Transportation Research Part B 80 (2015) 150–172

Contents lists available at ScienceDirect

Transportation Research Part B

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

Fine-grained OD estimation with automated zoning and sparsity regularisation

Aditya Krishna Menon [a,*], Chen Cai [b], Weihong Wang [b], Tao Wen [b], Fang Chen [b]

[a] National ICT Australia and the Australian National University, Canberra, ACT, Australia
[b] National ICT Australia and the University of New South Wales, Sydney, NSW, Australia

ARTICLE INFO

Article history:
Received 30 April 2015
Received in revised form 2 July 2015
Accepted 2 July 2015

Keywords:
OD estimation
Traffic analysis zones
Sparsity

ABSTRACT

Given a road network, a fundamental object of interest is the matrix of origin destination (OD) flows. Estimation of this matrix involves at least three sub-problems: (i) determining a suitable set of traffic analysis zones, (ii) the formulation of an optimisation problem to determine the OD matrix, and (iii) a means of evaluating a candidate estimate of the OD matrix. This paper describes a means of addressing each of these concerns. We propose to automatically uncover a suitable set of traffic analysis zones based on observed link flows. We then employ regularisation to encourage the estimation of a *sparse* OD matrix. We finally propose to evaluate a candidate OD matrix based on its predictive power on *held out* link flows. Analysis of our approach on a real-world transport network reveals that it discovers automated zones that accurately capture regions of interest in the network, and a corresponding OD matrix that accurately predicts observed link flows.

1. Problem statement

Origin–destination (OD) flows are a fundamental object of interest in the study of urban transportation networks (Ortuzar and Willumnsen, 2011, Chapter 5). In principle, these are the steady-state flow of traffic[1] between any pair of traffic intersections. In practice, one instead considers the steady-state flow of traffic between any pair of *traffic analysis zones* (henceforth simply *zones*). A zone represents an aggregation of intersections in the underlying network, and offers conceptually and computationally simpler modelling.

More formally, consider a directed graph $\mathcal{G}$ representing a road network, where the nodes in the graph represent traffic intersections, and the links represent road segments between intersections. We denote the set of nodes by $\mathcal{N}$ and the set of links by $\mathcal{L}$, and will often write $\mathcal{G} = (\mathcal{N}, \mathcal{L})$. Suppose also that there is a vector $\mathbf{y} \in \mathbb{N}_+^{|\mathcal{L}|}$, representing the count of steady-state traffic flow on each road segment over some time period (e.g. the morning rush hour). To analyse OD flows, one defines a special set of *virtual nodes* $\mathcal{N}_{\text{virt}}$, which are connected via a number of *virtual links* $\mathcal{L}_{\text{virt}}$ to nodes in $\mathcal{N}$. The set of traffic analysis zones $\mathcal{Z}$ is simply the set of virtual nodes and their corresponding virtual links, so that each virtual node represents the focal point of an individual zone. Given $\mathcal{Z}$, the *origin–destination matrix* (*OD matrix*) $\mathbf{X}$ of size $|\mathcal{Z}| \times |\mathcal{Z}|$ represents, for any pair of zones $(z, z') \in \mathcal{Z} \times \mathcal{Z}$, the steady-state flow of traffic that begins at zone z and ends at zone z'.

* Corresponding author.
E-mail addresses: aditya.menon@nicta.com.au (A.K. Menon), chen.cai@nicta.com.au (C. Cai), weihong.wang@nicta.com.au (W. Wang), tao.wen@nicta.com.au (T. Wen), fang.chen@nicta.com.au (F. Chen).

[1] More generally, one may be interested in time-varying OD matrices. While the topic of considerable research in its own right (Cascetta et al., 2013), we do not consider this problem here.

http://dx.doi.org/10.1016/j.trb.2015.07.003
0191-2615/

The OD matrix is a valuable tool for understanding and forecasting usage patterns of a network. Given the OD matrix, one can then make forecasts about traffic flows on a *different network* $\mathcal{G}' = (\mathcal{N}', \mathcal{L}')$, under the assumption that $\mathcal{G}$ and $\mathcal{G}'$ possess commensurate OD flows. For example, the network $\mathcal{G}'$ might be identical to $\mathcal{G}$, except that certain links are removed; the forecast flows may then be used to assess the impact this change has on the network. The predicted flows may be generated by any *route assignment* model, such as for example one based on a user equilibrium assumption (Sheffi, 1985, Chapter 3).

This paper is concerned with the *OD estimation problem*. Here, the aim is to recover $\mathbf{X}$ given the topology of the road network $\mathcal{G}$, observed link flows $\mathbf{y}$, and the definition of the traffic analysis zones $\mathcal{Z}$. Any attempt at OD estimation faces several entwined questions. We are interested in three basic questions:

- *Zoning construction*: given a network $\mathcal{G}$ and link flows $\mathbf{y}$, how does one define the traffic analysis zones $\mathcal{Z}$? As the choice of $\mathcal{Z}$ defines the precise pairwise flows we are interested in estimating, it plays a crucial role in determining whether the resulting OD matrix can be reliably estimated, and whether it is useful for analysis and forecasting.
- *Statistical OD estimation*: given a network $\mathcal{G}$, zones $\mathcal{Z}$, and link flows $\mathbf{y}$, how does one estimate the OD matrix $\mathbf{X}$ that best explains the flows $\mathbf{y}$? The OD matrix can be understood as the solution to a potentially ill-posed linear system. Its estimation thus requires some means of choosing amongst potentially multiple candidate OD matrices.
- *OD evaluation*: given an estimate of the OD matrix, $\widehat{\mathbf{X}}$, how does one evaluate its efficacy? As there is typically no direct ground truth for the OD matrix, any analysis of the quality of its estimate must rely on auxiliary measures.

In this paper, we explore techniques to answer all three questions.[2] In a nutshell, we propose:

- the *automated design* of *fine-grained* traffic analysis zones, based on the intuition that a good set of analysis zones results in an equilibrium assignment that can accurately explain observed link flows;
- an OD estimation procedure that encourages the estimation of *sparse* OD matrices, which mitigate ill-posedness, and additionally, offer interpretability[3];
- the use of *held-out flow predictions* to evaluate efficacy of an estimated OD matrix $\mathbf{X}$, by viewing OD estimation as a type of general regression problem.

We evaluate our approach on a real-world network, and find that we can discover an automated zoning of the network which has advantages over a manual zoning provided by a domain expert, and learn a sparse OD matrix over this zoning that reliably predicts link flows.

This paper is organised as follows. In Section 2, we give more detail on traditional approaches to the above three components of OD estimation. In Section 3, we discuss the above three challenges in more detail, and describe prior work in the literature on addressing these challenges. Then, in Section 4–6, we detail the elements of our solution, which attempt to employ machine learning to aid in solving the estimation problem. We then evaluate our method on a real-world network in Section 7. We conclude in Section 8 with some discussion on areas for future research.

Table 1 summarises some commonly used symbols in the paper.

2. Background: zoning and OD estimation algorithms

In this section, we describe in more detail each of the three constituent problems involved in OD estimation.

2.1. Zoning construction

In designing traffic analysis zones, there are two basic questions: how many zones to construct, and how to choose the constituent nodes in each zone. Suppose we have fixed the number of zones to be K. As mentioned in the Introduction, the standard mechanism for incorporating zones into the network is via the addition of virtual nodes and links. As the name suggests, these nodes (links) do not correspond to physical intersections (segments), and are simply mathematical constructs that facilitate simplified analysis. Conceptually, it may be useful to think of virtual nodes as corresponding to the centroids of the (physical) nodes they are connected to.

Formally, to define the K zones in the network, we define the set of virtual nodes $\mathcal{N}_{\text{virt}} = \{v_{\text{virt}}^{(n)}\}_{n=1}^{K}$, and the corresponding set of virtual links $\mathcal{L}_{\text{virt}} = \{\mathcal{L}_{\text{virt}}^{(n)}\}_{n=1}^{K}$ that connect these virtual nodes to those in $\mathcal{N}$. The set of traffic analysis zones is then simply

[2] This work extends the conference publication Menon et al. (2015).

[3] Interpretability is afforded since the matrix specifies trips between only a small subset of OD pairs. These pairs can be manually assessed based on domain knowledge, as a further test of the appropriateness of the estimated OD matrix. Interpretability may be achieved with other types of OD matrices, e.g. those that are low-rank; nonetheless, for a generic non-sparse OD matrix, one has trips between most pairs of nodes, it is more challenging to perform a manual inspection of the resulting estimates.

Table 1
Commonly used symbols.

Symbol	Meaning
$\mathcal{G}$	Road network
$\mathcal{N}$	Nodes (traffic intersections)
$\mathcal{L}$	Links (road segments)
$\mathcal{Z}$	Traffic analysis zones
$\mathbf{y}$	Link flows
$\mathbf{X}$	OD matrix
$\mathbf{x}$	OD matrix (vectorised form)
$\mathbf{A}$	Assignment map
$\mathbf{V}$	OD matrix prior covariance
$\mathbf{W}$	Link flow likelihood covariance
$\|\|\cdot\|\|_0$	ℓ_0 "norm" of vector
$\|\|\cdot\|\|_1$	ℓ_1 norm of vector
$\cdot \succeq \mathbf{0}$	Nonnegativity constraint on vector

$$\mathcal{Z} = \bigcup_{n=1}^{K} \left(v_{\text{virt}}^{(n)}, \mathcal{L}_{\text{virt}}^{(n)} \right).$$

Given $\mathcal{Z}$, all subsequent analysis is performed on the "zoned network" $\mathcal{G}_{\text{zoned}} = (\mathcal{N} \cup \mathcal{N}_{\text{virt}}, \mathcal{L} \cup \mathcal{L}_{\text{virt}})$.

In principle, there is no restriction on the choice of $\mathcal{N}_{\text{virt}}$. In particular, it is feasible for $\mathcal{N}_{\text{virt}} \subseteq \mathcal{N}$, so that virtual nodes correspond to particular nodes in the original graph. In the extreme case where $\mathcal{N}_{\text{virt}} = \mathcal{N}$ and $\mathcal{L}_{\text{virt}} = \emptyset$, the zoned network $\mathcal{G}_{\text{zoned}}$ is identical to the original network.

We will interchangeably refer to a zone $z \in \mathcal{Z}$ and its corresponding virtual node, for example to refer to the flow originating from a zone. Given a zone $z = (v_{\text{virt}}, \mathcal{L}_{\text{virt}})$, we view a virtual link to a non-virtual node $v \in \mathcal{N}$ to mean that v "belongs to" zone z. Each $z \in \mathcal{Z}$ thus represents an aggregation of nodes in the original graph.[4]

In conventional traffic modelling, one typically does not algorithmically determine any of the choices above, namely, the number of zones $|\mathcal{Z}|$, the position of the virtual nodes $\mathcal{N}_{\text{virt}}$, and the connections of the virtual links $\mathcal{L}_{\text{virt}}$. Instead, these are typically left as choices for a domain expert (Sheffi, 1985, p. 16). Several considerations often determine the precise zone boundaries and elements, most notably, alignment with suburban and provincial regions. While there are principles that underpin all such considerations, there is rarely any algorithmic maximisation of an explicit utility function.

2.2. Statistical OD estimation

Suppose we have fixed our traffic analysis zones $\mathcal{Z}$, so that the OD matrix $\mathbf{X}$ is of dimensionality $\mathbb{N}_+^{|\mathcal{Z}| \times |\mathcal{Z}|}$. For convenience, we shall interchangeably refer to the OD matrix by $\mathbf{X}$ and its vectorised form, $\mathbf{x} = \text{vec}(\mathbf{X}) \in \mathbb{N}_+^{|\mathcal{Z}|^2 \times 1}$. There are roughly three approaches to estimating the OD matrix (Cascetta, 1984):

- The simplest is *direct sample estimation*, wherein surveys or interviews are conducted to determine common origin–destination pairs for individuals. Aggregating these responses gives estimates of the OD matrix cells.
- Survey information is often incomplete, and thus may provide no (or highly biased) information about certain OD pairs. Inferences can nonetheless be made by performing *model estimation*, wherein a particular model is assumed to relate OD flow to several explanatory variables, such as the mean income of residents with a zone. A well-known instance of this approach is the *gravity model* (Ortuzar and Willumnsen, 2011, p. 182; Zhang et al., 2003a).
- Survey information is often based on small samples, and thus unreliable. A richer class of techniques are those based on *estimation from loop counts* (Willumsen, 1981). Loop counts embed information about OD and route choices, and are typically more plentiful and reliable than survey data. These are sometimes referred to as structured and unstructured methods, or parameter calibration and matrix estimation methods respectively (Tamin and Willumsen, 1989).

We will focus on the latter class of methods, noting that they may be extended to exploit survey information if it is available. Examples of methods that exploit link counts include those based on maximum entropy modelling (Van-Zuylen and Willumsen, 1980), maximum likelihood estimation (Vardi, 1996; Spiess, 1987), and Bayesian inference (Maher, 1983; Tebaldi and West, 1998; Hazelton, 2015). A basic fact that underlies these approaches is that the OD matrix $\mathbf{x}$ is related to the link flows $\mathbf{y}$ via the *flow-conservation* equation,

$$\mathbf{A}\mathbf{x} = \mathbf{y}, \tag{1}$$

[4] While virtual nodes aggregate the original nodes in the graph, one still retains the original nodes for all subsequent modelling and analysis. This is because the original nodes may be used as intermediate nodes for travelling from one zone to another.

where $\mathbf{A} \in [0,1]^{|\mathcal{L}| \times |\mathcal{Z}|^2}$ is the *assignment map*, whose entries denote the probability of a particular link being used for travel between an OD pair. Estimating the OD matrix is thus equivalent to solving this linear system. In practice, one expects there to be some noise in the observed link flows. To account for this, one can impose a noise distribution $\mathbb{P}(\mathbf{y}|\mathbf{A},\mathbf{x})$, and the compute the maximum likelihood estimate of $\mathbf{x}$,

$$\min_{\mathbf{x}} -\log \mathbb{P}(\mathbf{y}|\mathbf{A},\mathbf{x}). \tag{2}$$

For example, with isotropic Gaussian noise, we arrive at the familiar ordinary least squares objective,

$$\min_{\mathbf{x}} \sum_{e \in \mathcal{L}} (\mathbf{A}_{e:}\mathbf{x} - \mathbf{y}_e)^2. \tag{3}$$

There are at least two challenges with solving Eq. (1), or its probabilistic counterpart Eq. (2). First, the linear system is ostensibly strongly ill-posed or undetermined. (In Section 3.3, we justify why we use the term "ill-posed" to refer to the existence of multiple solutions.) Second, the assignment matrix $\mathbf{A}$ must itself be computed from the network. Each of these issues has been the subject of considerable research.

2.2.1. Regularisation of OD estimates

The ill-posedness of the linear system in Eq. (1) is apparent[5]: we have $|\mathcal{Z}|^2$ unknowns and $|\mathcal{L}|$ equations. This means there may not be a unique $\mathbf{x}$ satisfying Eq. (1). In particular, ill-posedness is guaranteed if there is some $\mathbf{x} \succeq \mathbf{0}$ such that $\mathbf{A}\mathbf{x} = \mathbf{0}$; this is because $\mathbf{x}$ can then be added onto any candidate solution to Eq. (1) without affecting the flow estimates.

One strategy to mitigate ill-posedness is to inject some prior or domain knowledge into the estimation problem. Typically, this is done by relying on prior OD estimates collected e.g. from a survey. Suppose $\mathbf{x}^{(\text{old})}$ denotes this prior estimate of the OD matrix. Then, the generalised least squares estimator (Cascetta and Nguyen, 1988; Bell and Iida, 1997, p. 155) aims to find

$$\min_{\mathbf{x}} (\mathbf{A}\mathbf{x} - \mathbf{y})^T \mathbf{W}^{-1} (\mathbf{A}\mathbf{x} - \mathbf{y}) + (\mathbf{x} - \mathbf{x}^{(\text{old})})^T \mathbf{V}^{-1} (\mathbf{x} - \mathbf{x}^{(\text{old})}), \tag{4}$$

for appropriate (typically diagonal) weighting matrices $\mathbf{W}, \mathbf{V}$. The first term in Eq. (4) is seen to seek a solution to Eq. (1), with some weighting of different links. By contrast, the second term does not depend on the link flows, but rather, penalises $\mathbf{x}$ based on distance to $\mathbf{x}^{(\text{old})}$. The second term is often referred to as a *regulariser*. Intuitively, regularisation aims to encourage solutions that explain the observed data well, without being overly "complex"; put another way, they attempt to make the model fit to only the signal in the data, and ignore the noise. In this case, regularisation is achieved by ensuring that the model only significantly deviates from the prior OD matrix if it is very confident that such a deviation is needed to explain the observed link flows.

As with ordinary least squares, the generalised least squares objective admits a closed-form solution. The GLS equation can be seen as a *maximum a posteriori* estimate of $\mathbb{P}(\mathbf{x}|\mathbf{A},\mathbf{y})$, under Gaussian models for the prior $\mathbb{P}(\mathbf{x})$ and likelihood $\mathbb{P}(\mathbf{y}|\mathbf{A},\mathbf{x})$, with covariance matrices $\mathbf{W}, \mathbf{V}$ respectively. From a Bayesian perspective, one may also obtain an estimate of the posterior covariance of the OD matrix (Cascetta and Nguyen, 1988).

A pleasant consequence of using a prior OD matrix $\mathbf{x}^{(\text{old})}$ is that the objective corresponding to e.g. Eq. (4) is strictly convex (assuming the diagonal entries of $\mathbf{V}$ are positive), meaning that there will in fact be a unique solution: amongst all OD matrices that have the same predicted link flows, we seek the one that is closest to the given prior OD matrix. Therefore, this is a viable approach to mitigating ill-posedness.

While GLS is a popular OD estimation technique, it is not the only one. Another popular class of techniques are based on the principle of *maximum entropy* (Van-Zuylen and Willumsen, 1980; Bell and Iida, 1997, p. 150). Here, the objective is typically written

$$\min_{\mathbf{x} \succeq \mathbf{0}} \mathrm{KL}(\mathbf{x} \| \mathbf{x}^{(\text{old})}) : \mathbf{A}\mathbf{x} = \mathbf{y}, \tag{5}$$

where $\mathrm{KL}(\cdot,\cdot)$ denotes the Kullback–Leibler (KL) divergence, or relative entropy, between two unnormalised distributions,

$$\mathrm{KL}(\mathbf{p} \| \mathbf{q}) = \sum_i \mathbf{p}_i \cdot \log \frac{\mathbf{p}_i}{\mathbf{q}_i}.$$

As the KL divergence is minimised when $\mathbf{p} = \mathbf{q}$, this approach also encourages the estimated OD matrix to be close to the prior one, while explaining the observed flows well.

2.2.2. Traffic assignment algorithms

While above we have assumed $\mathbf{A}$ fixed and known, in general it must also be computed from the network. The simplest approach to computing $\mathbf{A}$ is via "all-or-nothing" assignment (Sheffi, 1985, p. 73; Ortuzar and Willumnsen, 2011, p. 359):

[5] The system is potentially well-posed if one considers correlations in flows across *multiple* days. For example, Vardi (1996) showed that under mild assumptions, the OD matrix is identifiable for Poisson models. Hazelton (2001) showed that second-order information present due to temporal trends may also induce identifiability. In the sequel, we discuss the issue of ill-posedness further.

here, we simply compute the shortest path between any pair of zones, where the weight on each link in the graph represent some known impedances or travel times between the respective nodes. This has the advantage of simplicity, but the disadvantage of not capturing a basic fact about route choice, namely, that travel time between a link is intimately connected to the flow assigned to the link.

A more sophisticated approach is to directly compute an *equilibrium based assignment*, such as a deterministic user equilibrium assignment (Sheffi, 1985; Bell and Iida, 1997, p. 90), which explicitly considers the relationship between flow and travel time. Interestingly, the solution of this assignment problem by the Frank–Wolfe algorithm results in an iterative procedure wherein one repeatedly needs to solve an all-or-nothing assignment problem based on the current estimated travel times (Sheffi, 1985, p. 118).

Standard equilibrium based assignment approaches rely on knowledge of the OD matrix, so as to compute the flows along various paths. This can be alleviated to some extent by assuming the OD matrix depends in a known way on the travel times on paths connecting pairs of zones (known as variable demand models (Sheffi, 1985, Chapter 6)), or by directly estimating path flows in scenarios where one observes flow on every link (Sherali et al., 1994). However, in general, the estimation of the assignment map and OD matrix are entwined. One procedure is to cast the problem in the framework of bilevel programming (Yang et al., 1992; Bell and Iida, 1997, Section 7.4). Practically, what this amounts to is the alternating solution of the linear system in Eq. (1) with respect to $\mathbf{x}$, and an appropriate algorithm for computing $\mathbf{A}$ from $\mathbf{x}$, e.g. a deterministic user equilibrium.

2.3. OD evaluation

Much prior work on OD estimation evaluates the efficacy of the estimation procedure by applying it to a synthetic network where the ground-truth OD is known. While sensible, it is of interest to be able to compare different OD estimates on a real network where the ground-truth is of course unknown. The challenge is determining a suitable auxiliary measure of quality. We are not aware of much work that directly addresses this issue.

A natural idea is to assess its predictive power in forecasting link flows at future time periods, but the ill-posedness issue arises: suppose the flow-conservation equation (Eq. (1)) is ill-posed, so that $\mathbf{A}\mathbf{x}_1 = \mathbf{A}\mathbf{x}_2$ for some $\mathbf{x}_1 \neq \mathbf{x}_2$. Then several OD matrices will yield exactly the same link flows. An alternative is to resort to interpretability, but this may be difficult to achieve with more fine-grained estimates.

3. A motivating example and resulting challenges

In this section, we present the motivating example behind our work. We then review the ensuing challenges involved in each of the three items we described earlier. We also discuss existing work that we are aware of to deal with these challenges.

3.1. Motivation: forecasting flow under network change

Our interest is in forecasting the impact of a change to a large urban network, described by some $\mathcal{G} = (\mathcal{N}, \mathcal{L})$. Specifically, we have an urban network where a number of road segments are to be closed to general vehicles. Thus, we effectively have a new network $\mathcal{G}' = (\mathcal{N}, \mathcal{L}')$, where $\mathcal{L}' \subset \mathcal{L}$, so that certain links are deleted.

In lieu of any meaningful model of the change in OD, we will assume that the OD flows on $\mathcal{G}'$ will be similar to that on $\mathcal{G}$. With this assumption, we can perform an equilibrium assignment of the flows on the network. This gives estimates of the flows in the network after deletion of certain links. In order for these estimated flows to be meaningful, care is needed in each stage of the OD estimation procedure. We clarify these challenges.

3.2. Zoning construction

To appreciate the challenges inherent in the design of traffic analysis zones $\mathcal{Z}$, it is first worth noting the implications of two extreme choices of $\mathcal{Z}$. Ideally, one would like to operate on the original graph $\mathcal{G}$, without any zoning whatsoever, so that the OD matrix comprises flows between each pair of intersections. This corresponds to the choice $\mathcal{Z} = (\mathcal{N}, \emptyset)$. However, such a choice has the drawback of potentially leading to a highly ill-posed estimation for the OD matrix, as one needs to solve for $|\mathcal{N}|^2$ unknowns given only $|\mathcal{L}|$ equations. A computational drawback is that at increasing level of granularity, estimating $|\mathcal{N}|^2$ parameters may be infeasible on even moderately-sized networks.

Conversely, by choosing $|\mathcal{Z}|$ to be small, one mitigates ill-posedness and computational issues. However, this comes at a potentially significant cost: *intra-zonal flows* – i.e. flows between any pair of nodes v, v' that are in the same zone – are ignored. When the OD matrix is to be used for forecasting under changes to the network, such intra-zonal flow can be harmful, especially if it applies to trips that explain flow on high volume links.

Between these two extremes, then, one faces a tradeoff between statistical and predictive precision. The precise choice as to this tradeoff is often left as a specification for a domain expert (Ortuzar and Willumnsen, 2011). There have been alternate proposals that attempt to make the procedure more automated. A notable example is the work of Martínez et al. (2009), who

propose an optimisation framework that takes into account several desiderata for the design of analysis zones, including the minimisation of intra-zonal flows as mentioned, but also the geographic contiguity of the resulting partition of the road network. The framework relies on the availability of a sufficiently detailed prior OD matrix derived from survey data, however. As we now discuss, this is not present in our problem.

3.3. Statistical OD estimation

We have established earlier that Eq. (1) is ill-posed. Observe that for the task of predicting link flows on the *current* network, this ill-posedness is not an issue: all feasible solutions to Eq. (1) will result in the same flow estimates, and so for this purpose, the precise choice of OD matrix is irrelevant. However, for the task of predicting link flows on a *modified* network, as in our problem, ill-posedness is a significant issue: in general, one can expect the multiple solutions to Eq. (1) to yield different flow estimates for the modified network. If one happens to select a "wrong" OD matrix that gives very different estimates on the modified network than the "right" OD matrix, this could lead to misguide planning operations.

The above suggests to mitigate ill-posedness by regularising towards a prior OD matrix. A key challenge with OD estimation in our problem is the lack of a suitable prior OD matrix. Such matrices are usually derived from survey data. However, we do not have sufficiently many samples in order to construct a reliable estimate. Thus, the standard approach of regularising OD estimates towards a fixed prior OD, as outlined in Section 2.2.1, is not applicable. One can attempt to use these approaches with a naïve OD matrix, such as a uniform matrix. However, aside from being lacking strong justification, such a choice may not lead to good results, as one may predict nonnegligible flow on OD pairs that have few or zero trips.

Nonetheless, we still believe that it should be possible to mitigate the ill-posedness of the OD estimation. The use of a prior OD matrices can be thought of as a regularisation based on historical information. We can equally consider regularisation based on some domain knowledge about the space of plausible OD flows in a reasonable urban transportation network. The usefulness of such schemes will of course depend on how well the assumption encoded in the regulariser matches the actual network at hand. We call these *generic regularisers*; our interest is in evaluating whether there exist useful generic regularisers for our problem.

3.4. OD evaluation

Perhaps the ideal means of assessing the quality of OD estimates is in the ability to accurately forecast traffic flows under a changed network $\mathcal{G}'$ with commensurate demands; however, major network changes of this type are not common, making it difficult to acquire the necessary data.

Having outlined the key challenges involved in OD estimation for our problem, we now detail the elements of our solution to each of the challenges.

4. An automated zoning algorithm

Our design of traffic analysis zones is based on a few simple observations that are worth explicating. First, as noted previously, while a per-node zoning scheme poses statistical and computational challenges, it is the gold standard in terms of modelling capability: it contains at least as much information as any other zoning scheme, by virtue of node aggregation only ever losing (or at best maintaining) information. Therefore, it serves as a useful starting point from which to begin any attempt at zoning.

Second, it is not necessary to associate every node $v \in \mathcal{N}$ with a traffic analysis zone. Crucially, this does not preclude any links involving non-included nodes as being a component of the paths between some origin–destination pair. Consider the network shown in Fig. 1. Here, it is natural to associate only the nodes A, D, E with virtual nodes, as they represent the boundaries of the network. Nonetheless, we see that the link between $B \to C$ is a crucial component of all flow analysis, since it must be used when travelling from $A \to D$ or $A \to E$.

Third, our interest is in the use of the OD matrix to forecast the effect of changes to the network. Thus, our primary concern is in ensuring that the resulting zoning can account for the observed flows on network, i.e. that the optimal assignment and OD matrices explain the flow on the links.[6]

We use these facts to design an automated zoning algorithm. Our zoning will be "fine-grained" in the sense that it does not group together multiple nodes, but rather, selects *individual nodes* as centres of traffic analysis zones. The choice of the number of such "fine-grained zones" will be determined based on a well-defined objective function.

4.1. Zoning objective: intuitive formulation

We propose a simple zoning scheme which selects as virtual nodes a subset of the original graph nodes $\mathcal{N}$. The basic idea is to perform an initial per-node zoning, and then use the information derived from this to select a small subset of nodes that explain most of the traffic in the network. In particular, we wish to select a subset of nodes satisfying two properties:

[6] There may of course be other desiderata or constraints when constructing a zoning, for example, respecting suburban or demographic divisions. We do not directly consider these in this paper, as our goal is simply to be able to accurately forecast flows.

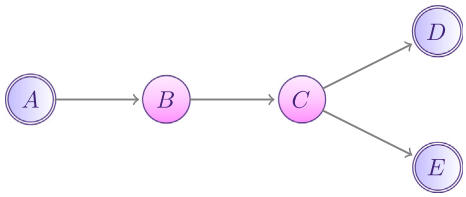


Fig. 1. Network where some nodes (B, C) are not associated to any zones.

- the subset is as small as possible; and,
- the travel between pairs of nodes in this subset accounts for most of the flow observed on the links.

For the second desiderata, we shall attempt to ensure that the flows produced by an equilibrium assignment on our chosen zoning matches the observed flows as well as possible. Doing so will, unsurprisingly, require a number of assumptions; but, we shall see that the resulting automated zoning will have favourable empirical performance compared to a manually defined zoning.

We now describe our zoning algorithm more formally.

4.2. Zoning objective: mathematical formulation

To formalise our zoning algorithm, let $\mathbf{s} \in \{0,1\}^{|\mathcal{N}|}$ be a selection variable, denoting whether or not a given node is included in the final set of zones. To maintain the first desiderata above, we want to minimise

$$\mathrm{nnz}(\mathbf{s}) = ||\mathbf{s}||_0.$$

To address the second desiderata, we need a way to explain the flow on a given link. We use as our reference the equilibrium flow predicted for a given zoning specified by nodes in $\mathbf{s}$. Suppose $\mathbf{A}^{(\mathbf{s})}$ represents the optimal equilibrium routing assignment map, and $\mathbf{x}^{(\mathbf{s})}$ represents the optimal OD matrix for this zoning. Then, we wish to minimise[7]

$$||\mathbf{A}^{(\mathbf{s})}\mathbf{x}^{(\mathbf{s})} - \mathbf{y}||_2^2.$$

so as to ensure maximal explanation of the flow on links in the network.

Combining the formalisations of the two desiderata, our aim is to find a Pareto optimal point for the bicriteria objective

$$\min_{\mathbf{s}\in\{0,1\}^{|\mathcal{N}|}} \left\{ ||\mathbf{s}||_0, ||\mathbf{A}^{(\mathbf{s})}\mathbf{x}^{(\mathbf{s})} - \mathbf{y}||_2^2 \right\}.$$

An immediate challenge arises in minimising this objective: it is highly non-convex, and requires searching over an exponentially large space of candidate solutions. Further, we face the problem of having to estimate both $\mathbf{A}^{(\mathbf{s})}$ and $\mathbf{x}^{(\mathbf{s})}$ for every candidate solution.

However, we now show how one can efficiently find a reasonable approximate minimiser to the objective by making the following assumptions:

Assumption A. The assignment map $\mathbf{A}^{(\mathrm{full})}$, produced by performing equilibrium assignment on a per-node zoning with a uniform OD matrix, is a reasonable reflection of the optimal map for this zoning.

Assumption B. Given a zoning specified by $\mathbf{s}, \mathbf{A}^{(\mathbf{s})}$ is simply a sub matrix of $\mathbf{A}^{(\mathrm{full})}$, where we only select the OD pairs corresponding to nodes in $\mathbf{s}$.

Assumption C. Given a zoning specified by $\mathbf{s}$, the optimal OD matrix is uniform, i.e. $\mathbf{x}^{(\mathbf{s})} = \frac{T}{||\mathbf{s}||_0} \cdot \mathbf{1}$, where T denotes the total number of trips in the network.

In practice, we cannot expect these assumptions to exactly hold; however, they greatly simply further analysis. We can partially justify them as follows:

[7] For simplicity, we do not including a weighting matrix $\mathbf{W}$ in this formulation, as we observed the construction of reasonable zoning without it. In general, this assumption may not be appropriate, as it does not allow for heteroskedastic noise, or for correlations amongst the link flows. We emphasise however that all subsequent analysis can easily incorporate such a $\mathbf{W}$.

- *Assumption A*: we can expect with a uniform OD that $\mathbf{A}^{(\text{full})}$ will minimally capture the top few shortest paths between pairs of nodes, with some compensation for any competition on bottleneck links. Thus, the result can be considered as a refinement of a simple "all or nothing" assignment.
- *Assumption B*: note that with equilibrium routing, if $\mathbf{A}^{(\text{full})}_{e,(v,v')} = 0$, then $\mathbf{A}^{(\mathbf{s})}_{e,(v,v')} = 0$ i.e. by selecting a subset of nodes, we never introduce new paths that were not present in the per-node zoning. Therefore, the difference between the two will only be in the relative values of the nonzero entries.
- *Assumption C*: this is arguably the most restrictive assumption. While the use of a uniform OD does affect the resulting zoning, we emphasise that given the zoning, we will perform statistical OD estimation in order to obtain a more reliable OD estimate. More sophisticated choices of $\mathbf{x}^{(\mathbf{s})}$ are possible via iterative schemes: for example, given a zoning generated by a uniform $\mathbf{x}^{(\mathbf{s})}$, and the resulting OD matrix $\mathbf{x}$, one could disaggregate the flow in this matrix uniformly across the sites in each zone, yielding a refined $\mathbf{x}^{(\mathbf{s})}$. We leave exploration such schemes to future work.

We shall henceforth take the assumptions to be true. With these, the second desiderata is to minimise

$$\left|\left|\frac{1}{||\mathbf{s}||_0}\mathbf{A}^{(\mathbf{s})}\cdot\mathbf{1}-\frac{1}{T}\cdot\mathbf{y}\right|\right|_2^2.$$

Thus, our bicriteria objective now simplifies to

$$\boxed{\min_{\mathbf{s}\in\{0,1\}^{|\mathcal{N}|}}\left\{||\mathbf{s}||_0,\left|\left|\tfrac{1}{||\mathbf{s}||_0}\mathbf{A}^{(\mathbf{s})}\cdot\mathbf{1}-\tfrac{1}{T}\cdot\mathbf{y}\right|\right|_2^2\right\}.} \tag{6}$$

We now turn to the question of attempting to optimise this objective.

4.3. Zoning objective: optimisation

To optimise Eq. (6), let us write it in constrained form

$$\min_{\mathbf{s}}\left|\left|\frac{1}{||\mathbf{s}||_0}\mathbf{A}^{(\mathbf{s})}\cdot\mathbf{1}-\frac{1}{T}\cdot\mathbf{y}\right|\right|_2^2 : ||\mathbf{s}||_0\leqslant K,$$

where K is some user-specified constant for the maximal number of nodes to be selected in the final zoning. This can be seen as an explicit means of trading off between the two desiderata.

As before, this objective is highly non-convex, and not amenable to standard gradient based optimisation. Consequently, we consider a simple greedy algorithm to approximately optimise the constrained objective. The algorithm proceeds as follows: at every iteration, we seek to add a *pair* of nodes to $\mathbf{s}$, such that there is the greatest overall reduction in the objective. To see how this can be done, note that the objective can be written

$$\sum_{e\in\mathcal{L}}\left(\frac{1}{T}\cdot\mathbf{y}_e-\frac{1}{||\mathbf{s}||_0}\cdot\sum_{v,v'}\mathbf{A}^{(\text{full})}_{e,(v,v')}\cdot\mathbf{s}_v\cdot\mathbf{s}_{v'}\right)^2$$

by recalling that Assumption B on the assignment map lets us work with the map constructed from a per-node zoning. Observe now that we can further simplify this to

$$\sum_{e\in\mathcal{L}}\left(\frac{1}{T}\cdot\mathbf{y}_e-\frac{1}{||\mathbf{s}||_0}\cdot\mathbf{s}^T\mathbf{A}^{(\text{full})}\mathbf{s}\right)^2.$$

Given some current solution $\mathbf{s}\in\{0,1\}^{|\mathcal{N}|}$, we are interested in solving

$$\min_{v,v'}\mathbf{C}_{vv'},$$

where

$$\mathbf{C}_{vv'}=\sum_{e\in\mathcal{L}}\left(\frac{1}{T}\cdot\mathbf{y}_e-\frac{1}{||\mathbf{s}+\mathbf{e}_v+\mathbf{e}_{v'}||_0}\cdot(\mathbf{s}+\mathbf{e}_v+\mathbf{e}_{v'})^T\mathbf{A}^{(\text{full})}(\mathbf{s}+\mathbf{e}_v+\mathbf{e}_{v'})\right)^2 \tag{7}$$

where $\mathbf{e}_v$ is an indicator vector with a 1 at the vth position, and zeros elsewhere. This problem can be solved by simply computing the entire $|\mathcal{N}|\times|\mathcal{N}|$ matrix $\mathbf{C}$. The entries of this matrix can easily be computed in $O(||\mathbf{s}||_0^2)=O(K^2)$ time. Thus, we can repeat this procedure as long as $||\mathbf{s}||_0\leqslant K$, so as to obtain a feasible zoning which greedily minimises the flow prediction error.

4.4. Zoning algorithm: summary

Our zoning algorithm can be summarised as follows.

1. Pick constants $K \in \mathbb{N}_+$ and $T \in \mathbb{N}_+$.
2. Compute $\mathbf{A}^{(\text{full})}$ by equilibrium assignment on a per-node zoning, using a uniform OD matrix with T trips.
3. Initialise $\mathbf{s} = \mathbf{0}^{|\mathcal{N}|}$.
4. Compute $\mathbf{C}$ via Eq. (7), and find
$$(v^*, v'^*) = \operatorname{argmin}_{v,v'} \mathbf{C}_{vv'}.$$
5. Let $\mathbf{s} \leftarrow \mathbf{s} + \mathbf{e}_{v*} + \mathbf{e}_{v'*}$.
6. Repeat steps (4) – (5) until $||\mathbf{s}||_0 \geqslant K - 1$.

Clearly, the choice of K plays an important role in determining the success of this algorithm. How do we determine the appropriate level of granularity? One way is to choose K based on the overall error in the equilibrium flow prediction, i.e. to choose the smallest K that results in an error of less than some fixed threshold: we expect larger K to result in lower error, but with K too large, we face problems with reliable OD estimation and interpretability. (The same procedure can also be used to determine a suitable value for the number of trips, T.)

5. Sparse OD estimation via ℓ_1 regularisation

Equipped with a means of computing fine-grained traffic analysis zones, we now turn to the problem of estimating the OD matrix $\mathbf{x}$. Recall that our interest is in mitigating the ill-posedness of the estimation problem with a generic regulariser, that acts in place of a prior OD matrix derived from historical data. Inspired by previous work (Zhang et al., 2005; Chawla et al., 2012; Mardani and Giannakis, 2013; Sanandaji and Varaiya, 2014), we shall explore sparsity of the OD matrix as our generic regulariser.

5.1. Sparsity as a generic regulariser

Without a prior OD matrix, to mitigate ill-posedness we must make some assumptions. The reasonableness of these assumptions can be argued and assessed intuitively, but ultimately must be demonstrated to result in good empirical performance.

We propose a generic OD regulariser that satisfies both these desiderata. Our basic assumption is that for all but the coarsest of zonings, the underlying OD matrix should be *sparse*: there should be exactly zero flow between most OD pairs. Our intuition is that the traffic observed in the network should largely be the result of the travel between a small subset of the $|\mathcal{Z}|^2$ candidate pairs, which e.g. reflects that during the morning rush hour, we expect there to be a few popular destinations (corresponding to office locations, parking lots, and so on), with most other zones seeing minimal in-flow. We may similarly expect most origins of flow in a business district to come from the boundaries of the network (corresponding to suburbs and residential areas).

In later sections, we shall assess the viability of the sparsity assumption. Before doing so, we must address the question of how we can achieve sparsity in the OD matrix. Following the GLS objective (Eq. (4)), we will consider the *sparse GLS objective*

$$\boxed{\min_{\mathbf{x} \succeq \mathbf{0}} (\mathbf{A}\mathbf{x} - \mathbf{y})^T \mathbf{W}^{-1} (\mathbf{A}\mathbf{x} - \mathbf{y}) + \lambda \cdot ||\mathbf{x}||_1,} \tag{8}$$

where $||\cdot||_1$ denotes the ℓ_1 norm of a (vectorised) matrix,

$$||\mathbf{x}||_1 = \sum_{z,z'=1}^{|\mathcal{Z}|} |\mathbf{X}_{zz'}|.$$

Here, $\mathbf{W}$ is a diagonal matrix whose entries are of the form

$$(\forall e \in \mathcal{L}) \mathbf{W}_{ee} = \mathbf{y}_e^{\beta} \tag{9}$$

for some $\beta \in \mathbb{R}_+$. This represents non-isotropic noise in the observed link flows, with higher link flows (corresponding to more heavily used roads) subject to higher errors.[8] When $\beta = 0$, this reduces to ordinary least squares. When $\beta = 1$, this is seen to mimic a Poisson model of the likelihood $\mathbb{P}(\mathbf{y}|\mathbf{x};\mathbf{A})$, which assumes the mean and variance are identical.

The term $||\mathbf{x}||_1$ encodes the belief that the true OD matrix is sparse. To see why this term induces sparsity, one can interpret it as a convex relaxation to the ℓ_0 "norm",

$$||\mathbf{x}||_0 = \sum_{z,z'=1}^{|\mathcal{Z}|} \mathbf{1}(\mathbf{X}_{zz'} \neq 0),$$

which is exactly the number of non zeros in $\mathbf{x}$. We shall refer to the term $\lambda \cdot ||\mathbf{x}||_1$ as an ℓ_1 *regulariser*. The use of ℓ_1 regularisation to discover sparse solutions has seen wide use in compressed sensing (Donoho, 2006) and in applications of the Lasso algorithm (Tibshirani, 1996). Theoretical results in these fields suggest that, under some assumptions, one can *exactly* recover a sparse solution by performing ℓ_1 regularisation.

The above objective operates in the scenario where there is assumed no prior OD matrix. However, it is equally applicable when there is such a matrix: we simply optimise

$$\min_{\mathbf{x} \succeq \mathbf{0}} (\mathbf{A}\mathbf{x} - \mathbf{y})^T \mathbf{W}^{-1} (\mathbf{A}\mathbf{x} - \mathbf{y}) + \lambda_1 \cdot ||\mathbf{x}||_1 + \lambda_2 \cdot ||\mathbf{x} - \mathbf{x}^{(\text{old})}||_2^2. \tag{10}$$

For $\lambda_1 > 0, \lambda_2 > 0$, the regulariser can be seen as a variant of the elastic net (Zou and Hastie, 2005). For the case where $\lambda_1 = 0$, this exactly matches the classical GLS objective, but for the explicit nonnegativity constraint. This constraint is intuitive, as negative OD flows do not have any easy interpretation. Nonnegativity is typically not considered in GLS because it disallows the simple derivation of a closed-form update rule (although simple updates are still possible (Bell, 1991)). However, as we shall see, there are several reasons to enforce this constraint in the objective. For future reference, we shall term the objective in Eq. (8) with $\lambda = 0$ be the "NN-GLS" objective, where "NN" denotes nonnegativity.

The idea of using ℓ_1 regularisation for problems involving OD estimation is not new. It has been proposed previously in at least Zhang et al. (2005),Chawla et al. (2012), Mardani and Giannakis (2013),Sanandaji and Varaiya (2014), albeit with slightly different contexts and motivations: the former three works are concerned with robustness to anomalies; the latter is concerned with sparsity in path flows, and considers a slightly different formulation that we shall discuss in Section 5.3. However, these works do not explicitly consider the impact of the constraint that the OD matrix elements must be nonnegative. While this constraint seems innocuous, as we shall see, it has important implications for the ill-posedness of the underlying system.

5.2. Optimisation of the sparse GLS objective

An apparent concern with the use of the ℓ_1 regulariser is that it is not differentiable at zero. This poses problems for optimisation using a gradient-following technique. However, observe that the non-negativity of $\mathbf{x}$ allows us to simplify the regulariser to yield

$$\min_{\mathbf{x} \succeq \mathbf{0}} (\mathbf{A}\mathbf{x} - \mathbf{y})^T \mathbf{W}^{-1} (\mathbf{A}\mathbf{x} - \mathbf{y}) + \lambda \cdot \mathbf{1}^T \mathbf{x}.$$

The objective now becomes differentiable everywhere, allowing for the easy application of gradient-based methods. Further, the nonnegativity constraint is relatively straightforward to enforce. We experimented with the LBFGS-B optimiser (Zhu et al., 1997), which performs quasi-Newton minimisation while respecting the constraint $\{\mathbf{x} \succeq 0\}$. Other approaches are of course possible, including the use of the general purpose quadratic programming solvers, or more general convex optimisation solvers such as the CVX toolbox (Grant and Boyd, 2014, 2008).

5.3. A sparse approximation viewpoint

There is another way to formulate the sparse GLS objective. By Lagrange duality, there exists some k_λ such that Eq. (8) is equivalent to

$$\min_{\mathbf{x} \succeq \mathbf{0}} (\mathbf{A}\mathbf{x} - \mathbf{y})^T \mathbf{W}^{-1} (\mathbf{A}\mathbf{x} - \mathbf{y}) : ||\mathbf{x}||_1 \leqslant k_\lambda. \tag{11}$$

We can interpret the resulting problem as follows: our goal is to minimise the (weighted) flow prediction error, subject to the constraint that the learned OD flows have small ℓ_1 norm (which induces them to further be sparse). We expect that this objective will select a sparse OD matrix that explains the link flows well.

[8] From a Bayesian perspective, this choice of $\mathbf{W}$ is not justified. The correct Bayesian choice of $\mathbf{W}$ follows directly from a well-defined prior distribution on $\mathbf{x}$; Cao et al. (2000) for example establish that under a Gaussian prior on $\mathbf{x}$ with covariance Σ, one has $\mathbf{W} = \mathbf{A}\Sigma\mathbf{A}^T$. By then setting Σ so as to mimic a Poisson model, however, the objective becomes non-convex in $\mathbf{x}$. Our choice of $\mathbf{W}$ can be seen as a heuristic that retains convexity of the objective, while attempting to partially account for noise in the flows. With a different choice of $\mathbf{W}$, the only change in all subsequent discussion is the details of the optimisation procedure.

Observe, however, that there is no guarantee in general that we will achieve a solution that has the *same* error in flow predictions as the solution when $\lambda = 0$. Intuitively, agreement with the $\lambda = 0$ case is desirable, since ill-posedness only means that the *spread* of flow across pairs of zones is incorrect, rather than the actual predicted flows. We can nonetheless arrive at a solution that achieves both sparsity and the same error as the reference solution when $\lambda = 0$. Suppose that

$$\mathbf{x}^{(\mathrm{NN})} \in \operatorname*{argmin}_{\mathbf{x} \succeq \mathbf{0}} (\mathbf{A}\mathbf{x} - \mathbf{y})^T \mathbf{W}^{-1} (\mathbf{A}\mathbf{x} - \mathbf{y})$$

represents a solution to the basic NN-GLS equation with no prior OD information. If the linear system is ill-posed, then there may be infinitely many possible $\mathbf{x}^{(\mathrm{NN})}$; however, for our purposes it suffices to just pick one. We shall assume that we are satisfied with the predicted flows from $\mathbf{x}^{(\mathrm{NN})}$, are unsure of the particular assignment of flow amongst the pairs of zones being accurate. To rectify the latter, we consider the objective

$$\boxed{\min_{\mathbf{x} \succeq \mathbf{0}} ||\mathbf{x}||_1 : \mathbf{A}\mathbf{x} = \mathbf{A}\mathbf{x}^{(\mathrm{NN})}.} \tag{12}$$

The constraint ensures that the predicted flows from our final solution agree exactly with that produced by $\mathbf{x}^{(\mathrm{NN})}$. However, the objective ensures that the OD matrix is sparse. This procedure can be interpreted as finding a sparse approximation to $\mathbf{x}^{(\mathrm{NN})}$ that retains the predicted flows of the latter. This objective can be seen as an instance of the basis pursuit principle (Chen et al., 2001), and may be solved efficiently via linear programming (Cheman, 2006). In the case where $\mathbf{A}\mathbf{x}^{(\mathrm{NN})} = \mathbf{y}$, Eq. (12) is exactly the proposal of Sanandaji and Varaiya (2014).

An advantage of the latter formulation over Eq. (11) is that there is no need to tune any parameters to control sparsity: this is done automatically once we have set the reference flows from $\mathbf{x}^{(\mathrm{NN})}$. However, a disadvantage is that we must perform *two* optimisations: one to find the reference solution $\mathbf{x}^{(\mathrm{NN})}$, and another to find the sparse approximation to this vector.

5.4. Nonnegativity, sparsity, and ill-posedness

In the above, we have made explicit that the objective must be optimised over OD matrices that are nonnegative. It is tempting to ignore this constraint during optimisation, and instead perform a *post hoc* clipping of OD flows. As the resulting optimisation is considerably simpler, for practical reasons, one may wish to ignore the constraint; but are there theoretical reasons it is important?

In fact, the non negativity constraint on $\mathbf{x}$ cannot be ignored in assessing the uniqueness of the linear system in Eq. (1). Recent work has shown that, while the unconstrained linear system may be strongly under-determined, the addition of a nonnegativity constraint may result in the system possessing a unique solution (Wang and Tang, 2009; Wang et al., 2011; Slawski and Hein, 2012; Meinshausen, 2013). The key finding is that, when there exists a sparse, nonnegative solution to the system, it may be the *only nonnegative solution*. In a sense, the non negativity constraint *by itself* may induce sparse solutions.

One can view this theoretical finding as saying that nonnegativity is itself a "generic regulariser" that helps mitigate ill-posedness. Further, this generic regulariser by itself induces sparsity if the ground truth OD is sufficiently sparse. With this view, one may ask whether there is reason to consider ℓ_1 regularisation. We can justify ℓ_1 regularisation as practised in the sparse approximation objective as only altering the solution to the nonnegative system *if it is ill-posed*. Formally, suppose $\mathbf{x}^{(\mathrm{NN})}$ attains minimal error in predicted flow. Denote the set of OD matrices with minimal error in predicted flow by

$$\mathcal{S} = \{\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{A}\mathbf{x}^{(\mathrm{NN})}, \mathbf{x} \succeq \mathbf{0}\}.$$

Observe that Eq. (12) can be seen as selecting an element from this set. Formally, it returns an element from the set of sparse approximations to $\mathbf{x}^{(\mathrm{NN})}$, denoted

$$\mathcal{S}_{\mathrm{sparse}} = \operatorname*{argmin}_{\mathbf{x} \in \mathcal{S}} ||\mathbf{x}||_1 \subseteq \mathcal{S}.$$

If $\mathcal{S} = \{\mathbf{x}^{(\mathrm{NN})}\}$, then trivially we also have that $\mathcal{S}_{\mathrm{sparse}} = \{\mathbf{x}^{(\mathrm{NN})}\}$. So, when we are in a regime where nonnegativity by itself induces a sparse, unique solution, additionally performing ℓ_1 regularisation does not alter the solution in any way. However, under regimes where nonnegativity alone does not induce a unique solution, ℓ_1 regularisation will help select a sparse OD matrix.

There is a further subtlety in the regime where $\mathcal{S} \supset \{\mathbf{x}^{(\mathrm{NN})}\}$, so that there are infinitely many possible OD matrices with minimal flow. Since $||\mathbf{x}||_1$ is not strictly convex, the set $\mathcal{S}_{\mathrm{sparse}}$ may *itself* contain infinitely many elements. Standard theoretical guarantees for sparse approximation only say that the set is a singleton under some assumptions. When these assumptions do not hold, we should not expect Eq. (12) to automatically select a unique optimal element. Indeed, in such cases, the precise element that is selected depends on the details of the underlying optimisation algorithm.

In practice, then, one can adopt the following simple strategy. First, compute $\mathbf{x}^{(\mathrm{NN})}$ with a standard nonnegative least squares solver. Next, compute $\mathbf{x}$ by minimising Eq. (12). Then, compare $||\mathbf{x}||_1$ and $||\mathbf{x}^{(\mathrm{NN})}||_1$. If $||\mathbf{x}||_1 < ||\mathbf{x}^{(\mathrm{NN})}||_1$, then we just return $\mathbf{x}$. However, if $||\mathbf{x}||_1 = ||\mathbf{x}^{(\mathrm{NN})}||_1$, then we also compare explicitly the number of nonzero elements in each of the matrices, and return the one that is more sparse.

5.5. Relation to other generic regularisers

There is an interesting contrast between the objective in Eq. (12), and that of the maximum entropy model in Eq. (5). Instantiating the latter for a uniform prior OD matrix, the objective reduces to

$$\min_{\mathbf{x}} -H(\mathbf{x}) : \mathbf{A}\mathbf{x} = \mathbf{y}, \tag{13}$$

where $H(\cdot)$ denotes the Shannon entropy of an unnormalised distribution. This can be optimised using Bregman's balancing method (Lamond and Stewart, 1981; Bell and Iida, 1997, p. 151). While Eq. (12) chooses the sparsest solution out of a set of candidate OD matrices, the maximum entropy model selects the solution with highest entropy. But entropy is maximised precisely when the solution is *not* sparse; in fact, it encourages the solution to be as close to uniform as possible. We expect that this is not as useful as a generic regulariser. (Given a more meaningful prior OD matrix, of course, the relative entropy selection rule is sensible.)

Other variants of the maximum entropy model are possible. For example, Zhang et al. (2003b) consider a problem of the form

$$\min_{\mathbf{X}} \mathrm{KL}(\mathbf{X} \| \mathbf{X}_r \mathbf{X}_c^T) : \mathbf{A}\mathbf{x} = \mathbf{y},$$

where $\mathbf{X}_r$ denotes the row-marginal of the OD matrix, and $\mathbf{X}_c$ the column-marginal. This solution encourages the selection of a *low rank* OD matrix, as without constraints, the optimal solution will be a rank one matrix of the form $\mathbf{X} = \mathbf{p}\mathbf{q}^T$. The low-rank assumption is also plausible as a generic regulariser, and does afford interpretability. However, its optimisation is more involved than that of sparse regularisation, which simply requires solving a linear program.

5.6. Relation to other work

Bierlaire (2002) proposed a surrogate measure of the degree of ill-posedness of the system, which has an interesting connection to our approach. Suppose we have assignment map $\mathbf{A}$, and a candidate OD vector, say $\mathbf{x}^{(\mathrm{NN})}$, the solution to

$$\mathbf{x}^{(\mathrm{NN})} \in \operatorname*{argmin}_{\mathbf{x} \succeq \mathbf{0}} ||\mathbf{A}\mathbf{x} - \mathbf{y}||_2^2,$$

which does not employ any regularisation towards a prior OD matrix. If the system is ill-posed, there will be many possible choices for $\mathbf{x}^{(\mathrm{NN})}$. We can quantify the degree of ill-posedness via the total demand scale,

$$\mathrm{TDS}(\mathbf{A}, \mathbf{x}^{(\mathrm{NN})}) = \phi_{\max}(\mathbf{A}, \mathbf{x}^{(\mathrm{NN})}) - \phi_{\min}(\mathbf{A}, \mathbf{x}^{(\mathrm{NN})}),$$

where

$$\phi_{\max}(\mathbf{A}, \mathbf{x}^{(\mathrm{NN})}) = \max_{\mathbf{x}} \mathbf{1}^T\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{A}\mathbf{x}^{(\mathrm{NN})}, \quad \mathbf{x} \succeq 0$$
$$\phi_{\min}(\mathbf{A}, \mathbf{x}^{(\mathrm{NN})}) = \min_{\mathbf{x}} \mathbf{1}^T\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{A}\mathbf{x}^{(\mathrm{NN})}, \quad \mathbf{x} \succeq 0.$$

If $\mathrm{TDS}(\mathbf{A}, \mathbf{x}^{(\mathrm{NN})}) > 0$, then we can conclude that the system is ill-posed, as there are multiple feasible solutions. If $\mathrm{TDS}(\mathbf{A}, \mathbf{x}^{(\mathrm{NN})}) = 0$, then we cannot conclude whether or not the system is ill-posed, but we can at least guarantee that the set of optimal OD flows all have the same *aggregate* demand, and only potentially vary in how they spread this across the various pairs of zones.

Observe that we can write

$$\phi_{\min}(\mathbf{A}, \mathbf{x}^{(\mathrm{NN})}) = \min_{\mathbf{x}} ||\mathbf{x}||_1 : \mathbf{A}\mathbf{x} = \mathbf{A}\mathbf{x}^{(\mathrm{NN})}, \mathbf{x} \succeq 0.$$

Therefore, we see that $\phi_{\min}$ involves the same minimisation as the sparse approximation version of our algorithm (Eq. (12)). However, the motivation for the total demand scale is very different from that of our approach.

6. Hold-out evaluation of OD estimates

Treating OD estimation for fixed $\mathbf{A}$ as a regression problem, a natural strategy to evaluate an OD matrix is based on prediction on flows that are *held-out* during the estimation procedure. This follows the general procedure for evaluating any model estimated from data (Hastie et al., 2009, Chapter 7). The idea is as follows:

1. partition the set of links $\mathcal{L}$ into two sets, $\mathcal{L}_1$ and $\mathcal{L}_2$,
2. estimate the OD matrix based *only on link flows from* $\mathcal{L}_1$,
3. evaluate the predictive performance *only on link flows from* $\mathcal{L}_2$.

This process may be repeated many times. The average of the performance computed in step (3) may be taken as an estimate of the predictive power of the estimated OD matrix $\hat{\mathbf{x}}$.

Formally, step (2) is equivalent to solving our objective in Eq. (8) with the vector $\tilde{\mathbf{y}}$, defined as

$$\tilde{\mathbf{y}}_e = \begin{cases} \mathbf{y}_e & \text{if } e \in \mathcal{L}_1 \\ 0 & \text{else,} \end{cases}$$

and the matrix $\tilde{\mathbf{A}}$, defined as

$$\tilde{\mathbf{A}}_{ep} = \begin{cases} \mathbf{A}_{ep} & \text{if } e \in \mathcal{L}_1 \\ 0 & \text{else.} \end{cases}$$

Put plainly, we simply ignore the links in $\mathcal{L}_2$ when estimating the OD matrix $\mathbf{x}$. Once we have an estimate $\hat{\mathbf{x}}$, we may of course compute the predicted flows on links in $\mathcal{L}_2$ via the full assignment map $\mathbf{A}$. Step (3) then requires that we summarise the fidelity of the predictions on these links alone.

A few comments are in order. First, the partitioning of the links into the two sets is independent of the zoning procedure. A simple strategy is to just perform a random partition. (Network-dependent partitions, e.g. ones that ensure path connectivity is not overly affected by excluding links, are of interest, but left for future work.) Second, it is essential that step (3) operate only on links in $\mathcal{L}_2$: as the links in $\mathcal{L}_1$ were used to estimate $\hat{\mathbf{x}}$, the performance on these links is a highly biased estimate of the true predictive performance of $\hat{\mathbf{x}}$. Third, the splitting is only performed for the OD estimation. It does *not* mean that the links are physically removed, and in particular, they are still a part of the assignment procedure employed to find the assignment map $\mathbf{A}$. The value of this split is that we can then evaluate the predictive performance of $\mathbf{X}$ on the second, or *held-out*, set of links.

A subtlety is that because of the risk of ill-posedness, it may well be that there are multiple OD matrices that have identical performance on held-out links as well. However, as we encourage agreement with the prior OD matrix, $\mathbf{x}^{\text{old}}$, and further regularise the matrix to possess sparsity (which is an application of our domain knowledge), we believe that good held-out performance of such a regularised estimate is indicative of reliable estimation. In our experiments, we shall further study the interpretability of the matrix, as well as perform a case-study on real link flows.

7. Evaluation of methods

We now present a range of results confirming the efficacy of our approach.

7.1. Experimental aims

In brief, the aims of our experimental study are:

- To assess the quality of the zoning induced by our approach described in Section 4. In particular, we assess how successful this approach is in attaining our goal of explaining the observed link flow, and how it compares to a manually defined zoning based on domain knowledge.
- To determine the value of imposing sparsity as a generic regulariser for OD estimates, and further, to assess the extent to which this can be achieved by enforcing non-negativity constraints during optimisation, and by explicitly considering ℓ_1 regularisation.
- To determine the value of assessing OD estimates using held-out flows, by for example demonstrating that it rejects estimates that grossly overfit to the observed link flows.

To answer the above, we shall largely operate with data from a real-world network, which we now describe.

7.2. Description of data

We conduct experiments on traffic counts obtained for an urban area during a two-week period in 2012. The data comprises observations for a network with 155 intersections and 310 road segments connecting them. The traffic counts are the aggregate of those observed during the period of 7 AM–10 AM.

Our motivation in studying this network, as per Section 3, is to understand what happens to this network when a subset of links are removed from the network. Recall that a major challenge with the data is the lack of a suitable prior OD matrix. Existing public survey data only provides a limited number of samples, which results in an insufficiently reliable prior matrix. In our experience, even simple smoothing models, such as the gravity model, do not offer significant improvement in the performance of these estimates.

The first step in analysing the network is to determine an appropriate zoning with which to analyse OD flows. We now describe the results of our automated zoning algorithm on this network.

7.3. Automated zoning: selecting the number of zones

Recall that the automated zoning algorithm requires specification of the number of zones K, and number of trips T. Our first question is thus how to go about selecting these parameters. We shall use a simple strategy: after performing k steps of the greedy algorithm, we compute an equilibrium assignment using a uniform OD matrix over these zones, with a set number of trips. We can then compute the error in the predicted flows under this assignment, compared against the ground truth flows. These errors are used to drive the selection of a suitable number of zones.

Fig. 2 shows the results of this procedure for $T = 50,000$ trips. (A similar trend is evident for varying numbers of trips.) We see that around $||\mathbf{s}||_0 = 21$ zones, we attain a local minimum in the errors. While the error does not increase significantly beyond this point, we favour selecting a fewer number of zones. Thus, we select $k = 21$ zones for our automated algorithm, and $T = 50,000$ trips. (We find a similar result when additionally performing OD calibration using NN-GLS, whose errors are also shown in Fig. 2.)

To illustrate the capabilities of our automated zoning algorithm, we contrast its results to a manual zoning provided by a domain expert. It comprises a total of 22 zones, which is essentially the same number as that discovered by the automated zoning. The manual design of these zones is guided by geographic contiguity, and domain knowledge as to certain regions of interest in the network (representing major office complexes, parking lots, shopping centres, *et cetera*).

The manual zoning has appeal because its results are largely intuitive. However, it is demonstrably suboptimal. We shall illustrate this both qualitatively and quantitatively.

7.4. Automated versus manual zoning: a qualitative comparison

Fig. 3 visualises the locations of the nodes selected by the automated zoning procedure (shown in red, large dots), as well as the centroids of the manually defined zones (shown in blue, small dots). The two zonings appear reasonably similar, and both largely span the entire area of the graph. However, there are some interesting differences in the specific nodes chosen by the automated procedure. For example, consider the two nodes at the bottom left of the graph. These are included in the same zone as per the manual zoning, which is intuitive. But in fact, the link connecting these two nodes has one of the highest observed volumes in the entire network. The manual zoning is thus unable to account for such flow.

Also of interest is the fact that the automated zoning identifies several extremal nodes in the graph as suitable for selection. This is intuitive, since these nodes can capture flow that has origin or destination exogenous to the network. While the manual zoning also accommodates several boundary zones, the automated zoning operates at a finer level of granularity. For example, it selects two nodes along the "fork" at the top right hand side of the network. These accommodate two very different paths of travel for vehicles that enter the network from the east side.

The above is illustrative of the differences between the automated and manual zoning. We now attempt to quantify the differences between them.

7.5. Automated versus manual zoning: a quantitative comparison

Fig. 4 shows a scatterplot of the observed link flows versus those predicted by an equilibrium assignment on the manual zoning, using a uniform OD matrix and 50,000 trips. We see that the resulting flows miss certain high volume links completely. This means that these links are simply not considered for travel between any of the defined zones, i.e. they are deemed as largely being necessary for intra-zonal travel.[9] By contrast, the automated zoning assigns non-neglible flow to each of these links. This shows that, as per our objective, the automated zoning can account better for the flow on the network, although it uses fewer zones than the manual scheme.

One objection to the above is that it operates with a uniform OD matrix, and the resulting assignment map. Does the manual zoning fare better when one considers more realistic choices for these inputs? Fig. 5 illustrates the flow predictions after performing 10 iterations of bilevel programming, where we alternately calibrate the OD matrix by solving the GLS equation with nonnegativity constraints (Eq. (8) with $\lambda = 0$), and then perform a deterministic user equilibrium assignment. It is evident that the results for both zonings improve considerably, which is unsurprising. However, we again see that the manual zoning continues to underperform when compared to the automated zoning. The former is seen to dramatically over predict the flow on certain links, while the predictions of the latter are very close to the observed link flows.

Another objection to the above is that the manual zoning can of course be updated in an iterative process, after studying the link flows that are not well modelled. While we do not dispute this, we would suggest that such an iterative scheme is implicitly performing an optimisation, with the objective of modelling the equilibrium flows accurately. The automated zoning directly seeks to find a zoning that minimises this objective. It can minimally be considered a tool that can serve as a foundation for manual definition of zoning, also incorporating e.g. geographical contiguity and alignment with suburban boundaries.

[9] By increasing the number of trips to a very large constant, these links will of course attain nonzero flow. But increasing the number of trips in this manner results in significant over prediction of flow on other links.

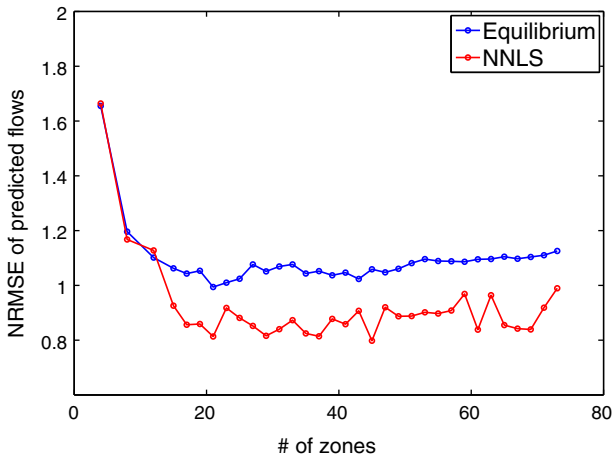


Fig. 2. Error in the equilibrium and NN-GLS flows output by the automated zoning procedure for varying numbers of zones.

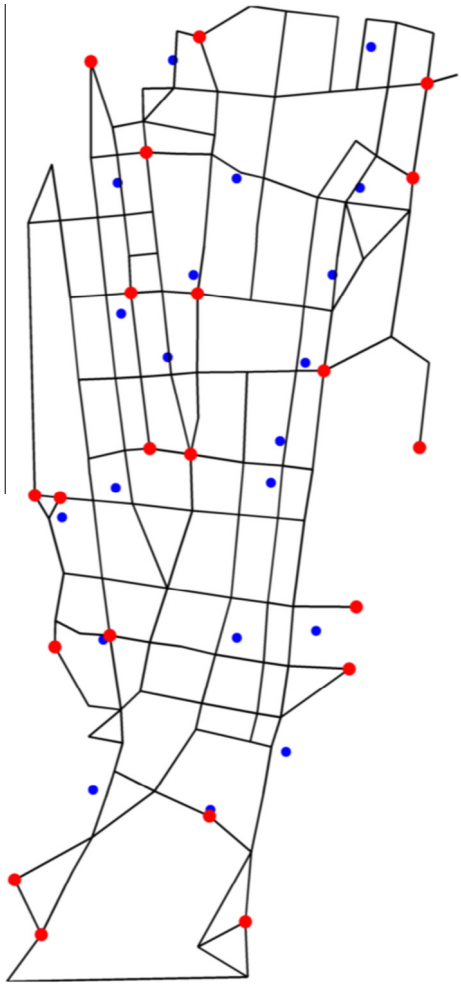


Fig. 3. Visualisation of results automated zoning procedure, and contrast with manual zoning. The red (large) dots denote nodes assigned to a separate zone as a result of the automated zoning. The blue (small) dots denote centroids of the zones as defined by the manual zoning. (The node layout corresponds to geographic location. Precise coordinates are omitted for data confidentiality reasons.) (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

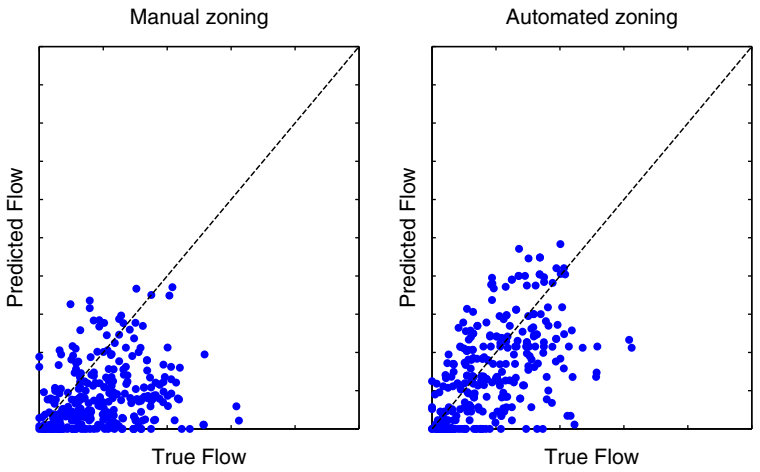


Fig. 4. Scatter plot of predicted versus true link flows, uniform OD and resulting assignment. The dashed black line denotes an optimal scatter plot. (The axes' scales are unlabelled for data confidentiality reasons.)

7.6. Sensitivity of automated zoning to the uniform OD assumption

We perform one final analysis of the automated zoning scheme, by assessing the impact of our Assumption C in Section 4.2. Recall that this assumption posited that the per-site OD matrix is uniform. In practice, we expect that this is far from true; thus, it is prudent to study how this affects the zoning discovered by our algorithm. In particular, does it lead to the discovery of many more zones than are actually needed to capture flow on the network?

To study this question, we performed the following experiment:

- We pick a given sparsity level $s \in [0, 1]$.
- We select a subset of nodes in the real-world network as the previous sections, based on the sparsity level.
- We generated a synthetic per-site OD matrix $\mathbf{x}^{(\text{synth})}$, where flows are only between the nodes selected in the previous step.
- We treat $\mathbf{x}^{(\text{synth})}$ as the ground truth OD matrix for the network, and generate equilibrium flows using this.
- We run the automated zoning algorithm on the resulting network and flows, and compare the discovered zones to the active nodes in $\mathbf{x}^{(\text{synth})}$.

In detail, for a given sparsity level s, we select a random subset I of nodes to be active for travel, where $|I| = s \cdot |\mathcal{N}|$; we call these the "active nodes". We then compute the set of feasible OD pairs P, by assigning each element in I as being a "source only" with probability $\frac{1}{3}$, a "destination only" with probability $\frac{1}{3}$, and both a source and destination with probability $\frac{1}{3}$; we then consider all possible combinations of sources and destinations.

Once the candidate OD pairs are selected, we select the proportion of flow by drawing an observation from a $|P|$-dimensional symmetric Dirichlet distribution[10] with parameter δ. We then set the total number of trips to be T. (Both δ and T will be specified shortly.) For a given synthetic OD matrix $\mathbf{x}^{(\text{synth})}$, we perform equilibrium assignment using this network, to derive the optimal assignment map $\mathbf{A}^{(\text{synth})}$. We treat the resulting equilibrium flows $\mathbf{y}^{(\text{synth})}$ to be the observed link flows in the network.

Then, we apply the automated zoning algorithm to this network. This algorithm returns a number of nodes that are deemed suitable to act as zones. We compare these returned nodes against the ground truth nodes in I, to assess whether or not we are able to correctly identify the true origins and destinations for travel. We do this for varying caps K on the number of desired zones: for each such K, we report the percentage of active nodes that are present in the automated zones. Ideally, when the number of automated zones K equals the number of active nodes, we would like the percentage to be 100%; smaller percentages indicate that the automated zoning incorrectly discovers some "wrong" nodes that are not actually the source or destination of travel.

We experiment with sparsity levels $s = 5\%$ and 25%, and for 100 random selections of active nodes. Different choices of the sparsity parameter s result in the selection of various numbers of active nodes; for each such selection, and each number of automated zoning K, we plot the percentage of active nodes that are correctly identified. For clarity, a vertical line marks

[10] Recall that a $|P|$-dimensional symmetric Dirichlet distribution with parameter 1 is simply a uniform distribution over all $|P|$-dimensional probability distributions. For parameters smaller than 1, we encourage concentration of the probability distribution on a few entries, which is what we expect of an OD matrix.

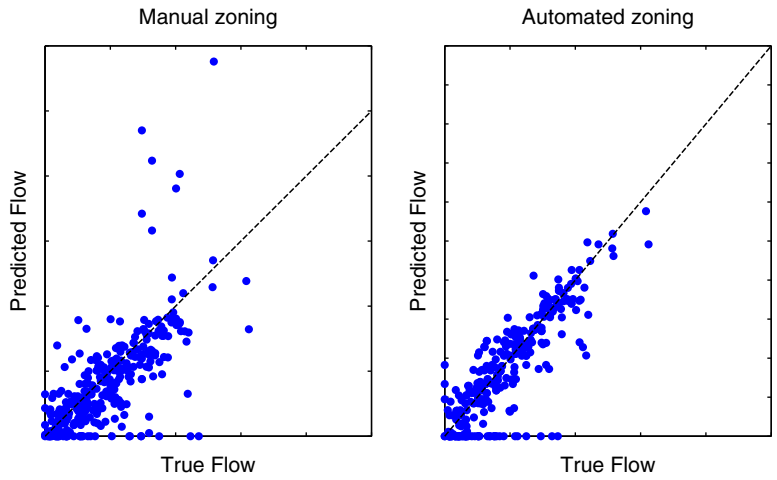


Fig. 5. Scatter plot of predicted versus true link flows, bilevel OD and assignment. The dashed black line denotes an optimal scatter plot. (The axes' scales are unlabelled for data confidentiality reasons.)

the point where the number of automated zones equals the number of active nodes; ideally, this would correspond to discovering 100% of the active nodes.

We present results in Fig. 6 for the setting $\delta = 0.75$ and $T = 800$. From the results, it is evident that even for a per-site OD matrix with a few number of active nodes, the zoning algorithm is able to recover close to 80% of these nodes when K equals the number of active nodes. Unsurprisingly, with a larger number of active nodes, this percentage increases.

The above results are encouraging, and indicate that even when the uniform OD assumption is violated, the automated zoning can perform reasonably. Of course, unsurprisingly, there are regimes where we can more severely degrade the algorithm's performance. We present results in Fig. 7 for the setting $\delta = 0.05$ and $T = 8000$. For this choice of δ, the per-site OD matrix is significantly more concentrated on a few entries. We see that for these parameter settings, significantly more automated zones are needed to correctly identify a large subset of active nodes. This suggests that a sparse, concentrated OD per-site matrix may cause the automated zoning algorithm to perform sub-optimally. In such cases, it may be prudent to investigate relaxations of Assumption C, as outlined in Section 4.2.

Having illustrated the benefits of our automated zoning scheme, we turn to the problem of estimation of the OD matrix. To illustrate the benefits of sparse regularisation, we first consider the problem of recovering a synthetically generated OD matrix.

7.7. The virtues of sparse regularisation: recovery of synthetic OD flows

As a preliminary demonstration of the virtues of the ℓ_1 penalty as a generic regulariser to mitigate ill-posedness of the OD estimation problem, we consider the problem of recovery of a synthetic OD matrix. As noted earlier, nonnegativity by itself may be able to recover sufficiently sparse OD matrices. To illustrate this, we consider a network of Sanandaji and Varaiya (2014), comprising 4 zones and 10 links, shown in Fig. 8. A fixed assignment is performed to construct an assignment map $\mathbf{A}$, such that only 3 OD pairs ($B \to C, B \to A, D \to A$) have nonzero flow, with a total of 14 potential paths employed. Sanandaji and Varaiya (2014) consider the estimation of a sparse path flow vector $\mathbf{z}^* \in \mathbb{R}^{14}_+$ with only 6 nonzero entries, namely,

$$\mathbf{z}^* = (0, 1000, 0, 0, 0, 0, 0, 500, 0, 0, 150, 0, 0, 450).$$

We assume that we have access to noise-free link flows resulting from $\mathbf{z}^*$, namely, $\mathbf{A}\mathbf{z}^*$. It is easy to check that[11]

$$\mathbf{z}^* = \underset{\mathbf{z} \succeq \mathbf{0}}{\operatorname{argmin}} \, ||\mathbf{A}\mathbf{z} - \mathbf{A}\mathbf{z}^*||_2^2,$$

where the right hand side has a unique minimiser. Therefore, by simply imposing nonnegativity, we can recover a sparse OD matrix. It follows then that minimising Eq. (12) will also recover this sparse matrix; but, in this instance, there appears little reason to use this more complicated objective.

The above suggests that in some regimes, nonnegativity alone suffices to recover a unique OD matrix. We now study instances where additional ℓ_1 regularisation may be useful. We use the real-world network as described above, and consider for simplicity the manual zoning of the network into 22 zones. We perform the following experiment, which is similar to that employed in Section 7.6, but at a coarser granularity:

[11] It MATLAB, the problem can be solved by `lsqnonneg`. The uniqueness of the optimal solution can be assessed by the total demand scale, which in this case is zero.

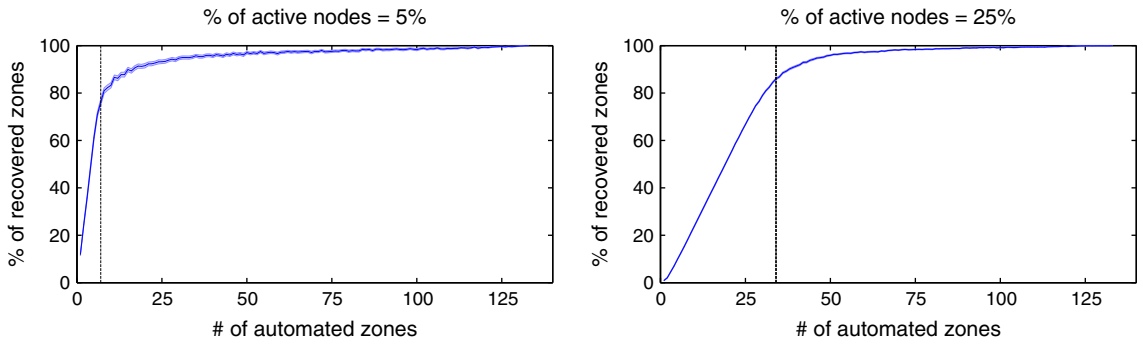


Fig. 6. Recovery of synthetic zones by automated zoning algorithm, $\delta = 0.75,\ T = 800$. Each panel shows the results for a varying number of synthetically chosen "active nodes", which serve as foci for travel. The black vertical line in each panel marks the point where the number of automated zones equals the number of active nodes; ideally, this corresponds to a 100% discovery rate. The shaded region in each plot represents an error band of one standard error across the 100 trials.

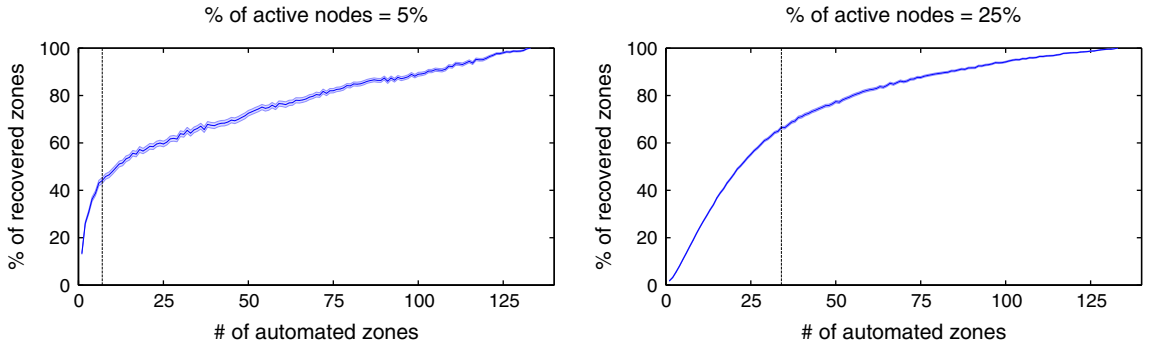


Fig. 7. Recovery of synthetic zones by automated zoning algorithm, $\delta = 0.05,\ T = 8000$. Each panel shows the results for a varying number of synthetically chosen "active nodes", which serve as foci for travel. The black vertical line in each panel marks the point where the number of automated zones equals the number of active nodes; ideally, this corresponds to a 100% discovery rate. The shaded region in each plot represents an error band of one standard error across the 100 trials.

- We pick a given sparsity level $s \in [0, 1]$ for the synthetic OD matrix.
- We generate a synthetic OD matrix $\mathbf{x}^{(\text{synth})}$, with sparsity level s, over the 22 zones.
- We treat $\mathbf{x}^{(\text{synth})}$ as the ground truth OD matrix for the network, and attempt to recover this using an OD estimation algorithm.

We repeat this procedure for varying draws of a synthetic OD matrix, sparsity levels, and OD estimation algorithms.

In detail, we generate a synthetic OD matrix $\mathbf{x}^{(\text{synth})}$ by first randomly selecting a subset I of possible OD pairs of appropriate size. For all pairs not in I, we set the flow to be zero. For all pairs in I, as per Section 7.6, we select the proportion of flow by drawing an observation from a $|I|$-dimensional symmetric Dirichlet distribution with parameter $\frac{3}{4}$. We then set the total number of trips to be 800 times the chosen sparsity level. (The number 800 was chosen heuristically so as to ensure that the final OD flows, which are typically real valued, are greater than 1.)

For a given synthetic OD matrix $\mathbf{x}^{(\text{synth})}$, we perform equilibrium assignment using this network, to derive the optimal assignment map $\mathbf{A}^{(\text{synth})}$. We treat the resulting equilibrium flows $\mathbf{y}^{(\text{synth})}$ to be the observed link flows in the network. Then, for various OD estimation algorithms, given as input the optimal assignment map and the equilibrium flows, we attempt to recover the ground truth OD. We report errors as measured by the root mean squared difference between the estimated and ground truth OD.

We compare the performance of three OD estimation algorithms:

1. generalised least squares as per Eq. (1), with post hoc thresholding of the estimates to ensure that they are nonnegative;
2. nonnegative least squares;
3. nonnegative least squares, followed by the sparse approximation problem in Eq. (12). As noted earlier, the sparse approximation problem may itself not have a unique solution. Thus, we only use its result when the ℓ_1 norm of the solution is smaller than that of the nonnegative least squares solution.

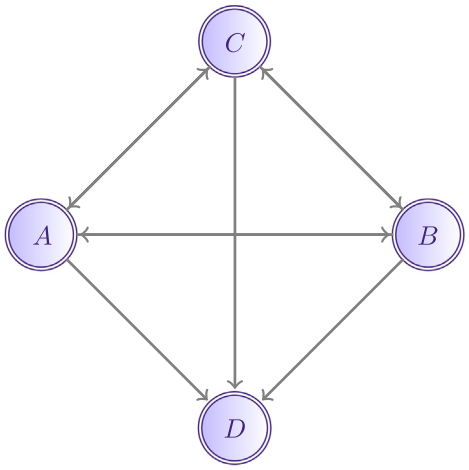


Fig. 8. Network considered in Sanandaji and Varaiya (2014).

Our hypotheses are: the first method should underperform, owing to the complete lack of regularisation of the solution; nonnegative least squares should offer good recovery as-is, owing to the implicit regularisation nonnegativity theoretically implies; and that the sparse approximation method should offer further advantages over nonnegative least squares.

We present results from 100 trials of synthetic OD generation, for sparsity levels $s \in \{0.01, 0.05, 0.10, \ldots, 0.50\}$, in Fig. 9. It is evident that the thresholding approach consistently under performs. This is unsurprising, but it is reassuring that the performance of this method is not unboundedly worse than the competitors, suggesting that it may be a reasonable heuristic if one has a tight computational budget.

Of interest is that for a range of sparsity regimes, the sparse approximation solution is able to attain lower average error than the nonnegative least squares solution. This illustrates the potential benefits of explicit sparsity regularisation. When the ground truth OD is very sparse, the two agree perfectly, which is in keeping with theoretical results on nonnegativity as a sparsity inducing regulariser. When the ground truth OD is very dense, the two similarly agree, which is intuitive because here we do not expect a sparse solution to exist in the first place.

The above is indicative of the virtues of sparse regularisation. To further study this point, we consider an alternate measure of performance in the setting where we do not know the ground truth OD matrix.

7.8. The virtues of sparse regularisation: held-out prediction results

We now report results on the quality of various OD estimation schemes. We vary three knobs, and report results for each ensuing combination:

- For the *Zoning*, we report results on both a manual zoning determined by a domain expert, and the automated zoning determined by our approach (Section 4).
- For the *Learner*, we either simply report the prior OD matrix (“None”), minimise the objective in Eq. (8) without a nonnegativity constraint (“GLS”), or minimise the objective in Eq. (8) with a nonnegativity constraint (“NN-GLS”). For GLS, the final solution is explicitly thresholded at 0.
- For the *Regulariser*, we test no regularisation ($\lambda_1 = \lambda_2 = 0$ in Eq. (10)), ℓ_2 regularisation to the prior OD matrix ($\lambda_1 = 0, \lambda_2 > 0$), ℓ_1 regularisation to induce sparsity ($\lambda_1 > 0, \lambda_2 = 0$), and a combination of ℓ_1 and ℓ_2 regularisation ($\lambda_1, \lambda_2 > 0$). Additionally, we evaluate the minimum sparsity (Eq. (12)) selection rule, over all OD matrices with the same predicted flow as the NN-GLS model. For brevity, we denote this learners as “BP”.

Following our discussion in Section 6, we report held-out link flow performance of the OD estimates derived from each method. As performance measures, we report the RMSE, MAE, and Spearman’s ρ of our predicted link flows $\hat{\mathbf{y}}$ against the true link flows $\mathbf{y}$. These are defined as

$$\mathrm{RMSE}(\mathbf{y}, \hat{\mathbf{y}}) = \sqrt{\frac{1}{|\mathcal{L}|} \sum_{e \in \mathcal{L}} (\mathbf{y}_e - \hat{\mathbf{y}}_e)^2}$$

$$\mathrm{MAE}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{|\mathcal{L}|} \sum_{e \in \mathcal{L}} |\mathbf{y}_e - \hat{\mathbf{y}}_e|$$

$$\rho(\mathbf{y}, \hat{\mathbf{y}}) = 1 - 6 \frac{\sum_{e \in \mathcal{L}} (\mathbf{r}_e - \hat{\mathbf{r}}_e)^2}{|\mathcal{L}|(|\mathcal{L}|^2 - 1)},$$

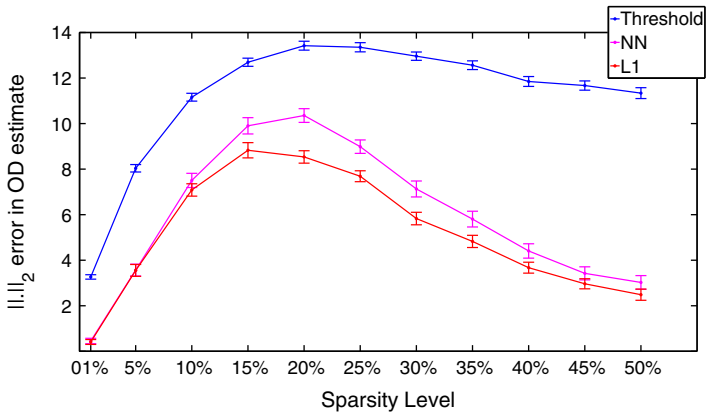


Fig. 9. Recovery of synthetic OD matrix by various algorithms. For each method, we plot the mean of the error estimate over 100 trials, as well as the standard error of this estimate.

where $\mathbf{r}, \hat{\mathbf{r}}$ represent the ranks of the links according to the flows $\mathbf{y}, \hat{\mathbf{y}}$. We normalise the RMSE and MAE metrics based on that of the trivial baselines $\mathbf{y}^{\text{mean}}, \mathbf{y}^{\text{median}}$, which predict the mean and median of the observed link flows respectively,[12] i.e. we report

$$\text{NRMSE}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{\text{RMSE}(\mathbf{y}, \hat{\mathbf{y}})}{\text{RMSE}(\mathbf{y}, \mathbf{y}^{\text{mean}})},$$

and similarly for the MAE. This scaling is the same for all methods, and thus does not change the ranking amongst them. However, it offers a clear point of reference, as any method must attain a normalised score of less than 1 to be considered useful.

Additionally, for each method, we consider the final OD matrix that is returned. We then report the NRMSE in the equilibrium flow predictions when loading this OD matrix onto the network. We report this as the "Equilibrium NRMSE" of the method, in contrast to the "H/O NRMSE" for the performance on the holdout set.

We tune the regularisation strengths where appropriate from $\lambda_1, \lambda_2 \in \{10^{-6}, 10^{-5}, \ldots, 10^{1}\}$. Similarly, we tune the exponent in the weight matrix $\mathbf{W}$ (Eq. (9)) from $\beta \in \{0, 0.5, 1, 1.5, 2\}$. This tuning is performed by creating a split *within* the set of links observed for training, and finding the hyperparameter settings that yield the best NRMSE.

For both constructing an initial assignment map, and for ℓ_2 regularisation towards a prior OD matrix, recall that we do not possess a reliable matrix from exogenous sources. Therefore, we chose to use a naïve *uniform* prior $\mathbf{x}^{(\text{uni})}$ as our prior OD matrix. All non-diagonal entries in this matrix are equal to $\frac{T}{|\mathcal{Z}| \cdot (|\mathcal{Z}|-1)}$, with T being the total number of trips in the network. The number T was fixed to be 50,000, as this was seen to yield the best performance for the automated zoning. Given this $\mathbf{x}^{(\text{uni})}$, we used equilibrium assignment to construct the assignment matrix used as input to all methods.[13]

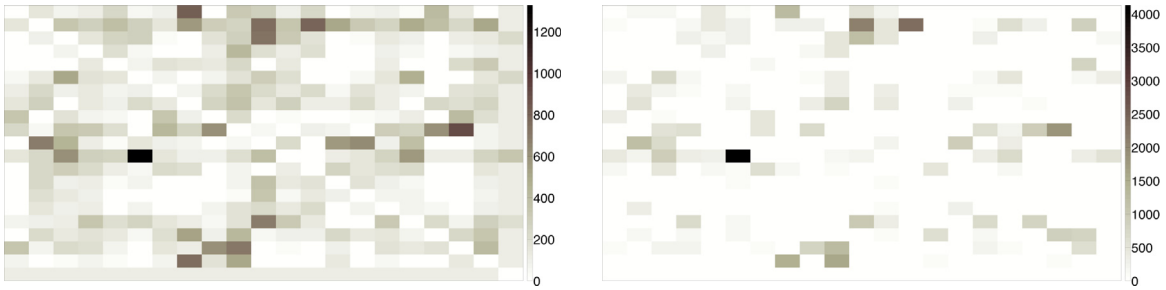
Table 2 summarises these performance measures from 5 independent trials of partitioning the links. Overall, we find that

- Simply relying on the prior OD matrix performs poorly in terms of NRMSE and NMAE, indicating the value of doing some form of optimisation based on link flows.
- Our automated zoning results in superior held-out predictions compared to the manually defined zoning when combined with all OD calibration algorithms. This is unsurprising, as the latter is simply unable to offer reasonable predictions for intra-zonal flows.
- Imposing a nonnegative constraint on the OD matrix during estimation has a non-trivial impact on performance: we find that NN-GLS outperforms plain GLS for both the manual and automated zoning, when both do not employ any additional regulariser. This agrees with previous study of the phenomenon (Bell, 1991).
- NN-GLS by itself is competitive with GLS and ℓ_1 regularisation for the automated zoning. We shall subsequently see that this is also true in terms of the sparsity of the solutions.
- ℓ_2 and ℓ_1 regularisation generally improve performance when used with GLS, with ℓ_2 being slightly more useful, by virtue of shrinking towards prior estimates of the OD. Their combination yields commensurate performance to either regulariser individually. Employing ℓ_1 regularisation has the advantage of improving sparsity of the resulting solution, as we shall see.

[12] This is akin to the R^2 coefficient of determination, which reports normalised *sum* of squared errors.
[13] Both the choice of T and $\mathbf{A}$ may be refined with techniques such as bilevel programming.

Table 2
Held-out flow prediction results of various zoning schemes and learners. The reported numbers are the mean and standard deviation over 5 independent trials. Lower values of the NRMSE and NMAE are better; higher values of ρ are better. "H/O" refers to performance on the holdout set, and "Eq." refers to performance of equilibrium predictions for the final OD estimate. See text for details of learners.

Zoning	Learner	Ω	Eq. NRMSE	H/O NRMSE	H/O NMAE	H/O ρ
Coarse	None	None	1.3468	1.3614 ± 0.0582	1.2348 ± 0.0792	0.2316 ± 0.1134
Coarse	GLS	None	1.0367	1.0557 ± 0.0994	0.9348 ± 0.1497	0.5585 ± 0.1110
Coarse	GLS	ℓ_2	1.1389	1.0855 ± 0.0643	0.9776 ± 0.0946	0.4882 ± 0.0868
Coarse	GLS	ℓ_1	0.9789	1.0497 ± 0.0702	0.9371 ± 0.1174	0.5857 ± 0.0676
Coarse	GLS	ℓ_1, ℓ_2	0.9807	1.0159 ± 0.0671	0.9084 ± 0.1284	0.5844 ± 0.0798
Coarse	GLS	BP	1.1236	8.7180 ± 1.2412	6.2041 ± 1.1723	0.0652 ± 0.1252
Coarse	NN-GLS	None	1.3321	2.5960 ± 1.8750	1.5561 ± 0.7082	0.3573 ± 0.2201
Coarse	NN-GLS	ℓ_2	1.1189	1.0307 ± 0.0634	0.9199 ± 0.1063	0.5566 ± 0.0778
Coarse	NN-GLS	ℓ_1	1.0098	1.0624 ± 0.0688	0.9527 ± 0.1173	0.5776 ± 0.0730
Coarse	NN-GLS	ℓ_1, ℓ_2	1.0036	1.0297 ± 0.0717	0.9190 ± 0.1298	0.5742 ± 0.0842
Coarse	NN-GLS	BP	1.3148	2.5956 ± 1.8783	1.5466 ± 0.7133	0.3619 ± 0.2315
Fine	None	None	1.0246	1.0415 ± 0.0657	0.9396 ± 0.0796	0.4694 ± 0.0594
Fine	GLS	None	0.7763	0.8701 ± 0.0182	0.7513 ± 0.0358	0.6584 ± 0.0188
Fine	GLS	ℓ_2	0.7352	0.8504 ± 0.0403	0.7337 ± 0.0391	0.6443 ± 0.0388
Fine	GLS	ℓ_1	0.6627	0.8655 ± 0.0632	0.7334 ± 0.0619	0.6633 ± 0.0466
Fine	GLS	ℓ_1, ℓ_2	0.7664	0.8466 ± 0.0327	0.7214 ± 0.0457	0.6687 ± 0.0332
Fine	GLS	BP	1.5542	10.0476 ± 1.3769	6.9549 ± 0.9683	0.3756 ± 0.0579
Fine	NN-GLS	None	0.7585	0.9029 ± 0.0744	0.7681 ± 0.0469	0.6322 ± 0.0664
Fine	NN-GLS	ℓ_2	0.7514	0.8475 ± 0.0409	0.7265 ± 0.0472	0.6625 ± 0.0370
Fine	NN-GLS	ℓ_1	0.6929	0.8736 ± 0.0738	0.7417 ± 0.0688	0.6554 ± 0.0549
Fine	NN-GLS	ℓ_1, ℓ_2	0.7514	0.8485 ± 0.0425	0.7270 ± 0.0478	0.6617 ± 0.0383
Fine	NN-GLS	BP	0.7041	0.8711 ± 0.0424	0.7450 ± 0.0515	0.6584 ± 0.0411



(a) GLS with ℓ_2.

(b) GLS with ℓ_1.

(c) NN-GLS.

(d) NN-GLS with BP.

Fig. 10. Heatmaps of the OD matrix estimates for various approaches. The rows represents origins, and the columns destinations, with cell colours representing the amount of flow between the appropriate OD pair. All methods assign large flow to a few dominant OD pairs, but differ in assignment of flow amongst the other OD pairs. Estimates derived from ℓ_1 and/or nonnegativity constraints are observed to be sparse. (Best viewed in colour.) (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

- The holdout the equilibrium flow NRMSE of a method are strongly correlated. However, the latter is generally lower, which is expected, as it employs an OD matrix that is learned given access to *all* link flows. Consequently, in some cases, the equilibrium NRMSE may give an overly optimistic estimate of performance.

7.9. Qualitative comparison of OD matrices

As a final study, we visualise the OD matrices produced by several of the estimation methods discussed above. We restrict attention to the automated zoning, and first consider the matrices produced by the GLS learner with ℓ_2 and ℓ_1 regularisation. Recall from Table 2 results that the ℓ_1 regulariser produces significantly superior performance in terms of the final equilibrium flow, although the held-out NRMSE of the two methods are similar. From Fig. 10a, we would nonetheless favour the ℓ_1 regularised solution in this case: we see that ℓ_2 solution produces a large number of OD estimates with a small, but non-zero flow. By contrast, the ℓ_1 solution favours concentration of the flow on only a few key pairs. This results in a more interpretable OD matrix.

There is however room to improve ℓ_1 regularised GLS solution: there are still a number of OD entries with a small but nonzero flow. This can be seen as a consequence of the fact that we do not explicitly enforce nonnegativity of flows during learner. Thus, we next turn attention to the NN-GLS learner, with no regularisation. In keeping with our earlier discussion, Fig. 10c illustrates that this method by itself induces sparse solutions. Indeed, the learned OD matrix is more than half as sparse as that produced by GLS with ℓ_1 regularisation, though both share the same dominant flow pairs. As this matrix is already sparse, performing an additional sparse approximation on top of this via basis pursuit (Fig. 10d) does not significantly change results.

8. Conclusion

This work proposed a strategy for the estimation of a sparse, fine-grained OD matrix. In particular, we proposed an algorithm for automatically constructing a fine-grained zoning, based on the intuition that we wish to consider OD pairs that best explain observed link flows under an equilibrium assignment. We then discussed how to encourage the estimation of a sparse OD matrix, using ℓ_1 regularisation. We also noted the non-trivial role that nonnegativity may play in the sparsity of OD estimates. We finally discussed how held-out link flow prediction can be used to assess the quality of OD estimates. Experimental results on a real-world network show encouraging results for our approach.

There are several avenues for future work. One is to do with the use of bilevel programming to estimate the OD matrix, which we expect will improve performance. A distinct approach is to directly estimate path flows. This brings a different set of advantages and disadvantages: one the one hand, one does not need to rely on an alternating optimisation, but on the other hand, one has to estimate a potentially exponential number of variables. The latter fact makes sparsity inducing regularisers a natural candidate, and indeed this has been explored in very recent work (Sanandaji and Varaiya, 2014). It is of also interest to see whether alternate generic regularisers may be useful for OD estimation, and to contrast them to ℓ_1 regularisation. For example, the trace norm regulariser has been explored in (Mardani and Giannakis, 2013); study of other regularisers such as the max-norm, which has been shown to yield empirical improvements over trace norm regularisation (Srebro et al., 2004), would be interesting.

Another important line of research is in the study of potentially more effective solvers to the automated zoning problem, as well as reformulations that loosen some of the imposed assumptions. As presented here, our automated zoning procedure is only concerned with accurate prediction of flows, and not e.g. agreement with geographical boundaries. Fusing the two desiderata would also be of interest.

Acknowledgements

We thank the Roads and Maritime Services (RMS) for providing data. This work was supported by NICTA. NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program.

References

Bell, M., Iida, Y., 1997. Transportation Network Analysis. John Wiley and Sons.

Bell, M.G., 1991. The estimation of origin-destination matrices by constrained generalised least squares. Transportation Research Part B 25 (1), 13–22, <http://www.sciencedirect.com/science/article/pii/019126159190010G>.

Bierlaire, M., 2002. The total demand scale: a new measure of quality for static and dynamic origin–destination trip tables. Transportation Research Part B 36 (9), 837–850.

Cao, J., Davis, D., Wiel, S.V., Yu, B., 2000. Time-varying network tomography: router link data. Journal of the American Statistical Association 95 (452), 1063–1075.

Cascetta, E., 1984. Estimation of trip matrices from traffic counts and survey data: a generalized least squares estimator. Transportation Research Part B 18 (4–5), 289–299, <http://ideas.repec.org/a/eee/transb/v18y1984i4-5p289-299.html>.

Cascetta, E., Nguyen, S., 1988. A unified framework for estimating or updating origin/destination matrices from traffic counts. Transportation Research Part B 22 (6), 437–455.

Cascetta, E., Papola, A., Marzano, V., Simonelli, F., Vitiello, I., 2013. Quasi-dynamic estimation of OD flows from traffic counts: formulation, statistical validation and performance analysis on real data. Transportation Research Part B 55, 171–187, <http://www.sciencedirect.com/science/article/pii/S0191261513001069>.

Chawla, S., Zheng, Y., Hu, J., 2012. Inferring the root cause in road traffic anomalies. In: Zaki, M.J., Siebes, A., Yu, J.X., Goethals, B., Webb, G.I., Wu, X. (Eds.), ICDM. IEEE Computer Society, pp. 141–150.

Cheman, K.M., 2006. Optimization Techniques for Solving Basis Pursuit Problems (Master's thesis). North Carolina State University.

Chen, S.S., Donoho, D.L., Saunders, M.A., 2001. Atomic decomposition by basis pursuit. SIAM Review 43 (1), 129–159. http://dx.doi.org/10.1137/S003614450037906X.
Donoho, D., 2006. Compressed sensing. IEEE Transactions on Information Theory 52 (4), 1289–1306.
Grant, M., Boyd, S., 2008. Graph implementations for nonsmooth convex programs. In: Blondel, V., Boyd, S., Kimura, H. (Eds.), Recent Advances in Learning and Control, Lecture Notes in Control and Information Sciences. Springer-Verlag Limited, pp. 95–110, <http://stanford.edu/boyd/graph_dcp.html>.
Grant, M., Boyd, S., Mar. 2014. CVX: Matlab software for disciplined convex programming, version 2.1. <http://cvxr.com/cvx>.
Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning: Data Mining, Inference, and Prediction, second ed. Springer.
Hazelton, M.L., 2001. Inference for origin-destination matrices: estimation, prediction and reconstruction. Transportation Research Part B 35 (7), 667–676, <http://www.sciencedirect.com/science/article/pii/S0191261500000096>.
Hazelton, M.L., 2015. Network tomography for integer-valued traffic. The Annals of Applied Statistics 9 (1), 474–506.
Lamond, B., Stewart, N., 1981. Bregman's balancing method. Transportation Research Part B 15 (4), 239–248, <http://www.sciencedirect.com/science/article/pii/0191261581900102>.
Maher, M.J., 1983. Inferences on trip matrices from observations on link volumes: a Bayesian statistical approach. Transportation Research Part B 17 (6), 435–447.
Mardani, M., Giannakis, G., May 2013. Robust network traffic estimation via sparsity and low rank. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 4529–4533.
Martínez, L.M., Viegas, J.M., Silva, E.A., 2009. A traffic analysis zone definition: a new methodology and algorithm. Transportation 36 (5), 581–599.
Meinshausen, N., 2013. Sign-constrained least squares estimation for high-dimensional regression. Electronic Journal of Statistics 7, 1607–1631, <http://dx.doi.org/10.1214/13-EJS818>.
Menon, A.K., Cai, C., Wang, W., Wen, T., Chen, F., 2015. An approach to sparse, fine-grained od estimation. In: 94th Annual Meeting of the Transportation Research Board (TRB).
Ortuzar, J.D., Willumnsen, L.G., 2011. Modeling Transport, fourth ed. John Wiley and Sons, New York.
Sanandaji, B.M., Varaiya, P.P., 2014. Compressive origin-destination matrix estimation. CoRR abs/1404.3263.
Sheffi, Y., 1985. Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods. Prentice Hall Inc., New Jersey.
Sherali, H.D., Sivanandan, R., Hobeika, A.G., 1994. A linear programming approach for synthesizing origin-destination trip tables from link traffic volumes. Transportation Research Part B 28 (3), 213–233, <http://www.sciencedirect.com/science/article/pii/0191261594900086>.
Slawski, M., Hein, M., 2012. Non-negative least squares for high-dimensional linear models: consistency and sparse recovery without regularization. <http://arxiv.org/abs/1205.0953>.
Spiess, H., 1987. A maximum likelihood model for estimating origin-destination matrices. Transportation Research Part B 21 (5), 395–412, <http://www.sciencedirect.com/science/article/pii/0191261587900373>.
Srebro, N., Rennie, J.D.M., Jaakkola, T., 2004. Maximum-margin matrix factorization. In: Advances in Neural Information Processing (NIPS).
Tamin, O., Willumsen, L., 1989. Transport demand model estimation from traffic counts. Transportation 16 (1), 3–26. http://dx.doi.org/10.1007/BF00223044.
Tebaldi, C., West, M., 1998. Bayesian inference on network traffic using link count data (with discussion). Journal of the American Statistical Association 93, 557–576, <http://ftp.stat.duke.edu/WorkingPapers/96-16.html>.
Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society, Series B 58, 267–288.
Van-Zuylen, H., Willumsen, L., 1980. The most likely trip matrix estimated from traffic counts. Transportation Research Part B 14 (3), 281–293.
Vardi, Y., 1996. Network tomography: estimating source-destination traffic intensities from link data. Journal of the American Statistical Association 91 (433), 365–377.
Wang, M., Tang, A., 2009. Conditions for a unique non-negative solution to an underdetermined system. In: 47th Annual Allerton Conference on Communication, Control, and Computing, Allerton. pp. 301–307.
Wang, M., Xu, W., Tang, A., 2011. A unique "nonnegative" solution to an underdetermined system: from vectors to matrices. IEEE Transactions on Signal Processing 59 (3), 1007–1016.
Willumsen, L., 1981. Simplified transport models based on traffic counts. Transportation 10 (3), 257–278. http://dx.doi.org/10.1007/BF00148462.
Yang, H., Sasaki, T., Iida, Y., Asakura, Y., 1992. Estimation of origin-destination matrices from link traffic counts on congested networks. Transportation Research Part B 26 (6), 417–434, <http://ideas.repec.org/a/eee/transb/v26y1992i6p417-434.html>.
Zhang, Y., Ge, Z., Greenberg, A., Roughan, M., 2005. Network anomography. In: Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement. IMC '05. USENIX Association, Berkeley, CA, USA, pp. 30–30, <http://dl.acm.org/citation.cfm?id=1251086.1251116>.
Zhang, Y., Roughan, M., Duffield, N., Greenberg, A., 2003a. Fast accurate computation of large-scale IP traffic matrices from link loads. In: Proceedings of the 2003 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems. SIGMETRICS '03. ACM, New York, NY, USA, pp. 206–217. http://dx.doi.org/10.1145/781027.781053.
Zhang, Y., Roughan, M., Lund, C., Donoho, D., 2003b. An information-theoretic approach to traffic matrix estimation. In: Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. SIGCOMM '03. ACM, New York, NY, USA, pp. 301–312. http://dx.doi.org/10.1145/863955.863990.
Zhu, C., Byrd, R.H., Lu, P., Nocedal, J., 1997. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. ACM Transactions on Mathematical Software 23 (4), 550–560.
Zou, H., Hastie, T., 2005. Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society, Series B 67, 301–320.