An Adaptive Method for Map Reconstruction

İ. Kuban Altınel, Ind. Eng. Dept., Boğaziçi Univ., İstanbul, 34342 Türkiye
Necati Aras, Ind. Eng. Dept., Boğaziçi Univ., İstanbul, 34342 Türkiye
B. John Oommen, School of Computer Sci., Carleton University, Ottawa, Canada

Abstract

We consider the fundamental problem of reconstructing a map when the given data is the set of road travel distances among cities in a region. This problem is the "inverse" of the distance estimation problem, in which the goal is to determine a good estimator for inter-city road travel distances. More specifically, given the road distances among cities in a geographical area, we attempt to determine the *locations* of the cities in a two dimensional map so that the Euclidean inter-city distances approximate the actual road distances as closely as possible. The reported solutions to this problem are few, and primarily involve multi-dimensional scaling techniques. We propose an adaptive method to overcome their distinct disadvantages. The new method has been rigorously tested on different data sets obtained from various countries. Our results have also been compared with the performance of the classical multi-dimensional scaling and ALSCAL. The accuracy of the proposed method is superior. It has also two additional desirable properties. First, we can obtain point configurations even if some of the input data are missing. Second, it becomes possible to determine configurations where points representing cities are located close to the original ones.

1 Introduction

One of the classic problems studied in operations research and geographic information systems is the *distance estimation problem*, which involves the determination of a good estimator for the inter-city road travel distances evaluated as a function of the city coordinates in the map. This problem has been extensively studied because of its importance in location, distribution and logistic problems [1, 2]. The distance estimation problem has an "inverse" problem which we call the *map reconstruction problem* (referred to as *MapRecon* in the following). It is based on resolving the following question: Given the road distances among a set of cities, determine locations for cities where distance relationships are preserved. Observe that this is a rather general problem, since the computed configuration for the cities can be regarded as an abstract map reflecting a hidden structure.

From the above perspective, the reader will observe that *MapRecon* has a close connection with multidimensional scaling (MDS). The main goal of MDS is to visualize and analyze similarities or dissimilarities between "objects" using a lower dimensional representation of their feature vectors, while, in turn, simultaneously preserving the hidden structure.

The first MDS method was proposed by Torgerson [3], and historically has its roots in psychometrics where it was used to discover people's judgments on the similarity of various objects. The literature on MDS is fairly extensive. In the context of this paper, the excellent book by Borg and Groenen [4] should suffice to update any interested reader.

In terms of nomenclature, if the MDS methodology is based on measured dissimilarities (at the ratio level of measurement), it is called a *metric* MDS. In this case the aim of the exercise is to find a configuration of points that preserves the scaled dissimilarities between the objects. Typically, this scaling is achieved by a transformation function. If the MDS is based on proximities obtained at the ordinal level of measurement (e.g., judgements, perceptions), the scheme is referred to as a *nonmetric* MDS. In nonmetric MDS [5] we assume that there is a monotonic relationship between the inter-point distances and observed proximities. This, in turn, means that the reproduced distances are constrained to preserve the rank order among the similarities or dissimilarities rather than the actual distances among objects.

MapRecon is equivalent to the metric MDS where objects are cities in a given region, and dissimilarities among objects are considered to be road distances among the cities. To formally formulate the problem, we first of all observe that if we are given a matrix $\mathbf{\Delta} = [\delta_{ij} : i, j = 1, ..., n]$ of road distances between all pairs of n cities, these cities can be represented by n points in a Euclidean space E^{n-1} such that the Euclidean inter-city distances d_{ij} are equal to δ_{ij} for all i, j provided that the elements of $\mathbf{\Delta}$ satisfy the metric inequality. The problem becomes both pertinent and non-trivial if we have to represent the cities by n points in a lower dimensional subspace E^r where $r << n-1$. It is clear that in the case of *MapRecon*, $r = 2$. An analytical technique, referred to as the *classical MDS* (CMDS) [6] has been shown to be capable of exactly reproducing the road distances when they are Euclidean (i.e., $\delta_{ij} = d_{ij}$). Indeed, with a little imagination, this technique can also be employed when the road distances are approximately Euclidean. Since typical road distances can be perceived as being Euclidean distances distorted by errors, CMDS has been used to solve the map reconstruction problem [7]. However, such solutions possess a fundamental "infirmity", namely, that the reproduced inter-city distances deviate *significantly* from the true road distances, resulting in a solution with poor accuracy. In this context, accuracy is measured in terms of the discrepancy between actual road distances and their estimates obtained by MDS.

Apart from the considerations mentioned above, MDS techniques possess a few other important drawbacks. The first one is related to the number of input dissimilarities or distances that are necessary to determine a feasible and meaningful location for the objects. Traditional MDS algorithms require an $n \times n$ matrix of input dissimilarities, where n is the number of objects. This could be problematic because of the resource limitations (cost, time etc.) involving the data collection procedure. Another important fact is that a map created by *any* MDS method is usually orientation-free, that is the orientation of the points representing the objects is arbitrary. Generally speaking, this does not pose a significant problem since most of the time we are only concerned with the proximity of the reproduced points on the lower dimensional map, which enables us to "visually" discover the hidden structure in the data. There may be occasions, however, where apart from the dissimilarity that exists between the "objects", we have further information about some of the objects. This information could be physical measurements about the objects along a set of dimensions such as the latitude and longitude of cities. In such a case the objective becomes to discover a relation between the coordinates of the objects (possibly in a high dimensional space) and the positions of the points representing these objects in a lower dimensional space by preserving the proximities. Steyvers [8] gives an example of this in a psychological context. He points out that when the physical representation of the features comprising the stimuli is ignored, as in a traditional MDS technique, it becomes difficult to *interrelate* the actual positions of stimuli with the points on the map that represent the stimuli. This happens because the orientation of the points on the reproduced map turns out to be different from the original map of objects due to the possible translation, rotation and reflection phenomena.

In this work we propose a new adaptive scheme for solving the map reconstruction problem. It can find spatial representations of cities even if some of the road distance data between cities are missing. The procedure is implemented and tested on several data sets where each data set consists of inter - city road travel distances in a given region. We have compared the performance of the new method with that of the CMDS and ALSCAL [9]. We use ALSCAL that is available in SPSS 10.0 for Windows. Experimental results have shown that the performance of the new method is better in terms of the scaling error measured by the stress function. Furthermore, the final locations of the cities tend to overlap with the original locations.

In the next section CMDS and ALSCAL are briefly explained. The new adaptive method is introduced in Section 3. Section 4 includes experimental results on the performance of the new method. Finally, Section 5 concludes the paper.

2 Map Reconstruction With Classical MDS and ALSCAL

If we know the coordinates of two points in a multidimensional coordinate system, the Euclidean distance between these two points is easily calculated. If the p-dimensional coordinates of n points are given in an $n \times p$ dimensional matrix, say $\mathbf{X}$, where each column corresponds to a dimension, and each row corresponds to a point, then the Euclidean distances between all pairs of points can be calculated by using the entries of $n \times n$ dimensional matrix $\mathbf{B} = \mathbf{X}\mathbf{X}^T$ where $b_{rs} = \sum_{j=1}^{p} x_{rj} x_{sj}$. When we put the center of gravity of the points at the origin, $\sum_{r=1}^{n} x_{rj} = 0$, $j = 1, ..., p$, the sum of any row or column of $\mathbf{B}$ will be zero. Since $\mathbf{B}$ is symmetric, we can write $\mathbf{B} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1}$ or $\mathbf{\Lambda} = \mathbf{Q}^{-1}\mathbf{B}\mathbf{Q}$ where $\mathbf{Q}$ is the matrix whose columns are the eigenvectors of $\mathbf{B}$. If $\mathbf{B}$ is positive definite, all the eigenvalues are greater than zero and it follows that

$$\mathbf{B} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1} = \mathbf{Q}\mathbf{\Lambda}^{1/2}\mathbf{\Lambda}^{1/2}\mathbf{Q}^{-1} = \mathbf{X}\mathbf{X}^{T} \qquad (1)$$

Hence the coordinates of n points may be found by setting $\mathbf{X} = \mathbf{Q}\mathbf{\Lambda}^{1/2}$.

If the distances obey the triangular inequality, $\mathbf{B}$ will be positive semidefinite. As mentioned before, unless the road distances are exactly Euclidean, there will always be a discrepancy between the reproduced (Euclidean) distances and true road distances. In most cases, road distances among cities in a region are larger than the Euclidean distances because of natural barriers such as lakes, rivers, mountains etc. In order to assess the performance of any other MDS technique we have to use a goodness-of-fit criterion that measures the deviation between the Euclidean distances among points on the map and original distances among objects. One of the most widely used measures is the stress function. There are different stress functions in the literature, normalized stress, raw stress, Kruskal stress and S-stress. We will use the Kruskal stress (also known as "Stress formula 1") which is provided in the output of ALSCAL. It was first proposed by Kruskal [11] in 1964 when he coined the term STRESS (*st*andardized *re*sidual *s*um of *s*quares). The formula for the stress is as follows:

$$\sqrt{\frac{\sum_{i<j}\left(f\left(\delta_{ij}\right) - d_{ij}\right)^2}{\sum_{i<j} d_{ij}^2}} \qquad (2)$$

In this formula d_{ij} refers to the Euclidean distance between points i and j on the map, $\|\mathbf{x}_i - \mathbf{x}_j\|$, where $\mathbf{x}_i$ and $\mathbf{x}_j$ are the coordinates of points i and j. $f\left(\delta_{ij}\right)$ is a linear transformation function. Note that when the original distances are preserved perfectly for all city pairs i and j, then $f\left(\delta_{ij}\right) = d_{ij}$ for all i, j and the stress is zero. Hence, a smaller stress value indicates a better representation. In the case of CMDS, the function f is the identity function, i.e., $f\left(\delta_{ij}\right) = \delta_{ij}$. In such a case the stress function becomes

$$\sqrt{\frac{\sum_{i<j}\left(\delta_{ij} - d_{ij}\right)^2}{\sum_{i<j} d_{ij}^2}} \qquad (3)$$

In other words, they are absolute MDS techniques. ALSCAL, on the other hand, uses a linear transformation function $f\left(\delta_{ij}\right) = \alpha\delta_{ij}$ for scaling the original distances.

In Section 4, different methods will be compared in terms of STRESS given with formula (3).

3 The New Method

Our aim is to determine a two-dimensional configuration for a set of n points in such a way that the inter-point Euclidean distances approximate the inter-city road distances as closely as possible. To achieve this, we will not use any matrix-based computations. Rather, we will perform operations on the points that "resemble" the operations done in vector quantization (VQ) and Kohonen's Self-Organizing Maps (SOM). As a consequence the new method can be said of having a "real time" flavor.

Rather than representing the entire data in a compressed form using estimated parameters, VQ [10] opts to represent the data in the actual feature space by compressing the information using a "small" set of vectors, called the code-book vectors. These code-book vectors are migrated in the feature domain so that they collectively represent the distribution under consideration. In both VQ and the SOM the polarizing algorithm is repeatedly presented with a point i from the set of points of a particular class. Each point i is represented by its feature vector $\mathbf{u}_i$. The neurons attempt to incorporate the topological information which is present in $\mathbf{u}_i$. This is done as follows. First of all, the closest neuron to i at time t, j^*, is determined (also called the "winner"). It is represented by the vector $\mathbf{y}_{j^*}(t)$. Here t is the discretized (synchronized) time index. This neuron and a group of neurons in its neighborhood, $B_{j^*}(t)$, are now moved in the direction of $\mathbf{u}_i$. The set $B_{j^*}(t)$ is called the "bubble of activity" at time t. The actual migration of the neurons is achieved by rendering the new point $\mathbf{y}_j(t+1)$ representing neuron j, to be a convex combination of $\mathbf{u}_i$ and $\mathbf{y}_j(t)$, $j \in B_{j^*}(t)$. The bubble of activity, $B_{j^*}(t)$, is the parameter which makes VQ differ from the SOM. Indeed, if the size of the bubble is always set to be zero, only the closest neuron is migrated, yielding a VQ scheme. However, in the SOM, the nearest neuron and the neurons within the bubble are also migrated, and it is *this* widened migration process which permits the algorithm to be both *topology preserving* and self-organizing.

Elegant though they are, VQ and SOM methods cannot be directly applied to the *MapRecon* problem. The reasons for this are the following. First of all, unlike in the fields of VQ and SOM, there is no question of presenting a city (a data point) to the network because, there is no available data point. All that is available is the set of inter-city road travel distances. Secondly, as a consequence of the latter, there is no question of a "winning neuron" which wins the competition. This, further disallows the entire issue of defining "neighbor neurons" and a "bubble of activity".

NeuroMapRecon operates by representing each city by a neuron in a two dimensional space. As a preprocessing step, the elements δ_{ij} of the distance matrix are normalized by dividing them by the largest element in the matrix. The initialization of the neurons are performed by assigning them weights (coordinates) that are uniformly distributed on the unit square (i.e., in the interval $[0, 1]$ along the x and y dimensions). The migration occurs as follows.

At each iteration t of the new method, a random pair of cities (i, j) is selected from among n cities. The road distance between these two cities is compared with the estimate obtained by calculating the Euclidean distance $d_{ij}(t)$ between the weights of neurons i and j representing cities i and j. If the distance between neurons i and j[1] is less than the road distance between cities i and j, i.e., $d_{ij}(t) < \delta_{ij}$, the neurons are clearly "too close for comfort". To compensate for this, the weights of the neurons are updated. To put it differently, the neurons are moved away from each other along the line connecting them by an amount proportional to the difference between $d_{ij}(t)$ and δ_{ij}. The proportionality factor is a parameter called the "step size", $\mu(t)$. As a result, both neuron i and j are moved by an amount equal to $\mu(t)\left(\delta_{ij} - d_{ij}(t)\right)/2$ so that the distance between the neurons increases by $\mu(t)\left(\delta_{ij} - d_{ij}(t)\right)$. The value of $\mu(t)$ is reduced once all $n(n-1)/2$ pairs of cities are presented twice and the corresponding neurons are updated as it is traditionally done in VQ and SOM. One such iteration is referred to as an cycle of the new method. In our implementation the initial value of $\mu(t)$ is equal to unity and it is decreased linearly according to the following rule: $\mu(t) = 1 - t/1000$. If the distance between neurons i and j is too far apart, the weights of the neurons are updated by an amount equal to $\mu(t)\left(d_{ij}(t) - \delta_{ij}\right)/2$ such that the distance between neurons i and j and the corresponding road distance come closer to each other by $\mu(t)\left(d_{ij}(t) - \delta_{ij}\right)$.

Note that we do not use the diagonal elements $\delta_{ii} = 0$ in the updates, and that in each cycle every pair is introduced twice to the algorithm in order to increase the rate of convergence. The termination criterion is based on the improvement in the stress of the configuration. If the reduction of the stress value in two consecutive cycles turns out to be less than a predetermined small value ϵ, the algorithm terminates and the current configuration of the neurons is accepted as the solution. The steps of the method are given below formally.

[1] The distance between two neurons is defined as the Euclidean distance between the weights of these neurons.

Input: Scaled distance matrix $\mathbf{\Delta}$ and $\epsilon = 10^{-6}$
Output: Spatial representation of cities in E^2

0 Generate n neurons with $\mathbf{y}_i(0) \in (0, 1)$,
$1 \le i \le n stress(0) \leftarrow \infty$ and $t \leftarrow 1$

1 **For** each city pair i and j
Do the following in a random order:
If $d_{ij}(t) \le \delta_{ij}$, move i and j apart
by an amount $\mu(t)\left(\delta_{ij} - d_{ij}(t)\right)/2$
If $d_{ij}(t) > \delta_{ij}$ move i and j closer
by an amount $\mu(t)\left(d_{ij}(t) - \delta_{ij}\right)/2$

2 **If** $stress(t) \le stress(t-1)$ and
$stress(t-1) - stress(t) < \epsilon$ **Then** stop
Else $t \leftarrow t+1$, decrease $\mu(t)$ and **GoTo** Step 1

The algorithm described above, clearly, has the effect that it moves *every* pair of neurons in such a manner that their Euclidean distance *after the migration* is closer to the road distance than it was before the migration. Thus, if the road distances are themselves Euclidean, as assumed in the CMDS, it is possible that the neurons will converge to a configuration for which the stress is zero. Numerical results supporting this claim are provided in the next section.

In addition to STRESS we use a second performance criterion referred to as the *average location error* in order to measure the deviation between the reproduced map and the original map of the cities. It is defined as follows: $LE = \left(\sum_{i \in S} e_i\right)/n$. Here, e_i is the Euclidean distance between city i and neuron i which represents city i on the reconstructed map, n is the number of cities, and $\mathcal{S}$ is the set of cities where $|\mathcal{S}| = n$. It is important to note that the location error can only be computed if the city coordinates are available as part of the data. In real-life applications where MDS is used, we do not have access to this information. Otherwise MDS would not be necessary. Our motivation in using this error measure is to show the value of the extra information (the coordinates of at most two cities) in reproducing maps as faithfully as possible to actual maps. Notice that when the input distance matrix consists of Euclidean distances *and* we obtain a city configuration with no distortions such as translation, rotation and reflection, the final coordinates of the neurons will overlap with the cities they represent. In other words, the positions of the cities in the reconstructed map will exactly match with the locations of the cities in the actual map. In such a case, $e_i = 0$ for all $i \in \mathcal{S}$ resulting in zero location error. However, this is only possible if the inter-city distances are Euclidean. This means that it is not possible to have a zero location error with a distance matrix consisting of road travel distances. Therefore reconstruction methods with small location errors are capable of producing maps preserving not only distance relations but also locational proximity.

4 Experimental Results

Using six different data sets consisting of inter-city road travel distances from various countries, we have compared the performance of three methods, i.e., CMDS, ALSCAL and the new method. Each method is employed to reproduce the locations of the cities for each data set. In the interest of nomenclature, the data sets are referred to by the name of the country from which they are sampled [13]. With the exception of Türkiye data, the number of cities in each data set is 15 resulting in a 15×15 distance matrix. The data for Türkiye consists of inter-city distances among 80 cities, which gives rise to an 80×80 distance matrix. We have two kinds of inter-city distances, Euclidean distances and true road travel distances among the cities.

The performance of the three methods is evaluated based on two criteria. The first one is STRESS given with formula (3). It is a very frequently used measure in the MDS literature, and also one of the two measures employed in ALSCAL (the other is S-stress). The other performance measure is the average location error.

It is possible to divide our experiments into three groups. In the first one, the road distances are Euclidean. The second set of experiments is carried out when the road distances are true. In many real life applications some of the inter-city road travel distances could be missing. Considering this, we perform a third group of experiments where the distance matrix is not complete. First 5, then 10 entries of the distance matrix are chosen randomly and they are not used as an input to the methods with the exception of the CMDS since the latter is not designed to handle missing input data. Therefore, we compare the new method only with ALSCAL in this case.

In all of the experiments the learning rate $\mu(t)$ is decremented linearly starting at unity, according to the update equation $\mu(t) = 1 - t/1000$. Clearly this allows for 1000 or less epochs.

4.1 Results when Road Distances are Euclidean

For each data set, the new method is employed to reconstruct the map when the road distances among the cities are considered to be Euclidean. STRESS values reported in Table 1 demonstrate the accuracy of the new method. Recall that, as the distances are Euclidean, CMDS and ALSCAL guarantee final configurations with a zero STRESS value. Therefore, no row is given for them in Table 1 (to save space we use the following abbreviations for countries:

A	C	E	F	T	U
4×10^{-8}	0.0024	4×10^{-8}	4×10^{-8}	6×10^{-8}	0.01

Table 1: Stress values for Euclidean data by the new method.

	A	C	E	F	T	U
1	3787	3690	614	855	312	3639
2	3787	3646	536	897	694	2933
3	1605	2739	408	457	787	2310

Table 2: Location error values for Euclidean inter-city distances.

A=Australia, C=Canada, E=England, F=France, T=Türkiye, U=United States). Since the final configuration obtained by the new method depends on the initial weights (coordinates) of the neurons, we performed 10 replications, where neurons were assigned random initial weights in each case. Reported numbers are the average stress values. The results obtained by the new method are very close to zero showing the success of the method. It undoubtedly performs almost as well as CMDS and ALSCAL on the Euclidean data.

In this setting, we would like to mention that there is no accepted standard as to what value of the stress can be used as an indicator of a good representation. As a rule of thumb, we have resorted to the classification given by Kruskal: Any value less than 0.05 is excellent, values between 0.05 and 0.1 are satisfactory, and anything above 0.15 is unacceptable. From the results given in Table 1 we conclude that the performance of the new method is excellent.

Average location errors (rounded off to the nearest integer) are presented in Table 2. (to save space we use the following convention: 1=CMDS, 2=ALSCAL, 3=New method). Since the results obtained by CMDS and ALSCAL do not depend on the initial conditions, there is no need for repeating the runs in either of these methods. On the other hand, we report the average values of 10 replications obtained by the new method. We observe that the location error usually decreases for all data sets as the method goes from classical MDS to NeuroMapRecon. Exception can be observed for France and Türkiye. For Türkiye data CMDS has the smallest location error. The second best is ALSCAL. As for France data, the location error of CMDS is the smallest, but the new method performs better then ALSCAL this time.

Figure 1 illustrates the final configuration that is obtained for the USA data with CMDS when intercity road distances are Euclidean, while Figure 2 displays the points generated by ALSCAL. Finally,

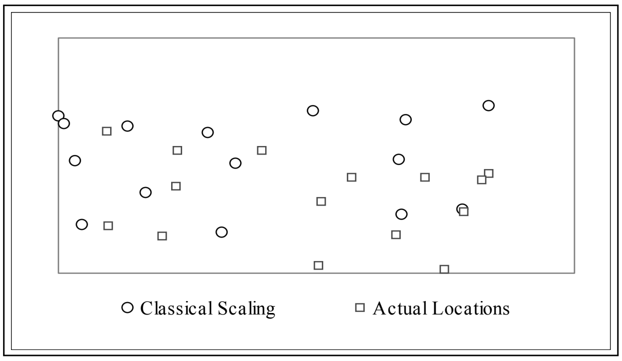

Figure 1: Final configuration obtained by classical MDS.

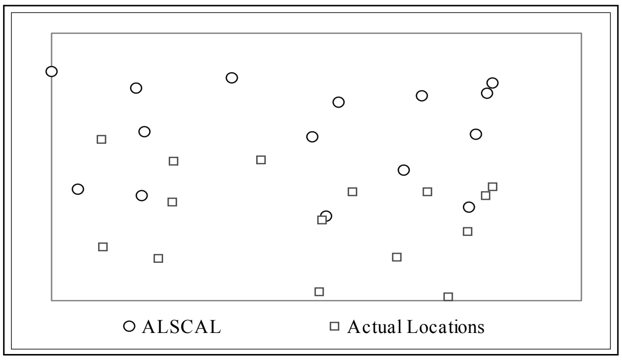

Figure 2: Final configuration obtained by ALSCAL.

Figure 3 shows the locations obtained by the new method. In all figures actual locations are also plotted in order to visualize final deviations.

4.2 Results when Inter-city Road Travel Distances are not Euclidean

When the distance matrix consists of inter-city road travel distances which are not Euclidean, the final configurations obtained by CMDS and ALSCAL have nonzero stress values. It is for such real-life data settings that the new method demonstrates its true advantages. Stress values and location errors obtained with all three methods are given in Tables 3 and 4, respectively. The values given for the new method are again the average values over 10 replications depending on the strategy. Based on the STRESS values reported in Table 3 we can say that ALSCAL performs better than CMDS while the new method is the best method with respect to both performance measures. According to the stress scale proposed by Kruskal the new method seems to be "excellent".

	A	C	E	F	T	U
1	0.0557	0.0252	0.0639	0.0345	0.0469	0.0419
2	0.0509	0.0230	0.0600	0.0312	0.0422	0.0376
3	0.0452	0.0131	0.0444	0.0276	0.0390	0.0295

Table 3: STRESS values for true inter-city road travel distances.

	A	C	E	F	T	U
1	3102	2761	639	866	793	2939
2	3896	3930	551	897	792	2936
3	1938	2869	452	472	806	2438

Table 4: Location error values for true inter-city road travel distances.

4.3 Results for Incomplete Data Sets

The new method can also reconstruct maps even if some of the distance data are missing. In order to demonstrate the efficiency of the new method with respect to this property, experiments are conducted by removing some elements of the distance matrix, $\mathbf{\Delta}$. In order to reduce the bias, for each data set the experiments are repeated 10 times by randomly removing first 5 and then 10 δ_{ij}'s from $\mathbf{\Delta}$. Since the final configuration given by the new method is dependent on the initial weights of the neurons, 10 replications are performed with random neuron initializations for each of the 10 runs. Note that a different set of distances are removed from $\mathbf{\Delta}$ in each run. Therefore, STRESS values reported in the second and fourth columns (new method columns) of Table 5 are the averages of 100 replications. However, random initialization is not possible for ALSCAL and thus the numbers given in the third and fifth columns of Table 5 are the averages of 10 stress values.

Results obtained with the new method are again

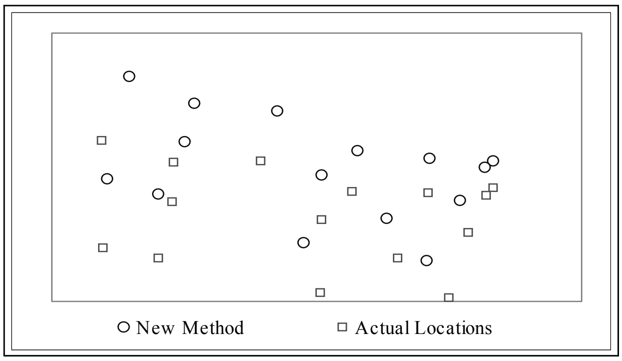

Figure 3: Final configuration obtained by the new method.

	5 missing δ_{ij}		10 missing δ_{ij}	
	New M.	ALSCAL	New M.	ALSCAL
A	0.04429	0.03898	0.04762	0.04805
C	0.01313	0.04126	0.01284	0.02126
E	0.04183	0.04130	0.04779	0.05788
F	0.03089	0.04068	0.02854	0.02902
T	0.03909	0.08008	0.03907	0.04212
U	0.04460	0.04304	0.03626	0.03401

Table 5: Performance of the new method when some input data is missing.

superior. Even in the presence of missing input data, it is capable of finding a final configuration with "excellent" stress values. ALSCAL performs slightly better for the Australia and England data with 5 missing values, and for the USA data with 5 and 10 missing values.

5 Conclusions

In this paper we have considered the fundamental problem of reconstructing a map when the given data is the set of distances among cities in a region. This is the "inverse" of the distance estimation problem where the goal is to determine a good estimator for inter-city road travel distances as a function of given city coordinates. In the map reconstruction problem our aim is to determine the location of the cities in a two dimensional map such that the Euclidean distances among the points in the obtained configuration approximate true road distances as closely as possible. The reported solutions to this problem are few, and primarily involve traditional techniques used in MDS.

The solution we propose, called the new method, is very accurate and does not involve any intricate matrix computations. It is also adaptive and can be said to be of a "real time" flavor. The new method has been rigorously tested on different data sets consisting of inter-city road travel distances obtained from various countries by comparing the results with those provided by CMDS method and ALSCAL. The accuracy of the proposed method is superior. The new method has also the following two desirable properties. First, it can reproduce configurations even if some of the input data are missing. Second, it is possible to obtain configurations without translation, rotation and reflection so that cities are located very close to their original locations.

References

[1] İ.K. Altınel, N. Aras, J. Oommen, "Vector quantization for arbitrary distance function estimation," *Informs Journal on Computing*, vol. 9, pp. 439–451, 1997.

[2] J. Brimberg and R.F. Love, "Estimating distances," *Facility Location; A Survey of Applications and Methods*, Z. Drezner, Ed., Springer Series in Operations Research, 1995.

[3] W.S. Torgerson, "Multidimensional scaling," *Psychometrica*, vol. 17, pp. 401–419, 1952.

[4] I. Borg and P. Groenen, *Modern Multidimensional Scaling, Theory and Applications*, New York, Springer–Verlag, 1997.

[5] R.N. Shepard, "Analysis of proximities: Multidimensional scaling with an unknown distance function I & II," *Psychometrica*, vol. 27, pp. 125–140, 219–246, 1962.

[6] W.S. Torgerson, "Multidimensional scaling of similarity," *Psychometrica*, 30, 379–393, 1965.

[7] C. Chatfield, *Introduction to Multivariate Analysis*, Chapman and Hall, 1980.

[8] M. Steyvers, "Multidimensional scaling" in *Encyclopedia of Cognitive Science*, 2002.

[9] Y. Takane, F.W. Young and J. de Leeuw, "Nonmetric individual differences multidimensional scaling: An alternating least squares method with optimal scaling features," *Psychometrika*, vol. 42, pp. 7–67, 1977.

[10] Y. Linde, A. Buzo and R.M. Gray, "An algorithm for vector quantization," *IEEE Transactions on Communications*, COM–28, pp. 61–71, 1980.

[11] J.B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrica*, vol. 29, pp. 1–27, 1964.

[12] M. Steyvers and T. Busey, "Predicting similarity ratings to faces using physical descriptions" in *Computational, geometric, and process perspectives on facial cognition: Contexts and challenges*, M. Wenger and J. Townsend Eds., Lawrence Erlbaum Associates, 2000.

[13] R.F. Love and J.H. Walker, "Distance data for eighteen geographic regions," Working Paper #383, Michael G. DeGroote School of Business, McMaster University, 1993.