Transportation Research Part C 144 (2022) 103882

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Cluster analysis of day-to-day traffic data in networks

Pengji Zhang [a], Wei Ma [c], Sean Qian [a,b,*]

[a] Department of Civil and Environmental Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, United States of America
[b] Heinz College of Public Policy and Information System, Carnegie Mellon University, Pittsburgh, PA 15213, United States of America
[c] Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hong Kong

ARTICLE INFO

Keywords:
Transportation network
Cluster analysis
Data driven

ABSTRACT

Day-to-day traffic data has been widely used in transportation planning and management. However, with the emerging of new technologies, one conventional assumption, on which many models rely, that all the day-to-day observations on the network follow a single pattern appears to be questionable. To better understand network flow patterns and their respective similarities, cluster analysis that partitions the day-to-day data into groups is an effective solution, but directly applying generic clustering algorithms may not always be appropriate identifying and interpreting day-to-day pattern changes due to the ignorance of the transportation network characteristics. In view of this practical issue, we propose a new clustering method that integrates network flow models, namely a statistical traffic assignment model and a probabilistic OD travel demand estimation model, into generic clustering algorithms. It essentially examines the probabilistic characteristics of traffic data by projecting those onto the dimensions of OD demands. For this reason, it can deal with traffic data where observations on some days and locations may be missing, or observing locations may change from day to day. The proposed algorithm embeds the domain knowledge of the transportation network, and is tested on two toy networks and one real-world network. Numerical experiments show the new clustering algorithm can effectively identify and interpret patterns that are hard to see by generic clustering algorithms otherwise, even with missing values or day-varying sensing locations.

1. Introduction

Day-to-day traffic data, such as daily traffic counts and travel speeds of links, are one of the most common for traffic analysis representing critical information for a transportation network. Such data can reflect recurrent traffic patterns of the network and thus are widely used in transportation planning and management. Over the past few decades, day-to-day traffic data sources have been used as inputs of numerous transportation applications such as Advanced Traveler Information Systems/Advanced Traffic Management Systems (ATIS/ATMS). However, emerging technologies in mobility systems bring new challenges to transportation system modeling. Ride-hailing vehicles, shared mobility, and new transportation alternatives and policies have led to an unprecedentedly complicated system revealing various traffic patterns. Moreover, incidents, roadworks, events, and many other factors would affect transportation network drastically. As a result, it appears to be questionable to assume that there exists a *single dominating daily traffic pattern* for a network over a period of time. The challenge is how we can differentiate daily traffic patterns from day to day provided with a large volume of data collected 24/7, analogous to unsupervised learning in data science. In view of this, we propose a data-driven clustering framework to discover traffic patterns from day-to-day traffic data. The clustering framework utilizes underlying network flow physics and system-level travel behaviors to group day-to-day traffic data, which can be further

* Corresponding author at: Department of Civil and Environmental Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, United States of America.
E-mail addresses: pengjiz@andrew.cmu.edu (P. Zhang), wei.w.ma@polyu.edu.hk (W. Ma), seanqian@cmu.edu (S. Qian).

https://doi.org/10.1016/j.trc.2022.103882
Received 3 April 2022; Received in revised form 7 July 2022; Accepted 3 September 2022
Available online 30 September 2022

used in downstream applications, e.g., estimating OD demands for each pattern, identifying the impact of policies or road closures, determining patterns from emerging sensors, just to name a few.

Cluster analysis (or clustering) for day-to-day traffic data is to partition a set of days T into several groups by certain criteria where each group present a typical daily traffic pattern. The traffic data for clustering are collected on each day over a course of a long period of time. That is, given a data set $X = \{X_i\}^{i \in T}$ where X_i is defined on a high dimension of spatio-temporal data, we need to find a cluster assignment for each day $i \in T$. For transportation network analysis, it can be used to detect patterns in the system and then to help at different stages of the whole data analysis workflow. For instance, with the clustering results we could summarize and compress data collected (Soriguera, 2012), clean noisy data and remove anomalous samples (Weijermars and Van Berkum, 2005), impute missing data (Ku et al., 2016), detect traffic anomalies, speed up algorithms by applying them in parallel on different clusters of data, and also design a better validation scheme for model performance evaluation (Asif et al., 2012).

In addition to be used as an auxiliary method in the data-driven analytics, the clustering results could be as well analyzed directly for transportation management. For example, from the clustering results we could select a few representative patterns and design different signal timing plans, travel demand management strategies, etc., individually for each pattern. For another example, clustering results for different periods of time could differ, which may indicate a shift in travelers' habits and thus guide the design of transportation improvement projects.

Cluster analysis is not a technique applicable only to traffic network modeling. It indeed has been widely used in data science across many domains – as part of the exploratory analysis, as a data pre-processing method, among many other applications. Therefore, it has been actively studied and there exist many generic or domain specific methods for cluster analysis. Based on those existing methods, classical methods have been adopted to discover patterns in various kinds of data for transportation systems through cluster analysis. Out of those existing methods, K-means (Hartigan and Wong, 1979), being a simple yet effective clustering method, has gained popularity in the transportation community. Xia and Chen (2007) applied the K-means algorithm on 15-minute traffic data of volume, average speed, and occupancy to define traffic flow phases. Gu et al. (2016) used it on arterial traffic flow data to detect disruptions of roadway network. Chen et al. (2017) applied the Davies–Bouldin index and Silhouette Coefficient to choose the number of clusters and then employed also the K-means algorithm for license plate recognition data to spatial travel patterns and temporal changes for different groups of vehicles. Similarly, Gace et al. (2021) used the K-means algorithm on contextually enriched automotive data to categorize the drivers' behaviors.

Other classical cluster analysis methods are also adopted for transportation system analysis. Weijermars and Van Berkum (2005) applied a hierarchical clustering method – Ward's hierarchical clustering – and discovered different patterns in the daily traffic profiles on holidays, weekdays, and weekends on a highway. Similarly, Soriguera (2012) used a three-stage hierarchical clustering method to explore the weekly and seasonal patterns with hourly traffic count data. Chung (2003) adopted the Small Large Ratio model to classify travel time data during different periods of a day, with the additional information on weather, events, etc. Guardiola et al. (2014) combined the dimension reduction method, Functional Principal Component Analysis, with the Partitioning Around Medoids algorithm to achieve better performance on discovering traffic patterns in the daily traffic profiles. Saha et al. (2019) compares eight clustering methods – K-means, K-prototypes, K-medoids, four variations of hierarchical clustering, and the combination of Principal Component Analysis for Mixed Data with K-means – for traffic pattern recognition, and provided recommendations on which method to use, how to choose the hyperparameters, and how to analyze the clustering results.

All existing research work provides valuable insights for applying clustering analysis for transportation management. However, despite the rich diversity of clustering methods and the data sets used, the information directly from the transportation network has been almost entirely overlooked. In other words, traffic flow in networks exhibits unique patterns following traffic flow physics and travel behaviors, which could provide additional information for achieving a better clustering outcome. Unfortunately, all existing approaches employ some generic distance metric, normally the Euclidean distance with weights, so that they are data-agnostic. With those algorithms, different features are treated equally (Jain et al., 1999), and the data set to be analyzed is the only source of information for clustering, without considering unique attributes of a transportation network. To better utilize the domain knowledge from the transportation network, in this study we inject network models of traffic flow into a classical clustering procedure, which brings non-trivial benefits for applications in system analysis, as we will show in the following sections.

It is worth noting that theoretically the data collected from a transportation network should inevitably contain information of the network already, at least to some extent. For example, the correlation among traffic observations on different links could be used to infer the general topological structure of the network. Therefore, with data pre-processing and augmentation, we could apply a clustering model without explicitly considering traffic flow attributes, but it may be aware of some information about the transportation network. Such augmentation work is tedious to do manually, but deep neural networks are well known to be applicable for such tasks. Li et al. (2018) developed an unsupervised model to cluster naturalistic driving encounters by combining an auto-encoder with K-means clustering algorithm, which outperformed the original K-means method. The auto-encoder part could be considered as a data pre-processing step that automatically discovers the underlying correlation among features and augments the data thereafter. Similarly, Markos and Yu (2020) attached a K-means clustering layer to a pre-trained deep convolutional auto-encoder and jointly fine-tuned the composite model on GPS trajectory data, which was used to discover transportation modes. There is no doubt that with sufficient data such sophisticated machine learning models could partially capture the attributes or information imposed by the network itself. However, using deep models have two major drawbacks comparing to the clustering method we propose to explicitly model network flow. First, deep models are usually viewed as black boxes and hard to explain. For transportation management, we often want not only a result, but also a reason and insights. Second, without the explicit constraints imposed by the network, the modeling tasks for transportation networks are mostly in the "low signal, high noise" category. Finding "sufficient data" to overcome this issue is generally challenging. Our method does not suffer from those two issues because

it combines a simple, interpretable clustering algorithm with a domain-specific model explicitly encapsulating characteristics of transportation networks and network flow.

As for the network flow, we choose to use a probabilistic way for modeling transportation networks. Classical traffic network models usually treat the system deterministically. For instance, traffic assignment models including the user equilibrium (UE) and the stochastic user equilibrium (SUE) (Fisk, 1980; Daganzo and Sheffi, 1977) represent a typical day's traffic pattern averaged from day to day, thus overlooking the day-to-day variances in OD demands and network flow. Such models indeed take the network structures and travel costs for links into consideration, but are "incompatible" with random nature of travels and traffic data. On the contrary, recent studies on transportation networks started to investigate statistical features of traffic data, and model network flow with probability distributions instead. For the static traffic assignment problem, Watling (2002a,b) proposed a modified SUE model called *stochastic demand generalized stochastic user equilibrium of order 2* which treated OD demands as a binomial distribution and applied a stochastic route choice model on top of probabilistic demand for traffic assignment. Shao et al. (2006) defined a reliability-based stochastic user equilibrium model, in which the OD demands were represented with a multivariate normal distribution. Those models enable the representation of traffic data with probabilistic distributions, which sheds light on understanding network flow with generic statistical models and makes it possible to design a clustering method tailored for transportation network data. For our purpose of analyzing day-to-day traffic count data, the *generalized statistical traffic assignment* (GESTA) framework proposed by Ma and Qian (2017) generalizes existing statistical traffic assignment models, and considers multiple sources of variances embedded in the traffic flow data. Besides, a corresponding probabilistic OD demand estimator is provided based on the GESTA framework (Ma and Qian, 2018). Therefore, we choose to use the GESTA framework as the building block of the clustering algorithm that represents underlying characteristics of network flow.

To summarize, this paper builds a cluster analysis framework for day-to-day traffic count data. The framework integrates traffic network models into generic clustering algorithms, with the goal to develop a clustering method tailored for transportation system. The major contributions of this paper are:

- It proposes a cluster analysis framework for discovering patterns regarding recurrent network traffic flows. The framework embeds probabilistic network flow models, such that the statistical features of a transportation network – OD demand probability distributions, link/path probability distributions, etc. – can be discovered for each pattern and utilized in the subsequent analysis.
- It sketches a way to guide general statistical models with network flow characteristics, so that the results are tailored for a transportation network. The incorporation of traffic models does not necessarily lead to superior results in all cases, but are proven to be useful and interpretable in clustering.
- It defines a novel way to measure the similarity of day-to-day general traffic data, particularly counts in this initial study. Instead of a generic distance metric in Euclidean space, we propose to project the data onto a subspace defined by OD demand probability distribution. The similarities among data points are then measured in that space where network flow characteristics are explicitly considered.
- It handles traffic pattern clustering or detection using day-to-day data where the sensing location can change from day to day or data can be incomplete/missing on some days, thanks to the underlying network flow characteristics.
- It examines the proposed framework on a large-scale network with real-world data to examine model performance and gain insights from the solutions.

The remainder of this paper is organized as follows. To begin with, we give three illustrative examples on clustering day-to-day traffic count data in Section 2. Those examples are meant to show why injecting traffic models into generic clustering algorithms could be useful. Then in Section 3 we formulate the whole problem as an optimization problem, and in Section 4 we give a solution algorithm for the problem based on the *Expectation-Maximization* algorithm (Moon, 1996). After that, in Section 5 we show numeric experiments for the framework on both hypothetical networks and synthetic data as well as a real-world network and real data. Finally, we draw conclusions on the behaviors and performance of the framework and discuss the existing issues of it in Section 6.

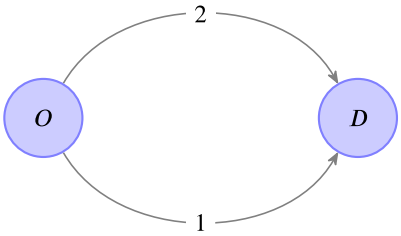
2. Illustrative examples

In this section, we give three examples on clustering day-to-day traffic count data to show why it is necessary to introduce traffic network models to generic clustering algorithms. Those examples are theoretical and a bit extreme, but reflect some features of real-world systems. As a note on the notation, in this section we use $\{(i, j, \dots), \dots\}$ to represent the clustering results, where each $(\cdot)$ is a cluster and within the parenthesis are the indices of days belonging to that cluster.

The first example is a two-link network shown in Fig. 1. In this hypothetical network, there are one origin node and one destination node, connected with two identical links. It is expected that the daily traffic counts on the two links would vary a lot by day. So we come up with a data set of four days' traffic counts for this network (shown in the table of Fig. 1). In real world this network structure is too extreme to exist. However, it is not uncommon to have multiple similar paths between one OD pair.

We may measure the distances among the four data points either by applying the Euclidean distance metric directly on the original data, or by transforming the data into daily OD demands and calculating the Euclidean distances of demands. The results obtained via both approaches are shown in Table 1.

With the first approach, directly measuring the Euclidean distances, we are focusing on the individual link flows. Based on the results in Table 1, we may conclude the clusters as $\{(1), (2, 4), (3)\}$. On the contrary, with the second approach, we are focusing on



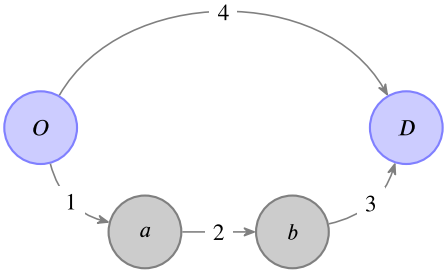
Traffic counts

Day	Link 1	Link 2
1	100	100
2	150	50
3	50	150
4	150	75

Fig. 1. A two-link network with two nodes and day-to-day traffic counts on it. The two links are identical so there is a large variance in the traffic counts observed.

Table 1
Distances among daily traffic counts of the four days in Fig. 1. Distances are measured with two approaches – directly applying the Euclidean distance metric or applying the Euclidean metric in the OD demand subspace. $d_{i,j}$ means the distance between Day i and Day j.

Approach	$d_{1,2}$	$d_{1,3}$	$d_{1,4}$	$d_{2,3}$	$d_{2,4}$	$d_{3,4}$
Direct Euclidean distance	70.7	70.7	55.9	141.4	25.0	125.0
Euclidean distance in OD subspace	0.0	0.0	25.0	0.0	25.0	25.0



Traffic counts

Day	Link 1	Link 2	Link 3	Link 4
1	100	100	100	50
2	110	110	110	50
3	100	100	100	60

Fig. 2. A two-node network with two paths and day-to-day traffic counts on it.

Table 2
Distances among daily traffic counts of the three days in Fig. 2. Distances are again measured with two approaches.

Approach	$d_{1,2}$	$d_{1,3}$	$d_{2,3}$
Direct Euclidean distance	17.3	10.0	20.0
Euclidean in OD subspace	10.0	10.0	14.1

the overall travel demands of the day, and the clusters are perhaps $\{(1,2,3),(4)\}$. Both approaches could be useful in transportation management. Clusters obtained by directly applying the Euclidean metric may help gain insights on travelers' route choice behaviors, while clusters in OD subspace usually represent recurrent patterns in the system, such as weekdays, holidays, etc.

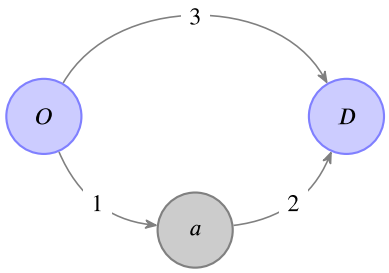
The second example is also a network with two nodes and two paths connecting them. However, this time one path consists of more links than the other, as shown in Fig. 2.

Again we may measure the distances among the three observations in the original space or the in OD demand subspace. The results are shown in Table 2.

In the network, Link 2,3 are linearly dependent on Link 1 entirely. Therefore, if we directly measure the Euclidean distance in the original space, we are in some sense artificially amplifying the differences on those links. In a real-world complex network, it can occur that some link flows are partially dependent due to network structure and/or route choices. However, in the subspace for OD demands, such issues are resolved automatically. Estimating OD demands could be viewed as an implicit way of examining dependency among those traffic measures. As for the clustering results, with the direct Euclidean distance we have $\{(1,3),(2)\}$, while in the OD demand subspace we have $\{(1),(2),(3)\}$.

The final example is also a three-link network with two nodes, as shown in Fig. 3. Not all links have traffic counts on a daily basis. On Day 2 there is no observation for Link 1 while on Day 3 there is no observation for Link 2. It can occur when data from probe sources do not come from the exact same location from day to day, or in some cases fixed location sensors can be placed in different locations at times, particularly when sensing resources are limited.

This time we cannot directly apply the Euclidean distance metric due to missing entries. However, those missing entries do not restrict the OD demand estimation at all – because Link 1 and Link 2 are on the same path, the path flow is equal to the link flow of either one. Therefore, we can calculate the distances among the data points without any issue, and get one single cluster $\{(1,2,3)\}$.



Traffic counts

Day	Link 1	Link 2	Link 3
1	100	100	50
2	-	100	50
3	100	-	50

Fig. 3. A three-link network with two nodes and day-to-day traffic counts on it.

The three examples above illustrate main differences between generic clustering methods and our new method, highlighting the need of integrating traffic models into generic models. Besides, note that it is possible to achieve similar clustering results by applying proper data pre-processing methods or manually modifying generic clustering algorithms. However, as mentioned in Section 1, those methods are ad-hoc and require much work from case to case, which would not generalize or scale in practice.

3. Formulation

In this section, we give a formal definition of the problem. We first list the frequently used notations in this paper, then briefly review the GESTA model (Ma and Qian, 2017) and Probabilistic OD demand Estimation (PODE) model (Ma and Qian, 2018), which are the two building blocks of the new clustering method. Finally, we formulate the clustering problem as an optimization problem, which assigns each data point a cluster.

3.1. Notations

The list of frequently used notations are shown in Table 3. Note that in this paper we use the hatted version of a symbol, $\hat{\cdot}$, to refer to the estimate of the original variable. Besides, in iterative algorithms or formulations, we use $\cdot^-$ to denote the variable in the previous iteration, and $\cdot^+$ to denote the variable for updating in the current iteration.

3.2. Assumptions

The proposed framework is built upon a few assumptions on OD demands and travelers' behaviors. Because they are essential to our models and results, we specifically list them here.

- OD demands follow a *mixture* of multivariate normal distributions. For each cluster there is a multivariate normal distribution that fully describes the OD demands of the cluster, and the realization of OD demands on a day is drawn from the distribution of the cluster that the day belongs to.
- Travelers are atomic players. On each day, all travelers between an OD pair independently and identically make route choices, which they learn through experience/information over a long time period.
- Route choice decisions are made solely based on the perception of the traffic conditions. Travelers are unaware of others' decisions. Travelers choose a route according to the experience/information on the generalized costs in the network.
- The variation of observed day-to-day traffic flows come from three sources as defined in Ma and Qian (2017): variation in OD demands, randomness in travelers' route choices, unobserved errors including measurement errors, non-recurrent events, etc. Unobserved errors follow an isotropic multivariate normal distribution.

3.3. Modeling and estimating probabilistic network conditions

We first need to briefly review the GESTA framework as well as the PODE algorithm. In combination, they are able to recover the probabilistic conditions of a transportation network based on day-to-day traffic count data. Although they are only applicable to days within a single cluster, they produce probability distributions of OD demands, path flows, link flows, etc., and will be used as the building blocks for clustering daily link traffic counts.

GESTA maps the distribution of OD demands Q to the distributions of path flow F, link flow X, and path costs C in the context of a transportation network, under a *statistical equilibrium* as defined in Definition 1.

Definition 1 (*Statistical Equilibrium*)**.** A transportation network is under a statistical equilibrium, if all travelers practice the following behavior: on each day, each traveler from origin r to destination s independently chooses route k with a deterministic probability p_{rs}^k. For a sufficient number of days, this behavior leads to a stabilized distribution, in turn, results in the deterministic probabilities $p = \psi(\theta)$ where $\psi(\cdot)$ is a general route choice function.

Table 3
List of notations.

Notation	Meaning
A	The set of all links.
A^o	The set of all links with flow observations.
Δ	Path/link incidence matrix.
Δ^o	Path/link incidence matrix for links in set A^o.
M	Path/OD pair incidence matrix.
Q	OD demands.
F	Path flows between all OD pairs.
X	Link flows on all link in set A.
X_a	Link flow on link a.
X^o	Link flows on links in set A^o.
X^m	Measurable link flows.
C	Path costs for all paths.
E	Unknown errors in X^m.
Z	Cluster labels for all daily records.
Z_i	Cluster label for the ith record.
D	Set of observed link flows $\{X_i^o\}^N$, or simply the data set.
N	Number of records in D.
D_i	The ith element of D.
$q = \mathbb{E}(Q)$	Mean values of Q.
Σ_q	Covariance matrix of Q.
$f = \mathbb{E}(F)$	Mean values of F.
Σ_f	Covariance matrix of F.
$x = \mathbb{E}(X)$	Mean values of X.
Σ_x	Covariance matrix of X.
$x^o = \mathbb{E}(X^O)$	Mean values of X^O.
Σ_{x^o}	Covariance matrix of X^O.
p	Route choice probabilities.
p_{rs}	Route choice probabilities for all paths between OD pair r, s.
p_{rs}^k	Route choice probability for choosing path k among all paths r, s.
$\tilde{p}$	Transformed p such that $\mathbb{E}(F \mid Q) = \tilde{p}Q$.
$c = \mathbb{E}(C)$	Mean values of C.
$\mathbf{x}^o$	Realization of the random variable X^o.
$\mathbf{x}_i^o$	Realization of the random variable X_i^o.
$\mathbf{x}^{o(k)}$	All $\mathbf{x}_i^o$ if Day i is in the cluster k.
$\mathbf{z}$	Realization of the random variable Z.
$\mathbf{z}_i$	Realization of random variable Z_i.

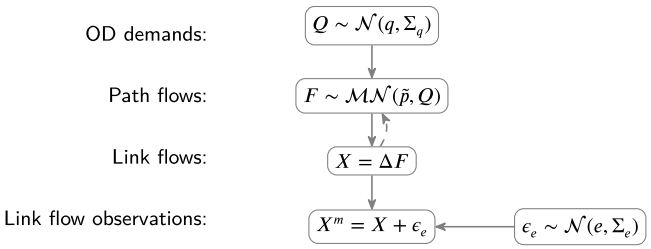


Fig. 4. GESTA as a hierarchical probabilistic model. $\mathcal{N}(\mu, \Sigma)$ represents a normal distribution with mean μ and covariance Σ. $\mathcal{MN}(p, n)$ means a multinomial distribution with probability vector p and number of trials n. $\tilde{p} \equiv \text{diag}(p)B$ where B is the transition matrix such that $\mathbb{E}(F \mid Q) = \tilde{p}Q$ (Watling, 2002a; Ma and Qian, 2017).

We may view the GESTA as a hierarchical probabilistic model. Fig. 4 shows the whole structure of such a model.

Although the whole graphic model seems innocent, the complicated interdependencies between F and X make it non-trivial. Unsurprisingly, X is dependent on F, and F is dependent on OD demand Q and route choice probability p. However, in GESTA p is dependent on the probability distribution of $C = t(X)$ where $t(\cdot)$ is a path cost function that maps link flows to path costs. For instance, if we use the random utility model (RUM) as the base route choice model, we have the probability of choosing route k among all routes between an OD pair (r, s) being

$$p_{rs}^k = \mathbb{P}\left(C_{rs}^k \leq \min_{i \neq k} C_{rs}^i\right). \tag{1}$$

As such, there is a circular dependency between X and F. Moreover, here p is not a random variable, but a property of the distribution of X.

That characteristic makes GESTA challenging to solve, and iterative solution algorithms are proposed to solve GESTA, particularly modified *Method of Successive Averages* algorithm. In that algorithm the marginal distribution of F is approximated with a

multivariate normal distribution,

$$F \sim \mathcal{N}(f, \Sigma_f), \tag{2}$$

where $f = \tilde{p}q$, $\Sigma_f = \Sigma_{f|q} + \tilde{p}\Sigma_q\tilde{p}^T$, and $\Sigma_{f|q}$ is the covariance matrix of path flow conditional on $Q = q$. With this simplification, the marginal distributions of X and X^m also follow multivariate normal distributions,

$$X \sim \mathcal{N}(x, \Sigma_x), \tag{3}$$

$$X^m \sim \mathcal{N}(x, \Sigma_x + \Sigma_e). \tag{4}$$

The GESTA model in Fig. 4 provides a way to map OD demands Q, a random variable, to the observed link flows X^m, another random variable of which the realization is the observed traffic counts. To project traffic count data to the OD demand subspace, we actually need to invert this model – given observed traffic counts, estimate the OD demand probability distribution. That can be done via the probabilistic OD demand estimation (PODE) method by Ma and Qian (2018).

PODE estimates the OD demand distribution by solving an optimization problem,

$$\min_{q,\Sigma_q} \alpha_1 g_1\left(x(q), \mathbf{x}^o\right) + \alpha_2 g_2\left(\Sigma_x\left(q, \Sigma_q\right), \Sigma_{x^o}\right), \tag{5}$$

where $x(\cdot), \Sigma_x(\cdot,\cdot)$ are functions that map the mean and covariance of OD demands to the mean and covariance of observed link flows, and $g_1(\cdot,\cdot), g_2(\cdot,\cdot)$ are functions measuring the discrepancy between two vectors/matrices while α_1, α_2 are weights for the two terms. They all can be set flexibly depending on the application, but in our case the $x(\cdot), \Sigma_x(\cdot,\cdot)$ come from the GESTA framework, $g_1(\cdot,\cdot), g_2(\cdot,\cdot)$ are Euclidean norm of the element-wise difference between the two vectors/matrices, and α_1, α_2 are set to equal.

To solve the optimization problem in Eq. (5), we choose *Iterative Generalized Least Squares* (IGLS). In each iteration, we first update $\hat{q}$ while keeping $\hat{\Sigma}_q$ by solving

$$\min_{f\in\Omega} n(\Delta^o f - \hat{x}^o)^T\left(\hat{\Sigma}_x^o\right)^{-1}(\Delta^o f - \hat{x}^o) + \left(q^H - Mf\right)^T\left(\Sigma_q^H\right)^{-1}\left(q^H - Mf\right), \tag{6}$$

where q^H is the historical mean of OD demands and Ω is the feasible set of f, which can be obtained with a traffic assignment model. After that, we need to update $\hat{\Sigma}_q$ with $\hat{q}$ fixed, which is simply solving another problem,

$$\min_{\Sigma_q} \quad \left\|S_x^o - \Sigma_{x^o}\right\|_F^2, \tag{7}$$

$$s.t. \quad \Sigma_{x^o} = \Delta^o\Sigma_{f|q}(\Delta^o)^T + \Delta^o\tilde{p}\Sigma_q\tilde{p}^T(\Delta^o)^T, \tag{8}$$

$$\Sigma_q \succcurlyeq 0. \tag{9}$$

The stopping criterion of the algorithm is defined with the discrepancy between the results from two successive iterations,

$$\tau = \mathcal{D}\left(\left(\hat{q}^n, \hat{\Sigma}_q^n\right), \left(\hat{q}^{n-1}, \hat{\Sigma}_q^{n-1}\right)\right), \tag{10}$$

where $\mathcal{D}(\cdot,\cdot)$ could be any discrepancy measure between two estimations, e.g. Hellinger distance. When τ is smaller than a chosen threshold, we consider the algorithm has converged and stop.

3.4. Modeling and estimating the mixture of probabilistic network conditions

GESTA together with PODE solves the problem of estimating the network conditions for one single pattern. An additional challenge is to identify multiple patterns on top of PODE for each pattern. The data generation process is similar to that of GESTA, with the exception that one new level of estimates is added to estimates of OD demands and path flows, which is clustering. The plate diagram for the probabilistic model is shown in Fig. 5.

For any day i, we introduce a new random variable z_i for its cluster. The random variable z_i follows a categorical distribution with parameter π. In addition, for each cluster j out of the K clusters we have a separate OD demand probability distribution parameterized by q_j and Σ_q^j, representing a recurrent traffic pattern.

The whole data generation process under this model is:

- For a single day i, draw a cluster index from the categorical distribution $z_i \sim C(\pi)$;
- Draw an OD demand for the day Q_i from the multivariate normal distribution $q_i \sim \mathcal{N}\left(q_{z_i}, \Sigma_q^{z_i}\right)$;
- With $\left(q_{z_i}, \Sigma_q^{z_i}\right)$, there exists a day-to-day stabilized route choice probability p_{z_i}, which travelers learn from experience/information;
- Draw a path flow F_i from the multinomial distribution $F_i \sim \mathcal{MN}\left(p_{z_i}, Q_i\right)$ (or from its multivariate normal approximation);
- Calculate the link flow $X_i = \Delta F_i$;
- Draw an unknown error term ϵ_i from the multivariate normal distribution;
- The final path flow observation is then $X_m^i = X_i + \epsilon_i$, which is also the link traffic counts observed on that day from a link set m (note that m can change from day to day).

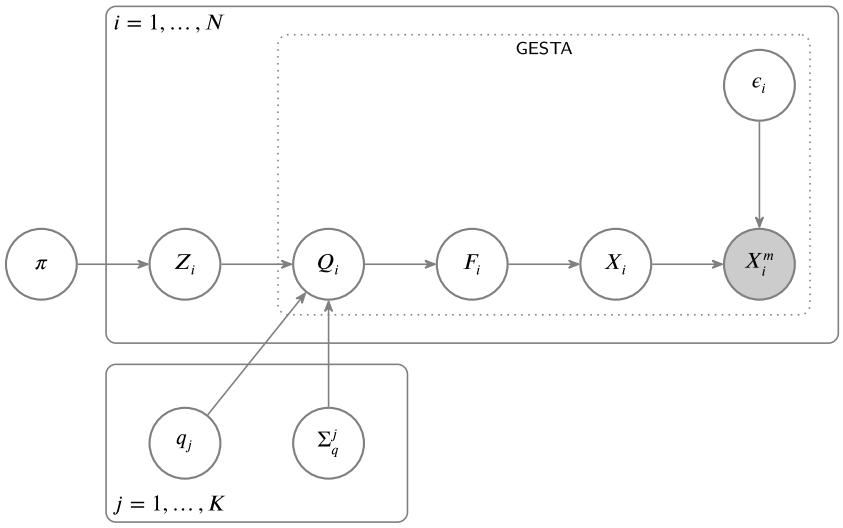


Fig. 5. Plate diagram for the mixture of GESTA to model transportation network conditions with multiple OD demand patterns. The part surrounded by the dotted frame is the same as the GESTA model in Fig. 4, while clusters together with the OD demand distributions for each cluster are added.

To estimate the unknown parameters $\Theta = (\pi, q, \Sigma)$, we formulate an optimization problem that maximizes the log likelihood of observations based on a mixture mode of patterns under GESTA,

$$\max_{\pi,q,\Sigma_q} \quad \log \prod_{i=1}^{N} \sum_{k=1}^{K} \pi_k p\left(X_i, z_i = k; q_k, \Sigma_q^k\right), \tag{11}$$

$$s.t. \quad f = \tilde{p} q, \tag{12}$$

$$f \in \Omega, \tag{13}$$

$$\Sigma_q \succcurlyeq 0, \tag{14}$$

where $p\left(\cdot; q_k, \Sigma_q^k\right)$ is the probability distribution of X_m obtained from the GESTA model under an OD demand probability distribution, and Ω is the feasible set of mean path flow f. With those estimates of parameters, the cluster assignment for any data point $\mathbf{x}_i^o$ is simply

$$z_i = \arg\max_{k \in \{1,\ldots,K\}} p\left(\mathbf{x}_i^o; q_k, \Sigma_q^k\right). \tag{15}$$

Eq. (11) is all about maximizing the data likelihood under the mixture model in Fig. 5. However, the latent cluster labels Z are coupled in the probability density function, imposing challenges to effectively solve this problem. Instead, we use the Expectation-Maximization (EM) algorithm (Moon, 1996). EM algorithm divides the original optimization problem into two sequential steps – an E-step and an M-step. In the E-step, the subproblem is to calculate the expected complete data likelihood (in log scale),

$$\mathcal{Q}(\Theta; \Theta^-) = \sum_{Z} p(Z \mid D, \Theta^-) \log p(D, Z \mid \Theta). \tag{16}$$

That is, we keep the current estimates of parameters Θ^- fixed and evaluate the probabilities of all cluster labels, which are later used to calculate the expected likelihood of the complete data. In the M-step, update the estimates of model parameters Θ by minimizing the expected likelihood of data in Eq. (16),

$$\Theta^+ = \arg\max_{\Theta} \mathcal{Q}(\Theta; \Theta^-). \tag{17}$$

Solving Eq. (17) is nothing more than solving the optimization problem $\max_\Theta \log p(D, Z \mid \Theta)$. By applying E-step and M-step repeatedly, the objective defined in Eq. (11) will gradually increase (Dempster et al., 1977) until convergence.

Particularly for this clustering problem, in the E-step, calculate

$$p\left(z_i = k; \mathbf{x}^o, \pi^-, q^-, \Sigma^-\right). \tag{18}$$

There are two ways to estimate this probability. The first one is usually called *soft assignment*,

$$p\left(z_i = k\right) = \frac{p\left(\mathbf{x}_i^o; q_k, \Sigma_q^k\right)}{\sum_{j=1}^{K} p\left(\mathbf{x}_i^o; q_j, \Sigma_q^j\right)}, \tag{19}$$

while the second one is called *hard assignment*,

$$p\left(z_i = k\right) = \begin{cases} 1, & \text{if } k = \arg\max_{j=1}^{K} p\left(\mathbf{x}_i^o; q_j, \Sigma_q^j\right), \\ 0, & \text{otherwise.} \end{cases} \tag{20}$$

Both are commonly used in clustering analysis. However, considering the complexity of traffic assignment models and the need for interpretability, we choose hard assignment – a single day always belongs to a cluster deterministically.

Besides, in the E-step we also need $p\left(X_i, z_i = k; q_k, \Sigma_q^k\right)$, which is nothing but an application of GESTA model for a specific cluster.

For the M-step, the update rule for π is

$$\pi_k^+ = \frac{\sum_{i=1}^{N} \mathbb{1}(z_i = k)}{N}, \tag{21}$$

which is the fraction of data points for each cluster and the update rules for q, Σ_q are

$$\max_{q_k, \Sigma_q^k} \sum_{i=1}^{N} \mathbb{1}(z_i = k) \log p\left(\mathbf{x}_i^o; q_k, \Sigma_q^k\right), \tag{22}$$

which is to find the optimal OD demand's mean value and covariance under the GESTA framework for each cluster, thanks to the hard assignment rule. However, finding the optimal q, Σ_q for a single cluster is non-trivial because the GESTA model is highly complex. In the GESTA framework, q and Σ_q are tightly coupled in a convoluted way – the route choice probability p is dependent on the distribution of X, which in turn depends on p, q, Σ_q. Therefore, the problem in the M-step is a non-convex problem without an analytical solution. Moreover, it is also hard to use a gradient-based optimization method because obtaining the gradient with respect to q, Σ_q is not easy either. Therefore, we resort to use the PODE algorithm reviewed in Section 3.3, a heuristic algorithm for the optimization problem in Eq. (22). That is not an ideal solution, but as what will be shown in Section 5, it can work very well in practice.

To summarize, in order to estimate the mixture of probabilistic network conditions and obtain the (hard) cluster assignments for data points, we use an EM algorithm:

- In the E-step, we update the cluster assignments via GESTA within each cluster using the current estimates of parameters;
- In the M-step, we update the estimates of parameters via PODE using the current cluster assignments.

4. Solution algorithm

Following Section 3.4, the solution algorithm, in its pseudocode, is listed in Algorithm 1.

As the initialization step, we first run an arbitrary clustering algorithm to initialize an initial cluster assignment $\mathbf{z}^{(0)}$. With $\mathbf{z}^{(0)}$ we estimate parameters $q^{(0)}, \Sigma_q^{(0)}$ to start the algorithm. Note that in Algorithm 1 we extend the PODE algorithm to take $\mathbf{z}$ and k for simplicity. $\mathbf{x}^o, k$ together act as a mask for the data $\mathbf{x}^o$ – only data points in $\mathbf{x}^o$ assigned to cluster k in $\mathbf{z}$ are considered for the OD demand estimation.

Note that the performance of PODE algorithm may be influenced by the stopping criteria as was mentioned in Section 3.3. However, for the clustering algorithm, the influence of that is only marginal according to our numerical experiments. Specifically, we tried the Hellinger distance and KL divergence, with and without early termination, and we found that the results of PODE might vary but the clustering results were always almost the same. The stopping criteria for PODE are about estimating one OD demand pattern, while clustering is about differentiating different patterns. So the clustering algorithm as a whole is less sensitive to the differences in the PODE results.

5. Numeric examples

In this section, we show three numerical examples to demonstrate the effectiveness and novelty of the new clustering algorithm. The first two examples are on synthetic data sets and small toy networks. They are designed to demonstrate the behaviors and special features of the new method. The last example is on a sizable real-world network to examine the performance of the new method for real-world applications.

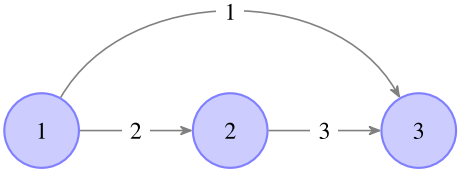
5.1. A three-link network

We first apply the algorithm on a three-link network shown in Fig. 6. There are two OD pairs: $(1, 3)$ and $(2, 3)$.

We generate two data sets on this network using the Probit-based GESTA traffic assignment model. The first one is well-separated and unambiguous while the second one is slightly harder for cluster analysis because some data points are mixed together in the link flow dimension.

Algorithm 1: Algorithm for clustering day-to-day traffic counts data using probabilistic network flow models.

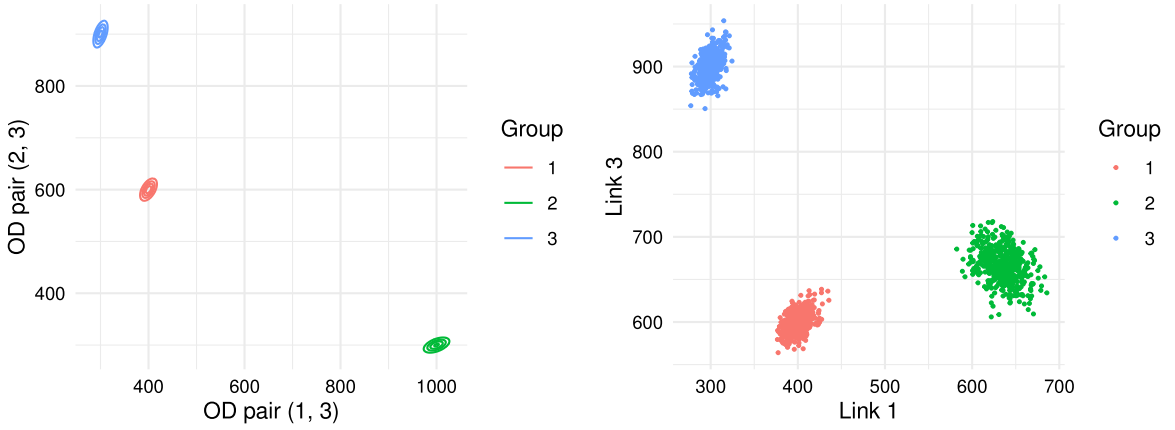
Data: Day-to-day traffic counts $\mathbf{x}^o$, number of clusters K
Result: Cluster assignments $\mathbf{z}$, parameters q, Σ_q of OD demand distributions
/* Initialization */
$n \leftarrow 0$;
/* Get initial clusters with another clustering method */
$\mathbf{z}^{(0)} \leftarrow \texttt{Cluster}(\mathbf{x}^o, K)$;
for $k \leftarrow 1$ **to** K **do**
  $q_k^{(0)}, \Sigma_q^{k(0)} \leftarrow \texttt{PODE}(\mathbf{x}^o, \mathbf{z}^{(0)}, k)$;
end
/* Main EM iteration */
repeat
  $n \leftarrow n + 1$;
  /* E-step: update cluster labels */
  for $i \leftarrow 1$ **to** N **do**
    for $k \leftarrow 1$ **to** K **do**
      $p_i^k \leftarrow \texttt{GESTA}(\mathbf{x}_i^o, q_k^{(n-1)}, \Sigma_q^{k(n-1)})$;
    end
    $\mathbf{z}_i^{(n)} \leftarrow \text{argmax}_{k=1,\dots,K} p_i^k$;
  end
  /* M-step: update parameter estimates */
  for $k \leftarrow 1$ **to** K **do**
    $q_k^{(n)}, \Sigma_q^{k(n)} \leftarrow \texttt{PODE}(\mathbf{x}^o, \mathbf{z}^{(n)}, k)$;
  end
until $\mathbf{z}^{(n-1)} = \mathbf{z}^{(n)}$;
return $\mathbf{z}^{(n)}, q^{(n)}, \Sigma_q^{(n)}$



Link parameters

Link	FFTT	Capacity
1	10	360
2	10	360
3	5	360

Fig. 6. A three-link toy network used in Section 5.1. "FFTT" means "free-flow travel time" of a link.



(a) OD demand distributions.

(b) Link traffic counts.

Fig. 7. OD demand distributions and the generated well-separated data of link traffic counts on the three-link toy network.

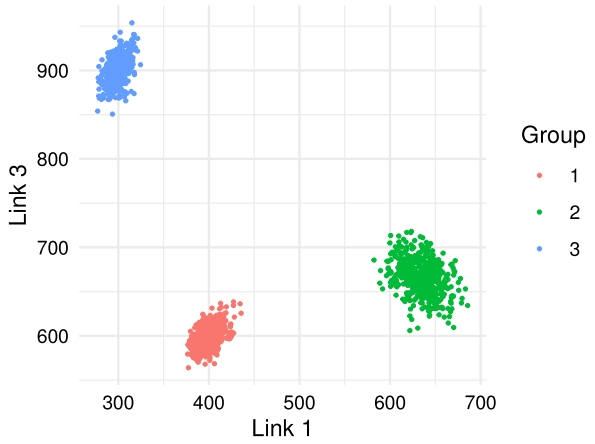


Fig. 8. Estimated groups of the K-means algorithm and the new clustering algorithm for the generated data shown in Fig. 7. Both algorithms give the same results.

Table 4
Estimated OD demand distributions from the data shown in Fig. 7. "Group" is the estimated group of the clustering algorithm and is manually matched to the real groups used for data generation.

Group	q	$\hat{q}$	Σ_q	$\hat{\Sigma}_q$
1	$(400.0 \quad 600.0)^T$	$(400.6 \quad 599.9)^T$	$\begin{pmatrix} 100.0 & 61.2 \\ 61.2 & 150.0 \end{pmatrix}$	$\begin{pmatrix} 100.3 & 59.4 \\ 59.4 & 146.4 \end{pmatrix}$
2	$(1000.0 \quad 300.0)^T$	$(983.3 \quad 318.1)^T$	$\begin{pmatrix} 250.0 & 68.5 \\ 68.5 & 75.0 \end{pmatrix}$	$\begin{pmatrix} 212.4 & 93.9 \\ 93.9 & 77.5 \end{pmatrix}$
3	$(300.0 \quad 900.0)^T$	$(299.8 \quad 899.2)^T$	$\begin{pmatrix} 75.0 & 65.0 \\ 65.0 & 225.0 \end{pmatrix}$	$\begin{pmatrix} 69.3 & 59.1 \\ 59.1 & 227.2 \end{pmatrix}$

5.1.1. Clustering well-separated data

The first data set is generated using three OD demand distributions shown in Fig. 7(a). For each pattern we generate 500 link traffic count observations on Link 1 and Link 3. The generated data are shown in Fig. 7(b).

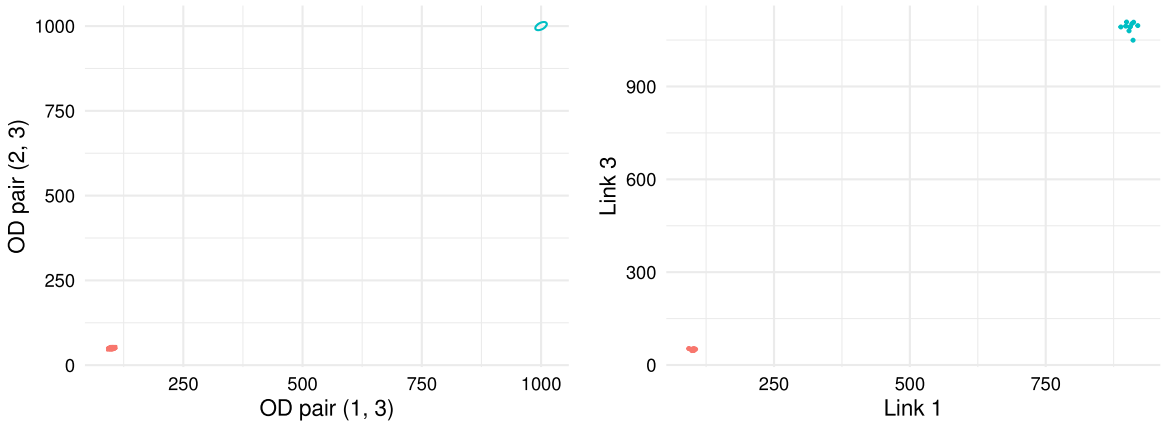
This data set is well-separated in both the link traffic count space and the OD demand space. Therefore, we expect that the results from both a generic clustering algorithm and the new algorithm should agree to each other if the parameters are properly chosen. Indeed, if we set the number of clusters $K = 3$, the K-means algorithm and the new algorithm give exactly the same cluster labels, which are shown in Fig. 8.

The estimated OD demand probability distributions for the three clusters are presented in Table 4. Those estimates are generally close to the real values. The clustering method indeed works as a reverse process of the data generation process depicted in Fig. 5. However, in practice we usually do not have the luxury of knowing the number of patterns and the ground-truth OD demand probability distributions. Moreover, the traffic flows may not conform to the mixed GESTA model, which is nothing more than yet another attempt to understand the complicated urban transportation system.

It is worth mentioning that there is a potential use case for the new clustering algorithm, to detect new OD travel demand patterns. In the case where we have collected sufficient day-to-day data, we may perform the cluster analysis to obtain a set of probability distributions of OD demand or flow that describe recurrent travelers' behaviors in this network. When new data on upcoming days are collected, they can be tested against those distributions to examine whether new data are part of existing clusters. For example, we generate a few new data points on the network with two different OD travel demand probability distributions, as shown in Fig. 9.

Now we can use the probability distributions of traffic counts on the two links to calculate the probability densities for those newly observed data points and compare them with those of the previously collected data in Fig. 7. The maximum log probability densities among the three estimated distributions of each data point are shown in Fig. 10.

We can see from Fig. 10 that the "old" data, which are used to estimate the link traffic count distributions, generally have the maximum log probability density at around −10, while the data points from the new emerging patterns have the maximum log probability density at around −800. Therefore, it can be concluded that those new data points represent different OD travel demand patterns from the previously estimated clusters. In practice, hypothesis test theories can be developed out of testing new data on existing patterns of probability distributions of OD demands or path/link flow, which would rigorously identify whether a new data point is likely a seen pattern or new pattern.



(a) OD demand distributions. (b) Link traffic counts.

Fig. 9. New emerging OD demand distributions and the resulting link traffic counts on the three-link toy network.

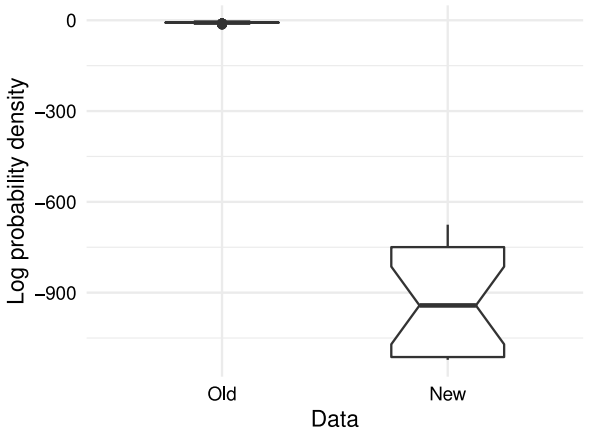


Fig. 10. Maximum log probability densities among the three estimated distributions for the data points in both Figs. 7 and 9.

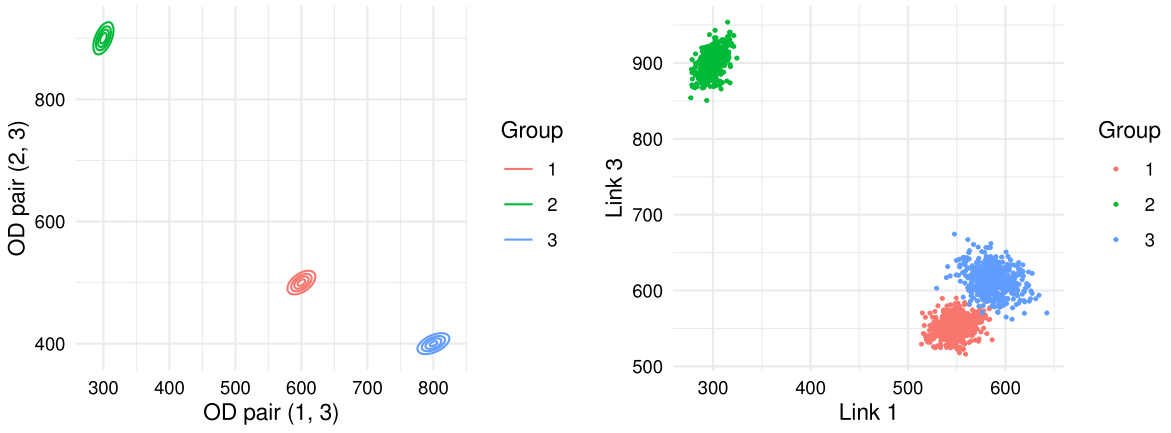
5.1.2. Clustering mixed data

Another interesting point to note is that in Table 4, the estimates for the second group is slightly worse than those of the other two groups. This is due to the non-unique solutions to the OD demand estimation problem. To see this we generate a new data set on the same network in Fig. 6 but with different OD demands. The OD demand probability distributions and the generated link flow data are shown in Fig. 11.

From Fig. 11, we can see that two OD demand probability distributions far apart (namely groups 2 and 3) could generate similar link flows on the observing links. Therefore, for this network under the "true" GESTA model, the regime near the lower right corner of Fig. 11(a) is an ambiguous zone. Many OD demand distributions could lead to similar blobs of data points in Fig. 11(b) – that is why in Table 4 the ODE is worse for group 3 than the other two. Unless additional information is provided, there is no way that the ODE can recover the "true" probability distribution in this case.

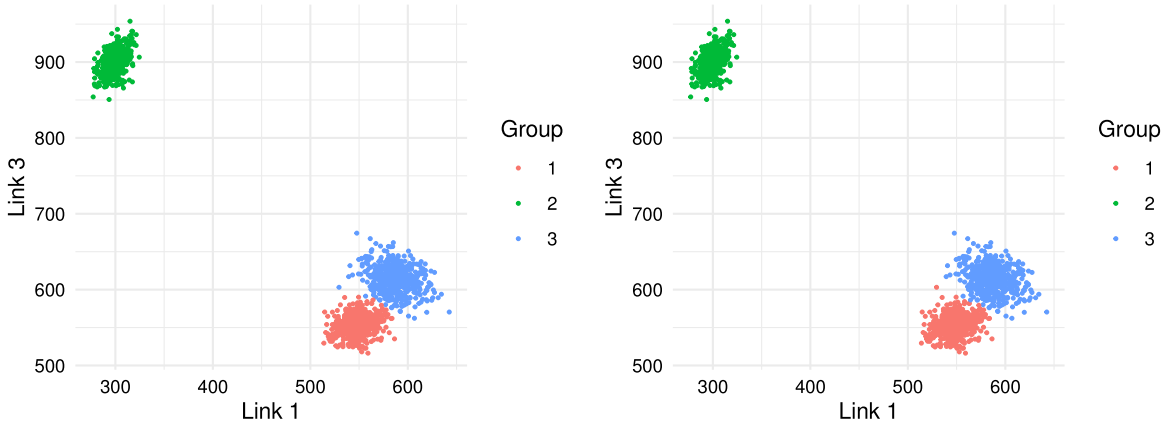
In addition, Fig. 11 also gives an interesting data set for examining a clustering algorithm. Some data points are mixed together in the link flow space, so we cannot expect a generic clustering algorithm to perfectly recover the original groups that those points belong to. That statement also holds for the new clustering algorithm because in the boundary regime between groups 2 and 3 in Fig. 11(b) the underlying OD probability distributions may not be perfectly recovered from mixed link flow data. Therefore, we conduct another cluster analysis on this data set and the results are shown in Fig. 12(a). The results from K-means algorithm are also included in Fig. 12(b) for comparison.

This time the results given by the two algorithms are similar except for a few points near the boundary of groups 1 and 3. Because of the introduction of probabilistic traffic assignment models, the new algorithm is aware of the correlation between the two links, which could "see" the differences in terms of the OD demand patterns, whereas the K-means algorithm would have difficulty placing those into the right group.



(a) OD demand distributions. (b) Link traffic counts.

Fig. 11. OD demand distributions and the generated data of link traffic counts on the three-link toy network. Note that this time two blobs of the data points for traffic counts are mixed together.



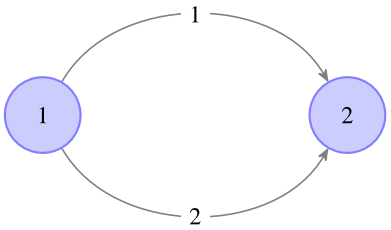
(a) Results using the new algorithm. (b) Results using the K-means algorithm.

Fig. 12. Clustering results using both the new algorithm and K-means algorithm. The number of desired clusters is again set to 3.

Note that this time neither of the two algorithms perfectly recovers the real groups shown in Fig. 11(b), which is expected as mentioned above. Recovering the "real groups" of traffic data – if that exists – is not the goal of cluster analysis and perhaps is not possible either using cluster analysis. When doing cluster analysis, we have no prior information at all on the desired clusters. The algorithm analyzes the given data and discover the patterns in the data based on how between/within group distances are defined. With the results, in Fig. 12 we are not trying to show that the new algorithm is always superior to the K-means algorithm. Instead, we would like to highlight the differences in the results from the two algorithms. That means the new algorithm is able to discover new and sensible patterns in the data and in certain cases it could be more appropriate and interpretable than other classical algorithms.

5.2. A two-link network

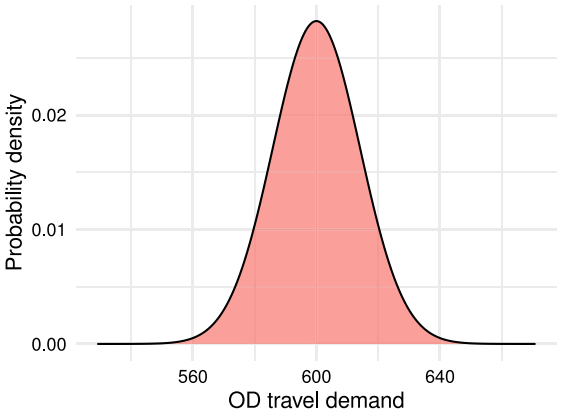
Now we apply the algorithm on another toy network shown in Fig. 13. There are only one OD pair $(1, 2)$ and two identical links. On this network, we expect to see high variances in the link count data because the two paths are interchangeable. We deliberately set the network so in order to understand and demonstrate the features and uniqueness of the new clustering algorithm. However, note that even though we probably will never see exactly such a network in the real world, it is not uncommon that multiple routes between an OD pair exhibit similar attributes.



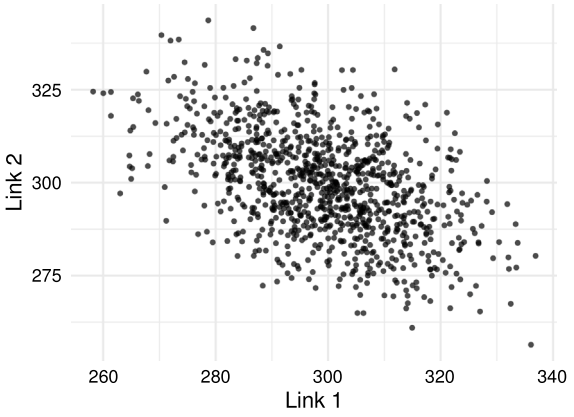
Link parameters

Link	FFTT	Capacity
1	20	300
2	20	300

Fig. 13. A two-link toy network used in Section 5.2. "FFTT" means "free-flow travel time" of a link.



(a) OD demand distribution. The parameters of the distribution are $q = 600, \Sigma = 100$.



(b) Link traffic counts.

Fig. 14. OD demand distribution and the generated data of link traffic counts on the two-link toy network.

We generate two different data sets on this network again using the Probit-based GESTA model. Each of the two data sets is designed to demonstrate certain interesting features of the clustering method.

5.2.1. Clustering data with a single pattern

The first data set is generated using only one OD demand distribution shown in Fig. 14(a). For this distribution we generate 1000 link traffic count observations on both links. The generated data are shown in Fig. 14(b).

Visually we may spot either one or two clusters from the data in Fig. 14(b), depending on how we define patterns. Therefore, it would be interesting to see what the algorithm gives if we ask it to return more than one clusters.

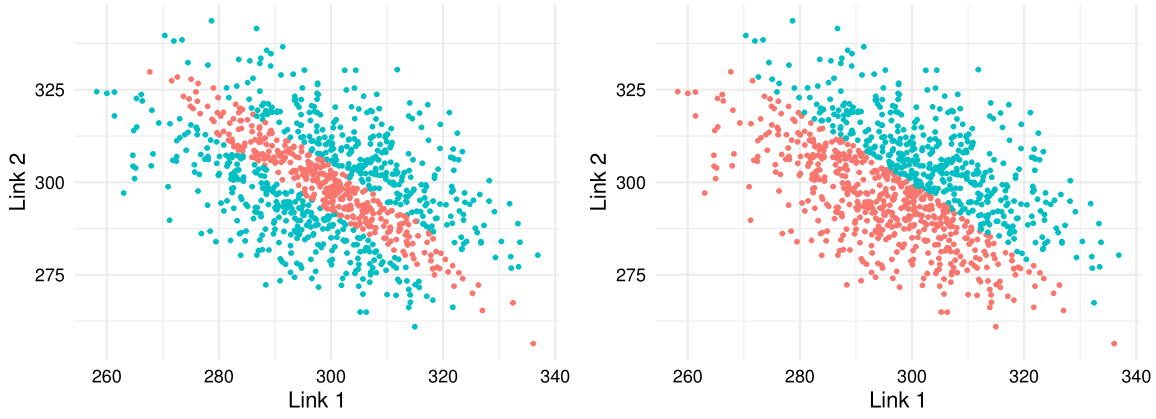
This time we set the desired number of clusters $K = 2$. For the initial assignment we use both random assignment and K-means clustering. Also, for comparison we run the K-means algorithm and the Gaussian mixture model on the same data set as well. The results for the four cases are shown in Fig. 15.

The results of K-means algorithm in Fig. 15(c) and the results of GMM in Fig. 15(d) are similar and match our expectation. However, the results from our network-based clustering algorithm shown in Figs. 15(a) and 15(b) are unusual – with random initialization it generates a non-compact cluster while with K-means initialization it generates two clusters differently from generic clustering methods. As was mentioned in Section 2, those results are because of the information imposed by the traffic assignment model. We now investigate the underlying probability distributions estimated during the clustering process to understand such behaviors of the algorithm.

Fig. 16 shows the estimated OD demand probability distributions. With random initialization, the two estimated distributions have almost the same mean value, but different variances, representing one way of differentiating traffic patterns. While with K-means initialization, the two probability distributions have similar variances but different mean values. Those are two ways to decompose the "true" demand probability distribution from Fig. 14(a).

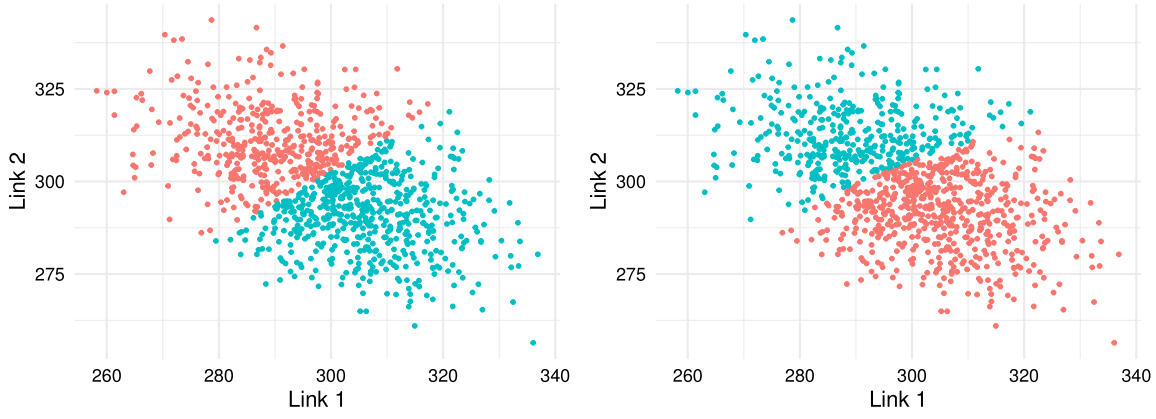
Fig. 17 shows the estimated link traffic count distributions. For comparison the results from GMM is also included. The embedded traffic assignment model injects the domain knowledge to the clustering process. Therefore, no matter how we initialize the clusters, the final clusters will always reflect the underlying correlation between the two links, instead of blindly maximize the data likelihood as what GMM does here in Fig. 17(c).

For Fig. 17(a), we can see that the "true" probability distribution is decomposed into two components – one that captures the strong correlation between the two links and one that captures the variability in the data. Fig. 17(b) instead shows a different way



(a) Estimated groups of the proposed algorithm with random initialization.

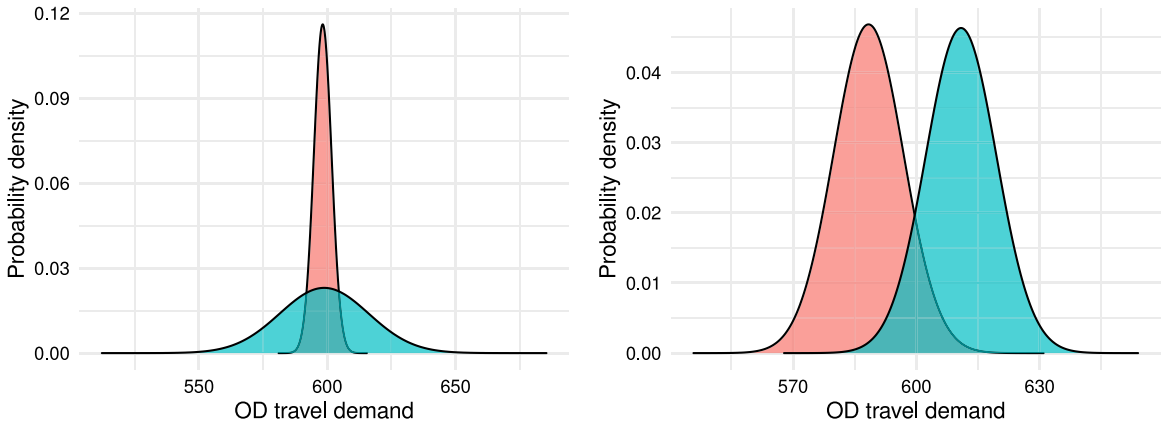
(b) Estimated groups of the proposed algorithm with K-means initialization.



(c) Estimated groups of the K-means algorithm.

(d) Estimated groups of the Gaussian mixture model. Here the GMM is set to estimate the covariance matrices fully independently for all clusters.

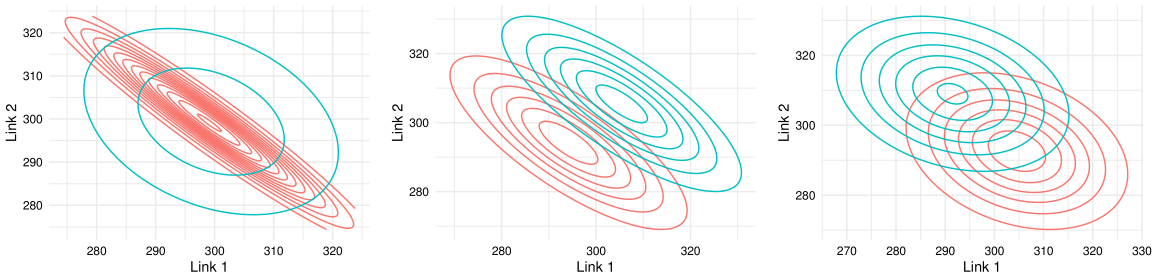
Fig. 15. Clustering results of the proposed algorithm with two different initialization methods, the K-means algorithm, and GMM for the data in Fig. 14.



(a) Estimated distributions with random initialization.

(b) Estimated distributions with K-means initialization.

Fig. 16. Estimated OD demand distributions of the proposed algorithm for the data in Fig. 14.

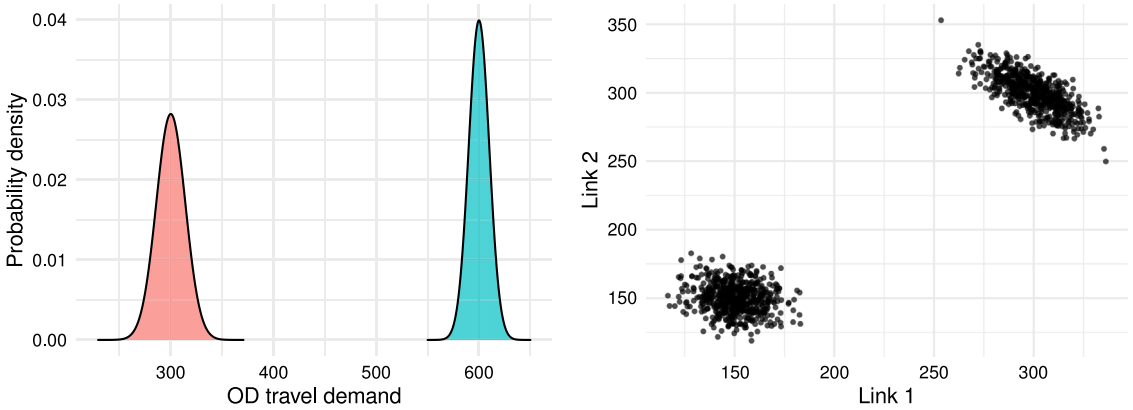


(a) Estimated distributions using the proposed algorithm with random initialization.

(b) Estimated distributions using the proposed algorithm with K-means initialization.

(c) Estimated distributions with GMM.

Fig. 17. Estimated link traffic count distributions of the proposed algorithm and GMM for the data in Fig. 14.



(a) OD demand distributions.

(b) Link traffic counts.

Fig. 18. OD demand distributions and the generated full data set of link traffic counts on the two-link network.

– each of the two is responsible for a subset of all data points. The two clusters with K-means initialization have almost the same shape and only the centroids differ. Both show that the targeted number of clusters $K = 2$ is larger than ideal. Note that specifically for this case, a non-parametric clustering algorithm (e.g., the mixture of Dirichlet process model Antoniak, 1974) could also give us such a hint, i.e., there is only one cluster in the data. Also, a non-parametric model might give more sensible results. However, using the knowledge on the network is still worthwhile in real world cases for clustering robustness. Besides, it might be possible to turn this algorithm into a non-parametric one.

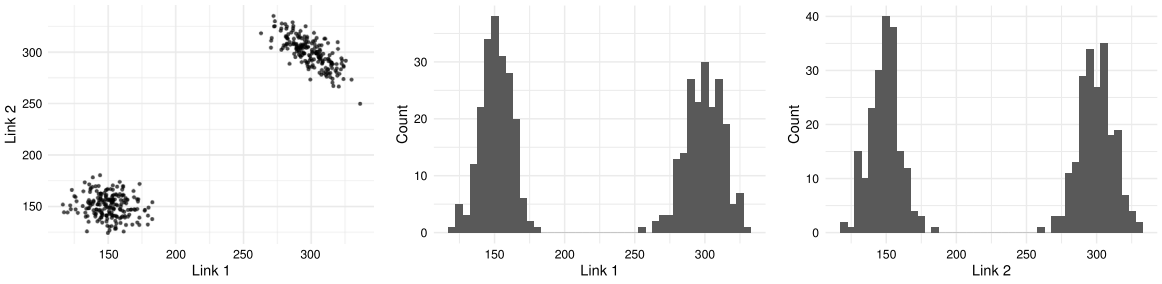
Also note that here with random initialization the algorithm may not always give the results in Fig. 17(a). Occasionally it would also cluster the data points in the way shown in Fig. 17(b) because of the randomness of the initialization, OD demands, and route choices. However, the results shown here is still representative and shows insights on the behavior of the algorithm. To reproduce the results here, multiple runs with different random states might be required.

5.2.2. Clustering incomplete data

Now we demonstrate another important feature of the proposed algorithm, i.e., its ability to handle an incomplete data set directly. As was mentioned in Section 2, it is not uncommon in real world that the set of observing links in the data set changes over time. To simulate that, we generate 1500 link traffic count observations using two OD demand probability distributions, shown in Fig. 18.

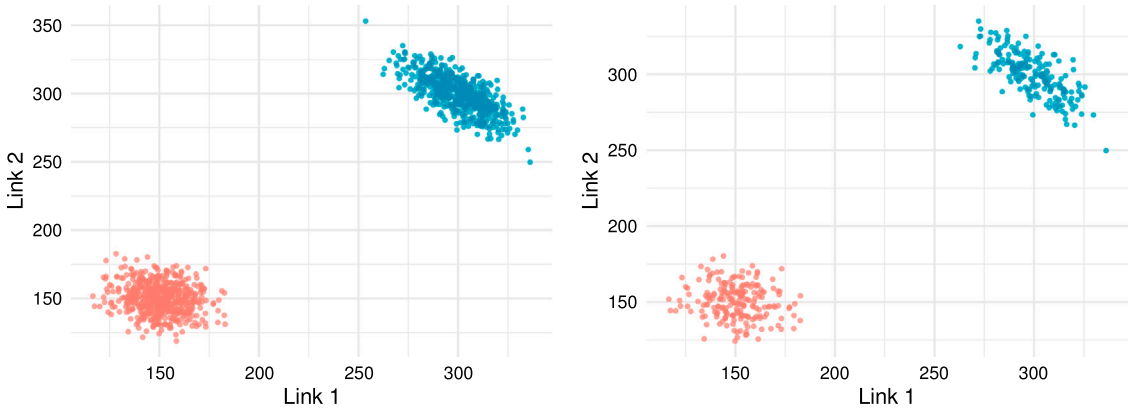
The two OD demand probability distributions are far apart and thus generated data points are clearly separated into two clusters. Then we randomly mask the observations on Link 2 for 500 data points, and the observations on Link 1 for another 500 data points. The remaining 500 points have traffic counts on both links. The resulting data set is shown in Fig. 19.

We can see that even after masking one dimension the data points are still well-separated, and it is not challenging to get a meaningful cluster assignment on each of the three subsets. However, instead we want to conduct clustering analysis on the combined data set of the three subsets as a whole. Generic clustering algorithms cannot handle this case directly. However, the proposed new clustering algorithm is able to infer demand patterns from partial observations, thanks to the embedded traffic network model and OD demand estimation. The results are shown in Fig. 20.

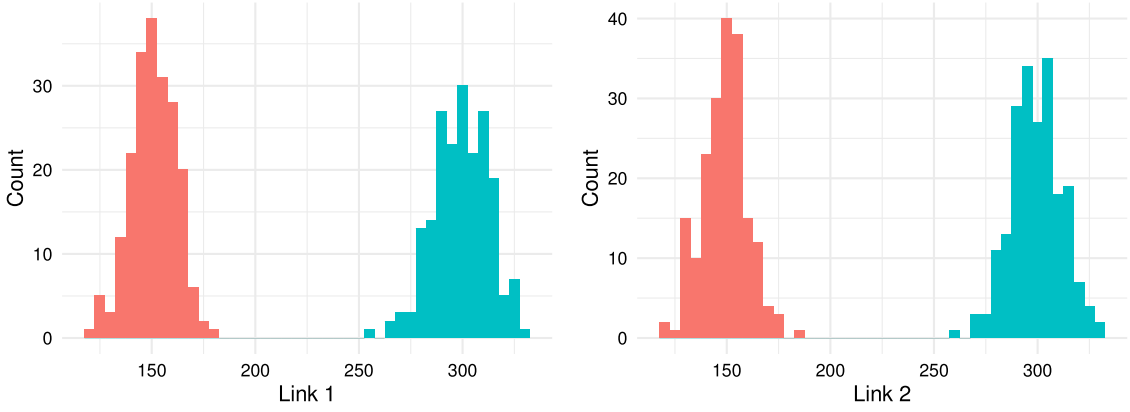


(a) Both links. (b) Link 1 only. (c) Link 2 only.

Fig. 19. Data sets of link traffic counts based on the data in Fig. 18(b). Two of those data sets have one of the two links masked.



(a) Estimated groups shown with the complete data. (b) Estimated groups for the data set in figure 19a.



(c) Estimated groups for the data set in figure 19b. (d) Estimated groups for the data set in figure 19c.

Fig. 20. Clusters returned by the proposed algorithm for the incomplete data in Fig. 19. The estimated groups are also shown with the original complete data for demonstration.

The algorithm is able to produce sensible clustering results without any specific pre-processing required to impute the data set. In addition to the convenience of no explicit pre-processing, combining different data sets could also improve the model performance, which can be seen from the estimated OD demand distributions using different data sets in Table 5.

In Table 5, combining the complete subset with the two incomplete data sets indeed helps with the accuracy in general. The additional information from the two incomplete subsets improves the estimates for Group 2, for which using only the complete subset does not perform well. Moreover, in our experiment, using only one of the incomplete subsets is even insufficient for an estimate, but combining the two subsets could provide, albeit not superior, reasonable results. Therefore, we suppose embedding traffic network models in the clustering process also enables better utilization of heterogeneous data sets.

Table 5
Estimated OD demand distributions for Fig. 18(a) using different data sets. "Complete data set" means the one in Fig. 18(b), without masking any features; "All three subsets" means to use all the three data sets shown in Fig. 19; "Complete subset only" means to use only the one in Fig. 19(a), which contains link traffic counts on both links; "Two incomplete subsets" means to use the two incomplete data sets in Figs. 19(b) and 19(c). KL divergence is calculated against the corresponding real distribution.

Data	Group	$\hat{q}$	$\hat{\Sigma}_q$	KL divergence
Complete data set	1	599.54	99.33	0.0011
	2	300.44	207.16	0.0008
All three subsets	1	598.53	92.80	0.0131
	2	300.29	169.02	0.0077
Complete subset only	1	599.09	102.61	0.0042
	2	301.19	249.78	0.0143
Two incomplete subsets	1	598.38	42.50	0.2795
	2	299.64	158.06	0.0154

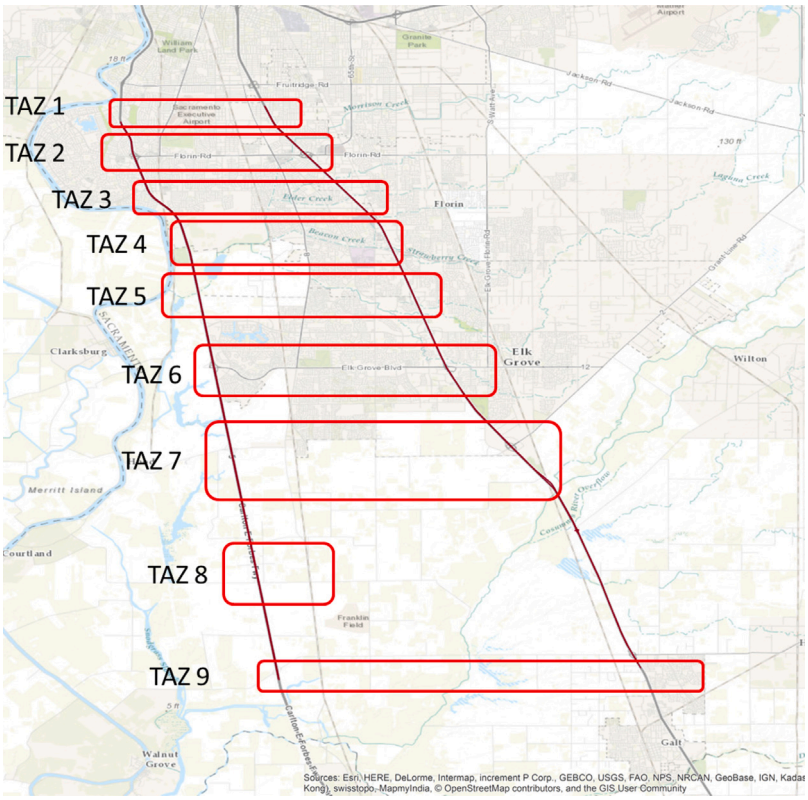


Fig. 21. TAZs in the Sacramento Regional Network.

5.3. A real-world network

Finally, we apply the clustering algorithm to conduct an analysis on a real-world network, the Sacramento Regional Network. Unlike the other numerical examples, this experiment is more of a demonstration of a common workflow for clustering analysis than a specifically constructed case to understand the proposed algorithm. Therefore, for this experiment we mainly focus on the high-level clustering results and how to use this algorithm for real-world applications.

The target network consists of two highways, I-5 and Hwy-90, towards Sacramento. The entire region is divided into 9 *traffic analysis zones* (TAZs) as shown in Fig. 21. Each of the 9 TAZs is treated as one origin as well as one destination in the network. Northern region of TAZ 1 is excluded from the network because there are many local roads in that region and our data do not suffice for modeling the travel demand profile. Considering there are few resident areas in the northern region of TAZ 1, this simplification should not impact the results much.

The raw daily link traffic counts are obtained through *Caltrans Performance Measurement System* (PeMS). The data set contains five-minute traffic counts on 86 road segments for three years – from January 1, 2014 to December 31, 2016 – with three days missing.

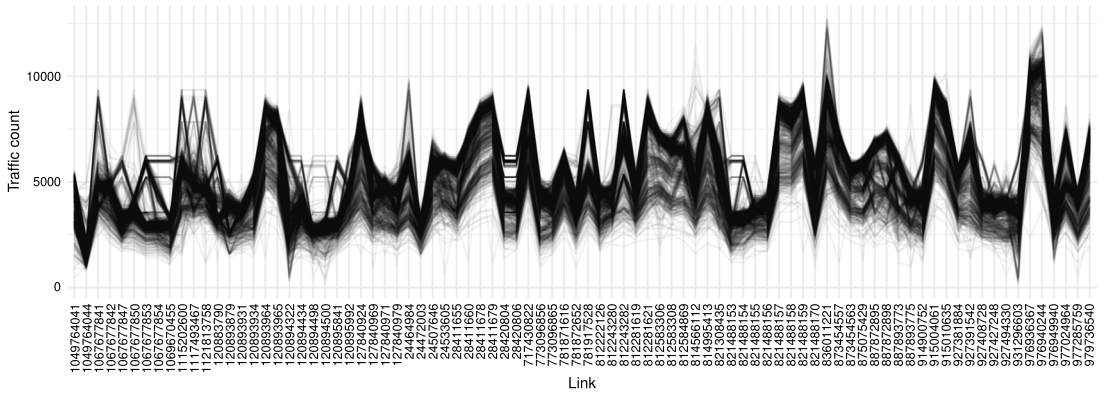
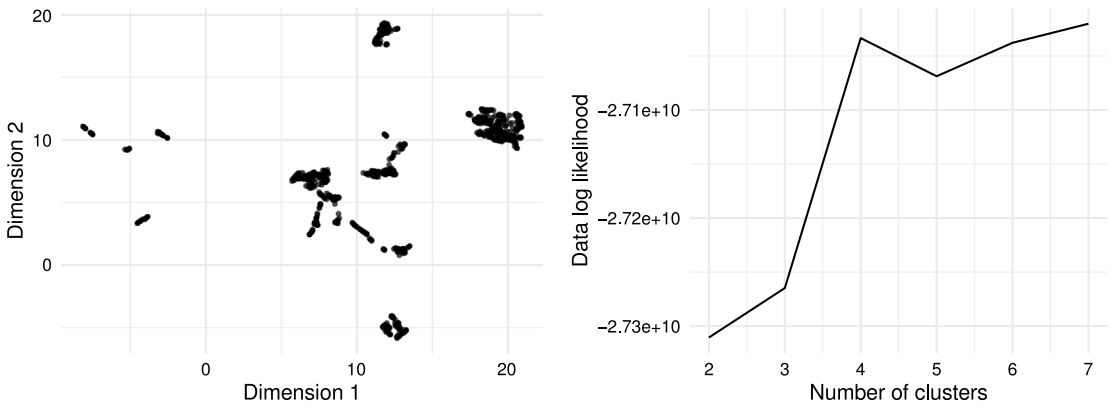


Fig. 22. Processed daily link traffic counts for all roadway segments in the Sacramento Regional Network. Each line represents a day.



(a) Daily link traffic count data in figure 22 after dimension reduction with the UMAP model.

(b) Data log likelihood for clustering data in figure 22 using different numbers of clusters.

Fig. 23. Results of exploratory analysis for data in Fig. 22. Those are meant to guide the choice of the desired number of clusters for the following cluster analysis.

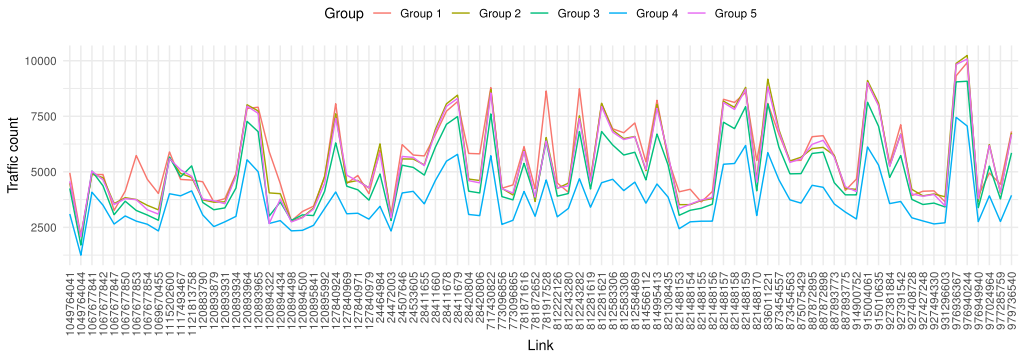
The raw data set is a combination of data from multiple sources including inductive loops, side-fire radar, and magnetometers. We keep only the data points within the morning peak (8 AM to 10 AM) and take the average for traffic counts from all available sources. The resulting data set contains 1094 entries of daily traffic counts on the 86 segments. Fig. 22 shows the processed daily link traffic count data.

The first step is to select a desired number of clusters. Visually based on Fig. 22, we can spot one dominating pattern and a few other minor patterns. However, on such a high dimension, it is unreliable to determine the number of clusters visually. Therefore, we use the *Uniform Manifold Approximation and Projection* (UMAP) model (McInnes et al., 2020) to reduce the dimension of the data to two. The results are shown in Fig. 23(a). Besides, we also run the clustering algorithms with number of clustering being 2 to 7 and examine the data likelihood as shown in Fig. 23(b).

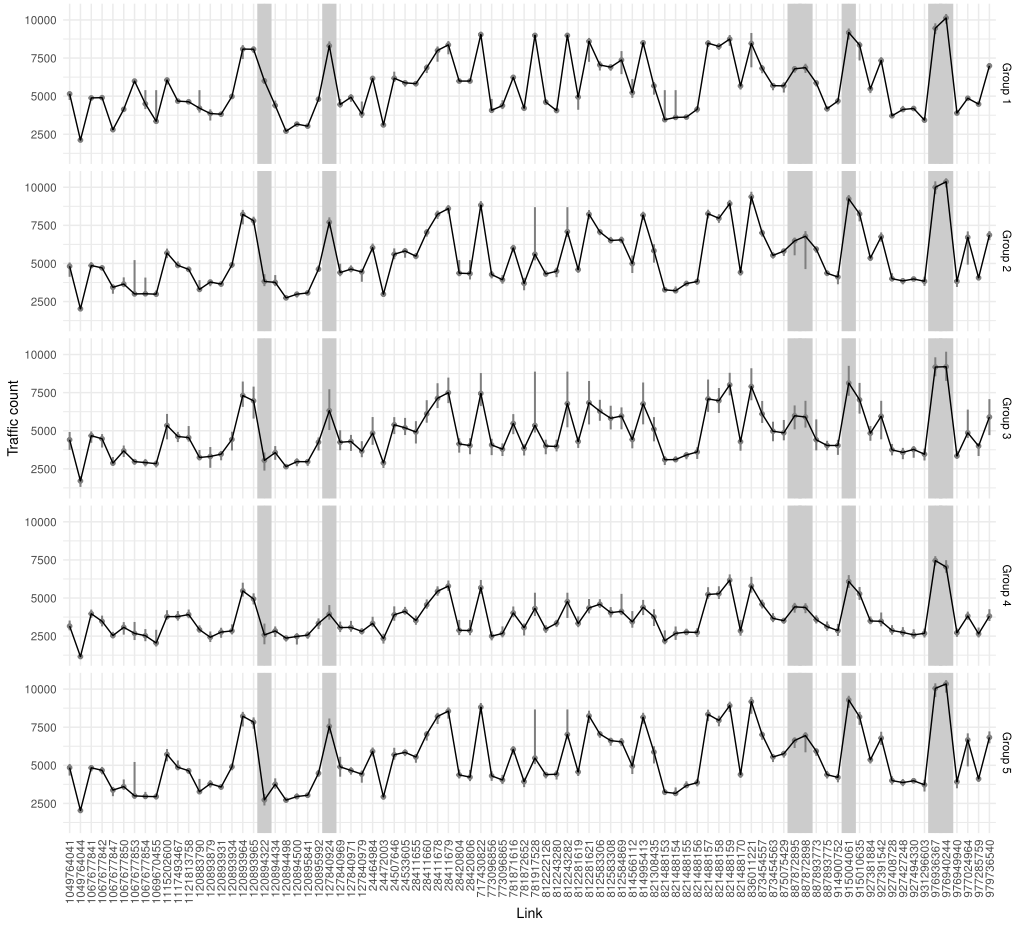
In the embedding space given by UMAP in Fig. 23(a) we can see roughly 5 to 7 clusters. Note that the UMAP model ignores the information from the network, so the embedding space discovered may not be necessarily desirable. Therefore, we look at the data log likelihood of the algorithm using different numbers of clusters in Fig. 23(b). Using 4 to 7 clusters gives similar outcomes. Normally we choose the value at the "elbow" of this curve. However, as was mentioned before, the proposed algorithm does not accurately maximize the data likelihood so we decide to combine both results in Fig. 23 and set $K = 5$. The resulting clusters are shown in Fig. 24.

Fig. 24(a) shows the mean values of day-to-day link traffic counts within each cluster, and Fig. 24(b) gives the ranges between the first and third quartiles of link counts for each cluster. Except for Group 4, all the groups are similar on the majority of the links while in Group 4 the link traffic counts are generally smaller. Besides, traffic counts on a few links are apparently different across groups, for example, on link 120894322, the traffic counts in Group 1 are larger than other groups.

For comparison, we also run K-means algorithm with $K = 5$ on this data set, and the results are shown in Fig. 25. Unsurprisingly, the results are similar to Fig. 24 considering that the Sacramento network is rather "regular" – the differences in the OD demand space are also well reflected in the link flow space. There are still some differences on certain links, which could be because of the



(a) Mean values of day-to-day link traffic counts for all clusters.

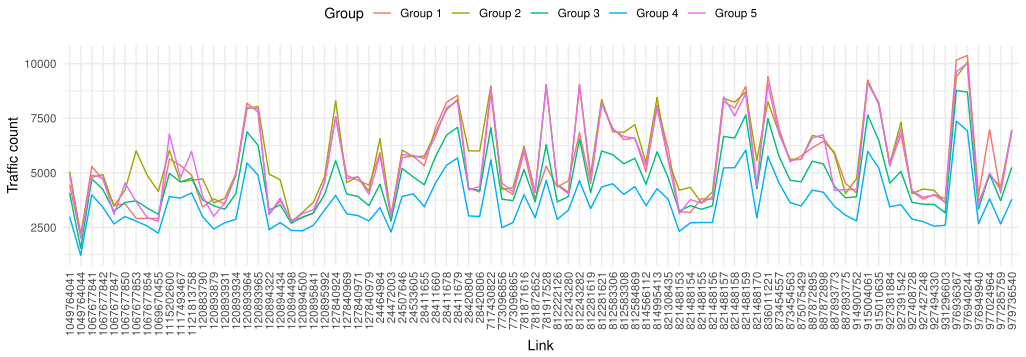


(b) Quartiles of day-to-day link traffic counts for all clusters. The lines and dots show the medians while the vertical bars give the ranges between the first and third quartiles. Certain links are shaded in gray to highlight differences among links.

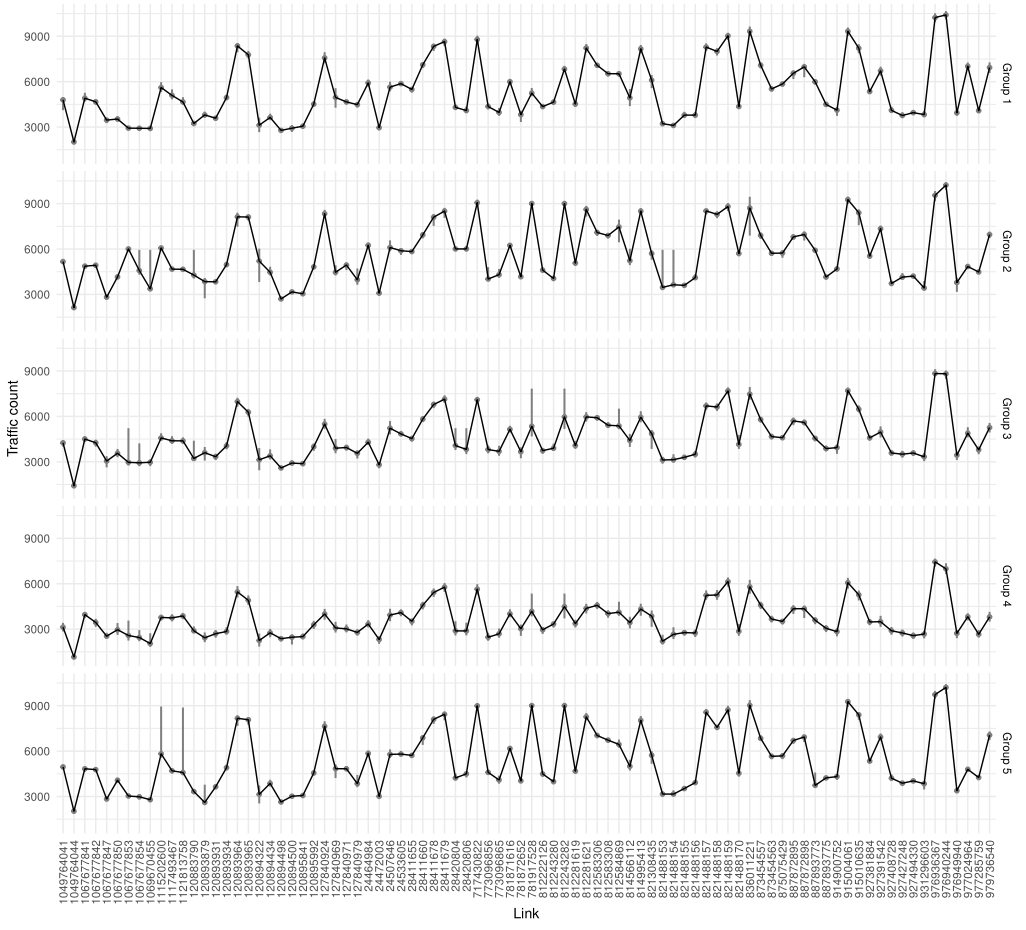
Fig. 24. Clustering results using the proposed algorithm for the data shown in Fig. 22.

embedded traffic network models. We cannot say the outcome from either model is superior to the other, but the differences in the results show that the proposed algorithm provides a more interpretable way consistent with network flow to understand the data and warrant its usefulness in real-world applications.

We also inspect the estimated OD demand probability distributions, which are shown in Fig. 26. Due to the high dimensionality, those figures are challenging to understand, but we may still spot a few interesting patterns. For example, according to Fig. 26(a)



(a) Mean values of day-to-day link traffic counts for all clusters discovered by K-means algorithm.

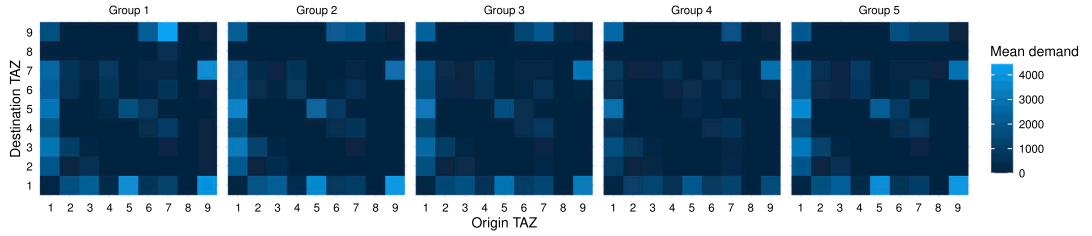


(b) Quartiles of day-to-day link traffic counts for all clusters discovered by K-means algorithm. The lines and dots show the medians while the vertical bars give the ranges between the first and third quartiles.

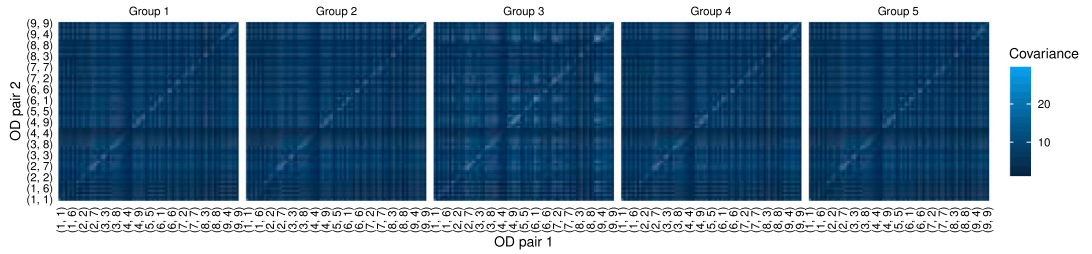
Fig. 25. Clustering results using K-means algorithm for the data shown in Fig. 22.

TAZ 1 is important in this network as both an origin and a destination, TAZ 5 has unusually high "loopback" demands, and TAZ 7 and TAZ 9 are relatively closely related. For another example, in Fig. 26(b), we can see that in Group 3 the OD demands have relatively high variability. Those observations, while are not directly from the clustering results, could be helpful when we are understanding, interpreting, and utilizing the results.

Finally, we show a small example on using the results for transportation management. Once we get the clusters, a natural next step is to find the similarities within each cluster and differences among clusters. In this case of analyzing recurrent traffic patterns,



(a) Mean values of OD demands among all TAZs.



(b) Covariance matrices of OD demands among all TAZ pairs. Certain axis labels are hidden for visualization, but for both axes the OD pairs are arranged in increasing order of the origin and destination TAZ numbers.

Fig. 26. Estimated OD demand mean values and covariance matrices for the results in Fig. 24.

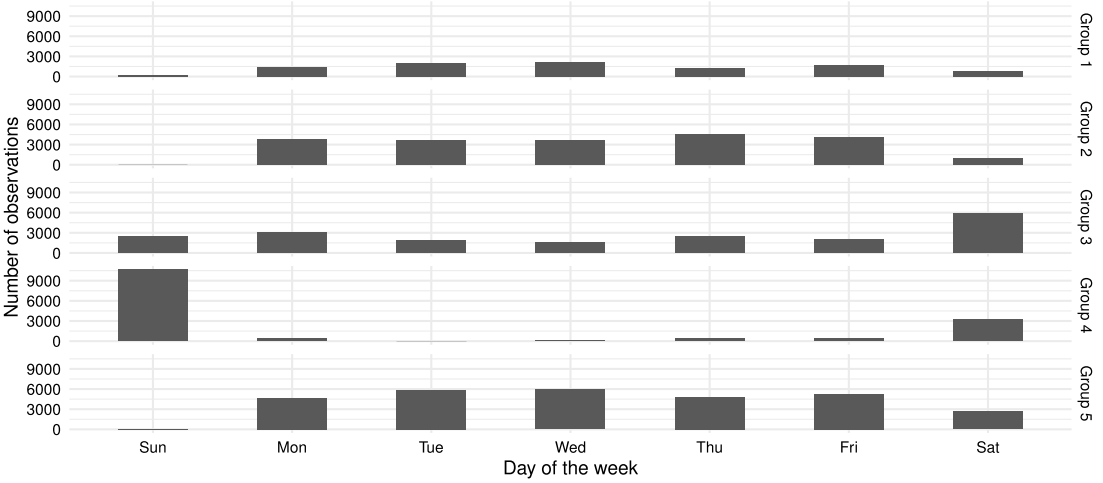


Fig. 27. Numbers of days of the week in each of the groups shown in Fig. 24(a).

oftentimes those patterns are associated with days of the week. For instance, people may tend to use certain roadways more on the weekends. Therefore, we examine the numbers of different days of the week in each of the five groups, shown in Fig. 27. We find that Group 1 and Group 2 contain mainly the weekdays while Group 4 contains almost exclusively weekends. Many Saturdays fall into Group 3 while Sundays are almost always in Group 3 and Group 4. In combination with the results in Fig. 24, those observations are presumably helpful for setting up traffic management strategies tailored for each pattern. For instance, in Group 4, which contains mostly Sundays, links have relatively smaller traffic volume than other groups, so new policies could be issued to migrate certain travel demands from weekdays to Sundays and help alleviate congestion.

6. Conclusion

We propose a novel clustering algorithm for analyzing day-to-day traffic data and discovering traffic patterns in the space of the OD demands, rather than in the space of traffic data directly. This is done through integrating the probabilistic OD demand estimation and statistical traffic assignment into classical generic clustering process. The proposed clustering method features awareness of the topological structure, roadway attributes, and travelers' route choices, of a transportation network and hence often gives new but

sensible ways to partition high-dimensional link traffic count data in comparison with generic clustering algorithms, such as K-means, GMM, etc. While patterns from unsupervised learning algorithm are somewhat subjective depending on how an algorithm sees the difference between data points, the new clustering algorithm provides an interpretable approach to better understand transportation network data, useful for transportation planning and management.

With the proposed algorithm, network conditions are characterized with a mixture of OD demand probability distributions and a probabilistic graphic model that maps OD demand probability distributions to link flow probability distributions. Then an EM algorithm is used to estimate the underlying OD demand probability distributions from daily traffic counts and a group assignment is performed in the OD demand space. In this paper, we use the GESTA model and the PODE model, respectively, in the E-step and M-step. However, the new clustering algorithm is flexible, and other traffic assignment models and OD demand estimation models could be applicable as well.

We examine the model with two hypothetical networks and four synthetic data sets as well as one real data set on a sizable real-world network, the Sacramento regional network. On the regular three-link toy network our algorithm is able to give sensible clusters for both a well-separated data set and a somewhat challenging data set. On the unusual two-link network the two numerical experiments demonstrate the uniqueness of the algorithm – it has knowledge on the correlation among links and is able to handle incomplete data directly. Finally, we conduct an experiment on the real-world network to demonstrate a typical cluster analysis workflow and to show that our algorithm is useful in practice.

Admittedly the algorithm is still in its imperfection and there are two limitations. First, it requires intensive computation due to the complexity of the embedded network models. Many transportation network models are computationally costly to use, let alone multiple such models are employed in this clustering framework. On the Sacramento regional network the algorithm takes around 20 min to discover five clusters while K-means takes less than one minute to run on the same machine. In real-world applications we often need to analyze data sets on much larger networks and identify much more clusters. Therefore, the performance issue is for sure not negligible. Second, the algorithm is hard to train and requires excessive hyperparameter tuning. In addition to the EM algorithm, it uses two non-trivial network models. Therefore, in order to output sensible results, we need to tune all the three sets of parameters together, leading to potentially the unstability of the EM algorithm, analogous to other classical clustering algorithms. In the experiments on the Sacramento network, it is common that the algorithm oscillates among a few local optima, and we have to tweak the termination condition specifically.

Both limitations are because the clustering algorithm amplifies the complexity of the embedded networks models. Considering the flexibility of the framework, it is possible to mitigate the two issues by using other simpler network models. For example, if run time complexity is more of a concern than model performance and intepretability, we can pre-train two black box models to approximate the GESTA and PODE models respectively, and use them in the clustering framework instead. Besides, regarding the desired number of clusters, it is possible and a future research direction to use the non-parametric clustering process with this framework and turn the hyperparameter, number of clusters, into a random variable by adding another level in the graphic model (Neal, 2000; Escobar and West, 1995). In that way, the number of clusters could also be automatically selected by balancing between lowering the number of clusters and maximizing data likelihoods.

To conclude, the algorithm frames a new way to inject domain knowledge of the transportation network into data-driven approaches. It demonstrates that it is possible, actionable, and valuable to tightly couple the transportation models with generic data analytics models. This would shed lights on future applications of data science to transportation planning and management in practice.

CRediT authorship contribution statement

Pengji Zhang: Literature review, Study conception and design, Data acquisition, Data analytics, Programming, Analysis and interpretation of results, Manuscript preparation. **Wei Ma:** Literature review, Data analytics, Programming, Analysis and interpretation of results, Manuscript preparation. **Sean Qian:** Initialize research idea, Study conception and design, Data acquisition, Analysis and interpretation of results, Manuscript preparation.

Acknowledgments

This research is supported by a National Science Foundation, United States grant CMMI-1751448, and a Department of Energy, United States grant DOE-EE0008466. The contents of this article reflect the views of the authors, who are responsible for the facts and accuracy of the information presented herein. The U.S. Government assumes no liability for the contents or use thereof.

References

Antoniak, C.E., 1974. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. Ann. Statist. 1152–1174.

Asif, M.T., Dauwels, J., Goh, C.Y., Oran, A., Fathi, E., Xu, M., Dhanya, M.M., Mitrovic, N., Jaillet, P., 2012. Unsupervised learning based performance analysis of n-support vector regression for speed prediction of a large road network. In: 2012 15th International IEEE Conference on Intelligent Transportation Systems. http://dx.doi.org/10.1109/ITSC.2012.6338917.

Chen, H., Yang, C., Xu, X., 2017. Clustering vehicle temporal and spatial travel behavior using license plate recognition data. J. Adv. Transp. 2017.

Chung, E., 2003. Classification of traffic pattern. In: Proc. of the 11th World Congress on ITS. pp. 687–694.

Daganzo, C.F., Sheffi, Y., 1977. On stochastic models of traffic assignment. Transp. Sci. 11 (3), 253–274.

Dempster, A.P., Laird, N.M., Rubin, D.B., 1977. Maximum likelihood from incomplete data via the EM algorithm. J. R. Stat. Soc. Ser. B Stat. Methodol. 39 (1), 1–22.

Escobar, M.D., West, M., 1995. Bayesian density estimation and inference using mixtures. J. Amer. Statist. Assoc. 90 (430), 577–588.
Fisk, C., 1980. Some developments in equilibrium traffic assignment. Transp. Res. B 14 (3), 243–255.
Gace, I., Pevec, D., Vdovic, H., Babic, J., Podobnik, V., 2021. Driving style categorisation based on unsupervised learning: A step towards sustainable transportation. In: 2021 6th International Conference on Smart and Sustainable Technologies. SpliTech, http://dx.doi.org/10.23919/SpliTech52315.2021.9566371.
Gu, Y., Qian, Z., Xie, X.-F., Smith, S., 2016. An unsupervised learning approach for analyzing traffic impacts under arterial road closures: Case study of East Liberty in Pittsburgh. J. Transp. Eng. 142 (9).
Guardiola, I.G., Leon, T., Mallor, F., 2014. A functional approach to monitor and recognize patterns of daily traffic profiles. Transp. Res. B 65, 119–136.
Hartigan, J.A., Wong, M.A., 1979. Algorithm AS 136: A k-means clustering algorithm. J. R. Stat. Soc. Ser C (Applied Statistics) 28 (1), 100–108.
Jain, A.K., Murty, M.N., Flynn, P.J., 1999. Data clustering: A review. ACM Comput. Surv. 31 (3), 264–323.
Ku, W.C., Jagadeesh, G.R., Prakash, A., Srikanthan, T., 2016. A clustering-based approach for data-driven imputation of missing traffic data. In: 2016 IEEE Forum on Integrated and Sustainable Transportation Systems. FISTS, http://dx.doi.org/10.1109/FISTS.2016.7552320.
Li, S., Wang, W., Mo, Z., Zhao, D., 2018. Cluster naturalistic driving encounters using deep unsupervised learning. In: 2018 IEEE Intelligent Vehicles Symposium. IV, http://dx.doi.org/10.1109/IVS.2018.8500529.
Ma, W., Qian, Z.S., 2017. On the variance of recurrent traffic flow for statistical traffic assignment. Transp. Res. C 81, 57–82.
Ma, W., Qian, Z.S., 2018. Statistical inference of probabilistic origin-destination demand using day-to-day traffic data. Transp. Res. C 88, 227–256.
Markos, C., Yu, J.J., 2020. Unsupervised deep learning for GPS-based transportation mode identification. In: 2020 IEEE 23rd International Conference on Intelligent Transportation Systems. ITSC, http://dx.doi.org/10.1109/ITSC45102.2020.9294673.
McInnes, L., Healy, J., Melville, J., 2020. UMAP: Uniform manifold approximation and projection for dimension reduction. arXiv:1802.03426.
Moon, T.K., 1996. The expectation-maximization algorithm. IEEE Signal Process. Mag. 13 (6), 47–60.
Neal, R.M., 2000. Markov chain sampling methods for Dirichlet process mixture models. J. Comput. Graph. Statist. 9 (2), 249–265.
Saha, R., Tariq, M.T., Hadi, M., Xiao, Y., 2019. Pattern recognition using clustering analysis to support transportation system management, operations, and modeling. J. Adv. Transp. 2019.
Shao, H., Lam, W.H., Tam, M.L., 2006. A reliability-based stochastic traffic assignment model for network with multiple user classes under uncertainty in demand. Netw. Spat. Econ. 6 (3), 173–204.
Soriguera, F., 2012. Deriving traffic flow patterns from historical data. J. Transp. Eng. 138 (12), 1430–1441.
Watling, D., 2002a. A second order stochastic network equilibrium model, I: Theoretical foundation. Transp. Sci. 36 (2), 149–166.
Watling, D., 2002b. A second order stochastic network equilibrium model, II: Solution method and numerical experiments. Transp. Sci. 36 (2), 167–183.
Weijermars, W., Van Berkum, E., 2005. Analyzing highway flow patterns using cluster analysis. In: Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005.. IEEE, pp. 308–313.
Xia, J., Chen, M., 2007. Defining traffic flow phases using intelligent transportation systems-generated data. J. Intell. Transp. Syst. 11 (1), 15–24.