of the vehicles on s_0.

Copyright © 2017 ASSA.

In the applications, average speed of a flow is calculated as follows:

$$\text{AvSp}_w(t) = \lambda \sum_{i=1}^{|L_w|} (1-\lambda)^{i-1} u_{l_i} + \lambda(1-\lambda)^{|L_w|}\text{AvSp}_l(t - \text{lag}(t));$$

These functions are very similar to A. To apply all of the results from thr previous section, slightly modify the problem. Form $\hat{u}$ of size $n+W$, which is u stacked with $\text{AvSp}_1(t-\text{lag}(t)), \ldots, \text{AvSp}_W(t-\text{lag}(t))$. Expand P with $W \times W$ identity matrix.

New formulation for risk function:

$$\mathcal{E}(i_1, \ldots, i_W) = \lambda^2(\mathcal{A}\mathbf{I}^T\hat{u})^2 + \lambda^2(\mathcal{A}\mathbf{I}^T\hat{u})(\mathcal{A}\pi^T\hat{u})^T + const$$

6.1. Optimization complexity

As a result of the analysis, the problem of optimization stage is equivalent to:

$$(\mathcal{A}\mathbf{I}^T\hat{u})^2 + (\mathcal{A}\mathbf{I}^T\hat{u})(\mathcal{A}\pi^T\hat{u})^T \longrightarrow \min$$
$$\begin{cases} \mathbf{I} \in I \times \cdots \times I \\ \|\mathbf{I}\|_0 = n \\ \mathbf{I}\mathbb{1} > \mathbb{0} \end{cases}$$

Where I is the set of extreme points of the convex polyhedron:

$$I = \text{extr}\left(\{x \in \mathbb{R}^{n+W} \middle| \lambda N^T x \leq \mathbb{1}; x \geq 0\}\right)$$

This problem belongs to the class of mixed-integer linear programming problems, which is proved to be NP-complete. There are several approaches to solving the problem. Some of them are considered in [16, 17].

Because of asymptotic complexity of calculation of π and possible errors, which can be caused by estimating P, we will not conduct further investigation of the optimization problem and algorithms of its solution.

6.2. Distribution estimation

P is the only unknown parameter of the model. Possible ways to estimate the probability for each vehicle depend on the data we have and include:

- without respect to the vehicle, as an unconditional probability.
- with respect to the type of a vehicle (a truck or a car)
- with respect to the short-term history of a vehicle (speed history)
- with respect to the long-term history of a vehicle (previous passages)
- with respect to the planned route of a vehicle (for routed vehicles)

7. PROPOSED SOLUTION

As it was already mentioned, if P consists of 0 and 1, the optimizational part becomes trivial. Another remarkable feature of such matrices is that it can be built by any classification method, which gives us a powerful tool of machine learning to perform the estimation. Therefore, for practical purposes we will work with the class of matrices that consist of 0 and 1.

In the following experimental part we will work with information that can be extracted from GPS tracks data. More specifically, every vehicle at each moment t is characterized by a

Copyright © 2017 ASSA.

vector, composed of its short-term speed history. We can use the metric space of these vectors to build a classifier on information about some vehicles from $G_1(t)$ as a training set and use it on a test set from $G_2(t)$.

8. EXPERIMENT

The data for the experimental part was provided by Yandex. In the model that was used for data collection the roads system was approximated with a graph of straight segments. An id was assigned to every vehicle that was available for data collection. Using this id we could follow the tracks through the transport system. Collected data included length of every actual segment, speeds of vehicles on it and times of their exits from the segment.

The data is very sparse because of low concentration of vehicles that were available for data collection. We use assumption that the concentration of vehicles that were available for observation is homogeneous and constant.

Also the data is noisy. Besides significant variance of speeds of vehicles, it was hard to build proper tracks. For example, a track could be interrupted in one part of the transport system and continued in another part.

In the experiments we used data collected during a weekday in the area of Third Ring Road in Moscow. We considered parts of the roads that are adjacent to crossroads or exits that meet three requirements:

- The number of recorded passages per day exceeds 5000.
- The number of vehicles in the classes doesn't differ more than 1.5 times.
- Average speeds of vehicles in two classes differ by more than 10%.

These conditions are considered as necessary for flow separation problem. All of the parts of the road have further separation into two, which means that in the further discussion $W = 2$.

The classification problem was set for every straight segment of the road and was solved for each time of injection of information into the system. In terms of section 2, for every moment t and every s_0 we composed train sample out of vectors of speeds on $s_0, s_{-1}, \dots s_{-b}$ of vehicles from $G_1(t)$. Test samples were built similarly for vehicles from $G_2(t)$.

8.1. $i.i.d.$ *algorithms*

Here we used an assumption about independency of the vehicles on the road. It is not always true, but we were dealing with sparse data, which means that most of the vehicles we studied did not follow each other immediately.

For each moment t for each segment s_0 we built an $i.i.d.$ classifier. For training set we select vehicles that passed s_0 in the $h = 20$ minutes interval. This time window is characteristic for transport processes of such scale [18].

In the Table 8.2 we provided information about used algorithms with tuned and constant parameters.

We used $Accuracy$ to measure the performance of the algorithms because it is proportional to a norm of $P - \hat{P}$ as both of the matrices consist of 0 and 1:

$$Accuracy = \frac{\sum\limits_{i=1}^{n} I(\text{estimated\_class}_i = \text{real\_class}_i)}{n} =$$

$$= \frac{1}{n}\sum_{i=1}^{n} I\{p_1^i = \hat{p}_1^i\} = \dots = \frac{1}{n}\sum_{i=1}^{n} I\{p_W^i = \hat{p}_W^i\} = \frac{1}{nW}\sum_{i=1}^{n}\sum_{w=1}^{W} I\{p_w^i = \hat{p}_w^i\} =$$

Copyright © 2017 ASSA.

Table 8.2. Hyperparameters of $i.i.d.$ classification algorithms

Algorithm	Constant parameters	Adjusted parameters
SVM [19]	RBF kernel with auto coefficienting, weights associated with classes, shrinking enabled	Penalty parameter
RandForest [20]	gini criterion, bootstraped sample	Number of trees-estimators; maximal depth of a tree
BaggingKNN [21] [22]	Base estimator is kNN, bootstraped sample	Number of estimators; The number of samples to train each base estimator
GradBoosting [23]	deviance loss (Log Reg)	Number trees-estimators; learning rate; maximal depth of a tree
LogReg [24]	l_2 norm of penalty, weights associated with classes	Inverse of regularization strength
NaiveBayes [25]	Hypothesis: multivariate Bernoulli distribution of binarized vectors.	Lidstone smoothing parameter
NeuralNet [26]	Two layers with 10 and 3 neurons, Activation function is logistic	Regularization term parameter; The initial learning rate

Table 8.3. $i.i.d.$ algorithms results

	SVM	RandFor	BagKNN	GradBst	LogReg	NB	NNet
1	0.722	0.879	0.873	0.886	0.717	0.727	0.880
2	0.750	0.857	0.866	0.865	0.768	0.860	0.864
3	0.744	0.751	0.765	0.765	0.731	0.548	0.796
4	0.907	0.920	0.929	0.925	0.912	0.715	0.925
5	0.769	0.799	0.799	0.801	0.761	0.722	0.800
6	0.711	0.772	0.777	0.778	0.703	0.692	0.769
7	0.725	0.742	0.749	0.753	0.750	0.690	0.751

$$= \frac{1}{nW}\sum_{i=1}^{n}\sum_{w=1}^{W}(p_w^i - \hat{p}_w^i)^2 = \frac{1}{nW}\|P - \hat{P}\|_F^2$$

The averaged throughout a weekday performance of the algorithms one can see in the Table 8.3. The results presented are obtained on test samples for 7 different two-lanes problematic crossroads in Moscow. It is easy to notice that results very much depend on the crossroad, which means that for some crossroads it is easier to solve the flow separation problem. Authors tried to find the reasons why it is easier to solve the problem on some crossroads, but apparent connections with the length of the segment or difference between speeds of the flows were not found. Probably, the data collected on some crossroads was simply more consistent then on the others.

9. RESULTS AND DISCUSSION

The results of the numerical experiment show that there is no general model that fits every crossroad. Complex models like neural networks showed the best results. However, the hyperparameters of each model were tuned differently for crossroads but performed nearly

Copyright © 2017 ASSA.

the same for each segment near a crossroad, which means that every case of transport flow separation has unique features that strongly depend on its position on the roadmap.

There are other possible approaches to estimate matrix P with matrices that do not consist of 0 and 1. In those approaches, we have to solve the optimization problem stated in Section 5. Despite its asymptotic complexity, we still can solve it because of its limited dimension. Numerical results show that n is up to 10 for real problems and depend on $\text{lag}(t)$).

Empirically we can say that the problem of traffic separation is characteristic for certain locations on the roadmap. As a result, there are other possible approaches for solving the problem more accurate. For example, cameras set on a problematic crossroad could give us exact solution to the problem without extra approximations and assumption.

However, the theory in Sections 3-5 is more general and can be applied to solve similar problems for networks, where it is possible to follow movements of single particles of the flow.

10. CONCLUSION

In the paper we propose a formal statement for traffic flow separation problem. We divide the problem into two: statistical and optimization.

The statistical part can be solved numerically using algorithms for recovery probability distribution. To simplify the optimizational part, we relax the problem and use standard classification algorithms. We trained and tested several types of prediction models on the constructed features space, compare their performance for several problematic parts of Moscow road system.

The optimization part was theoretically analyzed, and stated in a form suitable for relaxations and solution with standard optimizational techniques. Future works in this area might be to solve the optimization problem explicitly in the algorithm, without relaxations of the statistical part.

The theoretical results obtained in the work can find applications beyond the civil engineering, in other areas related to study of flow dynamics.

REFERENCES

1. French, S. P, Barchers, C. & Zhang W. (2017) How Should Urban Planners Be Trained to Handle Big Data? In Thakuriah P., Tilahun N., Zellner M. (Eds.) *Seeing Cities Through Big Data*, Springer, pp. 209–217.
2. Coogan S., Flores C. & Varaiya P. (2017) Traffic predictive control from low-rank structure. *Transportation Research Part B: Methodological*, *97*, 1–22.
3. Pan J. S., Popa I. S. & Borcea C. (2017) Divert: A distributed vehicular traffic re-routing system for congestion avoidance. *IEEE Transactions on Mobile Computing*, **16** (1), 58–72.
4. Fu T., Stipancic J., Zangenehpour S., Miranda-Moreno L. & Saunier N. (2017) Automatic Traffic Data Collection under Varying Lighting and Temperature Conditions in Multimodal Environments: Thermal versus Visible Spectrum Video-Based Systems. *Journal of Advanced Transportation*, **2017**, https://doi.org/10.1155/2017/5142732
5. Leduc G. (2008) Road traffic data: Collection methods and applications. *RC Technical Notes. Working Papers on Energy, Transport and Climate Change*, [Online]. Available: http://ftp.jrc.es/EURdoc/JRC47967.TN.pdf
6. Thomas R.W. & Vidal J.M. (2017) Toward detecting accidents with already available passive traffic information. *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 1–4.
7. D'Andrea E. & Marcelloni F. (2017) Detection of traffic congestion and incidents from GPS trace analysis. *Expert Systems with Applications*, **73**, pp. 43–56.

Copyright © 2017 ASSA.

8. Araki M., Kanamori R., Gong L. & Morikawa T. (2017) Impacts of Seasonal Factors on Travel Behavior: Basic Analysis of GPS Trajectory Data for 8 Months. *Serviceology for Smart Service System*, Springer, 2017, pp. 377–384.
9. Herrera J.C., Work D.B., Herring R., Ban Xu. J., Jacobson Q. & Bayen A. M. (2010) Evaluation of traffic data obtained via GPS-enabled mobile phones: The Mobile Century field experiment. *Transportation Research Part C: Emerging Technologies*, **73** (4), 568–583.
10. Fu M., Kelly J.A. & Clinch J P. (2017) Estimating annual average daily traffic and transport emissions for a national road network: A bottom-up methodology for both nationally-aggregated and spatially-disaggregated results. *Journal of Transport Geography*, **1**, 186–195.
11. Klepsch J., Klüppelberg C. & Wei T. (2017) Prediction of functional ARMA processes with an application to traffic data. *Econometrics and Statistics*, **1**, 128–149.
12. Xu B., Barkley T., Lewis A., MacFarlane J., Pietrobon D. & Stroila M. (2016) Real-time detection and classification of traffic jams from probe data. *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 79.
13. Wing M.G., Eklund A. & Kellogg L. (2005) D Consumer-grade global positioning system (GPS) accuracy and reliability. *Journal of forestry*, **103** (4), 169–173.
14. Taylor G., Brunsdon C., Li J., Olden A., Steup D. & Winter M. (2006) GPS accuracy estimation using map matching techniques: Applied to vehicle positioning and odometer calibration. *Computers, environment and urban systems*, **30** (6), 757–772.
15. Amini Z., Pedarsani R., Skabardonis A. & Varaiya P. (2016) Queue-length estimation using real-time traffic data. *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1476–1481.
16. Benders J.F. (1962) Partitioning procedures for solving mixed-variables programming problems. *Numerische mathematik*, 1962, **4** (1), 238–252.
17. Mohamed A.W. (2017) An efficient modified differential evolution algorithm for solving constrained non-linear integer and mixed-integer global optimization problems. *International Journal of Machine Learning and Cybernetics*, **8**(3), 989–1007.
18. Wang L., Abdel-Aty M., Wang Xu. & Yu R. (2017) Analysis and Comparison of Safety Models Using AADT, Hourly Traffic, and Microscopic Traffic. *TRB 96th Annual Meeting Compendium of Papers*.
19. Cortes C. & Vapnik V. (1995) Support-vector networks. *Machine learning*, **20**(3), 273–297.
20. Breiman L. (2001) Random forests. *Machine learning*, **45**(1), 5–32.
21. Breiman L. (1996) Bagging predictors. *Machine learning*, **24**(2), 123–140.
22. Cover T. & Hart P. (1967) Nearest neighbor pattern classification. *IEEE transactions on information theory*, **13**(1), 21–27.
23. Friedman J.H. (2002) Stochastic gradient boosting. *Computational Statistics & Data Analysis*, **38**(4), 367–378.
24. Yu H., Huang F. & Lin C. (2011) Dual coordinate descent methods for logistic regression and maximum entropy models. *Machine Learning*, **85**(1), 41–75.
25. Zhang H. (2004) The optimality of naive Bayes. *AA*, **1**(2), 3.
26. Kingma D. & Ba J. (2014) Adam: A method for stochastic optimization. *arXiv preprint No. 1412.6980*, [Online]. Available: https://arxiv.org/abs/1412.6980

Copyright © 2017 ASSA.