Transportation Research Part C 111 (2020) 513–530

Contents lists available at ScienceDirect

Transportation Research Part C

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

A novel approach for the structural comparison of origin-destination matrices: Levenshtein distance

Krishna N.S. Behara[a], Ashish Bhaskar[a,*], Edward Chung[b]

[a] School of Civil Engineering & Built Environment, Science and Engineering Faculty, Queensland University of Technology, 2 George St, GPO Box 2434, Brisbane, Qld 4001, Australia

[b] Department of Electrical Engineering, The Hong Kong Polytechnic University, Hong Kong

ARTICLE INFO

Keywords:
OD matrix structure
Structural comparison
Levenshtein distance
Destination choices
Trip distribution
Bluetooth OD matrices
Brisbane

ABSTRACT

Origin-Destination (OD) matrix is a tableau of travel demand distributed between different zonal pairs. Essentially, OD matrix provides two types of information: (a) the individual cell value represents travel demand between a specific OD pair; and (b) group of OD pairs provides insights into structural information in terms of distribution pattern of OD flows. Comparison of OD matrices should account both types of information. Limited studies in the past developed structural similarity measures, and most studies still depend on traditional measures for OD matrices comparison. Traditional performance measures are based on cell by cell comparison, and often neglect OD matrix structural information within their formulations.

We propose a methodology that adopts the fundamentals of Levenshtein distance, traditionally used to compare sequences of strings, and extends it to quantify the structural comparison of OD matrices. The novel performance measure is named as *normalised Levenshtein distance for OD matrices* (NLOD). The results of sensitivity analysis support NLOD to be a robust statistical measure for holistic comparison of OD matrices. The study demonstrates the practicality of the approach with a case study application on real Bluetooth based OD matrices from the Brisbane City Council (BCC) region, Australia.

1. Introduction

An origin destination (OD) matrix is a tableau representation of the travel demand distributed between different origin and destination locations of a study region. The OD matrix is a key input into most traffic simulation models and it essentially provides two major aspects of travel information. The cell values of the OD matrix represent flows between individual OD pairs and a group of OD flows represents the structure of demand distribution. Due to unavailability of ground truth, OD matrices are generally estimated from observed traffic counts. OD matrices are high dimensional datasets, and the estimation problem is usually under-determined because of many to one relationship between OD flows and link flows. Thus, most studies in this domain have focussed on developing new solution algorithms for OD estimation (Cipriani et al., 2011; Kim et al., 2001; Krishnakumari et al., 2019; Michau et al., 2017; Yang, 1995); methods to address the dimensionality problem (Djukic et al., 2012; Guo et al., 2012; Osorio, 2019; Yang et al., 2017); and performance measures to compute the (dis)similarity between the OD matrices; say, target OD and estimated OD (Bierlaire, 2002; Djukic et al., 2013; Tavassoli et al., 2016).

In literature, there is no formal definition for OD matrix structure. Researchers have considered a proxy for the OD matrix

* Corresponding author.
E-mail address: ashish.bhaskar@qut.edu.au (A. Bhaskar).

https://doi.org/10.1016/j.trc.2020.01.005
Received 16 May 2019; Received in revised form 6 January 2020; Accepted 8 January 2020
Available online 13 January 2020
0968-090X/

	D1	D2	D3	D4
O1	3	4	6	10
O2	7	4	5	11
O3	12	8	5	6
O4	13	7	9	6

OD matrix

(a)

	Dest. Choice-1	Dest. Choice-2	Dest. Choice-3	Dest. Choice-4
O1	D4	D3	D2	D1
O2	D4	D1	D3	D2
O3	D1	D2	D4	D3
O4	D1	D3	D2	D4

Skeleton/structure of OD matrix

(b)

	Dest. Choice-1	Dest. Choice-2	Dest. Choice-3	Dest. Choice-4
O1	10	6	4	3
O2	11	7	5	4
O3	12	8	6	5
O4	13	9	7	6

Mass/ OD flows on Skeleton/Structure

Fig. 1. Demonstration of (a) the skeleton/structure of OD and (b) corresponding mass/OD flows.

structural information in the OD estimation process. Gur (1980) proposed to use target trip matrix as a proxy, and many other OD estimation studies (Cantelmo et al., 2014; Cascetta, 1984; Doblas and Benitez, 2005; Lundgren and Peterson, 2008; Yang, 1995) have later adopted this approach. Other studies that have emphasised on the OD structure include: (a) Laharotte et al. (2015) have applied latent Dirichlet allocation (LDA) approach to cluster similar OD pairs (say, origins from high density residential zones and destinations from high density work zones) in order to identify latent structure within the OD matrix; and (b) Ashok (1996) preserved the structural integrity of OD flows (both estimated and predicted) by introducing deviations of OD flows from historical OD flows in the state space model formulation. However, these studies have not explicitly defined the term 'structure'. In general, the OD structure is captured from the normalised OD matrix where either each row of the matrix is normalized with total productions from the row (as by Kim et al. (2001)), or each column of the matrix is normalized with total attractions from the column (as by Bierlaire and Toint (1995)), or as the correlation between the normalized OD demand matrices (as by Djukic (2014)).

The term 'structure' in the Oxford dictionary is defined as: 'the arrangement of and relations between the parts or elements of something complex' (Oxford, 2018). To avoid ambiguity, in this research we establish the definition as follows:

(a) **Structure** is the skeletal framework of the OD matrix where the skeleton is expressed as the preference/arrangement of the destinations from each origin. For instance, refer to Fig. 1 where the skeleton/structure of the OD matrix (shown at the top) is illustrated in Fig. 1a. Here, the columns for each row (origin) is arranged in the order of destination preferences. The preference of destinations captures the order of trips distributed from each origin and thus, the definition of OD structure is trip production-based in this paper.
(b) The OD flows corresponding to the OD structure (skeleton) is termed as **mass.** The corresponding mass for the structure illustrated in Fig. 1a is presented in Fig. 1b.

Thus, the structure of OD and individual OD flows are analogous to the "skeleton" and "mass" on the skeleton, respectively.

Different methods are used to quantify the **similarity** between two OD matrices. If the structure of the OD matrices is also considered in the similarity estimation, then we term it as **structural similarity**. Two OD matrices have perfect structural similarity if their structures are similar with zero differences in the OD flows. Perfect structural similarity is possible only when the OD matrices are exactly the same.

In literature, traditional performance measures such as root mean square error (RMSE) and mean absolute error (MAE) are widely used to estimate similarity between OD matrices (see Section 2.1). For structural similarity, the literature is sparse and includes following measures:

(a) Mean structural similarity index (MSSIM) (Djukic et al., 2013);
(b) MSSIM variants i.e. 4D-MSSIM (Van Vuren and Day-Pollard, 2015) and geographical window based structural similarity index (GSSI) (Behara et al., 2019); and
(c) Wasserstein distance (Ruiz de Villa et al., 2014).

The covariance component of SSIM is the statistical measure that compares the structural (skeletal) differences between OD matrices. The other components (i.e. mean and standard deviation) take care of the mass (details in Section 2.2.1). Therefore, SSIM should be able to measure the structural differences between OD matrices. However, the application needs to define the window size for which the results can be sensitive and in literature, there is no consensus on the size of the window for OD matrices comparison. Moreover, the stability constants used in the formulation need to be defined and can be network specific ((Day-Pollard and Van

Vuren, 2015). The GSSI and 4D-MSSIM are only mere extensions to MSSIM that need further exploration (details in Section 2.2.2). Wasserstein metric formulation has an implicit consideration of the skeleton and mass (details in Section 2.2.3). However, the formulation is an optimization problem and computationally very expensive.

To this end, we propose a new measure based on Levenshtein distance for structural (dis)similarity of OD matrices. The formulation properly defines the structure and mass for OD matrices and is an alternative metric in the limited available literature on structural comparison of OD matrices.

Levenshtein distance (Levenshtein, 1966) is a measure, from computational linguistics and computer science, generally used to compare two strings based on the sequence of characters (details in Section 3). The sequence of characters defines the structure of a string. Metrics similar to Levenshtein such as sequence alignment method (Allahviranloo and Recker, 2015) were previously used in the transport applications like comparison of sequences of activity-travel patterns. However, due to two-dimensional array structure of OD flows between different origin and destination pairs, the application of these metrics on OD matrices is ignored. In this paper, we have uniquely identified the potential of Levenshtein measure to be extended for OD matrix comparison. We propose a methodology that adopts the fundamentals of Levenshtein distance and extends it to quantify the structural comparison of OD matrices. The novel performance measure is named as *normalised Levenshtein distance for OD matrices* (NLOD). The robustness of the measure for real world application is tested through its sensitivity analysis on the Bluetooth based OD matrices from the Brisbane City Council (BCC) region, Australia; and a case study is presented to interpret day-to-day variations within a typical week of March 2016 from the same study region.

Accounting for the structural dimension of OD matrix has many practical applications such as:

(a) *Constraints in OD estimation*: In the standard bi-level formulation (Doblas and Benitez, 2005), the OD matrix is generally estimated by minimizing the gap between the observed and assigned traffic counts. This is highly an under determined mathematical problem with multiple solutions satisfying the minimization objective. Traditionally, the deviation between the target and estimated OD matrices is added as a constraint to restrict the search space. This can be extended by also incorporating the structural consistency between target and estimated OD matrices, which should further result in better estimates.
(b) *OD matrix estimation model performance evaluation*: For holistic performance evaluation of the OD matrix estimation algorithm (on a benchmark network (Djukic et al., 2013)) we should incorporate the structural similarity of the estimated OD matrix with that of the true OD matrix.
(c) *Travel pattern identification*: Analysis of travel patterns through the comparison of the OD matrices from different time periods (such as day to day or within the day variations) (Andrienko et al., 2017; Behara et al., 2018; Guo et al., 2012; Yang et al., 2017) requires clustering OD matrices of similar nature. The key component of any clustering process is the comparison of OD matrices using (dis)similarity measure, which would be holistic if the structural component is also included.

The remainder of the paper is structured as follows: Section 2 reviews the literature on the use of statistical measures for OD matrices comparison; Section 3 presents the traditional Levenshtein distance method; Section 4 explains in detail the development process of the proposed measure – NLOD; Section 5 tests its robustness through sensitivity analysis; Section 6 presents a case study application using one week Bluetooth OD dataset from the BCC region; Section 7 discusses about the performance of NLOD; and finally, the paper concludes in Section 8.

2. Related studies on performance measures

Although there are a number of performance measures widely used in transport applications (Ciuffo and Punzo, 2010; Hollander and Liu, 2008), a limited number of studies focus on the development of measures suitable for the structural comparison of OD matrices. The review here focuses on (dis)similarity and structural (dis)similarity measures used in the literature.

2.1. The (dis)similarity measures

Some notable traditional measures widely used in the literature are: root mean square error (RMSE) (Ashok and Ben-Akiva, 2002; Barceló Bugeda et al., 2010; Tamin and Willumsen, 1989); normalised root mean square error (RMSN) (Antoniou et al., 2004); mean square error (MSE) (Cascetta, 1984); mean absolute error ratio (MAER) (Kim et al., 2005); mean absolute percent error (MAPE) (Cools et al., 2010); goodness of Theil's fit (GU) (Barceló et al., 2013); maximum possible absolute error (MPAE) (Yang et al., 1991); relative error (RE) (Gan et al., 2005); total demand deviation (TDD) (Bera and Rao, 2011); R-squared (R^2) (Tavassoli et al., 2016), and entropy measure (E) (Ros-Roca et al., 2018). Formulations of a few are shown in the Eqs. (1)–(3) where X_w and X_w are the OD flows of w^{th} OD pair from estimated ($\mathbf{X}$) and target ($\mathbf{X}$) OD matrices, respectively. The total number of OD pairs in both matrices is 'W' each.

$$\mathrm{RMSE}(\mathbf{X}, \mathbf{X}) = \sqrt{\frac{1}{W} \sum_{w \in W} (X_w - X_w)^2} \tag{1}$$

$$\mathrm{GU}(\mathbf{X},\mathbf{X}) = \frac{\sqrt{\frac{\sum_{w \in W}(X_w - X_w)^2}{W}}}{\sqrt{\frac{\sum_{w \in W} {X_w}^2}{W}} + \sqrt{\frac{\sum_{w \in W} {X_w}^2}{W}}} \tag{2}$$

$$\mathrm{E}(\mathbf{X},\mathbf{X}) = \sum_{w \in W} \left(X_w \log\left(\frac{X_w}{X_w}\right) - X_w + X_w \right) \tag{3}$$

The aforementioned measures have simple mathematical formulations and compute deviations between individual OD flows. However, they do not compare groups of OD pairs, and thus, are incapable of accounting the structural properties of OD matrices such as trip productions/attractions, destination choices, geographical correlations etc. (Behara et al., 2019).

2.2. The structural (dis)similarity measures

The literature on structural similarity measures is very limited and the measures are adopted from applications in other fields as discussed in the below sections.

2.2.1. Mean structural SIMilarity index (MSSIM)

MSSIM (Wang et al., 2004) is widely used to compare the structural degradation between two images where the pixels of the images are compared. Djukic et al. (2013) applied MSSIM on OD matrices by arguing that the cells of OD matrix are analogous to pixels of an image.

The structural similarity index (SSIM) statistics for two OD matrices is computed at various local windows (of dimensions n × n). The SSIM's formulation is composed of three components - luminance ($l(\mathbf{x},\mathbf{y})$), contrast ($c(\mathbf{x},\mathbf{y})$), and structure ($s(\mathbf{x},\mathbf{y})$) as expressed in Eqs. (4), (4a), and (4b), respectively. Here, $\mathbf{x}$ and $\mathbf{y}$ represent the group of OD pairs within local windows in OD matrices, $\mathbf{X}$ and $\mathbf{Y}$. These components contribute to the comparison of mean ($\mu_\mathbf{x}$ and $\mu_\mathbf{y}$), standard deviation ($\sigma_\mathbf{x}$ and $\sigma_\mathbf{y}$) and covariance ($\sigma_{\mathbf{xy}}$), respectively. While the mean and standard deviation components compare the mass/OD flows, covariance compares the skeletal/structural differences between matrices. The product of these three components in general form is presented in Eq. (5), and in the simplified form in Eq. (5a), respectively. The constants C_1, C_2 and C_3 are meant to stabilize the result when either mean or standard deviation is close to zero. Generally, C_3 is assumed to be $C_2/2$. Pollard et al. (2013) suggested values of 10^{-10} and 10^{-2} for C_1 and C_2, and are of the opinion that these values are network specific. The parameters α, β and Y are used to adjust the relative importance of mean, standard deviation, and structural components respectively, and are generally assumed to be equal to 1.

The average of all SSIM values (over '$\bar{W}$' local windows) yields MSSIM (refer to Eq. (6)). The values of both SSIM and MSSIM lie between −1 and 1. The value of 1 implies that matrices are exactly the same while the reverse is true when value is −1.

$$l(\mathbf{x},\mathbf{y}) = \frac{(2\mu_x\mu_y + C_1)}{(\mu_x^2 + \mu_y^2 + C_1)} \tag{4}$$

$$c(\mathbf{x},\mathbf{y}) = \frac{(2\sigma_x\sigma_y + C_2)}{(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{4a}$$

$$s(\mathbf{x},\mathbf{y}) = \frac{(\sigma_{xy} + C_3)}{(\sigma_x\sigma_y + C_3)} \tag{4b}$$

$$\begin{gathered}\mathrm{SSIM}(\mathbf{x},\mathbf{y}) = [l(\mathbf{x},\mathbf{y})^{\alpha}][c(\mathbf{x},\mathbf{y})^{\beta}][s(\mathbf{x},\mathbf{y})^{Y}];\ \alpha > 0, \beta > 0 \text{ and } Y > 0;\\ \text{Assuming } \alpha = \beta = Y = 1 \text{ and } C3 = C2/2\end{gathered} \tag{5}$$

$$\mathrm{SSIM}(\mathbf{x},\mathbf{y}) = \frac{(2\mu_\mathbf{x}\mu_\mathbf{y} + C_1)(2\sigma_\mathbf{xy} + C_2)}{(\mu_\mathbf{x}^2 + \mu_\mathbf{y}^2 + C_1)(\sigma_\mathbf{x}^2 + \sigma_\mathbf{y}^2 + C_2)} \tag{5a}$$

$$\mathrm{MSSIM}(\mathbf{X},\mathbf{Y}) = \frac{1}{\bar{W}} \sum_{\bar{w} \in \bar{W}} \mathrm{SSIM}(\mathbf{x},\mathbf{y}) \tag{6}$$

Application of MSSIM on OD matrices might need further exploration due to the following reasons:

1. MSSIM on OD matrices can result in different values based on the choice of local window size. In literature, no clear consensus has been reported on the level of acceptability of sliding window size and the resulting MSSIM values. On the other hand, choice of one single window might not be able to capture better correlation distortions (structural differences) within the matrices despite being statistically accurate.
2. Wang et al. (2004) used a combination of dynamic range of pixel values (255 for 8-bit grayscale images) and a small constant value ≪1 to estimate the values for stability constants. But for OD matrices, Pollard et al. (2013) opine that the stability constants could be network specific that needs further exploration.
3. MSSIM is sensitive to the order of OD matrix if computed using local windows and the order of matrix might not guarantee

geographic adjacency between consecutive zones of OD matrix (Pollard et al., 2013). This issue related to spatial adjacency does not arise in the comparison of images.

2.2.2. MSSIM's variants

Addressing the sensitivity of MSSIM towards local window size and geographical adjacency of zones, variants of MSSIM, namely GSSI (Behara et al., 2019) and 4D-MSSIM (Van Vuren and Day-Pollard, 2015) have been proposed in the literature. While both extensions use the same underlying SSIM formulation they differ as follows:

- in GSSI technique, geographical boundaries of higher-level zones are used to define the local windows; and
- in 4D-MSSIM, neighbourhood OD pairs are identified using spatial proximity between OD pairs. The spatial proximity is measured using Euclidian distance *see* Eq. (7) where $d(\tau_a^b, \tau_c^d)$ is the Euclidian distance from OD pair τ_a^b(flow from zone *a* to zone *b*) to OD pair τ_c^d (flow from zone *c* to zone *d*) and the coordinates of the centroid of zones *a*, *b*, *c* and *d* are (x_a, y_a), (x_b, y_b), (x_c, y_c) and (x_d, y_d), respectively.

$$d(\tau_a^b, \tau_c^d) = \sqrt{(x_a - x_c)^2 + (y_a - y_c)^2 + (x_b - x_d)^2 + (y_b - y_d)^2} \tag{7}$$

In above variants there is no ambiguity related to either selection of local window size or spatial adjacency of zones because OD pairs are geographically correlated in both cases. Since these variants use SSIM formulation, they consider both skeleton (through covariance) and mass (through mean and standard deviation). However, further exploration is required. For instance, the issue related to stability constants (as discussed previously) still persists; and the choice of windows (as in GSSI) and spatial proximity (as in 4D-MSSIM) for identifying geographically correlated OD pairs are network specific.

2.2.3. Wasserstein metric

Wasserstein distance (Ruiz de Villa et al., 2014) is popularly used in mass transportation problems such as optimal cost required to transfer iron-ore from many mining locations to several factories etc. In the context of OD matrices, the Wasserstein distance between two matrices is defined as the minimum total travel time required to assign the trips between OD pairs of matrix "**X**" (using an assignment compatible with matrix "**Y**"). Since the main goal of this metric is to transform one matrix into another, it implicitly considers both skeletal and mass differences while computing the transformation cost.

To explain the metric's formulation, consider an OD pair $\tau = (\tau_o, \tau_d)$ in **X**, and let $f_\tau^{\mathbf{X}}$ denote trips for τ in **X** i.e. from origin 'τ_o' to destination 'τ_d'. Now suppose that for τ, we have less trips in matrix **Y** than that in **X**; that is, $f_\tau^{\mathbf{Y}} < f_\tau^{\mathbf{X}}$. Thus, the difference in the trips i.e. $f_\tau^{\mathbf{X}} - f_\tau^{\mathbf{Y}}$ should penalise the comparison between **X** and **Y**. This is achieved by assigning vehicles to nearby OD pairs at the expense of some travel cost generally evaluated using the formulation as expressed in Eq. (8).

$$d(\tau, \gamma) = d(\tau_o, \gamma_o) + d(\tau_d, \gamma_d) \tag{8}$$

Here, $\gamma = (\gamma_o, \gamma_d)$ is another OD pair, and $v_\tau^\gamma$is the amount of trips assigned fromτtoγ. The cost $d(\tau, \gamma)$ is the average travel time required to go from $\tau_o$toγ_o, and then return to τ_d fromγ_d. Now, let's consider **X** and **Y** have total same mass/OD flows. Wasserstein distance between **X** and **Y** is expressed as Equation (9).

$$\text{Wasserstein}\ (\mathbf{X}, \mathbf{Y}) = \min\left(\sum_{\tau,\gamma} v_\tau^\gamma d(\tau, \gamma)\right);\ v_\tau^\gamma \geq 0 \tag{9}$$

In case of unequal masses between **X** and **Y**, Ruiz de Villa et al. (2014) suggested to create virtual OD pairs at the boundaries that can balance the difference. However, the solution depends on how (far) the new centroids are selected.

While this technique is able to capture the differences in trip distribution very well, but it is computationally very expensive. According to Ruiz de Villa et al. (2014) "One of the main drawbacks of this method is in computing the Wasserstein distance on large networks. If there are n centroids, then we have n^2 OD pairs and so the problem has n^4 variables".

2.3. Summary of literature review

To summarise, most studies focused on similarity measures only, and very limited studies such as MSSIM, its variants and Wasserstein have been proposed to compute structural (dis)similarity between OD matrices. Although these metrics compute differences in mass/OD flows and in skeleton/structure between OD matrices through explicit (as in MSSIM and its variants) and implicit (as in Wasserstein metric) formulation, no formal definition for OD matrix structure is provided in those studies. Also, these metrics needed further exploration as discussed above. There are avenues to explore alternate measures that can add to the body of literature to measure the structural (dis)similarity of the OD matrices.

3. Traditional Levenshtein distance measure

Levenstein distance, developed by Vladimir Levenstein in (1966), is a measure of proximity between two strings. It is mainly applied to compare sequences in linguistics domain such as plagiarism detection and speech recognition; in molecular biology for comparing sequences of macro molecules, etc.

Levenstein distance calculates least expensive set of *insertions*, *deletions* or *substitutions* that are required to transform one string

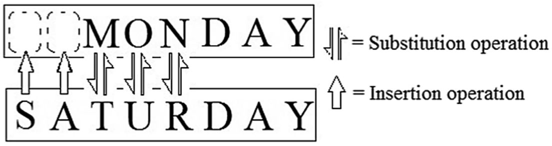


Fig. 2. Comparison of strings using GLD.

into another. For example, if we have to compare two strings such as "MONDAY' and 'SATURDAY', one of the optimum ways is to insert the letters "S" and "A" and substitute "M", "O" and "N" with "T", "U" and "R", respectively leading towards a generalised Levenstein distance (GLD) of 5 (assuming a unit distance for each operation) as shown in Fig. 2.

To understand the GLD technique and its formulation, let's say, $\mathbf{X}$ represents any string expressed as $\mathbf{X} = (x_1..x_i..x_q)$ where x_i is the i^{th} character of $\mathbf{X}$. The substring of $\mathbf{X}$ is represented as $\mathbf{X}_{i..j}$ that includes characters from x_i to x_j where $1 \leq i \leq j \leq q$. The length of $\mathbf{X}_{i..j}$ is expressed as $|\mathbf{X}_{i..j}|$ and is computed using Equation (10).

$$|\mathbf{X}_{i..j}| = j - i + 1 \tag{10}$$

A null string, with no character, is expressed as ε (with$|\varepsilon| = 0$). Any general edit operation for a pair of characters (a, b) is expressed as $a \rightarrow b$ where a and b are from separate strings.

If string $\mathbf{X}$ is the result of the operation $a \rightarrow b$ to string $\mathbf{Y}$, then it can be written as $\mathbf{Y} \Rightarrow \mathbf{X}$ via $a \rightarrow b$. The notations for the three operations are expressed as follows:

(1) *Insertion*: if $\varepsilon \rightarrow a$;
(2) *Deletion*: if $b \rightarrow \varepsilon$; and
(3) *Substitution*: if $a \rightarrow b$; $a \neq \varepsilon$ and $b \neq \varepsilon$

Let's define $\mathbf{S} = (S_o, S_1, ..S_k \cdots S_s)$ as the sequence of edit operations to transform $\mathbf{Y} \Rightarrow \mathbf{X}$ and then the cost associated with each edit operation as $\beta = (\beta_0, \beta_1. .\beta_k. .\beta_s)$. GLD is the minimum total cost required to transform $\mathbf{Y}$ to $\mathbf{X}$ as shown in Eq. (11).

$$\mathrm{GLD}(\mathbf{Y}, \mathbf{X}) = \min_{S}\left(\sum_{k=0}^{k=s} \beta_k\right) \tag{11}$$

The normalised Levenshtein distance (NLD) is the GLD normalised by the sum of the lengths of two strings (Eq. (12)). This measure always lies between 0 and 1 (Yujian and Bo, 2007).

$$\mathrm{NLD}(\mathbf{Y}, \mathbf{X}) = \frac{\mathrm{GLD}(\mathbf{X}, \mathbf{Y})}{|\mathbf{X}| + |\mathbf{Y}|} \tag{12}$$

Heeringa (2004) developed a pseudo code for computing GLD and NLD for comparing two strings. The *Algorithm 1* shown in Table 1 demonstrates the comparison between the strings, $\mathbf{X}$ and $\mathbf{Y}$. Here $\mathbf{X} = (x_1 \ldots x_q)$ and $\mathbf{Y} = (y_1 \ldots y_p)$. The lengths of strings i.e. $|\mathbf{X}|$ and $|\mathbf{Y}|$ are q and p, respectively.

The above pseudo code is explained with the help of an example presented in Fig. 3. Here, the matrix, $\mathbf{K}$, of size $(p + 1) * (q + 1)$ is defined to compare the strings $\mathbf{Y}$ (size p) and $\mathbf{X}$ (size q). The numbering of rows and columns of the matrix $\mathbf{K}$ commence with 0 (K (1,1)), and the values [0..q] and [0..p] are assigned to the first row and first column, respectively (see grey shaded column and row in Fig. 3). This is done to facilitate the comparison of the first character from both strings $\mathbf{X}$ and $\mathbf{Y}$. The comparison is made by traversing the matrix row by row, and then column wise until all characters in both strings are compared. Since, the overall comparison of all characters ends at the last cell of the matrix, $\mathbf{K}(p + 1, q + 1)$ is chosen as GLD value.

The presented arrows in Fig. 3 illustrates one possible path to reach to the cell $\mathbf{K}(p + 1,q + 1)$. There are multiple paths (i.e. different combination of arrows) possible to arrive at the final $\mathbf{K}(p + 1,q + 1)$. Each path is a combination of editing operations

Table 1
Algorithm 1 demonstrating computation of the Levenshtein distance for strings comparison.

Create an empty matrix "K" of size $(p + 1) * (q + 1)$ where the row and column headers correspond to characters of the string $\mathbf{Y}$ and $\mathbf{X}$, respectively.
Assign values [0..q] and [0..p] to the first row and first column, respectively
for j = 1 to q
for i = 1 to p
Estimate cost as$C_{i,j} = \begin{cases} 0 \, if \, y_i = x_j \\ 1 \, if \, y_i \neq x_j \end{cases} \forall \, i = [1. .p] and j = [1. .q]$
Set the cell $\mathbf{K}(i,j) = \min(\mathbf{K}(i - 1,j) + 1, \mathbf{K}(i,j - 1) + 1, \mathbf{K}(i - 1,j - 1) + C_{i,j})$ where:
$\mathbf{K}(i - 1,j) + 1$ represents the cell value immediately above the current cell plus 1
$\mathbf{K}(i,j - 1) + 1$ represents the cell value immediately to the left of current cell plus 1
$\mathbf{K}(i - 1,j - 1) + C_{i,j}$ represents the cell value immediately in diagonal above and to the left of current cell plus the cost
The GLD is the value of the cell $\mathbf{K}(p + 1, q + 1)$ and the NLD $= \frac{\mathrm{GLD}}{p + q}$

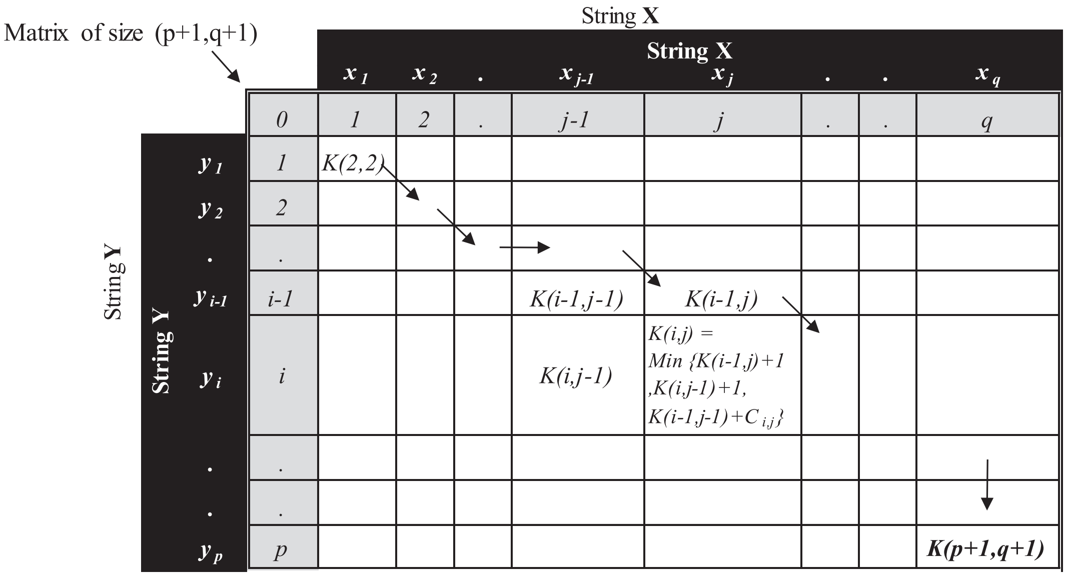


Fig. 3. Matrix demonstration of traditional Levenshtein approach (Algorithm 1).

represented as the following moves on the matrix grid: downward movement along the diagonal is for substitution operations, eastward movement is for deletion operation, and vertical downward movement is for insertion operation (Oliveira-Neto et al., 2012).

In literature, the use of Levenshtein distance for transport applications is relatively scarce. Oliveira-Neto et al. (2012), for instance, applied this technique to compare license plates. Here, the sequence of characters on the license plates observed at upstream and downstream stations were compared. Zhang et al. (2018) applied Levenshtein technique to compare the sequences of trip purposes and cluster activity-travel patterns. Markou et al. (2019) used Levenshtein distance as one of the algorithms to compare time series spatiotemporal event data from the internet and predict taxi demand hotspots. Other researchers have used similar techniques (such as sequence alignment method (SAM)) to compare any two activity-travel patterns (as by Allahviranloo and Recker, 2015); and sequence of trips (as by Crawford et al., 2018). The commonality among these studies is that they were similar to comparison of one-dimensional strings with unit cost for each operation. However, OD matrices are two-dimensional arrays consisting of OD flows between different origin and destination pairs, which means direct application of such traditional techniques is not possible. In light of this, the following section proposes a methodology to extend the applicability of traditional Levenshtein distance for the structural comparison of OD matrices.

4. Proposed Levenshtein distance for structural comparison of OD matrices

For the applicability of Levenshtein distance on OD matrices comparison we propose to:

(a) Consider each row of an OD matrix independently. The values in each row corresponds to the flow from an origin to different destinations. For a given origin we define a 'string' where each character is a destination ID arranged in the descending order of OD flows and is referred as 'sorted row'. This represents the 'skeleton' of the respective row of the OD matrix. To compare the structure of two OD matrices, we compare the order of destination IDs in each sorted row of the OD matrix.
(b) Include OD flows (mass) in the formulation of Levenshtein distance, the details for which are presented in the Section 4.1.1.

Hereon, the proposed modified approach is termed as *Levenshtein distance for OD matrices*.

4.1. Methodological formulation of Levenshtein distance for OD matrices

Before describing the proposed formulation, let us consider an example as shown in Fig. 4a where two OD matrices **X** (reference matrix) and **Y** (query matrix), each of dimensions M * M, are to be compared. Here, the origin IDs are expressed as O1, O2, O3 and O4 and destination IDs are expressed as N, E, W and S (M = 4 in this example). In Fig. 4b, the rows of each matrix are sorted individually in descending order of their OD volumes. For instance, for origin O1 (row-1) of matrix **Y**, the sequence of destinations (skeleton) in descending order of demand is S, W, N and E with 16, 12, 10 and 9 trips (mass), respectively (refer $\mathbf{R}_Y^1$ for matrix **Y** in Fig. 4b).

For OD matrix **Y**, the sorted set of destination IDs and the corresponding demand from nth origin is expressed as $\mathbf{R}_Y^n = (\mathbf{D}_Y^n, \mathbf{A}_Y^n) = [(D_{y_1}^n, A_{y_1}^n), ..(D_{y_i}^n, A_{y_i}^n), \cdots(D_{y_M}^n, A_{y_M}^n)]$. Here, $D_{y_i}^n$ and $A_{y_i}^n$ are the ith preferred destination and its corresponding demand value, respectively from nth origin of **Y**. Similarly, we express $\mathbf{R}_X^n = (\mathbf{D}_X^n, \mathbf{A}_X^n)$ for matrix **X**. The null pair is represented as (ε, 0). Length of the sets, $(\mathbf{D}_Y^n, \mathbf{A}_Y^n)$ and $(\mathbf{D}_X^n, \mathbf{A}_X^n)$ isM each. If$(\mathbf{D}_X^n, \mathbf{A}_X^n)$ is the result of any edit operations to $(\mathbf{D}_Y^n, \mathbf{A}_Y^n)$, then it can be written

(X) Reference Matrix

	N	E	W	S
O1	3	4	6	10
O2	7	4	5	11
O3	12	8	5	6
O4	13	7	9	6

(Y) Query Matrix

	N	E	W	S
O1	10	9	12	16
O2	17	10	13	11
O3	11	14	12	18
O4	12	13	19	15

a) Original OD matrices: Reference(**X**) and Query(**Y**)

	Origin \ Dest	(Dest., Trips) Choice 1	(Dest., Trips) Choice 2	(Dest., Trips) Choice 3	(Dest., Trips) Choice 4
R_X^1	O1	(S,10)	(W,6)	(E,4)	(N,3)
R_X^2	O2	(S,11)	(N,7)	(W,5)	(E,4)
R_X^3	O3	(N,12)	(E,8)	(S,6)	(W,5)
R_X^4	O4	(N,13)	(W,9)	(E,7)	(S,6)

	Origin \ Dest	(Dest., Trips) Choice 1	(Dest., Trips) Choice 2	(Dest., Trips) Choice 3	(Dest., Trips) Choice 4
R_Y^1	O1	(S,16)	(W,12)	(N,10)	(E,9)
R_Y^2	O2	(N,17)	(W,13)	(S,11)	(E,10)
R_Y^3	O3	(S,18)	(E,14)	(W,12)	(N,11)
R_Y^4	O4	(W,19)	(S,15)	(E,13)	(N,12)

b) Row sorted Reference (left) and Query matrices (right)

Fig. 4. Example to demonstrate Levenshtein distance application for OD matrices comparison.

as$(\mathbf{D}_Y^n, \mathbf{A}_Y^n) \Rightarrow (\mathbf{D}_X^n, \mathbf{A}_X^n)$.

4.1.1. Proposed edit operations

Compared to the traditional Levenshtein approach, the edit operations in the proposed *Levenshtein distance for OD matrices* is different in the following ways:

(a) We compute cost in each of the edit operations in terms of flows because OD demand is another attribute besides the destination IDs.
(b) Destination IDs in both the OD matrices are same, while their order varies, so we do not need any *substitution* operation.
(c) We propose additional edit operation –*absolute trips-difference* that accounts for the differences in the OD flows when the i^{th} preferred destination is same in both sorted rows.

Any edit operation towards the transformation of $\mathbf{R}_Y^n$ to $\mathbf{R}_X^n$ can be expressed as (D, A) → (D, A). Following are the possible operations:

(1) *Absolute trips-difference:* if the destination ID, D, is same in both $\mathbf{R}_X^n$ and $\mathbf{R}_Y^n$, i.e., (D, A) → (D, A), then associated cost is the absolute difference in the demand i.e. $|A - A|$.
(2) *Insertion of trips* i.e., $(\varepsilon, 0) \rightarrow (D, A)$: Here, the destination ID, D is inserted in$\mathbf{R}_Y^n$. The associated cost is the demand, A.
(3) *Deletion of trips,* i.e., $(D, A) \rightarrow (\varepsilon, 0)$: Here, the destination ID, D is deleted from $\mathbf{R}_Y^n$. The associated cost is the demand, A.

Let, $\mathbf{S} = (S_o, S_1, ..S_k \cdots S_s)$ be the sequence of edit operations to transform $\mathbf{R}_Y^n \Rightarrow \mathbf{R}_X^n$, and the cost (in terms of trips) associated with each edit operation are$\beta = (\beta_0, \beta_1. .\beta_k. .\beta_s)$, respectively. Then, *Levenshtein distance for OD matrices* computed for n^{th} row (LOD_n) is the minimum total cost needed for $\mathbf{R}_Y^n \Rightarrow \mathbf{R}_X^n$. (Eq. (13)). As the minimum cost is required, so it is an optimization problem. Section 4.1.2 demonstrates two possible combinations of edit operations for $\mathbf{R}_Y^2 \Rightarrow \mathbf{R}_X^2$ of the example shown in Fig. 4(b).

While the LOD_n is an absolute comparison of n^{th} rows, we can have a relative comparison by normalising LOD_n with the trip productions (sum of origin flows) for n^{th} row from both matrices. This normalised version of LOD_n, is referred as $NLOD_n$ and is expressed in Eq. (14). The values of $NLOD_n$ is between a scale of 0 and 1. If the number of origins is N, then we have N values of LOD_n and $NLOD_n$.

The overall comparison between the OD matrices is obtained through mean Levenshtein distance i.e. LOD is the average of all LOD_n values, and the mean normalised Levenshtein distance i.e. NLOD is the average of all $NLOD_n$ as expressed in Eq. (15) and Eq. (16), respectively.

$$\mathrm{LODn}(\mathbf{R}_Y^n, \mathbf{R}_X^n) = \min_S\left(\sum_{k=0}^{k=s} \beta_k\right) \tag{13}$$

$$\mathrm{NLODn}(\mathbf{R}_Y^n, \mathbf{R}_X^n) = \frac{\mathrm{LOD}_n(\mathbf{R}_Y^n, \mathbf{R}_X^n)}{\left(\sum_{j=1}^{j=M} A_{x_j}^n + \sum_{i=1}^{i=M} A_{y_i}^n\right)} \tag{14}$$

$$\mathrm{LOD}(\mathbf{Y}, \mathbf{X}) = \frac{\sum_{n=1}^{n=N} \mathrm{LOD}_n(\mathbf{R}_Y^n, \mathbf{R}_X^n))}{N} \tag{15}$$

$$\mathrm{NLOD}(\mathbf{Y}, \mathbf{X}) = \frac{\sum_{n=1}^{n=N} \mathrm{NLOD}_n(\mathbf{R}_Y^n, \mathbf{R}_X^n))}{N} \tag{16}$$

To explain the possible combinations of the edit operators, an example is presented in the following section.

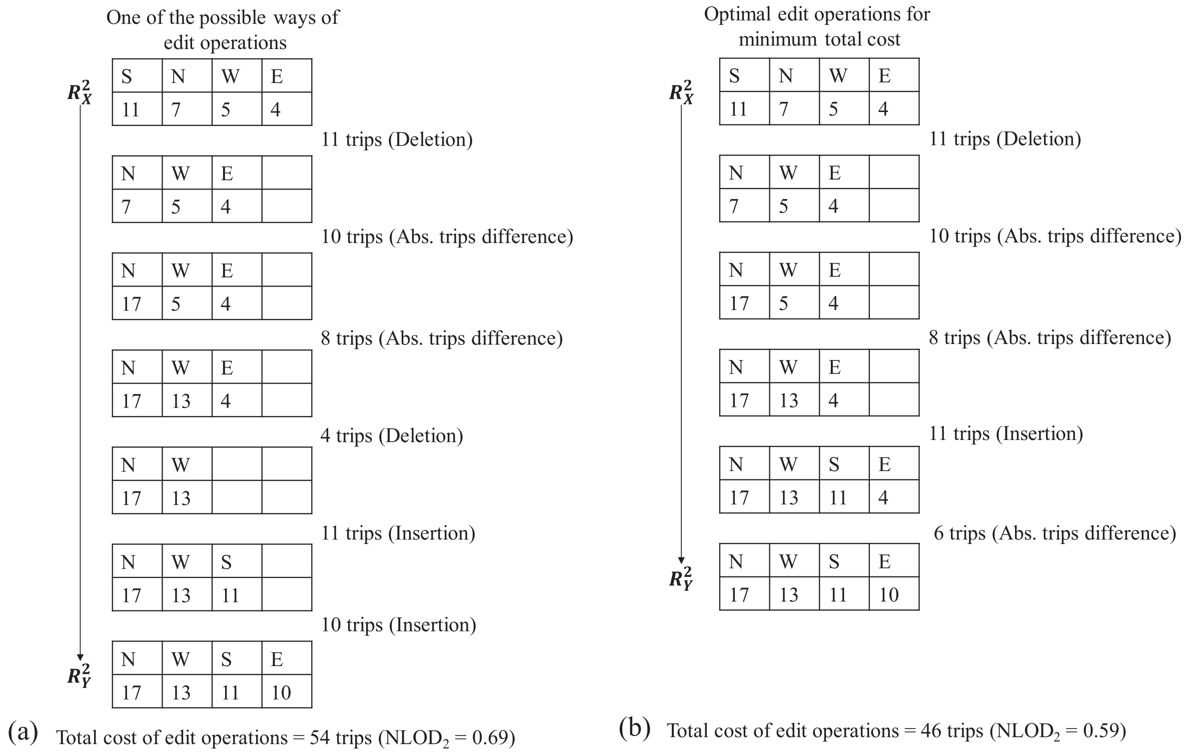


Fig. 5. A possible combination of edit operations vs minimum total cost of edit operations.

4.1.2. $R_Y^n \Rightarrow R_X^n$: An example

Consider the sorted rows, R_Y^2 and R_X^2 from the previous example (refer Fig. 4b). The transformation of $R_Y^2 \Rightarrow R_X^2$ can be achieved by multitude of edit operation combinations. Two such possible combinations are presented in Fig. 5. The cost of operation in Fig. 5a is higher than that of Fig. 5b. Operations in Fig. 5a and Fig. 5b have a total cost of 54 trips ($NLOD_2 = 54/(27 + 51) = 0.69$) and 46 trips ($NLOD_2 = 46/(27 + 51) = 0.59$), respectively.

4.1.3. Algorithm to compute Levenshtein distance for OD matrices

The *Algorithm 2* presented in Table 2 demonstrates the proposed approach to estimate Levenshtein distance for comparing OD matrices **Y** and **X** each of size M*M. LOD_n and $NLOD_n$ are estimated for each origin (n = 1 to M) individually that is later aggregated to estimate LOD and NLOD, respectively.

Table 2
Algorithm 2 demonstrating computation of the *Levenshtein distance for OD matrices.*

For each origin n (n = 1 to N)$LOD_n = 0$
Define $(\mathbf{D}_X^n, \mathbf{A}_X^n)$ and $(\mathbf{D}_Y^n, \mathbf{A}_Y^n)$ where,
$(\mathbf{D}_X^n, \mathbf{A}_X^n) = [(D_{x_1}^n, A_{x_1}^n), ..(D_{x_j}^n, A_{x_j}^n), \cdots(D_{x_M}^n, A_{x_M}^n)]$ and $(\mathbf{D}_Y^n, \mathbf{A}_Y^n) = [(D_{y_1}^n, A_{y_1}^n), ..(D_{y_i}^n, A_{y_i}^n), \cdots(D_{y_M}^n, A_{y_M}^n)]$,
Create an empty matrix **L** of size (M + 1) * (M + 1), where the row header and column header corresponds to$(\mathbf{D}_Y^n, \mathbf{A}_Y^n)$ and$(\mathbf{D}_X^n, \mathbf{A}_X^n)$, respectively (refer Fig. 6).
Assign cumulative flows $[A_{x_1}^n, \cdots, \sum_1^j A_{x_j}^n, \cdots, \sum_1^M A_{x_j}^n]$ and $[A_{y_1}^n, \cdots, \sum_1^i A_{y_i}^n, \cdots, \sum_1^M A_{y_i}^n]$ to the first row and column, respectively
for j = 1 to M
for i = 1 to M
Estimate cost as$C_{I,j} = \begin{cases} |A_{x_j}^n - A_{y_i}^n| \; if D_{x_j}^n = D_{y_i}^n \\ |A_{x_j}^n + A_{y_i}^n| if D_{x_j}^n \neq D_{y_i}^n \end{cases} \forall\, i = \{1, M\} and j = \{1, M\}$
Set the cell $\mathbf{L}(i,j) = \min(\mathbf{L}(i-1,j) + A_{y_i}^n, \mathbf{L}(i,j-1) + A_{x_j}^n, \mathbf{L}(i-1,j-1) + C_{i,j})$ Where:
$\mathbf{L}(i-1,j) + A_{y_i}^n$ represents the cell value immediately above the current cell plus $A_{y_i}^n$
$\mathbf{L}(i, j-1) + A_{x_j}^n$ represents the cell value immediately to the left of current cell plus$A_{x_j}^n$.
$\mathbf{L}(i-1, j-1) + C_{i,j}$ represents the cell value immediately in diagonal above and to the left of current cell plus the cost $C_{i,j}$.
The local Levenshtein distance i.e. $LOD_n = \mathbf{L}(M+1, M+1)$ and local Normalised Levenshtein distance is $NLOD_n = LOD_n/(\sum_{j=1}^{j=M} A_{x_j}^n + \sum_{i=1}^{i=M} A_{y_i}^n)$.
Mean Levenshtein distance values are computed as $LOD = (\sum_{n=1}^{n=N} LOD_n)/N$ and $NLOD = (\sum_{n=1}^{n=N} NLOD_n)/N$.

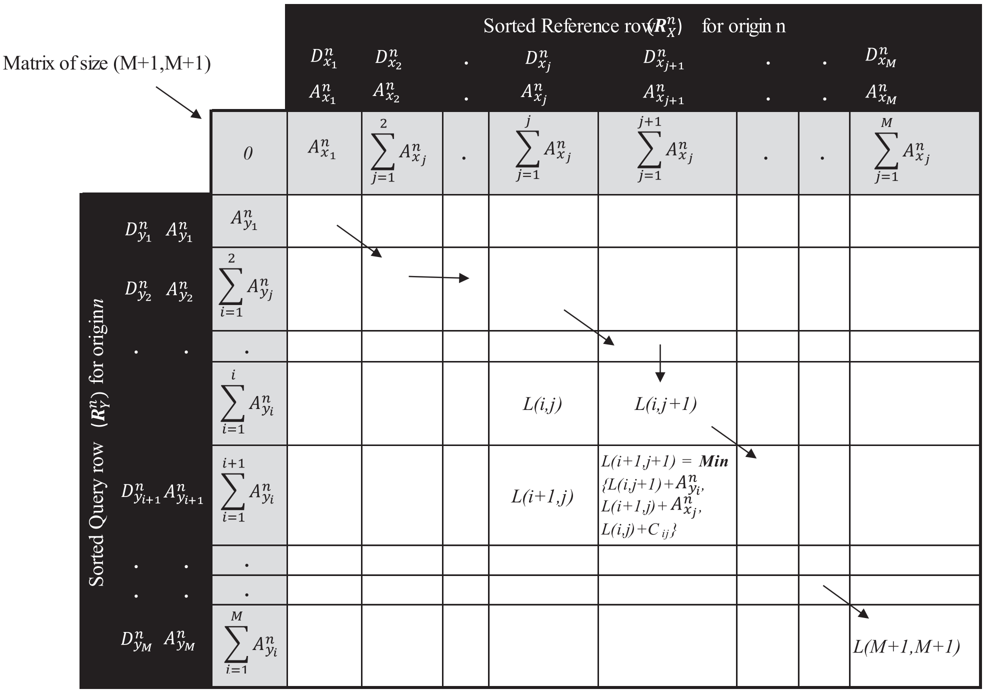


Fig. 6. Matrix demonstration of Algorithm 2 for sorted rows comparison.

Note that when destination IDs are different, the total cost ($C_{i,j}$) in *Algorithm 2* is estimated as the sum of demands i.e. $|A_{x_i}^n + A_{y_i}^n|$. One can argue why the cost is not the average of the two demands. Average is always lower than summation, and to be conservative we propose to have a higher cost for different destination IDs.

The self-explanatory matrix demonstration of *Algorithm 2* is illustrated in Fig. 6. Similar to traditional approach, the numbering of rows and columns of matrix (**L**) commence with 0. However, to account for the OD flows, in *Algorithm 2*, we replace the first row and column with cumulative sum of trips distributed to the destinations of sorted reference and query rows.

Similar to the traditional Levenshtein approach, we have multiple possible paths (i.e. different combination of arrows) to arrive at the final **L**(M + 1,M + 1). Each path is a combination of editing operations represented through the following moves on the matrix grid: downward movement along the diagonal is for *absolute-trips difference* operation, eastward movement is for *deletion* operation, and vertical downward movement is for the *insertion* operation.

The application of *Algorithm 2* on $R_Y^2 \Rightarrow R_X^2$ on the example in Fig. 4 is presented in Fig. 7. Here, the direction of arrows points towards the optimal combination of edit operations for minimum total cost. The last cell of matrix **L** i.e. L(5,5) is the value of LOD_2 = 46 trips. This value is same as the operations shown in Fig. 5b; that is, first we have a deletion operation (horizontal arrow pointing right); two consecutive absolute trips-difference operations (diagonal arrows pointing down); one insertion operation (vertical arrow pointing down); and finally one more absolute trips-difference operation (diagonal arrow pointing down).

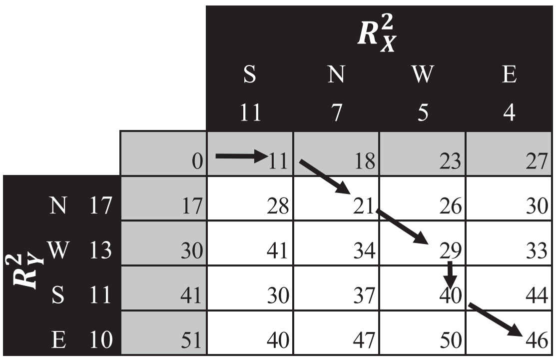


Fig. 7. $R_Y^2 \Rightarrow R_X^2$ using Algorithm 2.

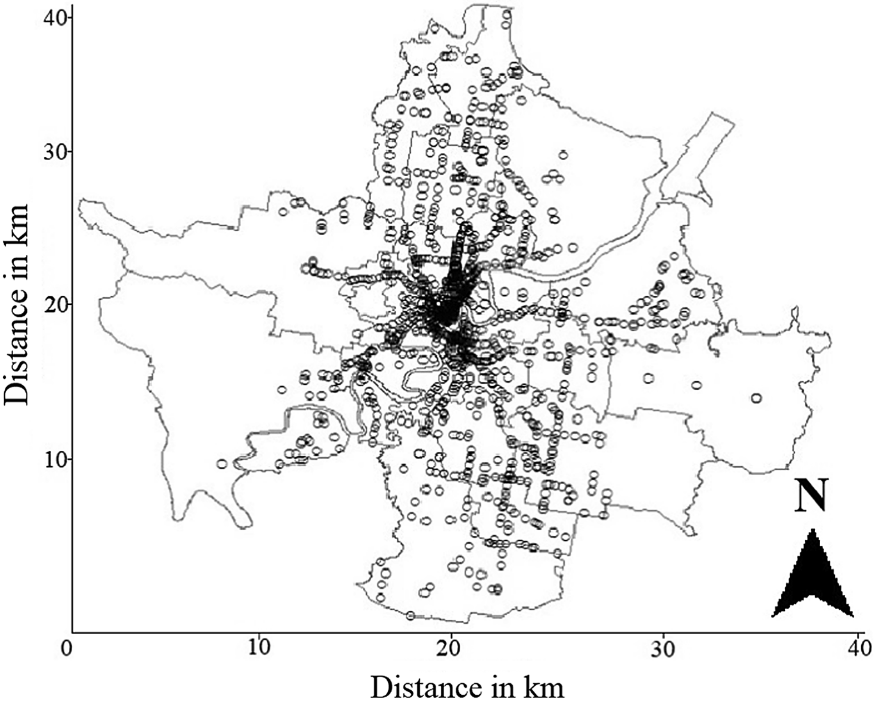


Fig. 8. Location of Bluetooth scanners within SA3 zones of the BCC region.

5. Sensitivity analysis

This section tests the robustness of NLOD. A statistical measure is considered to be robust if it is sensitive to both the 'skeleton' and the 'mass' of the OD matrix. The study site, data and the design of experimental set up are briefly discussed below.

Study site and data: The Brisbane City Council (BCC) region is the study area, and the data consists of Bluetooth observations observed from more than 845 Bluetooth scanners (Bhaskar and Chung, 2013) located within the study region (refer Fig. 8). The reference OD matrix, **X** is the Bluetooth based OD matrix (20 × 20) observed on Monday, 7th March 2016 (refer to Behara et al. (2018)). The OD pairs are represented at statistical area (SA)-3 level (QGSO, 2016). More details about the development of Bluetooth OD matrix from Bluetooth observations is described by Michau et al. (2014). The query OD matrices are developed specific to the experiments and are obtained by perturbing the reference OD matrix. The details for which are provided in Section 5.1.

Experiments: We generally encounter three possible situations as outlined in Table 3 while comparing OD matrices. The situations correspond to different combinations of skeleton and mass of OD matrices to be compared. Here, for situation-1 OD matrices are exactly the same (same skeleton and same mass); for situation-2, OD matrices has different mass component but same skeleton; for situation-3 OD matrices have different skeleton and different mass.

It would be interesting to see how the structure component of NLOD performs in all these situations. However, NLOD formulation does not have an explicit expression for its structural component. This can be addressed by normalising the OD matrices where the effect of OD flows (mass) on the estimates of NLOD can be nullified and the corresponding NLOD values should quantify the structural difference. Therefore, we term 'NLOD structural component' as the value corresponding the NLOD estimate on normalised OD matrices. The overall NLOD value is the estimate on the actual OD matrices.

We design following two experiments (as shown in Table 3) where each experiment tests the sensitivity of NLOD and its structural component:

(1) Uniform scaling effect- Here, the query matrices have the same skeleton/structure as that of reference OD matrix while the mass/OD flows may vary. If the uniform scale factor is one, then both OD matrices are exactly similar (*situation-1*), for other scaling

Table 3
Experimental set up for sensitivity analysis of NLOD.

	Skeleton	Mass	Experiments	Metric
Situation-1	Same	Same	Uniform scaling effects	NLOD and NLOD structural component
Situation-2	Same	Different		
Situation-3	Different	Different	Random scaling effects	

values it corresponds to *situation-2*.

(2) Random scaling effect- Here, the skeleton/structure and mass/OD flows vary between query and reference OD matrices. Thus, abovementioned *situation-3* is tested here.

In the following sections we first discuss the design of experimental set up for both experiments, and then present the results of sensitivity analysis.

5.1. Experimental set up for the sensitivity analysis of NLOD

5.1.1. Criteria for uniform scaling effects

Here, sensitivity of NLOD and its structural component are tested for different uniform scaling percentages. The reference OD matrix, $\mathbf{X}$ is compared with $\mathbf{Y_i}$ where $\mathbf{Y_i} = \varphi * \mathbf{X}$, and φ is chosen from [0.1, 0.2, 0.3…1.9, 2.0]. NLOD is a robust metric if:

(a) The NLOD structural component (i.e. NLOD on normalised OD flows) is zero for any value of φ.
(b) NLOD (i.e., NLOD applied on actual OD matrices) value is zero for $\varphi = 1$ and should increase as φ deviated from unity.

5.1.2. Criteria for random scaling effects

Here, sensitivity of NLOD and its structural component are tested for four different cases of random scaling percentages i.e. $\psi = [5\%, 10\%, 15\%, 20\%]$ over three types of demand scenarios. These demand scenarios are generally encountered in traffic demand modelling (refer Djukic et al. (2015)) and are as follows:

(1) Outdated surveys (low demand),
(2) The best historical estimates (medium demand), and
(3) Congested traffic conditions (high demand).

Note that the scenarios are named as low (l), medium (m), and high (h) in reference to the total daily demand on the network, and do not refer to the demands of individual OD pairs. In each case of the demand scenario, reference OD ($\mathbf{X}$) is compared with 100 replications of query ODs ($\mathbf{Y}$). The details of the demand scenarios are as follows:

Low demand scenario: Here, NLOD compares $\mathbf{X}$ and $\mathbf{Y}^{l}_{i,\psi}$ where$\mathbf{Y}^{l}_{i,\psi} = \mathbf{X} * (0.60 + \psi * \text{rand}[0, 1])$ and i$\in$ [1, 100]. For instance, if $\psi = 20\%$, then $\mathbf{Y}^{l}_{i,\psi}$ ranges between 60% and 80% of $\mathbf{X}$, and similarly for other values of ψ.

Medium demand scenario: Here, NLOD compares $\mathbf{X}$ and $\mathbf{Y}^{m}_{i,\psi}$ where$\mathbf{Y}^{m}_{i,\psi} = \mathbf{X} * (0.80 + \psi * \text{rand}[0, 1])$ and i$\in$ [1, 100]. For instance, if $\psi = 20\%$, then $\mathbf{Y}^{m}_{i,\psi}$ ranges between 80% and 100% of $\mathbf{X}$, and similarly for other values of ψ.

High demand scenario: Here, NLOD compares $\mathbf{X}$ and $\mathbf{Y}^{h}_{i,\psi}$ where $\mathbf{Y}^{h}_{i,\psi} = \mathbf{X} * (1.05 + \psi * \text{rand}[0, 1])$ and i$\in$ [1, 100]. For instance, if $\psi = 20\%$, then, $\mathbf{Y}^{h}_{i,\psi}$ ranges between 105% and 125% of $\mathbf{X}$ and, similarly for other values of ψ. The OD matrices for the high demand scenario represent demand during congested periods. Say, high daily demand can be witnessed during major events, such as Commonwealth games etc.

The condition for NLOD and its structural component to be robust towards random effects is that both should reflect the random differences that exist between the OD matrices; that is, the distance values should increase/decrease with increase/decrease in the magnitude of random scaling effects for all three demand scenarios.

5.2. Results of sensitivity analysis

5.2.1. Results of uniform scaling effects:

The results of uniform scaling are presented in Fig. 9. It can be seen that

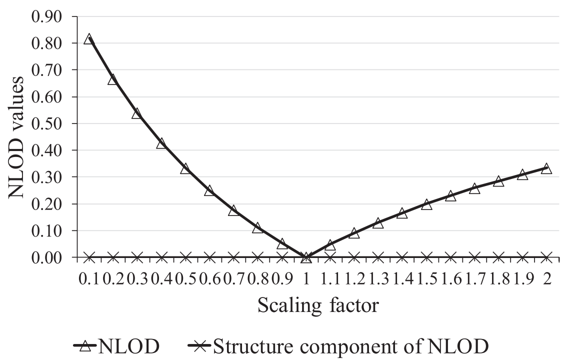


Fig. 9. Results of uniform scaling effects.

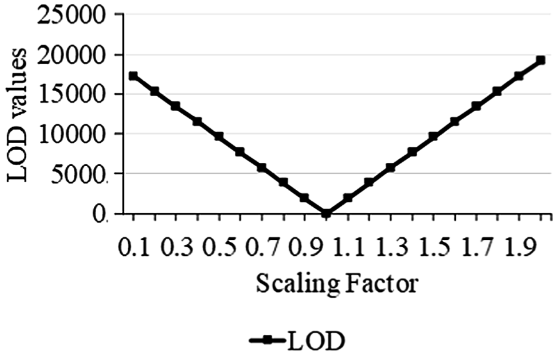


Fig. 10. LOD values for uniform scaling effects.

(a) The structure component of NLOD (line with crosses in Fig. 9) is zero for all values of the scaling factor.
(b) The NLOD values (line with triangles in Fig. 9) for different scaling factors are non-linear where the curve is concave for $\varphi < 1$, with value of zero for $\varphi = 1$, and convex for greater than 1φ. The LOD curve for the uniform scaling factor is piecewise linear and is presented in Fig. 10. NLOD is estimated by normalising the LOD values for each row (see Eq. (14)), which attributes to its non-linearity.

The above satisfies the robustness condition outlined in Section 5.1.2, and therefore we can conclude that NLOD is robust towards uniform scaling effects.

5.2.2. Results of random scaling effects

The plot shown in Fig. 11a demonstrates that as the magnitude of random fluctuations increase, the distance measure by NLOD and its structural component also increase. For instance, the median values for NLOD, as illustrated in Fig. 11a, for low demand scenario are 0.14, 0.24, 0.31 and 0.39 for $\psi = 5\%, 10\%, 15\%, 20\%$, respectively. The results showed similar increasing trend for all three demand scenarios (Fig. 11a and b). Thus, the results prove that NLOD and its structural component are robust towards random scaling effects.

It can be concluded from the above analysis that NLOD is sensitive to the structural differences within the OD matrices and is a robust statistical measure. Following the sensitivity test, we conducted a real case study analysis to demonstrate the practical application of NLOD.

6. Applying NLOD on real Bluetooth based OD matrices from Brisbane city council region

For the study area shown in Fig. 8, we consider OD matrices developed from Bluetooth observations during the period 7–13th March 2016, and compare each daily OD matrix with the reference Monday OD matrix (i.e. 7th March 2016) as shown in Fig. 12. Comparison among different combinations of OD matrices is shown in Table 4 using NLOD matrix.

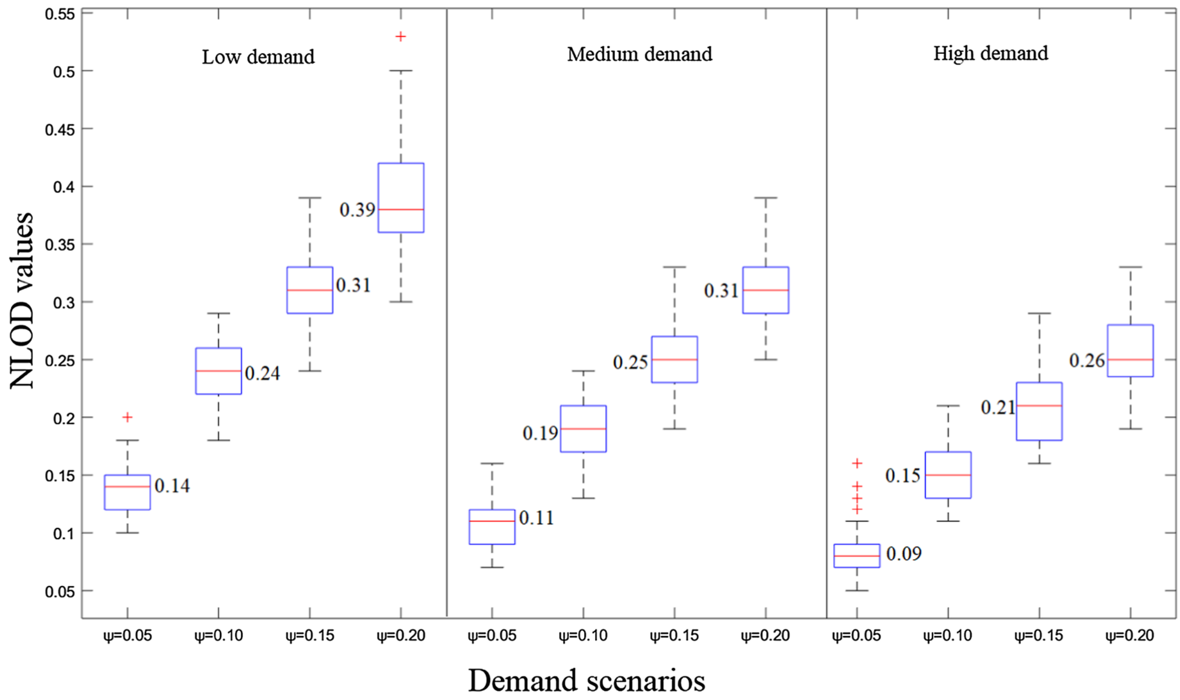
The following are the findings from this analysis:

(1) In Fig. 12, the comparison of Monday OD matrix with itself resulted in zero NLOD distance.
(2) The OD matrices during the weekdays are structurally more similar to the Monday OD matrix than the weekends (Saturday and Sunday). This is reflected in smaller NLOD values ($<=0.0769$) for weekdays in Fig. 12.
(3) Higher NLOD comparison of Monday with Saturday (0.1603) and Sunday (0.2742) in Fig. 12 indicate that the travel patterns during weekends are very different from that of weekdays. The travel patterns during Saturday and Sunday are also different from each other. This is because the percentage of work-related trips occurring during Saturdays is higher as compared to Sundays (Bhat and Misra, 1999).
(4) A closer observation of NLOD values in Table 4 reveal that OD matrices from consecutive weekdays are structurally similar in nature; for instance, the nearest neighbours are Monday-Tuesday, Tuesday-Wednesday, and Thursday-Friday.
(5) The empirical observations of the NLOD values in Table 4 indicates that the NLOD satisfies the following mathematical properties of distance measure (d) (Theodoridis and Koutroumbas, 2009):

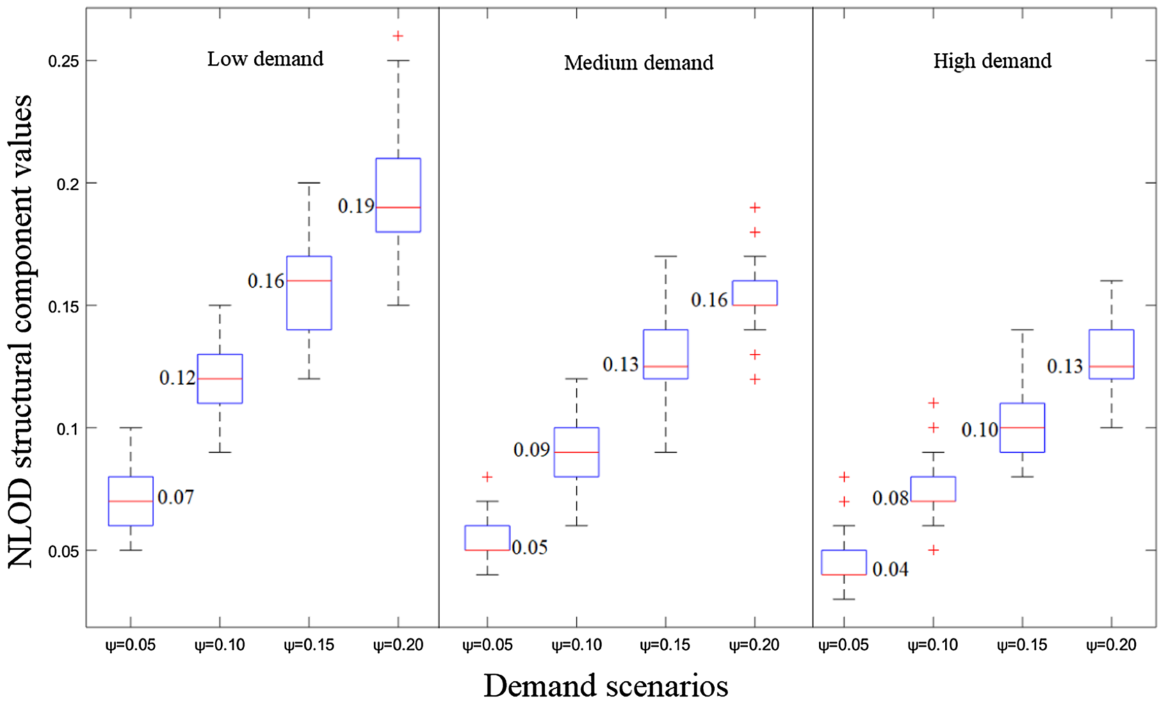
If x and y are datapoints in a dataset X, then:

1. $d(x, x) = d_o, \forall x \in X$;
2. $d(x, y) = d(y, x), \forall x,y \in X$; and $d(y, x) = d_o$ if and only if $x = y$;
3. $d(x, z) \leq d(x, y) + d(y, z), \forall x, y, z \in X$ (this property is also referred as triangular inequality)

We can see that the above three properties are satisfied in Table 4. Here X = {Mon, Tue, Wed, Thu, Fri, Sat}:



(a)



(b)

Fig. 11. Results of random scaling effects for (a) NLOD and (b) NLOD's structure component.

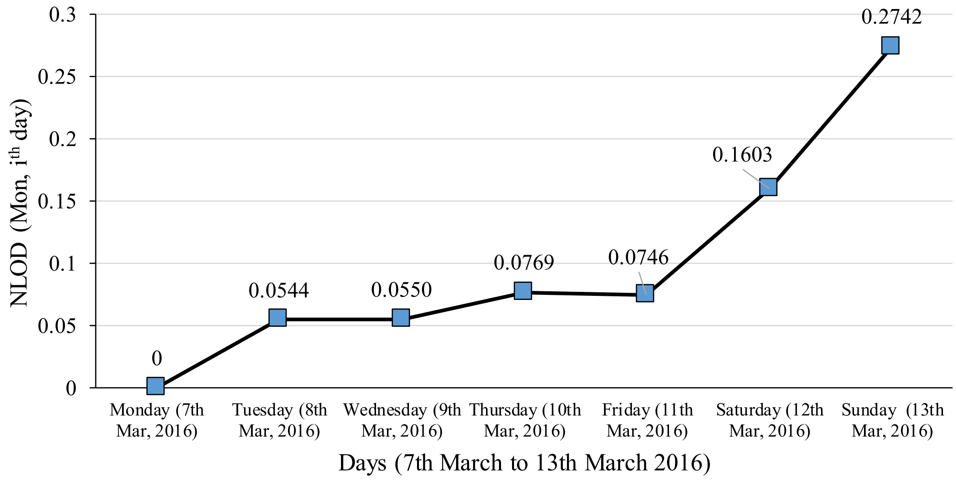


Fig. 12. NLOD comparison of Monday OD with ODs from all days in a week.

Table 4
NLOD matrix comparison for all OD combinations.

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Mon	0	0.0544	0.0550	0.0769	0.0746	0.1603	0.2742
Tue	0.0544	0	0.0345	0.0543	0.0519	0.1700	0.2934
Wed	0.0550	0.0345	0	0.0495	0.0506	0.1673	0.2953
Thu	0.0769	0.0543	0.0495	0	0.0453	0.1774	0.3067
Fri	0.0746	0.0519	0.0506	0.0453	0	0.1736	0.3087
Sat	0.1603	0.1700	0.1673	0.1774	0.1736	0	0.1777
Sun	0.2742	0.2934	0.2953	0.3067	0.3087	0.1777	0

a. The diagonal values of the matrix in Table 4 represents NLOD(x, x) and is equal to zero ($= d_o$); thus, satisfying $d(x, x) = d_o, \forall x \in X$;
b. The matrix in Table 4 is symmetrical, NLOD(x, y) = NLOD(y, x) $\forall x,y \in X$; and NLOD (y, x) = d_o if and only if x = y;
c. Triangular inequality is also satisfied: NLOD(x, z) ≤ NLOD(x, y) + NLOD(y, z), $\forall x, y, z \in X$. For instance, NLOD(Mon, Sun) ≤ NLOD(Mon, Tue) + NLOD(Tue, Sun); that is, 0.2742 ≤ 0.3478 (i.e. 0.0544 + 0.2934).

7. Discussion

The results of sensitivity analysis proved that the proposed NLOD approach is a robust statistical measure and is readily deployable to practical applications that involve OD matrices comparison.

The results of uniform scaling effects show that the structure component of NLOD does not change when the preference of destinations is the same in both OD matrices. This part of the analysis (i.e. uniform scaling effects) is more practically relevant to the application of NLOD where the preference of destinations between target OD and estimated OD (in OD estimation methods) or between ODs of similar travel patterns (say, from Monday and Tuesday) hardly change while they vary in the magnitude of OD flows.

Any structural (dis)similarity measure should be sensitive to random structural differences in the OD flows. The results of the random scaling effects illustrate the same phenomenon where higher NLOD values are observed for higher random variations in OD flows. This feature of NLOD is useful in checking the convergence/divergence of OD optimisation algorithms where the solution algorithm converges as the random structural differences between estimated and target ODs are minimised.

Additionally, the practical demonstration on Bluetooth OD matrices of the BCC region further highlighted the potential of NLOD and empirically proved that it satisfies the mathematical properties of a distance metric. Prior to the analysis it was expected that OD matrices during the weekends are different from that of weekdays due to differences in the daily activity-travel patterns. The NLOD is able to distinguish these structural differences because the destination preferences and OD flows distribution are different during weekends and weekdays. Also, NLOD technique is able to capture the subtle structural differences within the weekdays', and within the weekends' travel patterns.

8. Conclusion

The study proposes a novel technique – *normalised Levenshtein distance for OD matrices* (NLOD) for the structural comparison of OD matrices. We establish the following definition in this paper, *OD structure is the skeletal framework of the OD matrix where the skeleton is expressed as the preference/arrangement of the destinations from each origin, and the OD flows corresponding to the structure (skeleton) of the*

OD is termed as mass. A holistic comparison of OD matrices should include both deviations of individual OD flows and the structural (dis)similarity between the OD matrices using a robust statistical measure.

The traditional measures compare OD matrices through the deviations of individual OD flows, and thus fail to account the OD matrix structural information. Very limited studies in the past focussed on developing such structural (dis)similarity measures and they hardly define OD matrix ‘structure’. The comparison among different structural (dis)similarity metrics and the proposed metric is summarised in Table 5 of Appendix section.

The proposed NLOD is based on the traditional Levenshtein distance (widely applied in computational linguistics and computer science) that quantifies the differences between two sequences (strings). Levenshtein distance and other similar metrics such as sequence alignment method were previously used in the transport applications like comparison of sequences of activity-travel patterns. However, the technique has never been extended for OD matrices comparison. This is because the OD matrices are two-dimensional arrays consisting of OD flows between different origin and destination pairs, and due to which direct application of such traditional techniques is not possible. The proposed NLOD extends the capability of traditional Levenshtein as follows:

(a) We compute cost in each of the edit operations in terms of flows because OD demand is another attribute besides the destination IDs.
(b) We do not need any *substitution* operation because destination IDs in both OD matrices are same, while their order varies.
(c) We propose additional edit operation –*absolute trips-difference* that accounts for the differences in the OD flows when the i^{th} preferred destination is same in both sorted rows.

The sensitivity analysis and ability of NLOD to compare structural differences among OD matrices from a typical week for the BCC region proved that the proposed measure is robust statistical measure and can thus be used as another alternate metric for structural comparison of OD matrices. The empirical study conducted also indicates that the NLOD satisfies the mathematical properties of distance measure (NLOD for perfectly similar data points is zero, NLDO between x and y is same as between y and x; and NLOD satisfies triangular inequality).

In this paper, the structure of OD is defined from the perspective of trips distributed from each origin (i.e. trip production-based). However, we can also explore, in the future research, the OD structure based on trip attractions, and suggest the minimum of both as the final NLOD value between two OD matrices.

Acknowledgements

The authors are thankful to Brisbane City Council (BCC), Queensland Department of Transport and Main Roads (TMR) for providing the data and Queensland University of Technology (QUT) for supporting the research. The conclusions of this paper reflect understandings of the authors, who are responsible for the accuracy of the research findings.

Appendix A

See Table 5.

Table 5
Comparison of NLOD with other structural (dis)similarity metrics.

	MSSIM//GSSI/4D-MSSIM	Wasserstein	NLOD
Rationale behind	It is adopted from the comparison of two images. In OD matrices, the OD pairs are analogous to pixels of the image.	It is adopted from the mass transportation problem where the Wasserstein distance is the minimum cost required to transfer pile of sand (mass) to different holes on the network. For OD application, the difference of OD flows for a certain OD pair is analogous to mass, and rest of the OD pairs in the matrix are analogous to holes.	It is adopted from computational linguistics and computer science applications to quantify the dissimilarity between two sequences of strings (words). For OD application, the sequence/preference of destinations from each origin is considered analogous to the sequence of strings.
Mathematical formulation	Single mathematical expression that includes comparison of mean, standard deviation and covariance. Only 4D-MSSIM additionally computes Euclidian distance between geographical co-ordinates to identify nearby OD pairs.	It is an optimization problem to estimate minimum cost required to transform a query OD matrix into reference OD matrix considering the spatial correlations.	It is an optimization problem to estimate minimum cost required to transform each sorted row of query OD matrix into sorted row of reference OD.
Mass/OD flows and skeleton/structure comparison	Mean and standard deviation components compare masses, and covariance account for structural comparison. All three formulations are explicitly defined.	Mass and skeleton comparison are implicit in the formulation.	Mass and skeleton comparison are implicit in the formulation.
Unit of measurement	The scale of measurement is between −1 and 1. Thus no units.	It is measured either in vehicle-minutes or only in travel time units	The scale of measurement is between 0 and 1. Thus no units.

(*continued on next page*)

Table 5 (*continued*)

	MSSIM//GSSI/4D-MSSIM	Wasserstein	NLOD
Computational efficiency	MSSIM/GSSI are computationally effective as compared to Wasserstein and Levenshtein because it performs comparison in one single formulation and is not based on any optimisation problem. Comparison among MSSIM and its variants is as follows: if only one window is used then MSSIM is better than GSSI; otherwise GSSI is computationally effective because the number of fixed geographical windows are less than the number of sliding windows used in MSSIM. For 4D-MSSIM, there is one-time calculation of Euclidian distance for every OD pair in turn; otherwise it is similar to GSSI.	It is based on optimisation problem and the domain of integration is the whole space i.e. it compares each OD pair with rest of OD pairs within the matrix for an effective matrix transformation. Moreover, the travel time between each OD pairs need to be estimated. Thus, computationally it is more expensive as compared to MSSIM and Levenshtein.	It is based on optimisation problem and compares row by row. Only OD pairs within each sorted row are compared and not across the entire OD matrix. Thus, computationally it is more expensive as compared to MSSIM but less expensive as compared to Wasserstein.
Level of detail	SSIM compares at a local window level and the values have no physical meaning. On the other hand, consideration of a single window cannot explain the local structural differences (i.e. between subset of OD pairs within the matrix). Both GSSI and 4D-SSIM are able to compare OD pairs that are geographically correlated, and thus are able to compare local structural differences.	The Wasserstein metric cannot help in comparing local structural differences because the formulation yields the overall comparison.	The Levenshtein formulation is based on row to row comparison, thus can comparison of local structural differences with respect to each origin is possible.
Limitations	First, MSSIM is sensitive to the size of local window. Although GSSI and 4D-MSSIM propose to address this limitation but they need further exploration. Second, unless a single window is considered, MSSIM is sensitive to the order of OD matrix. The order of OD matrix might not guarantee geographic adjacency in MSSIM application. GSSI and 4D-MSSIM, on the other hand ensure geographical proximity using geographical boundaries and spatial proximity (through Euclidian distance), respectively. Third, the stability constants used in MSSIM/GSSI/4D-MSSIM are network specific and need further exploration.	First, the metric is computationally expensive because it is optimisation and we need additional information such as travel time between centroids. Second, the use of average travel time might not be appropriate in all situations; for instance, travel time could differ during a Monday and a Sunday between an OD pair, and use of average travel time might not be justified while comparing matrices from those day types (say, for OD clustering application).	The metric involves optimisation so computationally intensive as compared to MSSIM/GSSI/4D-MSSIM.

Appendix B. Supplementary material

Supplementary data to this article can be found online at https://doi.org/10.1016/j.trc.2020.01.005.

References

Allahviranloo, M., Recker, W., 2015. Mining activity pattern trajectories and allocating activities in the network. Transportation 1–19.

Andrienko, G., Andrienko, N., Fuchs, G., Wood, J., 2017. Revealing patterns and trends of mass mobility through spatial and temporal abstraction of origin-destination movement data. IEEE Trans. Visual. Comput. Graph. 1 1-1.

Antoniou, C., Ben-Akiva, M., Koutsopoulos, H., 2004. Incorporating automated vehicle identification data into origin-destination estimation. Transport. Res. Rec.: J. Transport. Res. Board 1882, 37–44.

Ashok, K., 1996. Estimation and Prediction of Time-Dependent Origin-Destination Flows. Institute of Technology, Massachusetts.

Ashok, K., Ben-Akiva, M.E., 2002. Estimation and prediction of time-dependent origin-destination flows with a stochastic mapping to path flows and link flows. Transport. Sci. 36 (2), 184–198.

Barceló Bugeda, J., Montero Mercadé, L., Marqués, L., Carmona, C., 2010. A Kalman-filter approach for dynamic OD estimation in corridors based on bluetooth and Wi-Fi data collection. In: 12th World Conference on Transportation Research WCTR, 2010.

Barceló, J., Montero, L., Bullejos, M., Linares, M., Serch, O., 2013. Robustness and computational efficiency of Kalman Filter estimator of time-dependent origin-destination matrices: exploiting traffic measurements from information and communications technologies. Transport. Res. Rec.: J. Transport. Res. Board 2344, 31–39.

Behara, K.N., Bhaskar, A., Chung, E., 2018. Classification of typical Bluetooth OD matrices based on structural similarity of travel patterns-Case study on Brisbane city. In: Transportation Research Board 97th Annual Meeting, Washington DC, United States.

Behara, K.N., Bhaskar, A., Chung, E., 2019. Geographical Window based Structural Similarity Index for OD Matrices Comparison [Working Paper]. Queensland

University of Technology.
Bera, S., Rao, K., 2011. Estimation of origin-destination matrix from traffic counts: the state of the art. Eur. Transp. - Trasporti Europei 49, 2–23.
Bhaskar, A., Chung, E., 2013. Fundamental understanding on the use of Bluetooth scanner as a complementary transport data. Transport. Res. Part C: Emerg. Technol. 37, 42–72.
Bhat, C.R., Misra, R., 1999. Discretionary activity time allocation of individuals between in-home and out-of-home and between weekdays and weekends. Transportation 26 (2), 193–229.
Bierlaire, M., 2002. The total demand scale: a new measure of quality for static and dynamic origin–destination trip tables. Transport. Res. Part B: Methodol. 36 (9), 837–850.
Bierlaire, M., Toint, P.L., 1995. Meuse: an origin-destination matrix estimator that exploits structure. Transport. Res. Part B: Methodol. 29 (1), 47–60.
Cantelmo, G., Viti, F., Tampêre, C., Cipriani, E., Nigro, M., 2014. Two-Step approach for correction of seed matrix in dynamic demand estimation. Transport. Res. Rec.: J. Transport. Res. Board 2466, 125–133.
Cascetta, E., 1984. Estimation of trip matrices from traffic counts and survey data: a generalized least squares estimator. Transport. Res. Part B: Methodol. 18 (4–5), 289–299.
Cipriani, E., Florian, M., Mahut, M., Nigro, M., 2011. A gradient approximation approach for adjusting temporal origin–destination matrices. Transport. Res. Part C: Emerg. Technol. 19 (2), 270–282.
Ciuffo, B., Punzo, V., 2010. Verification of traffic micro-simulation model calibration procedures: analysis of goodness-of-fit measures. In: Proceeding of the 89th Annual Meeting of the Transportation Research Record, Washington, DC.
Cools, M., Moons, E., Wets, G., 2010. Assessing the quality of origin-destination matrices derived from activity travel surveys: results from a Monte Carlo experiment. Transport. Res. Rec.: J. Transport. Res. Board 2183, 49–59.
Crawford, F., Watling, D.P., Connors, R.D., 2018. Identifying road user classes based on repeated trip behaviour using Bluetooth data. Transport. Res. Part A: Policy Practice 113, 55–74.
Day-Pollard, T., Van Vuren, T., 2015. When are origin-destination matrices similar enough? In: 94th Annual TRB Meeting, Washington DC.
Djukic, T., 2014. Dynamic OD Demand Estimation and Prediction for Dynamic Traffic Management. Delft University of Technology, TU Delft.
Djukic, T., Barceló Bugeda, J., Bullejos, M., Montero Mercadé, L., Cipriani, E., van Lint, H., Hoogendoorn, S., 2015. Advanced traffic data for dynamic od demand estimation: the state of the art and benchmark study. In: TRB 94th Annual Meeting Compendium of Papers, pp. 1–16.
Djukic, T., Flötteröd, G., Van Lint, H., Hoogendoorn, S., 2012. Efficient real time OD matrix estimation based on Principal Component Analysis. In: 2012 15th International IEEE Conference on Intelligent Transportation Systems. IEEE, pp. 115–121.
Djukic, T., Hoogendoorn, S., Van Lint, H., 2013. Reliability assessment of dynamic OD estimation methods based on structural similarity index. In: Transportation Research Board 92nd Annual Meeting.
Doblas, J., Benitez, F.G., 2005. An approach to estimating and updating origin–destination matrices based upon traffic counts preserving the prior structure of a survey matrix. Transport. Res. Part B: Methodol. 39 (7), 565–591.
Gan, L., Yang, H., Wong, S.C., 2005. Traffic counting location and error bound in origin-destination matrix estimation problems. J. Transport. Eng. 131 (7), 524–534.
Guo, D., Zhu, X., Jin, H., Gao, P., Andris, C., 2012. Discovering spatial patterns in origin-destination mobility data. Trans. GIS 16 (3), 411–429.
Gur, Y.J., 1980. Estimation of an origin-destination trip table based on observed link volumes and turning movements. Executive summary.
Heeringa, W.J., 2004. Measuring dialect pronunciation differences using Levenshtein distance. Citeseer.
Hollander, Y., Liu, R., 2008. The principles of calibrating traffic microsimulation models. Transportation 35 (3), 347–362.
Kim, H., Baek, S., Lim, Y., 2001. Origin-destination matrices estimated with a genetic algorithm from link traffic counts. Transport. Res. Rec.: J. Transport. Res. Board 1771, 156–163.
Kim, S.-J., Kim, W., Rilett, L., 2005. Calibration of microsimulation models using nonparametric statistical techniques. Transport. Res. Rec.: J. Transport. Res. Board 1935, 111–119.
Krishnakumari, P., van Lint, H., Djukic, T., Cats, O., 2019. A data driven method for OD matrix estimation. Transport. Res. Part C: Emerg. Technol. https://doi.org/10.1016/j.trc.2019.05.014.
Laharotte, P.-A., Billot, R., Come, E., Oukhellou, L., Nantes, A., El Faouzi, N.-E., 2015. Spatiotemporal analysis of Bluetooth data: application to a large urban network. IEEE Trans. Intelligent Transport. Syst. 16 (3), 1439–1448.
Levenshtein, V.I., 1966. Binary codes capable of correcting deletions, insertions, and reversals. Soviet Phys. doklady 707–710.
Lundgren, J.T., Peterson, A., 2008. A heuristic for the bilevel origin–destination-matrix estimation problem. Transport. Res. Part B: Methodol. 42 (4), 339–354.
Markou, I., Kaiser, K., Pereira, F.C., 2019. Predicting taxi demand hotspots using automated Internet Search Queries. Transport. Res. Part C: Emerg. Technol. 102, 73–86.
Michau, G., Nantes, A., Chung, E., Abry, P., Borgnat, P., 2014. Retrieving trip information from a discrete detectors network: The case of Brisbane Bluetooth detectors. In: 32nd Conference of Australian Institutes of Transport Research (CAITR 2014). University of New South Wales, Sydney, NSW.
Michau, G., Pustelnik, N., Borgnat, P., Abry, P., Nantes, A., Bhaskar, A., Chung, E., 2017. A primal-dual algorithm for link dependent origin destination matrix estimation. IEEE Trans. Signal Inform. Process. Over Networks 3 (1), 104–113.
Oliveira-Neto, F.M., Han, L.D., Jeong, M.K., 2012. Online license plate matching procedures using license-plate recognition machines and new weighted edit distance. Transport. Res. Part C: Emerg. Technol. 21 (1), 306–320.
Osorio, C., 2019. Dynamic origin-destination matrix calibration for large-scale network simulators. Transport. Res. Part C: Emerg. Technol. 98, 186–206.
Oxford, 2018. Structure, English Oxford living Dictionaries.
Pollard, T., Taylor, N., van Vuren, T., MacDonald, M., 2013. Comparing the quality of OD matrices in time and between data sources. In: Proceedings of the European Transport Conference, Frankfurt, Germany.
QGSO, 2016. Queensland statistical areas, level 3 (SA3), 2016.
Ros-Roca, X., Montero, L., Schneck, A., Barceló, J., 2018. Investigating the performance of SPSA in simulation-optimization approaches to transportation problems. Transport. Res. Proc. 83–90.
Ruiz de Villa, A., Casas, J., Breen, M., 2014. OD matrix structural similarity: Wasserstein metric. In: Transportation Research Board 93rd Annual Meeting.
Tamin, O., Willumsen, L., 1989. Transport demand model estimation from traffic counts. Transportation 16 (1), 3–26.
Tavassoli, A., Alsger, A., Hickman, M., Mesbah, M., 2016. How close the models are to the reality? Comparison of Transit Origin-Destination Estimates with Automatic Fare Collection Data, Australasian Transport Research Forum (ATRF), 38th, 2016, Melbourne, Victoria, Australia.
Theodoridis, S., Koutroumbas, K., 2009. Pattern Recognition.
Van Vuren, T., Day-Pollard, T., 2015. 256 shades of grey – comparing OD matrices using image quality assessment techniques. In: STAR Conference, Technology and Innovation Centre of the University of Strathclyde.
Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., 2004. Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. 13 (4), 600–612.
Yang, C., Yan, F., Xu, X., 2017. Daily metro origin-destination pattern recognition using dimensionality reduction and clustering methods. In: IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), 2017. IEEE, pp. 548–553.
Yang, H., 1995. Heuristic algorithms for the bilevel origin-destination matrix estimation problem. Transport. Res. Part B: Methodol. 29 (4), 231–242.
Yang, H., Iida, Y., Sasaki, T., 1991. An analysis of the reliability of an origin-destination trip matrix estimated from traffic counts. Transport. Res. Part B: Methodol. 25 (5), 351–363.
Yujian, L., Bo, L., 2007. A normalized Levenshtein distance metric. IEEE Trans. Pattern Anal. Mach. Intellig. 29 (6), 1091–1095.
Zhang, A., Kang, J.E., Axhausen, K., Kwon, C., 2018. Multi-day activity-travel pattern sampling based on single-day data. Transport. Res. Part C: Emerg. Technol. 89, 96–112.