

# NADeMaC: A Simple Non-Negative Decentralized Completion Algorithm for Internet Latency Matrix



Cong Wang  
School of Digital Media, Sichuan Normal University  
China  
[wongcong@gmail.com](mailto:wongcong@gmail.com)

**ABSTRACT:** In this paper, the non-negative decentralized completion problem of the in-complete Internet latency matrix is studied. On the basis of the low-rank approximation of the matrix, we componentized this problem into a couple of convex optimization problems by estimating the  $l_0$  norm of this matrix, and then solved it by alternative direction algorithm. Owing to the asymmetry and the negative definite characteristic of the matrix caused by the difference between autonomous system routing strategies, some negative entries inevitably exist in the completed matrix. Unlike traditional non-negative completion algorithms, this paper does not try to prevent the generation of the negative entries. As a replacement, this paper presents a novel and much simpler non-negative ensuring scheme named NADeMaC, which calibrates the negative entries by a prior positive estimation value after they appeared. Theoretical analysis shows that the accuracy of our algorithm is at least no less than traditional methods, and furthermore our experiments show that our method is far better than traditional non-negative ensuring scheme.

**Keywords:** Latency Matrix Completion, Sparse Model, Internet Measurement, Optimization

**Received:** 18 June 2015, Revised 20 July 2015, Accepted 26 July 2015

© 2015 DLINE. All Rights Reserved

## 1. Introduction

Many kinds of Internet applications, including Peer-to-Peer Networks[1], Multiplayer online games[2,3], Cloud computing[4,5], Wireless sensor networks[6] and even The Onion Router(Tor) [7,8,9], their performances rely a lot on the latencies between each couple of nodes. To optimize the system performance, a key issue is how to get the latency matrix of the system. Although nodes can ping or trace route each other to fit all the entries of the matrix, the complexity of these measure methods is up to  $O(n)^2$ . As a result, these methods cannot update the matrix real-timely while only about 500 nodes exist in the system [10], while for many Internet applications, there are always millions of nodes are online simultaneously.

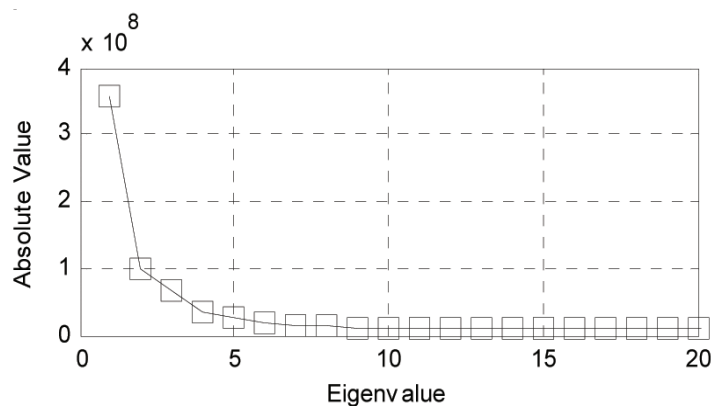
It has been widely concerned about how to complete the latency matrix of the system from limited measurement data accurately. Early studies of the latency matrix completion problem embeds all the nodes of the system into a specific metric space, then the real latency between any couple of nodes can be estimated by the corresponding spatial distance of the space. In general, this kind of studies is well known as Network Coordinate System, NCS. NCS can estimate all the latencies with acceptable accuracy, and has been deployed in some Internet applications [11]. A potential assumption of NCS is, nodes can be embedded into the metric space ideally. But for Internet, packages between a couple of nodes sometimes will be routed on a longer path while a shorter path between those two nodes is available. This phenomenon always causes the Triangle Inequality Violations, TIVs. Which means that for node a, b, and c, the latency between node a and b,  $d_{a,b}$  maybe larger than the sum of

$d_{a,c}$  and  $d_{b,c}$ . Another problem of NCS is the asymmetry of the Internet routing polices, e.g. a routing path is always different from the reverse path, which yields an inequation  $d_{a,b} \neq d_{b,a}$  while in NCS there must be  $d_{a,b} = d_{b,a}$ . And even worse, the non-convexity of the optimization problem in NCS always leads to get a local optimal value. Recent studies show that the Internet latency matrix is approximately sparse [12]. The relationship between the sparsity of the latency matrix and the metric space dimension of NCS is also studied well in [13], which indicates that NCS is a special kind of completion method of latency matrix. In full distributed environment, we can assign each node  $i$  a  $n$ -dimensional row vector  $\mathbf{u}_i$  and a  $n$ -dimensional column vector  $\mathbf{v}_i$ , then the empty entry in the Internet latency matrix  $d_{i,j}$ , e.g. the latency between node  $i$  and  $j$  can be estimated by  $\mathbf{u}_i \cdot \mathbf{v}_j$ . Comparing with NCSs, this method has three advantages: Firstly, it can be reformulated into a couple of convex problems to solve; Secondly, the TIVs can be easily handled since the restricted condition  $d_{a,b} + d_{b,c} \geq d_{a,c}$  in NCS does no longer exist; Thirdly, the completion matrix is not necessarily a symmetry one, thus the asymmetry of the Internet routing polices can be captured well. IDES [14] is the first algorithm which based on this idea. But the multi-manifold property of the Internet latency space [15] leads to ill-posed matrixes inversions computing inevitably, and then the accuracy of the latency estimation is dropped significantly. DMF [16] and DMFSGD [17] algorithm imports a regularization factor to get a biased estimation of the latency through minimize the mean square error of the estimation error, which enhances the robustness substantially. On the basis of DMF, Phoenix algorithm proposes a non-negative ensuring scheme which can eliminate the negative entry existing in the completed matrix [18]. But, the non-negative ensuring algorithms will drop the latency estimation accuracy [19]. While for some specific applications, such as the Dijkstra shortest path finding algorithm, even only one negative latency estimation that also crashes the whole iterative process. Thus it is necessary to study how to complete the latency matrix without losing any accuracy while the non-negative property is still be ensured.

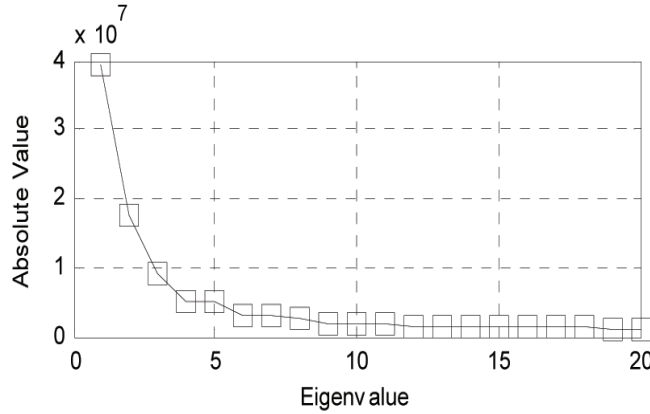
In this paper, we firstly analyze the non-negative latency matrix completion problem theoretically, and draw a conclusion that traditional non-negative completion algorithms cannot bring us higher accuracy. And then, we propose a new Non-negative Adaptive Decentralized Matrix Completion algorithm, NADeMaC, to deal with the negative entries in the completed matrix. Our scheme does not try to avoid the appearance of the negative entries, as a replacement, we calibrate the negative entries with a specific value while it appears. Theoretical analysis proves that the accuracy of NADeMaC is at least no less than that of traditional methods. Moreover, for those negative entries in the completed matrix, our algorithm achieves much higher accuracy than tridational methods without dropping any other performance indexes, while the computational cost of our algorithm is also negligible.

## 2. Related works

Firstly we review the basic idea of the Internet latency matrix completion problem: For an Internet application which contains  $P$  nodes, there must be a  $P$  row and  $P$  column latency matrix  $\mathbf{D}$  with  $P \times P$  entries. Each entry  $d_{i,j}$  of  $\mathbf{D}$  represents the latency between node  $i$  and  $j$ . Be limited by the system scale and the computing ability, node can hardly get the latencies to all other nodes directly and realtimely. Thus we can only get an incomplete matrix  $\mathbf{D}'$  since some entries in  $\mathbf{D}$  are missing. But the approximately sparsity of the latency matrix provides another possibility. The approximately sparsity means that only a few eigenvalues of the matrix have significant absolute values, while other eigenvalues are close to zero. The top 20 largest eigenvalues of the latency matrixes of Planetlab[20] and King[21] datasets are shown in Figure.1:



(a) King dataset



(b) Planetlab dataset

Figure 1. The top 20 largest eigenvalues of the latency matrixes of King and Planetlab datasets

By utilizing the limited information provided by  $\mathbf{D}'$ , we can get a completed matrix  $\hat{\mathbf{D}}$  as a good fitting of  $\mathbf{D}$ . Then, each entry  $d_{i,j}$  in  $\mathbf{D}$ , e.g. the real latency between node  $i$  and  $j$ , can be estimated by the corresponding entries  $\hat{d}_{i,j}$  in  $\hat{\mathbf{D}}$ .

In full decentralized environment, extracting a global consistent latency matrix is difficult. But in consideration of the approximate sparsity of the latency matrix  $\mathbf{D}$ , in general it is much easier for reaching a consensus about the prior estimation  $N$  of the  $I_o$  norm of the completed latency matrix  $\hat{\mathbf{D}}$ , which can guarantee the rigid sparsity of  $\hat{\mathbf{D}}$ . In this time, the latency matrix can be completed by solving the following optimization problem:

$$\min_{\hat{\mathbf{D}}} \left( L \left( P_{\Omega} (\hat{\mathbf{D}} - \mathbf{D}') \right) \right), \text{ s.t. } \|\hat{\mathbf{D}}\|_0 = N \quad (1)$$

Where  $P_{\Omega}(\cdot)$  extracts those entries which exist in  $\mathbf{D}'$  and the corresponding entries in  $\hat{\mathbf{D}}$  to join the computing process;  $L(\cdot)$  is the pre-defined loss function;  $\|\cdot\|_0$  computes the  $I_o$  norm of a matrix, e.g. the number of non-zero eigenvalues. It is known that  $\hat{\mathbf{D}}$  can be expressed as a multiplier of two  $P \times N$  matrixes  $\mathbf{U}$  and  $\mathbf{V}$ :

$$\hat{\mathbf{D}} = \mathbf{U} \cdot \mathbf{V}^T \quad (2)$$

Hence the entry  $d_{i,j}$  in  $\mathbf{D}$  can be estimated by the corresponding entry  $\hat{d}_{i,j}$  in  $\hat{\mathbf{D}}$ :

$$d_{i,j} \approx \hat{d}_{i,j} = \mathbf{u}_i \cdot \mathbf{v}_j^T = u_{i,1} \times v_{j,1} + \dots + u_{i,N} \times v_{j,N} \quad (3)$$

Where  $u_{i,n}$ ,  $v_{j,n}$  is the  $n$ -th corresponding entry in vector  $\mathbf{u}_i$  and  $\mathbf{v}_j$ . Then,  $\mathbf{U}$  and  $\mathbf{V}$  can be dispatched to each node, and be maintained by all the nodes: We can assign each node  $i$  the corresponding vectors  $\mathbf{u}_i$  and  $\mathbf{v}_j$ , then node  $i$  can estimate the latency between any other node  $j$  and itself by getting the vectors holded by  $j$ . Correspondingly, Eq(1) can be factorized to the following couple of sub-problems:

$$\left\{ \begin{array}{l} \min_{\mathbf{u}_i} \left( \sum_{(i,j) \in \Omega} l(d_{i,j} - \mathbf{u}_i \cdot \mathbf{v}_j^T) \right) \\ \min_{\mathbf{v}_j} \left( \sum_{(j,i) \in \Omega} l(d_{j,i} - \mathbf{u}_j \cdot \mathbf{v}_i^T) \right) \end{array} \right\} \quad (4)$$

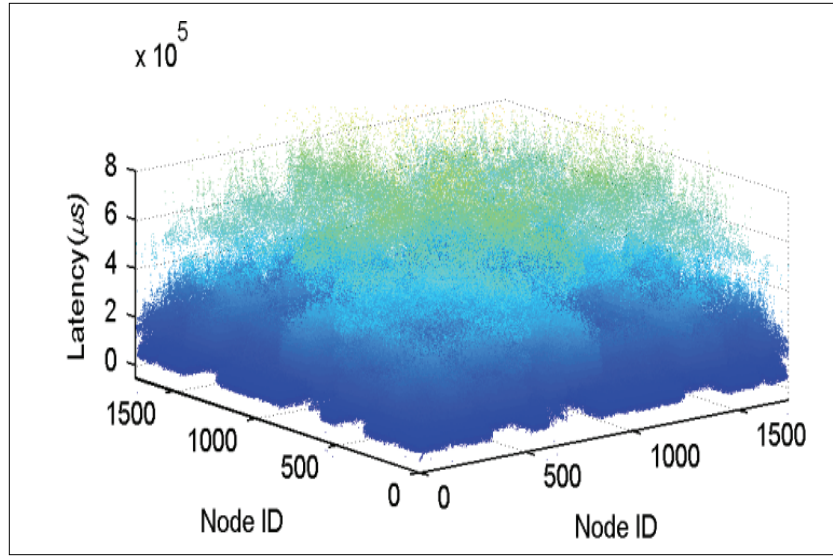
While  $l(\cdot)$  is a convex function, the global optimum of this couple of problems can be obtained numerically. When taking the robustness of the numerical solutions into consideration, a regularized factor can be introduced:

$$\left\{ \begin{array}{l} \min_{\mathbf{u}_i} \left( \lambda \mathbf{u}_i \cdot \mathbf{u}_i^T + \sum_{(i,j) \in \Omega} l(d_{i,j} - \mathbf{u}_i \cdot \mathbf{v}_j^T) \right) \\ \min_{\mathbf{v}_j} \left( \lambda \mathbf{u}_j \cdot \mathbf{u}_j^T + \sum_{(j,i) \in \Omega} l(d_{j,i} - \mathbf{u}_j \cdot \mathbf{v}_i^T) \right) \end{array} \right\} \quad (5)$$

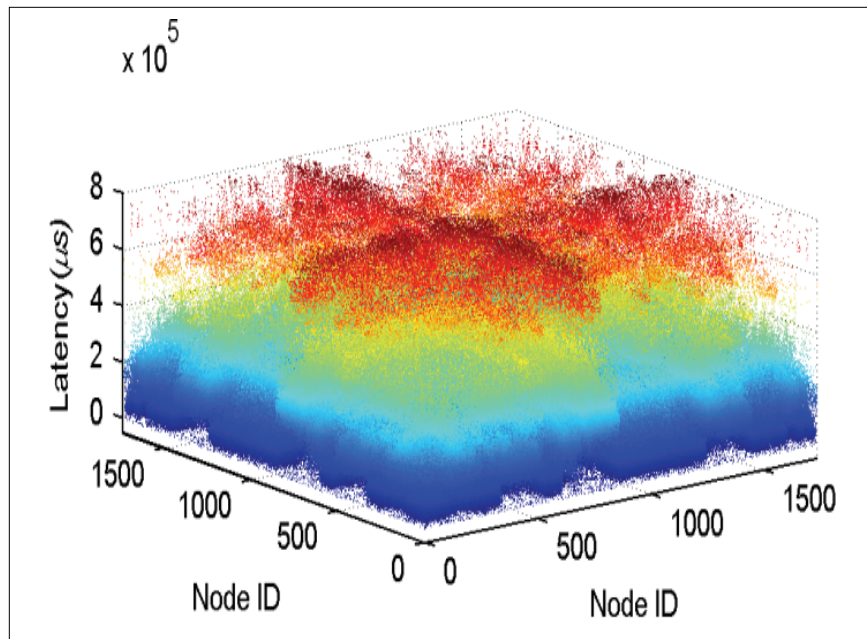
Obviously that couple of optimization problems is also a convex one though the regularized factors  $\lambda \mathbf{u}_i \mathbf{u}_i^T$  and  $\lambda \mathbf{v}_j \mathbf{v}_j^T$  are introduced. It can be easily solved through the alternating direction method of multipliers. DMFSGD algorithm proposes a kind of alternating direction sub-gradient descending method to update  $\mathbf{u}_i$  and  $\mathbf{v}_j$  alternatively:

$$\left. \begin{aligned} \mathbf{u}_i^{(t+1)} &= (1 - \eta_u \lambda) \mathbf{u}_i^{(t)} + \eta_u \sum_{(j,l) \in \Omega} \frac{\partial l(\mathbf{u}_i)}{\partial \mathbf{u}_i} \\ \mathbf{v}_j^{(t+1)} &= (1 - \eta_v \lambda) \mathbf{v}_j^{(t)} + \eta_v \sum_{(i,l) \in \Omega} \frac{\partial l(\mathbf{v}_j)}{\partial \mathbf{v}_j} \end{aligned} \right\} \quad (6)$$

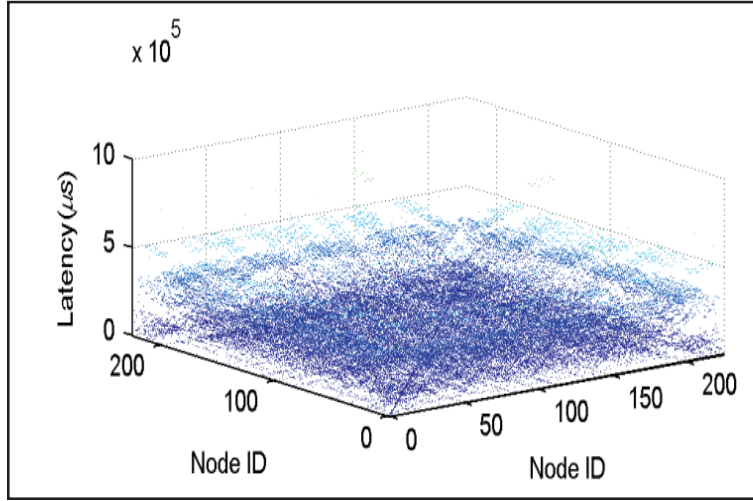
Where  $\eta_u$  and  $\eta_v$  is the sub-gradient descending length. Through this method node can easily complete the latency matrix. The original and the completed latency matrixes are shown in Figure. 2. Intuitively, the completed matrixes are very similar with the originals.



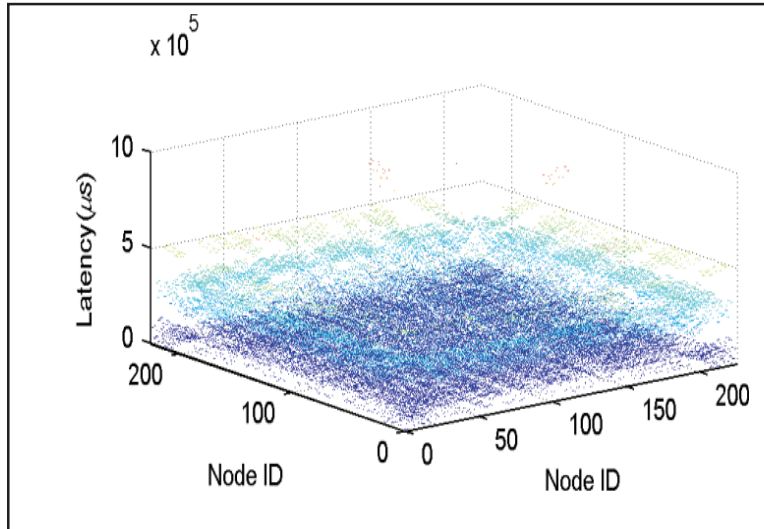
(a) The original latency matrix of King



(b) The completed latency matrix of King



(c) The native latency matrix of Planetlab



(d) The completed latency matrix of Planetlab

Figure 2. The native and completed latency matrixies of King and Planetlab datasets

### 3. Traditional Non-negative Ensuring Scheme

When it comes to the non-negative matrix completion problem, the following constrained optimization problem should be solved:

$$\begin{aligned} & \min_{\mathbf{u}_i \in \mathbb{R}^{+N}} \left( \lambda \mathbf{u}_i \cdot \mathbf{u}_i^T + \sum_{\langle i,j \rangle \in \Omega} l_2(d_{i,j} - \mathbf{u}_i \cdot \mathbf{v}_j^T) \right) \\ & \min_{\mathbf{v}_i \in \mathbb{R}^{+N}} \left( \lambda \mathbf{v}_i \cdot \mathbf{v}_i^T + \sum_{\langle j,i \rangle \in \Omega} l_2(d_{j,i} - \mathbf{u}_j \cdot \mathbf{v}_i^T) \right) \end{aligned} \quad (7)$$

Phoenix solves these problems through interior points method. But the high computational cost limits the method. To deduce the cost, a kind of non-negative matrix factorization algorithm, NMF is adpoted in DMFSGD. It forces the negative entries exist in  $\mathbf{u}_i$  and  $\mathbf{v}_i$  to be zero during the whole updating process:

```

For (n=0;n<N;n++)
  If  $u_{i,n} < 0$  then
     $u_{i,n} = 0$ ;
  End
  If  $v_{i,n} < 0$  then
     $v_{i,n} = 0$ ;
  End
End

```

Algorithm 1. The non-negative ensuring scheme of DMFSGD

Where  $u_{i,n}$  and  $v_{i,n}$  are the n-th entry of vector  $\mathbf{u}_i$  and  $\mathbf{v}_i$  respectively. Obviously this is a variation of the gradient projection method. It can be proved that both Phoenix and DMFSGD deduce the estimation accuracy:

We assume the optimums of Equations(5) are  $\mathbf{u}_i^{(opt)}$  and  $\mathbf{v}_i^{(opt)}$ , and the corresponding value of the loss function is  $E_{opt}$ ; correspondingly we also assume the optimums of Equations (7) are  $\mathbf{u}_i^{(opt)}$  and  $\mathbf{v}_i^{(opt)}$ , and the corresponding value of the loss function is  $\hat{E}_{opt}$ . Thus we can get the following relationship since  $E_{opt}$  is the global optimum while  $\hat{E}_{opt}$  isn't:

$$E_{opt} \leq \hat{E}_{opt} \quad (8)$$

This is to say, the non-negative ensuring scheme cannot improve the latency estimation accuracy. This conclusion is also confirmed experimentally by [19]. But, in consideration of some specific applications mentioned before, to develop a new non-negative ensuring scheme without losing any accuracy is still necessary.

#### 4. Research Method

We propose a novel and simple non-negative ensuring algorithm NADeMaC. Unlike traditional methods, we focus on how to calibrate the negative estimation values after they are generated in the updating process rather than prevent the generations of them. Our calibrating method is very simple: Replace the negative entries with a small positive value or even zero. Theoretical analysis shows the upper estimation error bound of NADeMaC is not higher than the lower bound of traditional method. The whole process of NADeMaC is shown in Algorithm 2:

Where  $\tau$  is a pre-defined positive value which is as small as the shortest latency appears in real environment. We set  $\tau$  equal to a small positive value rather than zero since for any couple of nodes, the latency between them should not be zero absolutely. Obviously our algorithm is much simpler than any previous algorithms. In programmatic view, only just a simple ternary operator  $\hat{d}_{i,j} = (\hat{d}_{i,j} \geq 0) ? \hat{d}_{i,j} : \tau$  can express our algorithm well, hence its computation cost are almost negligible. Although the non-negative ensuring scheme of DMFSGD looks also simple, it must have to run throughout the whole updating process while ours needn't. The effectiveness of NADeMaC can be proved as follows:

```

Updating  $\mathbf{u}_i$  and  $\mathbf{v}_i$  periodically;
  While estimating  $d_{i,j}$ 
     $d_{i,j} = \mathbf{u}_i \cdot \mathbf{v}_j^T$ ;
    If  $d_{i,j} > 0$ 
      Continue;
    Else
       $d_{i,j} = \tau$ ;
    End
  End
End

```

Algorithm 2. The NADeMaC Algorithm



Let's assume that there exists an entry  $\hat{d}_{i,j}$  in the completed matrix  $\hat{\mathbf{D}}$  which satisfies  $\hat{d}_{i,j} < 0$ . If the corresponding real latency is  $d_{i,j}$ , we can get:

$$\begin{cases} d_{i,j} > 0 \\ \hat{d}_{i,j} < 0 \end{cases} \quad (9)$$

Then, if the non-negative ensuring scheme is not adopted, the estimation error of  $d_{i,j}$  can be expressed as:

$$e_{i,j} = \text{abs}(d_{i,j} - \hat{d}_{i,j}) \quad (10)$$

We assume the estimation error of NADeMaC is  $\tilde{e}_{i,j}$ , then we can get the following relationship since  $\hat{d}_{i,j} < 0$  and  $r < d_{i,j}$ ;

$$e_{i,j} = \text{abs}(d_{i,j} - \hat{d}_{i,j}) > d_{i,j} > \text{abs}(d_{i,j} - r) = \tilde{e}_{i,j} \quad (11)$$

If and only if  $\hat{d}_{i,j} \geq 0$  we can get:

$$e_{i,j} = \tilde{e}_{i,j} \quad (12)$$

Then the relationship between the total error  $\tilde{E}$  of NADeMaC and that of traditional method  $E$  is:

$$\tilde{E} = \sum_{\forall i,j} \tilde{e}_{i,j} \leq \sum_{\forall i,j} e_{i,j} = E \quad (13)$$

Equality will be achieved if and only if all the entries in the completed matrix are non-negative. Along with  $U$  equation (8) we get:

$$\tilde{E} \leq E \leq \hat{E} \quad (14)$$

Q.E.D.

## 5. Experiments

This section evaluates the NADeMaC algorithm and compares it to the baseline algorithms: DMFSGD with/without NMF algorithm.

At first Relative Error, RE is introduced as an evaluation criterion to evaluate the estimation accuracy:

The relative error value  $RE_{i,j}$  between nodes  $i$  and  $j$  is defined as:

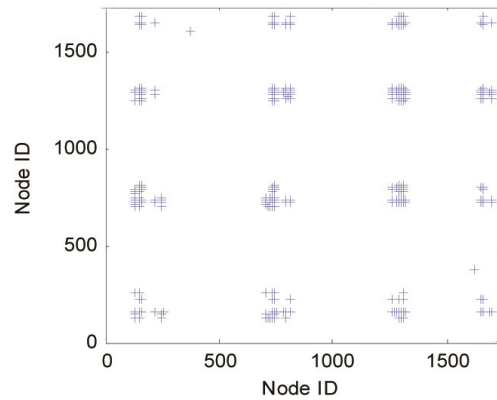
$$RE_{i,j} = \text{abs}\left(\frac{\hat{d}_{i,j} - d_{i,j}}{d_{i,j}}\right) \times 100\% \quad (15)$$

The Ninetieth Percentile Relative Error (NPRE) is a criteria derived from RE. It is the smallest RE value which cumulative distribution function, CDF value is no smaller than 90%.

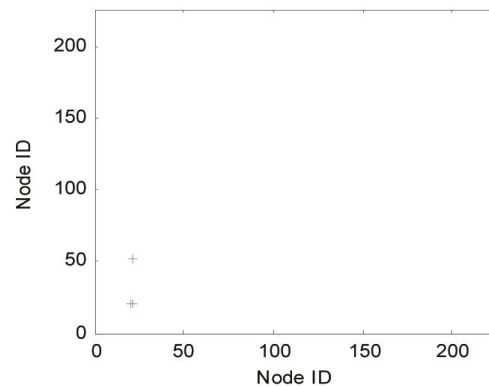
This article performs the experiments on the King and Planetlab Datasets. All the experiments of this article set the dimension of vector  $\mathbf{u}_i$  and  $\mathbf{v}_i$  to be 5.

From Figure 3, we can study how many negative entries exist in the completed latency matrix. For King dataset, there are 326 entries, only about 0.011% of the whole completed matrix entries are negative. Similarly, only 3 entries, about 0.005% of the whole completed matrix entries of the Planetlab dataset are negative. This phenomenon proves that the negative entries occupy a very small fraction of the completed matrix. Thus the accuracy of the non-negative preserving algorithms cannot be improved obviously. We tend to agree with the conclusion drawn by [18]: The meaning of the non-negative ensuring scheme is limited to the reasonable physical meaning, and the applications in some specific algorithms.

Figure 4 shows the CDF of the RE values of the DMFSGD, DMFSGD-NMF and the NADeMaC algorithm. It can be seen that the traditional non-negative ensuring scheme drops the estimation accuracy slightly. In King dataset, the NPRE is risen from 32.03% to 33.91% while the non-negative ensuring scheme is adopted. Similarly, in Planetlab dataset, the NPRE is also



(a) King dataset

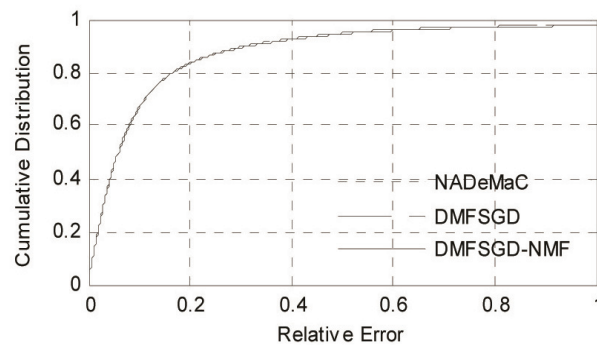


(b) Planetlab dataset

Figure 3. The locations of the negative entries in the completed matrixes

rised from 28.32% to 31.31% by the non-negative ensuring scheme. Experiments results confirm our theoretical analysis: the traditional non-negative ensuring scheme cannot improve the estimation accuracy. But on the other hand, from Fig. 4 we can also see that NADeMaC cannot improve the estimation accuracy significantly either since the negative entries are very few while our NADeMaC algorithm only act on this part of entries.

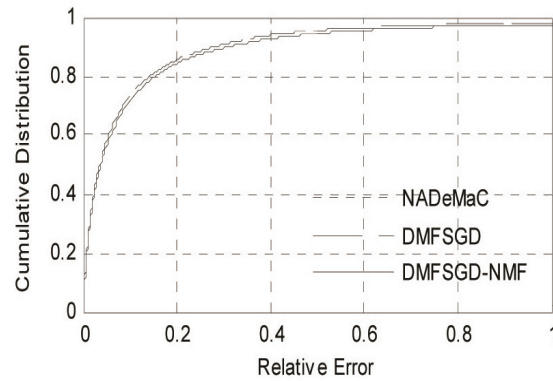
Now we study how the NADeMaC algorithm affects the negative entries. Fig. 5 shows the comparison of the CDF of the RE of those parts of entries between NADeMaC and the baseline algorithms. It can be seen clearly that for those negative entries, the estimation values made by NADeMaC are much closer to the real latency than DMFSGD-NMF. While comparing with DMFSGD, our method has strict non-negative property.



(a) King dataset

(a) King dataset





(b) Planetlab dataset

Figure 4. The CDF of the RE

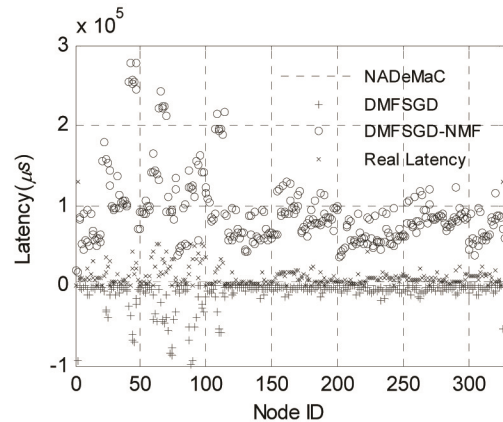


Figure 5. The comparison on the negative entries between different algorithms

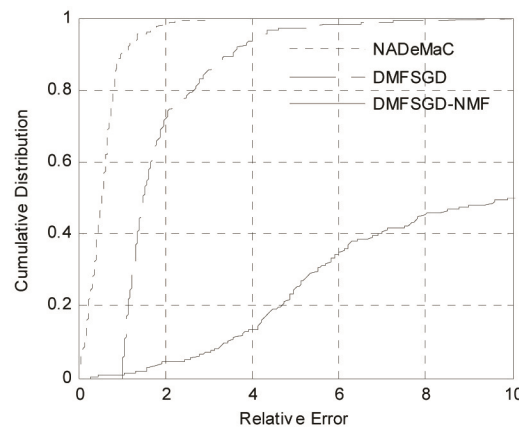


Figure 6. The CDF of the REs on the negative entries of different algorithms

As shown in Figure 5, negative estimations always occur on short latencies. In this case, the RE value up to about 100% can also be accepted since for short latencies the absolute error is still small. As the comparison shown in Fig. 6, for NADeMaC, there are 14.23% latency estimations have no more than 20% RE values while 89.74% estimations have no more than 100% RE values, while these two indexes of DMFSGD-NMF are only 0% and 0.99% respectively. As for DMFSGD, all the relative errors of estimations are more than 100% since the estimation value are all negative. For Planetlab dataset, there are only 3 negative entries in the completed matrix. We list the 3 entries in Table 2. Being similar with King dataset, NADeMaC shows far better performance than DMFSGD and DMFSGD-NMF.

Planetlab entries	[20,21]	[21,20]	[21,52]
Real latency(is)	1999	2042	51192
NADeMaC	5000	5000	5000
DMFSGD	-4443	-4604	-2135
DMFSGD-NMF	9619	11013	4154

Table 2. The negative entries in the completed matrix of Planetlab dataset

## 6. Conclusion

This paper argues the necessity of the non-negative preserving scheme runs throughout the whole latency matrix completion process. To satisfy the needs of some specific applications, this paper proposes a simple non-negative ensuring algorithm, NADeMaC. Unlike traditional methods, NADeMaC don't try to avoid the appearance of the negative entries in the completed matrix. By using some prior knowledge of the real latency, NADeMaC focus on how to calibrate the negative entries in the completed matrix after they are generated. Both the theoretical analysis and the experiments show that, the computation cost of NADeMaC is negligible, while the estimation accuracy of it is higher than traditional methods.

## References

- [1] Nazanin, M., Reza, R., Ivica, R., et al. (2014). ISP-Friendly Live P2P Streaming. *IEEE/ACM Transactions on Networking*, 22 (1) 244-256.
- [2] Armitage, G., Heyde, A. (2012). REED: Optimizing first person shooter game server discovery using network coordinates. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 8 (2) 20.
- [3] Jiang, JR., Wu ,JW., Fan, JY., et al. (2014). Immersive voice communication for massively multiplayer online games. Peer-to-Peer Networks Applications, (Already online but not assigned to an issue yet).
- [4] Agarwal, S., Dunagan, J., Jain, N., et al. (2010). Volley: Automated Data Placement for Geo-Distributed Cloud Services. In Proceedings of the 7th USENIX Symposium on Networked Systems Design & Implementation, San Jose, CA, USA, April 28-30, 1-16.
- [5] Ding, C., Chen, Y, Xu, T., et al. (2012). CloudGPS: a scalable and ISP-friendly server selection scheme in cloud computing environments, *In: Proceedings of the 2012 IEEE 20th International Workshop on Quality of Service*, Coimbra, Portugal. IEEE Press, 2012: 5.
- [6] Boon-Chong, S., Qin,g Z., Chuan, HF., Alvis, C. M., Fong. (2012). Hybrid RF Mapping and Kalman Filtered Spring Relaxation for Sensor Network Localization. *IEEE Sensors Journal*, 12 (5) 1427-1435.
- [7] Klein, A., Ishikawa, F., Honiden, S. (2013). Towards Network-Aware Service Composition in the Cloud. *In: Proceedings of the 21<sup>st</sup> International Conference on World Wide Web*, Lyon, France, 959-968.
- [8] Sherr, M., Mao, A., Marczak, WR, et al. (2010). A3: An Extensible Platform for Application-Aware Anonymity. *In: Proceedings of the Network and Distributed Security Symposium*, San Diego, U.S.A.
- [9] Wacek, C., Tan, H., Bauer, K., et al. (2013). An Empirical Evaluation of Relay Selection in Tor. *In: Proceedings of the Network and Distributed Security Symposium*, San Diego, U.S.A.
- [10] All Pair Ping Project . [http://pdos.csail.mit.edu/~strib/pl\\_app/](http://pdos.csail.mit.edu/~strib/pl_app/).
- [11] Frank, D., Russ, C., Frans, K., Robert, M. (2004). Vivaldi: A Decentralized Network Coordinate System. In proceedings of 2004 SIGCOMM, Portland, OR, USA,15-26.
- [12] Lee, S., Zhang, ZL, Sahu, S., et al . (2010). On Suitability of Euclidean Embedding for Host-Based Network Coordinate Systems. *IEEE/ACM Transactions on Networking*, 18 (1) 27-40.
- [13] Candes, EJ, Plan, Y. (2011). Tight Oracle Inequalities for Low-Rank Matrix Recovery from a Minimal Number of Noisy Random Measurements. *IEEE Transactions on Information Theory*, 57 (4) 2342-2359.

- [14] Mao, Y., Saul, LK., Smith, JM. (2006). IDES: An Internet Distance Estimation Service for Large Networks. *IEEE Journal on Selected Areas in Communications*, 24 (12) 2273-2284.
- [15] Wang, Z., Chen, M., Xing, C, et al. (2013). Multi-Manifold Model of the Internet Delay Space. *Journal of Network and Computer Applications*, 36 (1) 211-218.
- [16] Liao, Y., Geurts, P., Leduc, G. (2010). Network Distance Prediction Based on Decentralized Matrix Factorization. NETWORKING 2010. Berlin Heidelberg: Springer, 2010: 15-26.
- [17] Liao, Y., Du, W., Geurts, P., et al (2013). DMFSGD: A Decentralized Matrix Factorization Algorithm for Network Distance Prediction. *IEEE Transactions on Networking*, 21 (5) 1511-1524.
- [18] Chen, Y., Wang, X., Shi, C, et al. (2011). Phoenix: A Weight-Based Network Coordinate System using Matrix Factorization. *IEEE Transactions on Network and Service Management*, 8 (4) 334-347.
- [19] Cheng, J., Guan, X., Ye, Q., et al. (2014). RNC: A High-Precision Network Coordinate System. *In* : Proceedings of 22<sup>nd</sup> International Symposium of Quality of Service (IWQoS), Hong Kong, China, May 26-27, 228-237.
- [20] Planetlab Dataset. <http://www.eecs.harvard.edu/~syrah/nc/sim/pings.4hr.stamp.gz>.
- [21] King Dataset. <http://pdos.csail.mit.edu/p2psim/kingdata/>.