

On Suitability of Euclidean Embedding of Internet Hosts

Sanghwan Lee
Dept. of Computer Science
Kookmin University, Seoul, Korea
sanghwan@kookmin.ac.kr

Zhi-Li Zhang
Dept. of Comp Sci and Engg.
Univ. of Minnesota, Minneapolis
zhzhang@cs.umn.edu

Sambit Sahu, Debanjan Saha
IBM T.J. Watson Research
Hawthorne, NY
sambits,dsaha@us.ibm.com

ABSTRACT

In this paper, we investigate the suitability of embedding Internet hosts into a Euclidean space given their pairwise distances (as measured by round-trip time). Using the classical scaling and matrix perturbation theories, we first establish the (sum of the) magnitude of *negative* eigenvalues of the (doubly-centered, squared) distance matrix as a measure of suitability of Euclidean embedding. We then show that the distance matrix among Internet hosts contains negative eigenvalues of *large magnitude*, implying that embedding the Internet hosts in a Euclidean space would incur relatively large errors. Motivated by earlier studies, we demonstrate that the inaccuracy of Euclidean embedding is caused by a large degree of *triangle inequality violation* (TIV) in the Internet distances, which leads to negative eigenvalues of large magnitude. Moreover, we show that the TIVs are likely to occur *locally*, hence, the distances among these close-by hosts cannot be estimated accurately using a *global* Euclidean embedding, in addition, increasing the dimension of embedding does not reduce the embedding errors. Based on these insights, we propose a new hybrid model for embedding the network nodes using only a 2-dimensional Euclidean coordinate system and small *error adjustment terms*. We show that the accuracy of the proposed embedding technique is as good as, if not better, than that of a 7-dimensional Euclidean embedding.

Categories and Subject Descriptors

C.2 [Computer Systems Organization]: Computer - Communication Networks; C.2.1 [Computer-Communication Networks]: Network Architecture and Design

General Terms

Algorithms, Measurement, Performance

Keywords

Euclidean Embedding, Triangle Inequality, Suitability

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGMetrics/Performance'06, June 26–30, 2006, Saint Malo, France.
Copyright 2006 ACM 1-59593-320-4/06/0006 ...\$5.00.

1. INTRODUCTION

Estimating distance (e.g., as measured by round-trip time or latency) between two hosts (referred as nodes hereafter) on the Internet in an accurate and scalable manner is crucial to many networked applications, especially to many emerging overlay and peer-to-peer applications. One promising approach is the *coordinate (or Euclidean embedding) based network distance estimation* because of its simplicity and scalability. The basic idea is to embed the Internet nodes in a Euclidean space with an appropriately chosen dimension based on the pairwise distance matrix. The idea was first proposed by Ng *et al* [1]. Their scheme, called GNP (Global Network Positioning), employs the least square multi dimensional scaling (MDS) technique to construct a low dimensional Euclidean coordinate system, and approximate the network distance between any two nodes by the Euclidean distance between their respective coordinates. To improve the scalability of GNP, [2] and [3] propose more efficient coordinate computation schemes using the principal component analysis (PCA). Both schemes are in a sense centralized. Methods for distributed construction of Euclidean coordinate systems have been developed in [4, 5].

While many efforts have focused on improving the accuracy and usability of the coordinate based distance estimation systems, studies have demonstrated the potential limitations of such schemes. For example, [6] shows that the amount of the triangle inequality violations (TIVs) among the Internet hosts are non-negligible and illustrates how the routing policy produces TIVs in the real Internet. They *conjecture* that TIVs make Euclidean embedding of network distances less accurate. [7] proposes new metrics such as relative rank loss to evaluate the performance and show that such schemes tend to perform poorly under these new metrics. In addition, [8] claims that the coordinate based systems are in general inaccurate and incomplete, and therefore proposes a light weight *active* measurement scheme for finding the closest node and other related applications.

In spite of the aforementioned research on the coordinate based network distance estimation schemes regardless of whether they advocate or question the idea, no attempt has been made to systematically understand *structural* properties of Euclidean embedding of Internet nodes based on their pairwise distances: what contributes to the estimation errors? Can such errors be reduced by increasing the dimensionality of embedding? More fundamentally, how do we quantify the suitability of Euclidean embedding? We believe that such a systematic understanding is crucial for charting the future research directions in developing more

accurate, efficient and scalable network distance estimation techniques. Our paper is a first attempt in reaching such an understanding, and proposes a simple new *hybrid* model that combines global Euclidean embedding with local non-Euclidean error adjustment for more accurate and scalable network distance estimation.

The contributions of our paper are summarized as follows. First, by applying the classical scaling and matrix perturbation theory, we establish the (sum of the) magnitude of *negative* eigenvalues of the (doubly-centered, squared) distance matrix as a measure of suitability of Euclidean embedding. In particular, existence of negative eigenvalues with **large magnitude** indicates that the set of nodes cannot be embedded well in a Euclidean space with small absolute errors.

Second, using data from real Internet measurement, we show that the distance matrix of Internet nodes indeed contains negative eigenvalues of large magnitude. Furthermore, we establish a connection between the degree of triangle inequality violations (TIVs) in the Internet distances to the magnitude of negative eigenvalues, and demonstrate that the inaccuracy of Euclidean embedding is caused by a large degree of TIVs in the network distances, which leads to negative eigenvalues of large magnitude.

Third, we show that a majority of TIVs occur among nodes that are close-by, suggesting a strong *local* non-Euclidean effect. By clustering nodes based on their distances, we find that while the distances between the nodes in the different clusters (the *inter-cluster* node distances) can be fairly well-approximated by the Euclidean distance function, the *intra-cluster* node distances are significantly more *non-Euclidean*, as manifested by a much higher degree of TIVs and the existence of negative eigenvalues with considerably larger magnitude. In addition, increasing the dimensionality of Euclidean embedding does not significantly improve its accuracy, in particular, for intra-cluster node distances. Based on these results we conclude that estimating network distances using coordinates of hosts embedded in a *global* Euclidean space is rather inadequate for close-by nodes.

As the last (but not the least) contribution of our paper, we develop a new hybrid model for embedding the network nodes: in addition to a low dimensional Euclidean embedding (which provides a good approximation to the inter-cluster node distances), we introduce a locally determined (*non-metric*) adjustment term to account for the non-Euclidean effect within the clusters. The proposed hybrid model is mathematically proven to always reduce the estimation errors in terms of *stress* (a standard metric for fitness of embedding). In addition, this model can be used in conjunction with any Euclidean embedding scheme.

The remainder of the paper is organized as follows. In Section 2 we provide a mathematical formulation for embedding nodes in a Euclidean space based on their distances, and apply the classical scaling and matrix perturbation theories to establish the magnitude of negative eigenvalues as a measure for suitability of Euclidean embedding. In Section 3, we analyze the suitability of Euclidean embedding of network distances and investigate the relationship between triangle inequality violations and the accuracy. We show the clustering effects on the accuracy in section 4. We describe the new hybrid model for the network distance mapping in Section 5 and conclude the paper in Section 6.

2. EUCLIDEAN EMBEDDING AND CLASSICAL SCALING

In this section we present a general formulation of the problem of embedding a set of points (nodes) into a r -dimensional Euclidean space given the pairwise distance between any two nodes. In particular, using results from classical scaling and matrix perturbation theories we establish the (sum of the) magnitude of negative values of (an appropriately transformed) squared distance matrix of the nodes as a measure for the *suitability* of Euclidean embedding.

2.1 Classical Scaling

Given only the ($n \times n$, *symmetric*) distance matrix $D = [d_{ij}]$ of a set of n points (nodes) (from some arbitrary space), where d_{ij} is the *distance*¹ between two points \mathbf{x}_i and \mathbf{x}_j , $1 \leq i, j \leq n$, we are interested in the following problem: can we embed the n points $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ in an r -dimensional space for some $r \geq 1$ with *reasonably good accuracy*? To address this question, we need to first determine what is the appropriate dimension r to be used for embedding; given r thus determined, we then need to map each point \mathbf{x}_i into a point $\tilde{\mathbf{x}}_i = (\tilde{x}_{i1}, \dots, \tilde{x}_{ir})$ in the r -dimensional Euclidean space to minimize the overall error of embedding with respect to certain criterion of accuracy.

Before we address this problem, we first ask a more basic question: Suppose that the n points are actually from an r -dimensional Euclidean space, given *only* their distance matrix $D = [d_{ij}]$, is it possible to find out the original dimension r and recover their original coordinates in the r -dimensional space? Fortunately, this question is already answered by the theory of classical scaling [9]. Let $D^{(2)} = [d_{ij}^2]$ be the matrix of squared distances of the points. Define $B_D := -\frac{1}{2}JD^{(2)}J$, where $J = I - n^{-1}\mathbf{1}\mathbf{1}^T$, I is the unit matrix and $\mathbf{1}$ is a n -dimensional column vector whose entries are all 1. J is called a centering matrix, as multiplying J to a matrix produces a matrix that has 0 mean columns or rows. Hence B_D is a doubly-centered version of $D^{(2)}$. A result from the classical scaling theory gives us the following theorem.

THEOREM 1. *If a set of n points $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ are from an r -dimensional Euclidean space. Then B_D is semi-definite with exactly r positive eigenvalues (and all other eigenvalues are zero). Furthermore, let the eigen decomposition of B_D is given by $B_D = Q\Lambda Q^T = Q\Lambda^{1/2}(Q\Lambda^{1/2})^T$, where $\Lambda = [\lambda_i]$ is a diagonal matrix whose diagonal consists of the eigenvalues of B_D in decreasing order. Denote the diagonal matrix of the first r positive eigenvalues by Λ_+ , and Q_+ the first r columns of Q . Then the coordinates of the n points are given by the $n \times r$ coordinate matrix $Y = Q_+\Lambda_+^{1/2}$. In particular, Y is a translation and rotation of the original coordinate matrix X of the n points.*

Hence the above theorem shows that if n points are from an Euclidean space, then we can determine precisely the original dimension and recover their coordinates (up to a translation and rotation). The *contrapositive* of the above theorem states that if B_D is not semi-definite, i.e., it has *negative* eigenvalues, then the n points are *not* originally from

¹We assume that the distance function $d(\cdot, \cdot)$ satisfy $d(x, x) = 0$ and $d(x, y) = d(y, x)$ (symmetry), but may violate the *triangle inequality* $d(x, z) \leq d(x, y) + d(y, z)$; hence d may not be *metric*.

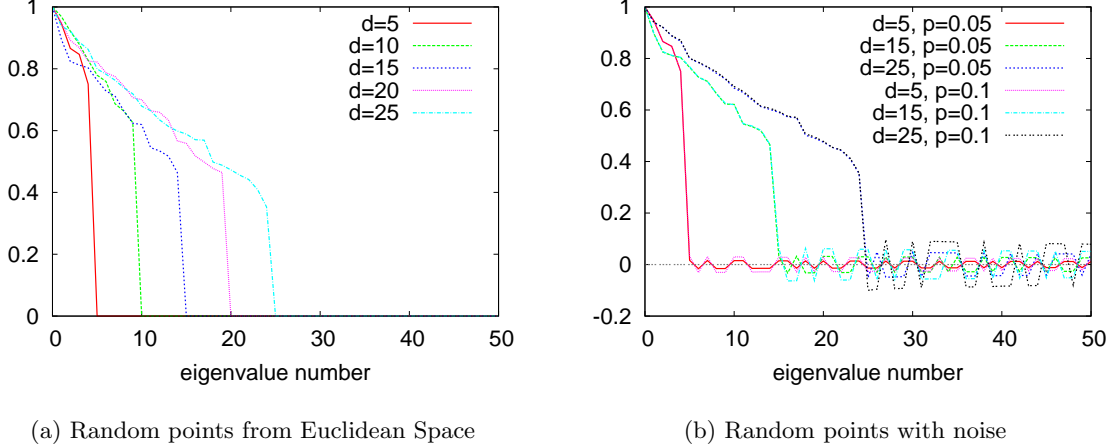


Figure 1: Scree plots of the eigenvalues on data sets. Random points are generated in d -dimensional Euclidean space. The noise is computed as $dnoise_{xy} = d_{xy} + d_{xy} \times f$, where f , the noise factor, is uniformly randomly selected from a range of $[0, p)$. $p = 0.05$ and $p = 0.1$ are used.

an Euclidean space. A natural question then arises: *does the negative eigenvalues of B_D tell us how well a set of n points can be embedded in a Euclidean space?* In other words, can they provide an appropriate measure for *suitability* of Euclidean embedding? We formalize this question as follows. Suppose the n points are from an r -dimensional Euclidean space, but the actual distance \tilde{d}_{ij} between two points \mathbf{x}_i and \mathbf{x}_j is “distorted” slightly from their Euclidean distance d_{ij} , e.g., due to measurement errors. Hence, intuitively if the total error is small, we should be able to embed the n points into an r -dimensional Euclidean space with small errors. Using the matrix perturbation theory, we show that in such a case the (doubly centered) squared distance matrix must have small negative eigenvalues.

Formally, we assume that $\tilde{d}_{ij}^2 = d_{ij}^2 + e_{ij}$, where $|e_{ij}| \leq \epsilon/n^2$ for some $\epsilon > 0$. Hence $\tilde{D}^2 := [\tilde{d}_{ij}^2] = D^{(2)} + E$, where $E := [e_{ij}]$. A frequently used matrix norm is the *Frobenius norm*, $\|E\|_F := \sqrt{\sum_i \sum_j |e_{ij}|^2} \leq \epsilon$. Then $B_{\tilde{D}} := -\frac{1}{2}J\tilde{D}^{(2)}J = B_D + B_E$, where $B_E := -\frac{1}{2}JEJ$. It can be shown that $\|B_E\|_F \leq \epsilon$. For $i = 1, 2, \dots, n$, let $\tilde{\lambda}_i$ and λ_i be the i th eigenvalue of $B_{\tilde{D}}$ and B_D respectively, where $\tilde{\lambda}_1 \geq \dots \geq \tilde{\lambda}_n$ and $\lambda_1 \geq \dots \geq \lambda_n$. Then the Wiedlandt-Hoffman Theorem [10] states that $\sum_{i=1}^n (\tilde{\lambda}_i - \lambda_i)^2 \leq \|B_E\|_F^2$. Since $\lambda_i \geq 0$, we have

$$\sum_{\{i: \tilde{\lambda}_i < 0\}} |\tilde{\lambda}_i| \leq \sum_{\{i: \tilde{\lambda}_i < 0\}} (-\tilde{\lambda}_i + \lambda_i) \leq \sum_{i=1}^n (\tilde{\lambda}_i - \lambda_i) \leq \|B_E\|_F \leq \epsilon.$$

Hence the sum of the squared absolute values of the *negative* eigenvalues is bounded by the squared Frobenius norm of the (doubly-centered) error matrix $\|B_E\|_F^2$, which is the sum of the (doubly-centered) squared errors. In particular, the absolute value of any negative eigenvalue $|\tilde{\lambda}_i|$ is bounded by $\|B_E\|_F$. Hence if the total error (as reflected by $\|B_E\|_F$) is small and bounded by ϵ , then the negative eigenvalues of $B_{\tilde{D}}$ are also small and their magnitude is bounded by ϵ . Hence the *magnitude* of negative eigenvalues (and their sum) pro-

vides a measure of the *suitability* of Euclidean embedding: if a set of n points can be well-approximated by a Euclidean space with an appropriate dimension, then their associated doubly-centered squared distance matrix only has negative eigenvalues of small magnitude, if any. On the other hand, the contrapositive of the above proposition leads to the following observation: if the doubly-centered squared distance matrix has negative eigenvalues of *large* magnitude, then the set of n points cannot be embedded into a Euclidean space with a small total error (as measured by $\|B_E\|_F$), hence they are less amenable to Euclidean embedding.

More generally, when a set of n points are originally from an Euclidean space, embedding them into an r -dimensional Euclidean space would introduce some errors. *Multidimensional scaling* (MDS) [9] is a generalization of classical scaling for embedding a set of points into an r -dimensional (metric) space (with not necessarily Euclidean distance function) to minimize certain pre-specified error function, e.g., the stress (1). The GNP method proposed in [1] uses MDS for embedding Internet nodes in an r -dimensional Euclidean space given the distance matrix to minimize the so-called the overall *relative error* (see the next section). The dimension r is essentially determined by trial-and-error. In [2] and [3] an more efficient (but somewhat less accurate) approach is proposed using Liptchiz embedding and principal component analysis (PCA) or singular value decomposition (SVD) of the distance matrix, where the dimension of the embedding is the number of singular values with relative large magnitude.

2.2 Illustration

We now generate some synthetic data to demonstrate how classical scaling can precisely determine the original dimensionality of data points that are from a Euclidean space. First, we generate 360 random points in a unit hyper cube with different dimensions and compute the corresponding distance matrix for each dataset. Fig. 1(a) shows the *scree plot* of the eigenvalues obtained using classical scaling. The eigenvalues are normalized by the largest value (This will be

Data Set	Nodes	Date
King462 ([11])	462	8/9/2004
King2305 ([12])	2305	2004
PlanetLab ([13])	148	9/30/2005
GNP ([14])	19	May 2001

Table 1: The data sets used in our study. The number of nodes is chosen to make the matrix complete and square.

the same for the rest of the paper). We see from Fig. 1(a) that the eigenvalues vanish right after the dimensionality of the underlying Euclidean space where the data points are from, providing an unambiguous cut-off to uncover the original dimensionality. We now illustrate what happens when distances among data points are not precisely Euclidean (e.g., due to measurement errors). We add noise to the synthetically generated Euclidean datasets as follows: The noise component in the data is $d \times f$, where d is the original Euclidean distance and f is a randomly selected number from $[0, p)$. We use $p = 0.05$ and $p = 0.1$ for the illustration below. We observe in Fig. 1(b) that the first r eigenvalues are positive, and are nearly the same as in the case without noise, where r represents the actual dimension of the dataset. Beyond these eigenvalues, we observe only small negative eigenvalues. As the noise increases, the magnitudes of negative eigenvalues increase slightly. It is clear that as the data set deviates from Euclidean more, the magnitudes of the negative eigenvalues become larger.

3. EUCLIDEAN EMBEDDING OF NETWORK DISTANCES

In this section, we begin by examining the accuracy of Euclidean embedding of network distances for a wide range of datasets. We consider four different metrics for this error evaluation that we believe are useful for a variety of delay sensitive applications. We apply eigenvalue analysis to show that the (doubly-centered, squared) distance matrices of the datasets contain negative eigenvalues of relatively large magnitude. We then attribute existence of the negative eigenvalues of relative large magnitude to the large amount of triangle inequality violations existing in the datasets by showing: i) embedding a subset of nodes without triangle inequality violations in a Euclidean space produces higher accuracy, and the associated distance matrix also contains only negative eigenvalues of much smaller magnitude; ii) by increasing the degree of TIVs in a subset of nodes of the same size, the performance of Euclidean embedding degrades and the magnitude of the negative eigenvalues also increases.

We use four different datasets which we refer to as *King462*, *King2305*, *PlanetLab* and *GNP*, as listed in 1. *King462* is derived from the dataset used by Dabek et al. [11] after removing the partial measurements to derive a 462×462 complete and square distance matrix among 462 hosts from the original 2000 DNS server measurements. Using the same refinement over the dataset used in [12], we derive *King2305*, which is a 2305×2305 complete and square distance matrix. *PlanetLab* is derived from the distances measured among the Planetlab nodes on Sep 30th 2005 [13]. We chose the minimum of the 96 measurement (one measurement per 15 minutes) data points for each measurement between node pairs. After removing the hosts that have missing distance

information, we obtain a 148×148 distance matrix among 148 nodes. *GNP* dataset is obtained from [14] that contains a 19×19 distance matrix. Even though the number of hosts is small in this dataset, we have chosen this dataset in order to compare with the results in other papers.

3.1 Performance of Euclidean Embedding

We consider four performance metrics, namely, *stress*, (*cumulative*) *relative errors*, *relative rank loss* (RRL), and *closest neighbor loss* (CNL) that have been introduced across various studies in the literature. We compute the embedding errors for these four metrics for the datasets mentioned earlier. We consider two Euclidean embedding, namely, Virtual Landmark and GNP that are suggested in the literature. The four metrics are stress, relative error, relative rank loss, and closest neighbor loss. Stress and relative errors are used for many Euclidean embedding literatures. Relative rank loss and closest neighbor loss have been recently introduced in [7], where they focus on finding nearest neighbor. The four metrics are defined as follows :

- Stress: This is a standard metric to measure the overall fitness of embedding, originally known as *Stress-1* [9]:

$$\text{Stress-1} = \sigma_1 = \sqrt{\frac{\sum_{x,y} (d_{x,y} - \hat{d}_{x,y})^2}{\sum_{x,y} d_{x,y}^2}}, \quad (1)$$

where $d_{x,y}$ is the actual distance between x and y , and $\hat{d}_{x,y}$ is the estimated one.

- Relative error: This metric is introduced in [1] that is defined as follows: $\frac{|d_{x,y} - \hat{d}_{x,y}|}{\min(d_{x,y}, \hat{d}_{x,y})}$. Note that the denominator is the minimum of the actual distance and the estimated one².
- Relative rank loss (RRL): RRL denotes the fraction of pair of destinations for which their relative distance ordering, i.e., rank in the embedded space with respect to a source has changed compared to the actual distance measurement. For example, for a given source node, we take a pair of destinations and check which one is closer to the source in the real distances and the estimated distance. If the closest one is different, then the relative rank is defined to be lost. We compute the fraction of such relative rank losses for each source and compute the average among all the sources to get the overall RRL for the embedding.
- Closest neighbor loss (CNL): For each source, we find the closest node in the real distances and the estimated distances. If the two nodes are different, the closest neighbor is lost. We compute the fraction of such losses for each source and compute the average for overall CNL. However, unlike the original CNL in [7], we introduce δ , which is a sort of margin for CNL. So if the closest node in the estimated distances is within δ ms, we consider them as non-loss. The original CNL is when $\delta = 0$. We expect that as δ increases, the CNL decreases.

²In some literatures, instead of $\min(d_{x,y}, \hat{d}_{x,y})$, $d_{x,y}$ is used. This usually produces smaller relative errors.

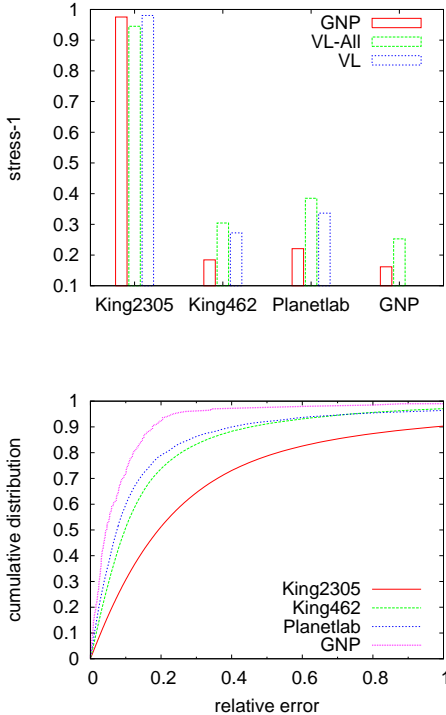


Figure 2: Top (a): Stress, Bottom (b) : Relative Error. Performance Embedding. The number of dimensions is 7. Only the results from GNP method are shown for the relative error.

Following the results in [1, 2, 3], we choose 7 as the dimension of Euclidean embedding for both GNP and Virtual Landmark (using the first 7 largest singular values). Since there is no embedding that is best for all the cases, we try different types of embedding. Fig. 2(a) shows the stress of embedding the network distances in a 7-dim Euclidean space using the GNP and Virtual Landmark methods. In GNP and VL, we use 20 landmarks randomly selected from the data set. In VL-All, we use all the nodes as landmarks. We see the overall stress is 0.2 to 0.5 except King2305 dataset, which indicates that on the average the estimations deviate from the original distances from 20 % to 50%. In King2305 data set, there are many links which have more than 90 seconds RTT, which might be an outlier, but we just use it as it is. It is clear from (1) that the metric stress can be affected by outliers. Fig. 2(b) shows the cumulative distribution of relative errors obtained from GNP method. For about 20% to 50 % of the estimations, the relative errors are more than 0.5. In short, both results show a non-negligible amount of estimation errors. Also as can be seen in Fig. 2(a) and 2(b) the stress and the relative errors are correlated.

Fig. 3 shows the cumulative distribution of the individual relative rank losses. Most of them are below 0.3, which is relatively good. However, the performance in CNL is quite bad as can be seen in Fig. 4. The maximum CNL is almost 1 in some cases, which means that most of the nodes could not find the closest node based on the distance estimation. Even when δ increases, CNL does not increase much.

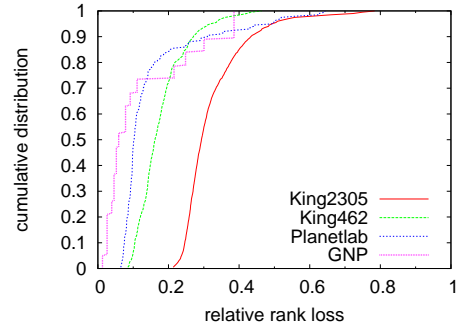


Figure 3: The cumulative distribution of Relative Rank Loss. GNP method is used.

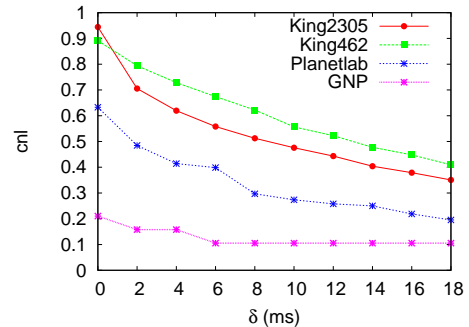


Figure 4: Closest Neighbor Loss (GNP method).

3.2 Analyzing Suitability of Euclidean Embedding

To understand the possible sources of Euclidean embedding of network distances, in this section we perform eigenvalue analysis of the distance matrices and investigate how triangle inequality violations (TIVs) affect the suitability of Euclidean embedding and accuracy of the embedding.

Eigenvalue Analysis. First, we perform eigenvalue analysis of the doubly-centered, squared distance matrix $B_D = -JD^{(2)}J$. Fig. 5 shows the scree plot of the resulting eigenvalues, normalized by the eigenvalue of the largest magnitude $|\lambda_1|$, in decreasing order in the magnitude of the eigenvalues. We see that each of the datasets has one or more negative eigenvalues of relatively large magnitude that are at least about 0.2% of $|\lambda_1|$, and the negative eigenvalue of largest magnitude is among the second and fourth largest in terms of magnitude). This suggests that the network distances are fairly strong “non-Euclidean”, and the nodes are somewhat less suitable for Euclidean embedding. Hence it is expected that embedding the nodes in a Euclidean would produce considerable amount of errors.

TIV Analysis. Motivated by earlier studies (e.g., [6]) which shows that there is a significant amount of TIVs in the Internet distance measurement, and attributes such TIVs to routing policies. Here we investigate how the amount of TIVs in the datasets affect the suitability and accuracy of Euclidean embedding of network distances. In particular, we establish a strong correlation between the amount of TIVs and the magnitude of negative eigenvalues of the associated distance matrix. First we analyze the amount of TIVs in the four data sets of real Internet distances. For each data set,

Data Set	GNP	King2305	King462	Planetlab
fraction	0.116	0.233	0.118	0.131

Table 2: The fraction of TIVs over all triples of nodes

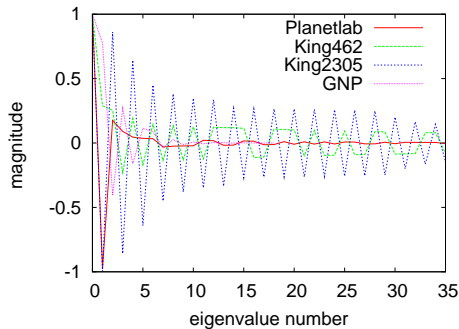


Figure 5: Scree plot of the eigenvalues on network distance measurement data set.

we take a triple of nodes and check whether they violate triangle inequality. We then compute the fraction of such TIVs over all possible triples. Table 2 shows that the King2305 data set (where the fraction of TIVs in the King2305 data set is about 0.23, while for the other three datasets, it is around 0.12). Hence the triangle inequality violations are fairly prevalent in the data sets.

To investigate how the amount of TIVs affect the suitability and accuracy of Euclidean embedding, in particular, its impact on the magnitude of negative eigenvalues, we start with a subset of nodes without any triangle inequality violation (we refer to such a subset of nodes as *TIV-free set*). Ideally we would like this subset to be as large as possible – the *maximal TIV-free (sub)set*. Unfortunately, given the distance matrix of a node set, finding the maximal TIV-free subset is NP-hard, as is shown in the following theorem (proof of which is described in the appendix).

THEOREM 2. *The maximal TIV-free set problem is NP-complete.*

Hence we have to resort heuristics to find a relatively large TIV-free sets. Here we describe three heuristic algorithms. The basic algorithm (referred to as *Algo 0*) is to randomly

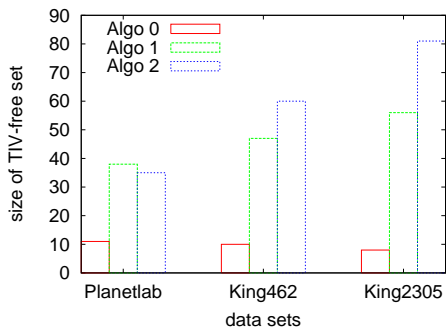


Figure 6: Performance of the 3 heuristic algorithms. Algo 2 finds the largest size TIV-free sets for King462 and King 2305 set.

choose k nodes from a given set of n nodes and check whether any three nodes of these randomly selected k nodes violates the triangle inequality. If the triangle inequality is violated, the process is repeated again by randomly selecting another set of k nodes. If we find a TIV-free set of size k , we increase k by one and try again to attempt to find a larger set. Otherwise the algorithm terminates after a pre-specified number of failed tries, and returns the TIV-free set of size $k - 1$. The second heuristic algorithm (*Algo 1*) is as follows. We start with a TIV-free set with two randomly selected nodes, and a (remaining) node set of $n - 2$ nodes. We then select a random node and check to see whether it violates the triangle inequality with the existing TIV-free set. If yes, this node is removed from the node set. Otherwise it is added to the TIV-free set and removed from the remaining node set. The process is repeated until the remaining node set becomes empty. The third heuristic algorithm (*Algo 2*) is slightly more sophisticated, and works in a similar fashion as *Algo 1*, except that we do not choose nodes randomly for consideration. We start with an initial TIV-free set A of two nodes, where the two nodes are chosen such that the pair of nodes has the least number of TIVs with nodes in the remaining node set C . Given this pair of nodes, we remove all nodes in the remaining node set C that violate the triangle inequality with this pair of nodes. For each node c in C , we compute the number of nodes in C that violates triangle inequality with c and any node in A .

We add the node c that has the smallest such number, add it to A and remove it from C . We then purge all the nodes that violate the triangle inequality with c and a node in A . We repeat the above process until C becomes empty.

The size of largest TIV-free sets found by the three heuristic algorithms is shown in Fig. 6. For the three datasets considered, Algo 0 only finds a TIV-free set of 10 nodes. Algo 2 finds the largest TIV-free sets for the King462 and King 2305 datasets, while Algo 1 finds the largest TIV-free set for the Planetlab dataset. For the following analysis, we use the largest TIV-free set found for each dataset. Fig. 7(a) shows the scree plot of the eigenvalues for the associated (doubly-centered, squared) distance matrix of the TIV-free node sets. We see that they all have only a small number of negative eigenvalues and the magnitudes of all the negative eigenvalues are also in general fairly small. Comparing to Fig. 5, we can see the reduction in both the number and magnitude of negative eigenvalues. Consequently, comparing the results in Fig. 7(b) with those in Fig. 2(b), we see that the Euclidean embedding of the TIV-free sets has a much better overall accuracy.

Correlation between Negative Eigenvalues and

Amount of TIVs. Next, we show how the amount of TIVs in a dataset contributes the magnitude of negative eigenvalues, thereby the suitability and accuracy of Euclidean embedding. We use the King2305 dataset as an example. The largest TIV-free set we found has 81 nodes. We fix the size of the node sets, and randomly select *six* other node sets with exactly 81 nodes, but with *varying* amount of TIVs. The scree plots of the eigenvalues for the six node sets are shown in Fig. 8(a), and the cumulative relative error distributions of the corresponding Euclidean embedding are shown in Fig. 8(b). We see that with the increasing amount of TIVs, both the magnitude and number of negative eigenvalues increase; and not surprising, the overall accuracy of the Euclidean embedding also degrades. In fact, we can

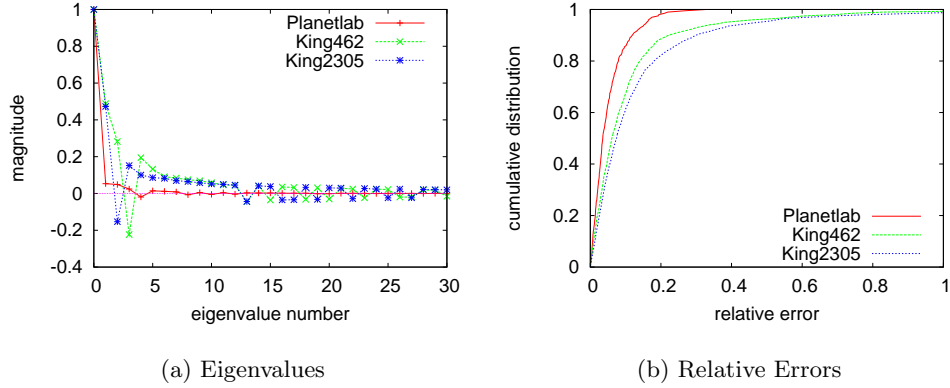


Figure 7: Embedding of TIV-free node sets. GNP method with 7 dimension is used. 20 landmarks are randomly selected among the set and the relative error is computed over all the nodes.

mathematically establish a relation between the amount of TIVs and the sum of squared estimation errors as follows.

LEMMA 1. *If the distances a, b, c among 3 nodes violate the triangle inequality, i.e., $c > a + b$, the minimum squared estimation error of any Euclidean embedding of the 3 nodes is $\frac{(c-a-b)^2}{3}$.*

THEOREM 3. *The sum of squared estimation errors of any Euclidean embedding of n nodes is larger than or equal to $\frac{1}{3(n-2)} \sum_{t \in V} (t_c - t_a - t_b)^2$, where V is the set of triples that violates triangle inequality, t_a, t_b , and t_c are the 3 distances of a triple v , and $t_c > t_a + t_b$.*

The proofs are delegated to the appendix. Theorem 3 states that as the amount of TIVs increases, the sum of the squared estimation errors also increases. It should be noted that this argument can be directly applied to any metric space not only to the Euclidean space. In addition, a similar result can be established for the sum of squared relative errors, the details of which is omitted here.

4. LOCAL NON-EUCLIDEAN EFFECT

Our analysis in the previous section illustrated that distance matrix of Internet hosts contains large negative eigenvalues and large number of TIVs. In addition, we established a correlation between the degree of TIVs in the network distance and the magnitude of negative eigen values. However, we observe that the magnitude of embedding errors vary across the four performance metrics we have considered. In this section, we dissect the dataset further to find out whether clustering of Internet hosts contribute to the errors in the embedding, which nodes are likely to contribute to the higher degree of TIVs and whether increasing the dimension of the embedding helps improve the embedding performance.

4.1 Clustering Effect

The hosts in the Internet are clustered due to many factors such as geographical location and ASes. This clustering causes many hosts to have short distances among themselves. To investigate the effect of clusters on accuracy, we

first look at the existence of clusters in the network distances. To identify clusters within the network distances, we apply the spectral clustering algorithm [15] to King462 data set with the outliers³ removed. In this experiment, 28 nodes out of 462 are removed. The algorithm⁴ obtains 4 clusters for the King462 dataset. We use a *gray scale* plot to show the existence of clusters in the King462 dataset with outliers removed.

In Fig. 9, the vertical axis represents the source nodes and the horizontal axis represents the target nodes (both are grouped based on the clusters they belong to). The cross point between the vertical and horizontal axis represents the distance between the corresponding two nodes. The distance is represented in a gray scale: White color represents distance 0 and black color represents the distance larger than the 95th percentile distance. The interval between 0 and the 95th percentile distance is divided into 10 gray scales (with a total of 11 gray scales), with increasing darkness from white to black (beyond 95th percentile distance). We can clearly see that there are about 4 clusters. The table in Fig. 9 shows the median distances between nodes within and across clusters. As can be expected, the intra-cluster median distances are much smaller than the inter-cluster median distances.

To illustrate the characteristics of the individual clusters, in Fig. 10, we show the scree plot of the eigenvalues of classical scaling on the 4 clusters of the King462 data set. The magnitudes of the negative eigenvalues are larger than those of the whole data sets (compared to Fig. 5). The “non-Euclidean-ness” amplifies within each cluster. It suggests that the intra-cluster distances are much harder to embed into the Euclidean space. This can be easily observed by looking at the relative errors of the embedding. Fig. 11 shows the relative errors in a gray scale matrix for the King462 dataset, where the Virtual Landmark method is used for the embedding. The pure black color represents the relative error of 1.0 or larger, and 10 gray scales are used

³Outliers are defined as the nodes of which distance to 8th nearest nodes are larger than a threshold.

⁴The algorithm takes as input a parameter K , the number of clusters, and produces *up to* K as a result. We have experimented with $K = 3$ to 7, and the algorithm in general produces 3-4 “relatively big” clusters for the three datasets.

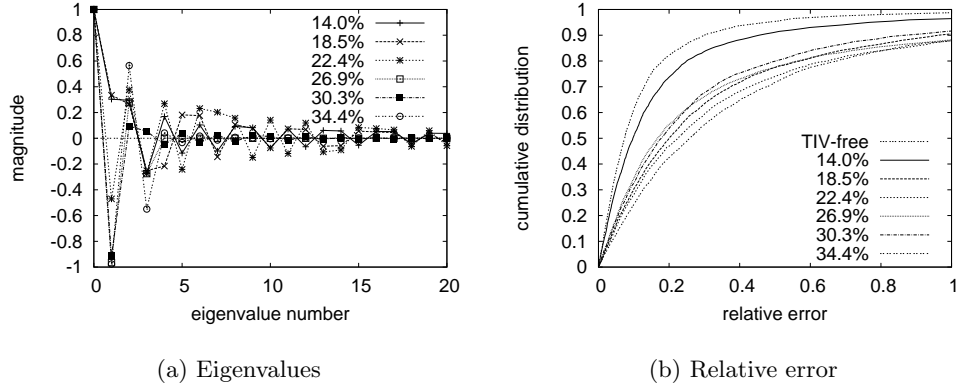
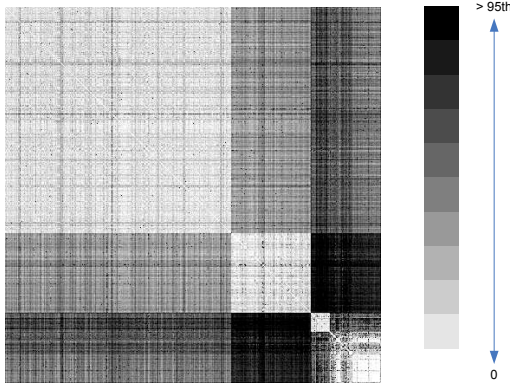


Figure 8: The change of eigenvalues and relative errors over the fraction of TIVs.



	c1	c2	c3	c4
c1	62.099	154.287	254.469	212.538
c2	154.287	60.681	376.146	321.508
c3	254.469	376.146	61.194	238.938
c4	212.538	321.508	238.938	61.950

Figure 9: Distances between each pair of nodes in King462 data set after removing outliers. White represents distance 0 and black represents 95th percentile or higher distances. Median distances (in ms) among the nodes of the intra and inter clusters are shown in the table.

for relative errors between 0 and 1. We see that the relative errors of the intra-cluster estimations are larger than those of inter-cluster estimations.

We next examine which nodes are more likely to contribute towards the TIVs. As we shall illustrate next, the high errors in the intra cluster distance estimation and the large magnitudes of the negative eigenvalues can be explained by the varied number of TIVs over the different distances. Intuitively, a TIV is likely to occur if distance between two nodes is very short or very large compared to the other two distances for a given set of three nodes. Using this intuition we proceed with our data analysis as fol-

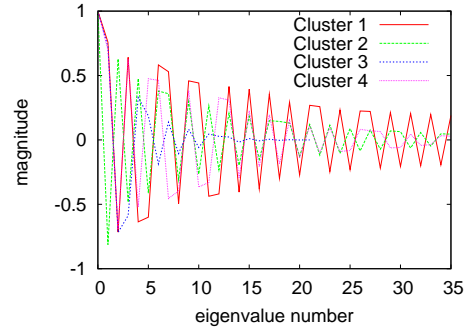


Figure 10: Screen plot of the eigenvalues of CS on the 4 clusters of the King462 data set after removing 28 outliers : Cl 1 (261 nodes), Cl 2 (92 nodes), Cl 3 (22 nodes), and Cl 4 (59 nodes).

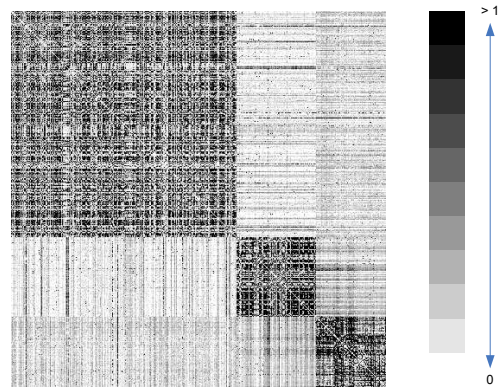


Figure 11: Relative errors between each pair of nodes in King462 data set without outliers. White represents relative error 0 and black represents relative error 1 or larger. Virtual Landmark method with 7 dimension is used.

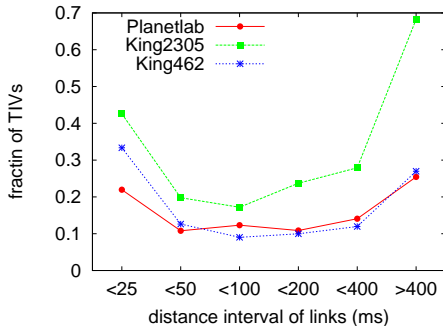


Figure 12: Average fraction of TIVs at each distance interval.

lows: We divide the distances into 6 intervals : $[0ms, 25ms)$, $[25ms, 50ms)$, $[50ms, 100ms)$, $[100ms, 200ms)$, $[200ms, 400ms)$, and $[400ms, \infty)$. We group all the pairs of nodes by their distance intervals. Then, for each pair of nodes, we compute the fraction of TIVs in conjunction with the rest of the nodes, i.e. we count how many nodes violate triangle inequality with the given pair of nodes. Finally, we compute the average of the fractions of all the pairs in each interval. Fig. 12 shows the average fraction of TIVs in each distance interval. We observe that higher fractions of TIVs occur in the intervals $[0, 25ms)$ and $[400, \infty)$ compared to other intervals. Since the fractions of pairs in $[400, \infty)$ are quite small in all the data set and are not much interest to any application, reducing the errors in short distance estimations is much more crucial for overall performance.

The above analysis illustrated that the distances among the inter-cluster nodes are more likely to be better approximated by their Euclidean coordinates, whereas Euclidean embedding of nodes within a cluster would poorly estimate the distance. This seems to suggest that there is much stronger *local* “non-Euclidean effect” on the network distances.

4.2 Effect of Increasing the Embedding Dimension

The results in the previous subsection suggests that to improve the overall performance, the accuracy of the intra-cluster embedding should be improved. We examine whether increasing the dimension of the embedding would help improve the embedding accuracy. Fig. 13 shows the cumulative distributions of relative errors of the King462 data set embedded using the Virtual Landmark method over various dimensions. We see that there is a significant improvement in relative errors when we increase the embedding dimension from 2 to 3. However, in general, increasing the embedding dimension beyond 3 dimensions does not yield considerable gain. The distributions are very similar from $d=3$ to $d=7$.

Next, we look at the effect of increasing dimensions on the *intra- vs. inter-cluster* node distance estimation. We use the embedding of the King462 dataset using the Virtual Landmark method as an example. For each node, we compute the fraction of good estimates (when the relative error is less than a threshold p , we consider the estimation good. We use $p = 0.15$.) to other nodes within the same cluster as well as to other nodes in different clusters. We then compute the *average* fraction of good estimates for each combination

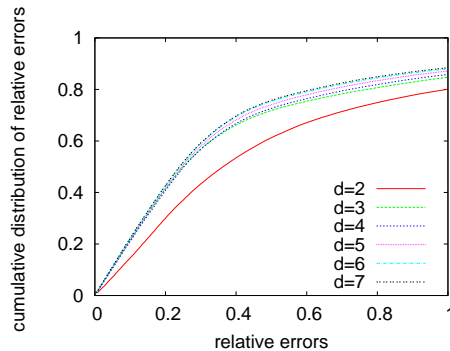


Figure 13: Cumulative distributions of relative error over the number of dimensions. After 3 dimensions, the performance does not decrease much.

of cluster pairs. The results are shown in Fig. 14, where the plot “ $cx-cy$ ” is the average fraction of good estimates between nodes cluster x and nodes in cluster y for the Virtual Landmark embedding of varying dimensions.

Fig. 14 illustrates and confirms several important observations. First, regardless of the embedding dimension, the inter-cluster node distance estimation (the top six curves) has far better performance than intra-cluster node distance estimation (the bottom four, $c1-c1$, $c2-c2$, $c3-c3$ and $c4-c4$). Second, for inter-cluster node distance estimation, increasing d beyond 3 dimensions does not improve the performance, in fact, often degrades the performance. Note that cluster 1 (275 nodes) and cluster 3 (102 nodes) are large clusters. For them, increasing the embedding dimension yields somewhat better performance, indicating that higher dimensions are needed to embed the nodes in the clusters.

These results clearly illustrate the strong local “non-Euclidean” effect of the datasets, and demonstrate that simply increasing the dimension of embedding is *not* the right approach to improve the performance of network embedding methods (in fact a dimension of 3 or 4 seems to suffice). We also have investigated whether using a non-Euclidean distance metric can improve the overall performance of network distance embedding. For this purpose, we have used the Minkowski p -norm, $\|x\|_p = (\sum_{i=1}^n |x_i|^p)^{1/p}$, where p can be fractional. However, this does not improve the performance either. In general, since the TIVs do not disappear no matter how to embed the nodes into the Euclidean space (more generally the metric space where triangle inequality holds), it is hard to improve the performance for the nodes where many TIVs exist. The theorem 3 clearly states this.

5. A HYBRID MODEL FOR LOCAL ADJUSTMENT

The results from previous sections show that the existence of TIVs highly affects the accuracy of the Euclidean embedding. Furthermore, the network distances exhibit strong *local* non-Euclidean effect. In particular, Euclidean embedding is fairly good at estimating network distances between nodes that are far-away (in different clusters), whereas it is rather poor at estimating local network distances (distance between nodes within a cluster). These observations inspire us to develop a hybrid embedding model which incorporates a (non-Euclidean) localized adjustment term (LAT)

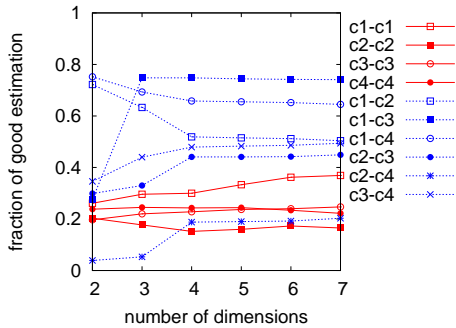


Figure 14: The average fraction of good estimations within/between clusters. Virtual Landmark on the King462 dataset

into the distance estimation. We show that using only a 2-dimensional Euclidean embedding plus the localized adjustment terms, we can obtain better performance than a pure Euclidean embedding with 7 dimensions.

5.1 The Hybrid Model

The basic ideas behind our hybrid model are as follows: we first embed the network distances in a Euclidean space of d dimensions, and then for each node we compute an adjustment term to account for the (local) non-Euclidean effect. Hence in our hybrid model, each node x has a d -dim Euclidean coordinate, (x_1, x_2, \dots, x_d) , and a (non-Euclidean) adjustment e_x : we use $(x_1, x_2, \dots, x_d; e_x)$ to denote the total “coordinate” of node x . The distance d_{xy} between two nodes x and y is then estimated by $\hat{d}_{xy} := d_{x,y}^E + e_x + e_y$, where $d_{x,y}^E = \sqrt{\sum_{k=1}^d (x_k - y_k)^2}$ is the Euclidean distance between x and y in the embedded d -dim Euclidean space. e_x is similar to the height vector in Vivaldi system [4], but actually it is quite different as can be discussed later in this section. The key question in this model is how to define and determine e_x for each node x . Ideally, we would like e_x to account for the “non-Euclidean” effect on the distance estimation errors to nodes within its own cluster. However, this requires us to know which cluster node x is in as well as the other nodes in its cluster. For simplicity, we derive e_x using all nodes as follows. We first compute ϵ_x , which minimizes the error function $E(x) = \sum_y (d_{xy} - (d_{xy}^E + \epsilon_x))^2$, where d_{xy} is the actual distance between x and y . It can be shown that the optimal ϵ_x is given by the average error in estimation:

$$\epsilon_x = \frac{\sum_y (d_{xy} - d_{xy}^E)}{n}. \quad (2)$$

We then set e_x to the half of ϵ_x , namely, $e_x = \epsilon_x/2$. In other words, \hat{d}_{xy} can be re-written as $d_{x,y}^E + \frac{(\epsilon_x + \epsilon_y)}{2}$. In short, we adjust the Euclidean estimation by the average of the two error terms of x and y . We have the following theorem that establishes the advantage of the hybrid model. The sketchy of the proof is in the appendix.

THEOREM 4. *The hybrid model using a d -dim Euclidean space and the adjustment term defined above reduces the squared stress of a pure d -dim Euclidean embedding by*

$$\frac{4n \sum_x e_x^2 + 2n^2 \text{Var}(e_x)}{\sum_{x,y} d_{x,y}^2} \geq 0,$$

where $\text{Var}(e_x) = \sum_x e_x^2/n - \left(\frac{\sum_x e_x}{n}\right)^2$.

Hence the larger the individual adjustment term, $|e_x|$ (thus the average estimation error for each node x using the pure Euclidean embedding), the more performance gain the hybrid model attains. It should be noted that e_x can be positive or negative⁵.

In (2), e_x is determined by the measurement to all the other nodes in the system. In practice, however, this is not feasible nor scalable. Instead, we compute \tilde{e}_x based on *sampled* measurements to a small number of randomly selected nodes. Let S denote the set of randomly sampled nodes. Then

$$\tilde{e}_x = \frac{\sum_{y \in S} (d_{xy} - d_{xy}^E)}{2|S|}, \quad (3)$$

Hence in practice the hybrid model works as follows: a) A number of landmarks are pre-selected and perform distance measurements among themselves to obtain a distance matrix. Using either Virtual Landmark or GNP a d -dim Euclidean embedding of the landmarks is obtained and their coordinates are determined. b) Each node x measures their distance to the landmarks and computes its d -dim Euclidean coordinate (x_1, x_2, \dots, x_d) ; it then measures its distance to a small number of randomly selected nodes, and computes \tilde{e}_x using eq. (3).

Note that in a sense, the adjustment term is similar to the “height vector” introduced in Vivaldi [4]. However, there are several key differences. First of all, the computation of the local adjustment term is very simple, and does not depend on the adjustment term of other nodes. Hence it does not require any iterative process to stabilize the adjustment term. In contrast, in Vivaldi, partly due to its distributed nature, a small change in the height vector of a node would affect the height vectors of the other nodes, and requires an iterative process to stabilize the height vectors of all nodes. Second, the local adjustment terms *provably* improve the performance of network distance embedding, as shown in the above theorem. Another good feature of the local adjustment term is that it can be used with any other schemes, not just the coordinate based schemes. As long as d_{xy}^E is the estimated distance based on the original scheme, the adjustment term can be computed as described above. In this sense, LAT is an *option* that can be used in conjunction with other schemes rather than a totally new scheme. Note that LAT can be used even with Vivaldi.

5.2 Evaluation

We evaluate the performance gain obtained by using the localized adjustment term (LAT) option in network distance embedding. For this purpose, we compare the stress of the Virtual Landmark method without LAT and the Virtual Landmark method with LAT, where the local adjustment term is computed using all the nodes. We vary the number of dimensions from 2 to 7. As can be seen in Fig. 15, the use of adjustment term (keys with LAT) reduces the stress significantly compared to the VL-All without LAT. In particular, when the original Euclidean embedding has high stress (large error), the reduction of stress is significant, which is expected from Theorem 4. In fact, increasing

⁵It is possible that the estimated distance is negative due to negative LAT. In this case, we use the estimation of the Euclidean part as the estimated distance.

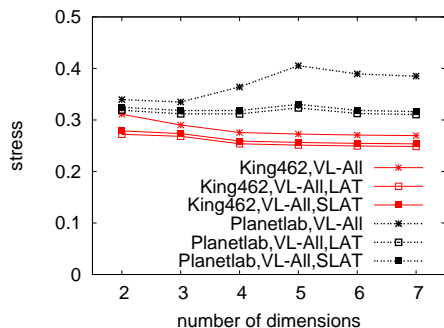


Figure 15: Stress of Virtual landmarks method over the number of dimensions. Both LAT and SLAT options are shown together.

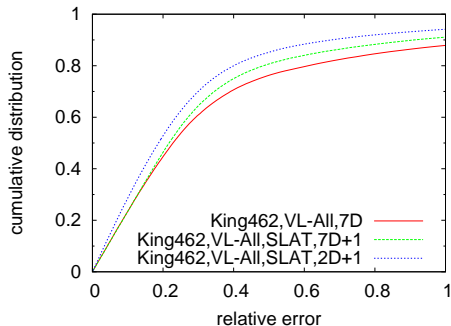


Figure 16: Cumulative distribution of the relative errors with different method on King462 data set.

the dimensionality of the Euclidean space does not help very much; a lower dimension Euclidean embedding plus the local adjustment terms is sufficient to improve the accuracy of the embedding significantly.

Next, we evaluate the performance of LAT using only a small number of randomly selected nodes as in (3); we call this option “SLAT (Sampled LAT)”. Fig. 15 shows the stress of embedding using SLAT (keys with SLAT) over different number of dimensions, where the adjustment term is computed using the measurement to 10 randomly selected nodes. We see that the performance between LAT and SLAT are very close. This is quite expected because the average of a randomly sampled set is an unbiased estimation of the average of the entire set. This result indicates that the adjustment term can actually be computed quickly with a small number of additional measurements.

In addition to the improved overall stress, the local adjustment terms also improve the relative errors. As an example, Fig. 16 compares the cumulative distribution of the relative errors of the pure Virtual Landmark with 7 dimensions (denoted as VL-All) with that using the same methods with *only 2 dimensions plus the SLAT* (denoted as SLAT (2D+1)) and 7 dimensions plus the SLAT (denoted as SLAT (7D+1)) for the King462 dataset⁶. The SLAT (2D+1) attains better performance than that of pure Virtual Landmark with 7 dimensions. For example, 90 percentile relative error of

⁶The Euclidean coordinates of the SLAT (2D+1) are the first 2 coordinates of the Virtual Landmark 7 dimension embedding.

SLAT (2D+1) is less than 0.6, but that of pure VL-All is larger than 1.0. The SLAT (2D + 1) is even better than SLAT (7D + 1), where all the 7 dimensions of the Virtual Landmark embedding is used for SLAT. This suggests that adding an adjustment term can perform better than adding additional dimensions. We have looked at the result more carefully and have seen that the performance gain comes largely from improved distance estimation for nodes within the same cluster.

6. CONCLUSION

This paper investigates the suitability of the Euclidean embedding of the network distances. We show that based on matrix perturbation theory, the existence of the large negative eigenvalues in classical scaling indicate that the data set cannot be embedded into the Euclidean space without considerable errors. By looking at the eigenvalues and the amount of TIVs, we show that network distances do not naturally arise from the Euclidean space. Furthermore, we show that the intra-cluster distances tend to have more TIVs, which shows strong local non-Euclidean effect.

Based on these insights, we have proposed and developed a simple hybrid model that incorporates a localized (non-Euclidean) adjustment term for each node on top of a low-dimensional Euclidean coordinate system. Our hybrid model preserves the advantages of the Euclidean coordinate systems, while improving their efficacy and overheads (by using coordinates with lower dimensions). This model is proven to reduce the estimation errors in terms of stress. In addition, our model can be incorporated into any embedding system (not necessarily Euclidean embedding).

7. ACKNOWLEDGMENTS

Sanghwan Lee and Zhi-Li Zhang were supported in part by the National Science Foundation grants ITR-0085824 and CNS-0435444. The authors would also like to thank Prof. Dan Boley of University of Minnesota for valuable discussion on matrix perturbation theory, in particular, for pointing out the Wiedlandt-Hoffman Theorem. Part of this work was done when Sanghwan Lee was in IBM T.J. Watson Research Center .

8. REFERENCES

- [1] T.S. Eugene Ng and Hui Zhang. Predicting Internet network distance with coordinates-based approaches. In *Proc. IEEE INFOCOM*, New York, NY, June 2002.
- [2] Liying Tang and Mark Crovella. Virtual landmarks for the Internet. In *Proceedings of the Internet Measurement Conference (IMC)*, Miami, Florida, October 2003.
- [3] Hyuk Lim, Jennifer C. Hou, and Chong-Ho Choi. Constructing Internet coordinate system based on delay measurement. In *Proceedings of the Internet Measurement Conference (IMC)*, Miami, Florida, October 2003.
- [4] Frank Dabek, Russ Cox, Frans Kaashoek, and Robert Morris. Vivaldi: A decentralized network coordinate system. In *Proceedings of ACM SIGCOMM 2004*, Portland, OR, August 2004.
- [5] Manuel Costa, Miguel Castro, Antony Rowstron, and Peter Key. Pic: Practical Internet coordinates for

distance estimation. In *Proceedings of International Conference on Distributed Computing Systems (ICDCS)*, Tokyo, Japan, March 2004.

- [6] Han Zheng, Eng Keong Lua, Marcelo Pias, and Timothy G. Griffin. Internet routing policies and round-trip-times. In *The 6th annual Passive and Active Measurement Workshop*, Boston, MA, March 2005.
- [7] Eng Keong Lua, Timothy Griffin, Marcelo Pias, Han Zheng, and Jon Crowcroft. On the accuracy of embeddings for internet coordinate systems. In *Proceedings of the Internet Measurement Conference(IMC)*, Boston, MA, April 2005.
- [8] Meridian: A lightweight network location service without virtual coordinates. Philadelphia, PA, August 2005.
- [9] Ingwer Borg and Patrick Groenen. *Modern Multidimensional Scaling : Theory and Applications*. Springer, 1997.
- [10] Gene H. Golub and Charles F. van Loan. *Matrix Computation*. the John Hopkins University Press, 3rd edition, 1996.
- [11] King462 data set. <http://pdos.lcs.mit.edu/p2psim/kingdata>.
- [12] King2305 data set. <http://www.cs.cornell.edu/People/egs/meridian/data.php>.
- [13] Jeremy Stribling. Rtt among planetlab nodes. http://www.pdos.lcs.mit.edu/strib/pl_app/.
- [14] Global network positioning (gnp). <http://www-2.cs.cmu.edu/eugeneng/research/gnp/>.
- [15] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering : Analysis and an algorithm. *Advances in Neural Information Processing Systems (NIPS)*, 14, 2002.
- [16] Douglas B. West. *Introduction to Graph Theory*. Prentice Hall., second edition, 2001.

APPENDIX

A. PROOFS

Proof of Theorem 2 Verifying whether there exist a maximal TIV-free set of size k among a set of nodes with a distance matrix can be done in polynomial time by enumerating all set of size k and checking the TIVs in a non-deterministic machine. Hence this problem is *NP*.

Now we prove that the problem is NP-hard. We prove NP-hard by transforming CLIQUE (a problem to find the maximal clique in a graph G , which is known to be NP-complete in Karp[1972] of [16]), to the maximal TIV-free set problem.

Let G be a connected undirected graph with $n > 2$ nodes. We assume that the size of maximal clique of G is $k > 2$. When $k = 2$, it is trivial case such that any pair of vertices with edges is a maximal clique. We construct a distance matrix $D = (d_{ij})$ among the set of vertices of G as follows, where d_{ij} will be the defined distance between vertices i and j . For each vertex i , we set $d_{ii} = 0$. For each edge e_{ij} between vertices i and j , we set $d_{ij} = 1$ and $d_{ji} = 1$. Note that for any triangle in G , the corresponding distances in D do not violate triangle inequality. For the pair of vertices i and j that does not have an edge between them in G , we set $d_{ij} = \text{undefined}$. Now, we define all the undefined d_{ij} as follows. For an undefined d_{ij} , we compute $c = \max_k (d_{ik} + d_{kj})$ for all k such that d_{ik} and d_{kj} are already defined. If no such c can be computed because d_{ik} and d_{kj} are undefined for all k , we set $c = 0$. Then, we set $d_{ij} = d_{ji} = c + 1$. We define the undefined d_{ij} 's until all d_{ij} is defined. This transformation takes

polynomial time, $O(n^3)$, because there are n^2 entries in D and for each entry $O(n)$ computation is required.

It can be easily shown that that a triple of nodes (i, j, k) in G forms a triangle if and only if i, j , and k do not have triangle inequality violations with d_{ij} , d_{ik} , and d_{jk} in D (we omit the details for the sake of space). This means that the maximal TIV-free set whose distance are defined as D is the maximal clique in G . We conclude that maximal TIV-free set problem is NP-hard. Since maximal TIV-free set problem is NP and NP-hard, it is NP-complete. ■

Proof of Lemma 1 Note that $c > a + b$. Let \hat{a} , \hat{b} , and \hat{c} be the distances among the 3 nodes after the Euclidean embedding. Then the squared estimation error e is

$$e = (a - \hat{a})^2 + (b - \hat{b})^2 + (c - \hat{c})^2 \quad (4)$$

Suppose $|a - \hat{a}| + |b - \hat{b}| + |c - \hat{c}| = k$. Then, $e \geq (\frac{k}{3})^2 + (\frac{k}{3})^2 + (\frac{k}{3})^2 = \frac{k^2}{3}$, where

$$|a - \hat{a}| = \frac{k}{3}, \quad |b - \hat{b}| = \frac{k}{3}, \quad |c - \hat{c}| = \frac{k}{3} \quad (5)$$

The sum of squared error e is minimized when k is minimized. The range of k is determined by the triangle inequality constraints among \hat{a} , \hat{b} , and \hat{c} . In other words, it should satisfy

$$\hat{a} + \hat{b} \geq \hat{c}, \quad \hat{a} + \hat{c} \geq \hat{b}, \quad \hat{b} + \hat{c} \geq \hat{a} \quad (6)$$

Based on (5), (6), and $(c > a + b)$, we have the lower bound of $k \geq (c - a - b)$, when $\hat{a} = a + \frac{k}{3}$, $\hat{b} = b + \frac{k}{3}$, and $\hat{c} = c - \frac{k}{3}$. So the squared estimation error $e \geq \frac{(c-a-b)^2}{3}$. ■

Proof of Theorem 3 Let E be the sum of squared error of n nodes. $E = \sum_i (\hat{d}_i - d_i)^2$, where d_i is a distance between a pair of nodes (called i) and \hat{d}_i is the embedded distance of the pair i . It should be noted that there are $n(n-1)/2$ pairs. Since there are $n(n-1)(n-2)/6$ triples among n nodes, E can be rewritten by the triples of nodes as follows. $E = \frac{1}{n-2} \sum_{t \in T} ((\hat{t}_a - t_a)^2 + (\hat{t}_b - t_b)^2 + (\hat{t}_c - t_c)^2)$, where T is the set of triples and t_a, t_b , and t_c are the three distances of a triple t , and \hat{t}_a, \hat{t}_b , and \hat{t}_c are the corresponding embedded distances. E should be larger than the sum of squared errors among the the triples of TIV and embedding each triple independently has smaller errors than embedding all the node at the same time, $E \geq \frac{1}{3(n-2)} \sum_{t \in V} (t_c - t_a - t_b)^2$ based on Lemma 1, where V is the set of triples with TIVs. ■

Proof of Theorem 4 We just describe a sketchy of the proof. Let s_1 be the stress of using the pure Euclidean based scheme. Let s_2 be the stress of using the pure Euclidean based scheme with the adjustment term.

$$s_1^2 = \frac{\sum_{x,y} (d_{xy} - d_{xy}^E)^2}{\sum_{x,y} d_{xy}^2} \quad (7)$$

$$s_2^2 = \frac{\sum_{x,y} (d_{xy} - d_{xy}^E - e_x - e_y)^2}{\sum_{x,y} d_{xy}^2} \quad (8)$$

Since the denominators are the same, we compute $(\sum_{x,y} d_{xy}^2)(s_1^2 - s_2^2)$ to compute $s_1^2 - s_2^2$. Using (2) and some reformatting the formula, the final result can be easily obtained. ■