PAPER

# Fast and Accurate PSD Matrix Estimation by Row Reduction

**Hiroshi KUWAJIMA**[†a)], ***Nonmember***, **Takashi WASHIO**[††b)], ***Member***, ***and*** **Ee-Peng LIM**[†††c)], ***Nonmember***

**SUMMARY** Fast and accurate estimation of missing relations, *e.g.*, similarity, distance and kernel, among objects is now one of the most important techniques required by major data mining tasks, because the missing information of the relations is needed in many applications such as economics, psychology, and social network communities. Though some approaches have been proposed in the last several years, the practical balance between their required computation amount and obtained accuracy are insufficient for some class of the relation estimation. The objective of this paper is to formalize a problem to quickly and efficiently estimate missing relations among objects from the other known relations among the objects and to propose techniques called "*PSD Estimation*" and "*Row Reduction*" for the estimation problem. This technique uses a characteristic of the relations named "*Positive Semi-Definiteness* (PSD)" and a special assumption for known relations in a matrix. The superior performance of our approach in both efficiency and accuracy is demonstrated through an evaluation based on artificial and real-world data sets.

***key words:*** *similarity, Positive Semi-Definite (PSD) matrix, Positive Semi-Definite (PSD) Estimation, row reduction, incomplete Cholesky decomposition*

## 1. Introduction

With the recent growth of network society and ubiquitous sensing environments, a lot of real data sets representing the relations among massive objects are now available, where each relation between the two objects is represented by a value such as similarity, distance and kernel. However, for various reasons, it is not always true that the relation information is fully given in the data sets. On the other hand, the missing information of the relations is needed in many applications. For example, the estimation of the economic trading amounts among some underdeveloped or small countries that have no record in the world trading statistics is important to manage global economy in economics [1], [2]. Estimation of the missing elements in pairwise comparison matrices acquired in psychological experiments is an important technique to complete the data without applying exhaustive questionnaires, which is not feasible due to the burden of human workload [3]. Automated friend introduction in a network community, which requires the estimation of potential friendships, is an essential service in a social network to enhance the communication activity in the community. Accordingly, the development of a scalable technique to quickly and accurately estimate the missing information of the relations among the objects in an objective data set is now an important issue.

Although some general-purpose approaches can be utilized for the estimation as described in the Sect. 3, they are not preferable for large-scale and real-world problems because of their high computational complexity and insufficient accuracy. Thus, we set the objective of this paper to carefully formalize the problem setting of the aforementioned estimation problem and to propose techniques which are tailored to the estimation problem and achieves short computation time and high accuracy at the same time. Based on the observation that most of the relations among the objects are transformed into Positive Semi-Definite (PSD) relations as described later, we propose the following novel techniques to extend one of conventional approaches, namely, "*PSD Approximation*," based on "*Incomplete Cholesky Decomposition*" [12], [13] [†].

(1) "*PSD Estimation*" which provides the estimation of each missing element with its admissible value interval, and
(2) "*Row Reduction*" which is an efficient Pivoting criterion to estimate the missing elements within a given error tolerance under lower computational complexity.

The secondary objective of this paper is to characterize the performance of our proposed techniques in both computational efficiency and estimation accuracy. A program named "*PERCH* (Psd Estimation by row Reduction based on incomplete CHolesky decomposition)" has been developed, and its performance has been compared with a SVM based regression, a PSD Completion approach named *dualcomp*, and a NNLS approach named *APGL* through numerical experiments using artificial and real world data. Moreover, the practicality of our PERCH has been assessed in an application to estimate potential friendships in a social network

Manuscript received June 7, 2011.
Manuscript revised March 26, 2012.

[†]The author is with DENSO CORPORATION, Kariya-shi, 448–8661 Japan.
[††]The author is with the Department of Reasoning for Intelligence, The Institute of Science and Industrial Research, Osaka University, Ibaraki-shi, 567–0047 Japan.
[†††]The author is with the School of Information Systems, Singapore Management University, Singapore.
a) E-mail: hiroshi_kuwajima@denso.co.jp
b) E-mail: washio@ar.sanken.osaka-u.ac.jp
c) E-mail: eplim@smu.edu.sg

DOI: 10.1587/transinf.E95.D.2599

[†]"Incomplete Cholesky Decomposition" became to be used in rather new literatures of machine learning and data mining, and it originally has an alternative name "Truncated Cholesky Factorization" in computational mathematics [11]. We use the terminology "Incomplete Cholesky Decomposition" according to the original paper of PSD Approximation [12], [13].

community.

## 2. Problem Setting

The problem to estimate missing relations among the objects in a set is mathematically formalized as follows. Given a set of objects $OB$ ($|OB| = n$), let $OA^h$ ($\subset OB$) ($h = |OA^h| < n$) be a set where all pairwise relations among the objects in $OA^h$ and all bipartite relations of the objects between $OA^h$ and $OB^{n-h} = OB - OA^h$ are known, and all pairwise relations among the objects in $OB^{n-h}$ are not known. Upon this definition, our problem is to estimate the unknown relations among the objects in $OB^{n-h}$ by the known relations among the objects in $OB$. In the aforementioned examples, an object in $OB$ is a country, a pairwise experiment, or a user in the social network where the relations among the objects in $OB^{n-h}$ are the missing information. An advantageous representation to enable rigorous and flexible manipulation of this problem is to use a relation matrix for the objects in $OB$. Let $A$ be an $n \times n$ matrix where each element represents a pairwise relation between two objects in $OB$. $A$ is represented by the following expression.

$$A = \begin{pmatrix} A^h & B^{n-h^T} \\ B^{n-h} & X^{n-h} \end{pmatrix}. \quad (1)$$

where $A^h$ is a $h \times h$ principal submatrix of $A$, representing the known relations of the objects in $OA^h$, and $(n-h) \times h$ submatrix $B^{n-h}$ represents the known bipartite relations of the objects between $OA^h$ and $OB^{n-h}$. Moreover, $X^{n-h}$ is a $(n-h) \times (n-h)$ principal submatrix of A, representing the missing relations among objects in $OB^{n-h}$. Our problem is now to estimate $X^{n-h}$ from the rest of $A$. To the best of our knowledge so far, no solution has been proposed to take the practical balance between their required computation amount and obtained accuracy for this problem while guaranteeing the absolute error less than a tolerance. Hence, this study proposes a novel technique to achieve a practical balance between the two.

To address this problem, we focus on mathematical characteristics of the relation matrix $A$. Due to the symmetric property of the majority of relations, most of the relation matrices $A$ are square and symmetric. Moreover, in many cases, $A$ is Positive Semi-Definite (PSD) or can be converted into a PSD matrix by a transformation to another matrix such as a signless Laplacian. In this paper, we explore a principle and an algorithm to estimate the missing part of the relation matrix from its known part by exploiting the PSD property in order to advance the accuracy of the estimation and the scalability in terms of computational complexity.

## 3. Related Work

Although there are not special techniques for the problem setting as described in the Sect. 2, in this section, we introduce how conventional approaches designed for a general purpose can solve the PSD Estimation problem.

A conventional way to address this problem is to develop a classification or regression model by assuming that the relation between two objects $p$ and $q$, *i.e.*, an element $a_{p,q}$ in $A$, can be estimated from the known relations of $p$ with the other objects and those of $q$. In the model, the objective variable is $a_{p,q}$, and its explanation vector is one of ordered concatenations, $[\boldsymbol{a}_p^{h^T} \boldsymbol{a}_q^{h^T}]$ and $[\boldsymbol{a}_q^{h^T} \boldsymbol{a}_p^{h^T}]$ where $\boldsymbol{a}_p^{h^T}$ and $\boldsymbol{a}_q^{h^T}$ are the $p$-th row and the $q$-th row of the first $h$ columns of $A$ respectively which represent the known relations of the two objects with the other objects. The model is trained by the known part of the matrix $A$, *i.e.*, a data set consisting of the $2hn - h^2$ relations $([\boldsymbol{a}_p^{h^T} \boldsymbol{a}_q^{h^T}], a_{p,q})$ ($p = 1, \ldots, n$, $q = 1, \ldots, h$, and $p \geq q$) and $([\boldsymbol{a}_q^{h^T} \boldsymbol{a}_p^{h^T}], a_{p,q})$ ($q = 1, \ldots, n$, $p = 1, \ldots, h$, and $q > p$). Every unknown element $a_{p,q}$ in $X^{n-h}$ is estimated by this trained model. Since this framework does not use the PSD property of the relations, the accuracy of the estimation is supposed to be limited. This is shown in our experimental demonstration later. In addition, its computational complexity is quite high. For example, the computational complexity of regression and its estimation is $O(M^3N + MN')$ where $M$, $N$ and $N'$ are respectively the number of explanation elements, the number of training examples and the number of objective elements to be estimated. Under $N = 2hn - h^2 = O(hn)(n > h)$, $M = 2h = O(h)$, $N' = (n-h) \times (n-h) = O(n^2)$ and the assumption that the number of the known relations is almost proportional to the total number of relations, *i.e.*, $h^2 \propto n^2$, the computational complexity of the conventional framework is eventually $O(n^5)$ which is not very tractable in large-scale real world applications. Accordingly, more accurate and efficient approaches than this conventional framework should be explored by taking into account the mathematical characteristic of the problem such as Positive Semi-Definiteness (PSD).

"*Positive Semi-Definite (PSD) Completion*" is known to be a subclass of Semidefinite programming, and derives a matrix $\tilde{A}$ in which missing elements of an original PSD matrix $A$ are completed by maximizing the determinant $\det(\tilde{A})$ under PSD constraints [5], [6]. Its program named *dualcomp* is opened to the public [7]. PSD Completion is applicable to any PSD matrix representing a chordal graph where the matrix is interpreted to be a symmetric adjacency matrix having none zeros at known elements and zeros at missing elements. This has been applied to the completion of missing elements in a kernel matrix under a recent study [8]. However because they maximize the determinant and do not adapt to an original PSD matrix $A$ in the form of Eq. (1), *i.e.*, the missing elements make up a principal submatrix of $A$, the computational complexity of these approaches is $O(n^6)$ [6], and the number of known elements required in the matrix is almost $O(n^2)$ since they can complete the elements with sufficient accuracy only when most of the elements are known in advance. Thus, PSD Completion is not very suitable to the case under limited known relations. Another relevant approach is to use a parametric function to

complete the elements in a PSD kernel matrix [9]. It completes missing elements in an auxiliary kernel matrix by fitting a parametric function based on a complete primary kernel matrix. However, this is not applicable when the appropriate primary kernel matrix well followed by the missing elements is not available. A more recent approach applies a Gaussian parametric function to complete missing features of objects to construct a kernel matrix [10]. However, this is also limited to the case that both the distribution of the missing value and the kernel matrix are Gaussian. In addition to high computational complexities described above, PSD Completion approaches cannot guarantee the absolute error less than a tolerance because these techniques intended to recover an original matrix as far as possible using available information.

Another recent approach named "*Nuclear norm regularized linear least squares problem (NNLS)*" is a convex relaxation of "*affine rank minimization problem*," which finds the matrix of minimum rank subject to linear equality constraints [25]. NNLS derives a matrix $Y$ in which missing elements of an original PSD matrix are completed by minimizing the sum of the square error for known elements in the estimated matrix and the nuclear norm, *i.e.*, $|\mathcal{A}(Y) - \boldsymbol{b}|^2/2 + \mu\|Y\|_*$ where $\mathcal{A}(Y) = \boldsymbol{b}$ represents the linear equality constraints for known elements, $\|Y\|_*$ is the nuclear norm of $Y$ and $\mu > 0$ is a given regularization parameter. An algorithm named *APGL* was developed to solve NNLS in [25] and its program is opened to the public [26]. Unlike PSD Completion, NNLS under PSD constraints is applicable to any PSD matrices, *i.e.*, NNLS under PSD constraints has no requirement for positions of known elements and missing elements in an original PSD matrix. The iteration complexity is $O(1/\sqrt{\epsilon})$ with an $\epsilon$-optimal solution and the computational complexity for each iteration including eigenvalue decomposition is $O(n^3)$. Assuming $\epsilon$ is a constant independent from $n$, the total computational complexity of APGL is considered as $O(n^3)$. APGL can solve our matrix estimation problem as a special case of NNLS under PSD constraints which we compare to PERCH later. Although it has low computational complexity, APGL cannot guarantee the maximum absolute error less than a tolerance.

"*PSD Approximation*" based on "*Incomplete Cholesky Decomposition*" is another relevant approach [12], [13]. This technique has been developed to produce a good approximation of a known PSD matrix by using an appropriately chosen subset of rows and columns of the matrix. Given a PSD matrix $A$ where all of its elements are known, the $k$ rows and their symmetric $k$ columns of $A$ are sampled according to some criteria to obtain a good approximation $\tilde{A}$ of $A$ within the small number $k$. This sampling of the rows and the columns in $A$ is called "*Pivoting*," and the objects corresponding to these rows and columns are called "*Pivots*." PSD Approximation samples the Pivots under a criterion to ensure the small value of $\mathrm{tr}(A - \tilde{A})$. This Pivoting criterion enables efficient factorization of the relation matrix $A$ into a lower $n \times k$ trapezoid matrix $L$ and its transposed $L^T$ with $O(k^2 n)$ computational complexity, and derives the approximation $\tilde{A} = LL^T$ with $O(kn^2)$. Another similar work introduced the other two Pivoting criteria [14]. The first criterion is to sample the Pivots which ensure the small Frobenius norm of the residual matrix of Incomplete Cholesky Decomposition, but requires $O(kn^3)$ computational complexity. The second is a relaxation of this criterion to ensure the small upper bound of the norm, and requires $O(kn^2)$. An underlying assumption of these approaches is that "*the majority of relation matrices have a rapidly decaying spectrum, i.e., a low rank*" [15]. This assumption ensures their good approximation under a small $k$.

However, the application of the present PSD Approximation to the estimation problem of the missing elements in $A$ has three difficulties. First, all of the aforementioned criteria of the Pivoting need to know the actual values of the missing elements of $A$ in advance. This is essentially impractical for the aforementioned problems. Second, the computational complexity of the matrix factorization under these Pivoting criteria is $O(k^2 n) \sim O(kn^3)$. An efficient Pivoting is desired for the fast and accurate estimation of the missing elements. The third is that they do not provide the admissible error bound of each element estimation, and do not guarantee the absolute error less than a tolerance. This guarantee is practically required in many applications.

## 4. Background Principles

In this section, a past study named "*PSD Approximation*" which provides technical bases to our proposal is explained [12], [13]. First, we describe the normalization of the relation matrix to simplify the computation scheme of PSD Approximation without losing its generality. Given a set of objects $OB$ ($|OB| = n$), let $A$ be an $n \times n$ PSD relation matrix, denoted by $A \succeq 0$, representing the relations among all of the objects in $OB$. By definition, $A$ is a symmetric matrix where all of its diagonal elements are nonzero and known by default. Thus, through $W = diag(A)$, any $A \succeq 0$ has the following invertible transformation into the corresponding $A^* \succeq 0$ where all of its diagonal elements are normalized to unity.

$$A^* = {W^{-1/2}}^T A W^{-1/2} = W^{-1/2} A W^{-1/2}.$$

Accordingly, our discussion in the rest of this paper focuses on $A^*$ without loss of generality, and we use the symbol $A$ instead of $A^*$ to represent a normalized PSD relation matrix.

PSD Approximation efficiently approximates the values of some elements in a PSD relation matrix $A$ by using a limited number of the other element values [12], [13]. In contrast to PSD Completion, it does not maximize $\det(A)$, but approximates $A$ to minimize a trace associated with $A$, *e.g.*, $\mathrm{tr}(A - \tilde{A})$, while maintaining $\det(A) = 0$ by its progressive and low rank factorization based on Incomplete Incomplete Cholesky Decomposition. Hence, PSD Approximation is not a subclass of PSD Completion.

Assuming an element $a_{p,q}$ in $A$ is the relation between

two objects $p$ and $q$ in $OB$, let $A^k$ be a $k \times k$ principal submatrix of $A$ ($1 \leq k \leq n$) representing relations among $k$ objects in a subset $OA^k \subseteq OB$ ($k = |OA^k|$). Moreover, let $B^{n-k}$ be a $(n-k) \times k$ submatrix of $A$. $B^{n-k}$ consists of the relations between every object in $OA^k$ and every object in $OB^{n-k} = OB - OA^k$. The objects in $OA^k$ are called "*Pivots*." Upon these definitions, we consider the following $A(k)$ where the elements which are not included in $A^k$ and $B^{n-k}$ in $A(k)$, *i.e.*, $Z^{n-k}$, are to be approximated.

$$A(k) = \begin{pmatrix} A^k & B^{n-k\,T} \\ B^{n-k} & Z^{n-k} \end{pmatrix}. \tag{2}$$

PSD Approximation approximates all elements in a $(n-k)\times(n-k)$ matrix $Z^{n-k}$ by appropriately selecting $A^k$ and $B^{n-k}$ from $A$ to make the trace $\mathrm{tr}(A - A(k))$ within a tolerance.

This approximation is performed by applying the $k$ step Incomplete Cholesky Decomposition to $A(k)$ as follows[†].

$$A(k) = L(k)L(k)^T \tag{3}$$

where

$$L(k) = \begin{pmatrix} L^k \\ L^{n-k} \end{pmatrix} = \left(\begin{array}{ccc} \ell_{1,1} & & 0 \\ \vdots & \ddots & \\ \ell_{k,1} & \cdots & \ell_{k,k} \\ \hline \ell_{k+1,1} & \cdots & \ell_{k+1,k} \\ \vdots & & \vdots \\ \ell_{n,1} & \cdots & \ell_{n,k} \end{array}\right), \tag{4}$$

$$\ell_{i,j} = \frac{a_{i,j} - \sum_{h=1}^{j-1} \ell_{i,h}\ell_{j,h}}{\ell_{j,j}}, \quad (j = 1, 2, \cdots, min(i-1, k)) \tag{5}$$

$$\ell_{i,i} = \pm\sqrt{a_{i,i} - \sum_{h=1}^{i-1} \ell_{i,h}^2} = \pm\sqrt{1 - \sum_{h=1}^{i-1} \ell_{i,h}^2}. \tag{6}$$

and $a_{i,j}$ is an $i,j$-element of $A(k)$. In addition, the following "*residual*" of each row for $k+1 \leq p \leq n$ in $L(k)$ is defined for the later use.

$$e_p^{(k)} \equiv 1 - \sum_{h=1}^{k} \ell_{p,h}^2 \tag{7}$$

The trace $\mathrm{tr}(A - A(k))$ under this approximation of $A(k)$ is known to be $\sum_{p=k+1}^{n} e_p^{(k)}$ which is also equal to the sum of the eigenvalues of $A - A(k)$.

We construct $L(k)$ in bottom up manner starting from $k = 1$. Toward the less $\mathrm{tr}(A - A(k))$, an object $p$ is selected from $OB^{n-k}$ and moved to $OA^k$ in the next $k+1$-th step. This selection is important, and $p_{max} \in OB^{n-k}$ yielding the maximum residual $e_{p_{max}}^{(k)} = \max_{p \in OB^{n-k}} e_p^{(k)}$ is selected. Because $e_p^{(k)}$ is considered to be the component which is independent of $L^k$ in the $p$-th row of $L(k)$, the move of $p_{max}$ from $OB^{n-k}$ to $OA^k$ is expected to effectively reduce the residuals of the other rows in $L^{n-k}$ in the next step. This selection which is called "*Pivoting*" yields the renewed $OA^{k+1}$ and $OB^{n-k-1}$. Subsequently, $L(k)$ is extended to an $n \times (k+1)$ matrix by applying the $k+1$-th step of Incomplete Cholesky Decomposition as follows.

$$L(k+1) = \begin{pmatrix} L^{k+1} \\ L^{n-k-1} \end{pmatrix} = \left(\begin{pmatrix} L^k & \mathbf{0} \\ \boldsymbol{\ell}_{p_{max}}^{kT} & \sqrt{e_{p_{max}}^{(k)}} \end{pmatrix}\right), \\ L_{\bar{p}_{max}}^{n-k-1}$$

where $\boldsymbol{\ell}_{p_{max}}^{kT}$ is the row of $p_{max}$ moved from $L^{n-k}$, and $L_{\bar{p}_{max}}^{n-k-1}$ is a $(n-k-1) \times (k+1)$ submatrix in which each row consists of the row of an object $q$ in $L^{n-k}$ and $\ell_{q,k+1}$ for all objects $q \in OB^{n-k-1}$ except $p_{max}$. $e_{p_{max}}^{(k)}$ is provided by Eq. (7). This is essentially equal to $\ell_{k+1,k+1}$ to be derived by Eq. (6) when the row of $p_{max}$ is attached to the $k+1$-th row. Though either $\pm\sqrt{e_{p_{max}}^{(k)}}$ can be selected for this place, we assign $\sqrt{e_{p_{max}}^{(k)}}$ only since Incomplete Cholesky decomposition $A(k+1) = L(k+1)L(k+1)^T$ is not affected by this selection due to its quadratic symmetry. $L_{\bar{p}_{max}}^{n-k-1}$ is obtained by computing the last element $\ell_{q,k+1}$ through Eq. (5). Note that all data needed to compute $\ell_{q,k+1}$ for all $q \in OB^{n-k-1}$ is the elements $a_{q,p_{max}}$ in $A$ corresponding to the relations between the moved object $p_{max} \in OA^{k+1}$ with all objects $q \in OB^{n-k-1}$, and thus $L(k+1)$ is progressively computed only from $A^{k+1}$ and $B^{n-k-1}$ by Eq. (5) and (7). After this one step progress to derive $A(k+1) = L(k+1)L(k+1)^T$, we obtain less $\mathrm{tr}(A - A(k+1))$. Accordingly, a more accurate approximation of $Z^{n-k-1}$ in $A$ is derived from the additional relations between the moved object $p_{max}$ with the others in $OB^{n-k-1}$ through this Pivoting operation. Under the low rank assumption, this progressive approximation will stop within a small number of steps $k \ll n$ under a tolerance on the trace.

The adjacency matrix corresponding to $A^k$ and $B^{n-k}$ in Eq. (2) represents a complete split graph which is a subclass of the chordal graph. Accordingly, the arrangement of the matrix elements used in PSD Approximation is required to be more restricted than PSD Completion. However, the number of the elements used in PSD Approximation is $O(kn)$ since the known part of the matrix is limited to $A^k$ and $B^{n-k}$. Moreover, the computational complexity to compose $A^k$ and $B^{n-k}$ is $O(kn)$, that of the Incomplete Cholesky Decomposition using the aforementioned Pivoting is $O(k^2n)$, and the computation of $A(k)$ by Eq. (3) is $O(kn^2)$. These are far lower than those of PSD Completion. In this regard, PSD Approximation is more practical. Its major drawback is that any method to ensure the estimation error bound of the individual element within a certain limit has not been established. In addition, it can not be directly applied to the data where some elements are missing, since every new Pivot must be selected from the set of all objects which are not Pivots in $OB$.

[†]The ordinary expression of Incomplete Cholesky Decomposition is $A(k) = L(k)L(k)^T$ or $A(k) = L(k)D(k)L(k)^T$. We use $A(k) = L(k)L(k)^T$ for the ease of our explanation.

## 5. Proposed Techniques

To overcome the drawbacks of PSD Approximation, the objective (1) mentioned at the last part of the first section is achieved by our proposing novel principle named "*PSD Estimation*" under the aforementioned assumption that "*the majority of relation matrices have a rapidly decaying spectrum, i.e., a low rank*." Moreover, to reduce the error bound within a tolerance while optimizing the computation scheme for less computational complexity, we achieve the aforementioned objective (2) by introducing a novel Pivoting principle and algorithm named "*Row Reduction*."

### 5.1 PSD Estimation

The principle of our PSD Estimation comes from the following basic theorem [16].

**Theorem 1** (Sylvester's criterion): A square matrix $A$ is PSD, *i.e.*, $A \succeq 0$, iff the determinants of all principal submatrices of $A$ are more than or equal to 0. A square matrix $A$ is PD, *i.e.*, $A \succ 0$, iff the determinants of all leading principal submatrices of $A$ are more than 0. □

Based on this theorem, our PSD Estimation requires the PSD admissibility condition, *i.e.*, the nonnegative determinant of the principal submatrices of $A$, in the estimation of every element of $A$. For preliminary analysis, let's consider the following $(k+1)\times(k+1)$ principal submatrix $A_p^{k+1}$ of $A(k)$ in Eq. (2) which contains the $k\times k$ principal submatrix $A^k$ representing the relations among Pivots in $OA$ and the $k$ dimensional vector $\boldsymbol{a}_p^k$ in $B^{n-k}$ representing the relations between every Pivots and the objects $p \in OB^{n-k}$.

$$A_p^{k+1} = \begin{pmatrix} A^k & \boldsymbol{a}_p^k \\ \boldsymbol{a}_p^{k\,T} & 1 \end{pmatrix}$$

By Theorem 1 and $A \succeq 0$, $A_p^{k+1} \succeq 0$ holds, and thus it can be decomposed by ordinary Cholesky Decomposition. The submatrix $A^k$ is decomposed by Eq. (5), and the last row $\boldsymbol{a}_p^{k\,T}$ and $a_{k+1,k+1} \equiv a_{p,p} = 1$ are processed as follows by Eq. (5) and (7).

$$\ell_{p,i} = \frac{a_{p,i} - \sum_{h=1}^{i-1} \ell_{i,h}\ell_{p,h}}{\ell_{i,i}}, (i \neq p)$$

$$e_p^{(k)} \equiv \ell_{k+1,k+1}{}^2 \equiv \ell_{p,p}{}^2 = 1 - \sum_{h=1}^{k} \ell_{p,h}^2.$$

This results the following decomposition.

$$A_p^{k+1} = L_p^{k+1} L_p^{k+1\,T},$$

where

$$L_p^{k+1} = \begin{pmatrix} L^k & 0 \\ \boldsymbol{\ell}_p^{k\,T} & \sqrt{e_p^{(k)}} \end{pmatrix},$$

$$\boldsymbol{\ell}_p^k = (\ell_{p,1} \;\; \ell_{p,2} \;\; \ldots \;\; \ell_{p,k})^T.$$

$\boldsymbol{\ell}_p^k$ plays an important role in our PSD Estimation as shown shortly. Here, we assume a stronger constraint $A^k \succ 0$ and hence $\det(A^k) > 0$ which is achieved without loss of generality as shown later. Then, we obtain the following lemma on the residual $e_p^{(k)}$.

**Lemma 1:** Given $A^k \succ 0$. If $A_p^{k+1} \succeq 0$, then $0 \leq e_p^{(k)} \leq 1$. If $e_p^{(k)} > 0$, then $A_p^{k+1} \succ 0$. □

**Proof 1:** From Eq. (7), $e_p^{(k)} \leq 1$ is trivial. Based on the property of determinants for triangular matrices and $A_p^{k+1} \succeq 0$,

$$(*) \qquad \det(A_p^{k+1}) = \det(L_p^{k+1}) \det(L_p^{k+1\,T}) = e_p^{(k)} \prod_{i=1}^{k} \ell_{i,i}^2 \geq 0.$$

Because of $A^k \succ 0$,

$$\det(A^k) = \det(L^k) \det(L^{k\,T}) = \prod_{i=1}^{k} \ell_{i,i}^2 > 0.$$

According to these inequalities, $e_p^{(k)} \geq 0$. Moreover, by Theorem 1, the determinant of any leading principal submatrix of $A^k$ is positive, *i.e.*,

$$\det(A^h) = \prod_{i=1}^{h} \ell_{i,i}^2 > 0 \quad (h = 1, \ldots, k).$$

In addition, from Eq. (*), if $e_p^{(k)} > 0$, then $\det(A_p^{k+1}) > 0$. Thus, the determinants of all leading principal submatrices of $A_p^{k+1}$ are positive, and this proves $A_p^{k+1} \succ 0$. ■

Now, we derive our main result to estimate a missing element of $X^{n-h}$ in Eq. (1) by considering the following $(k+2)\times(k+2)$ principal submatrix $A_{p,q}^{k+2}$ of $A(k)$ in Eq. (2) where $k \leq h$ and the condition that $X^{n-h}$ is a principal submatrix of $Z^{n-k}$ are assumed.

$$A_{p,q}^{k+2} = \begin{pmatrix} A^k & \boldsymbol{a}_p^k & \boldsymbol{a}_q^k \\ \boldsymbol{a}_p^{k\,T} & 1 & x_{p,q} \\ \boldsymbol{a}_q^{k\,T} & x_{q,p} & 1 \end{pmatrix}, \tag{8}$$

where $x_{p,q}(= x_{q,p})$ in $X^{n-h}$ is a missing element representing the relation between $p$ and $q$. Similarly to $A_p^{k+1}, A_{p,q}^{k+2} \succeq 0$ holds by Theorem 1 and $A \succeq 0$. Hence, $A_{p,q}^{k+2}$ can be factorized by Incomplete Cholesky Decomposition up to the $k$-th step as follows.

$$A_{p,q}^{k+2} = L_{p,q}^{\prime k+2} L_{p,q}^{\prime k+2\,T}, \tag{9}$$

where

$$L_{p,q}^{\prime k+2} = \begin{pmatrix} L^k & \mathbf{0} & \mathbf{0} \\ \boldsymbol{\ell}_p^{k\,T} & & \\ \boldsymbol{\ell}_q^{k\,T} & \mathrm{R}_{p,q} & \end{pmatrix},$$

and $\mathrm{R}_{p,q}$ is a $2\times 2$ submatrix. From Eq. (9), $\mathrm{R}_{p,q}$ must satisfy

the following relation.

$$\mathrm{R}_{p,q}\mathrm{R}_{p,q}{}^{T} = \begin{pmatrix} 1 & x_{p,q} \\ x_{p,q} & 1 \end{pmatrix} - \begin{pmatrix} \boldsymbol{\ell}_p^{k\,T}\boldsymbol{\ell}_p^k & \boldsymbol{\ell}_p^{k\,T}\boldsymbol{\ell}_q^k \\ \boldsymbol{\ell}_q^{k\,T}\boldsymbol{\ell}_p^k & \boldsymbol{\ell}_q^{k\,T}\boldsymbol{\ell}_q^k \end{pmatrix} = \begin{pmatrix} 1-\boldsymbol{\ell}_p^{k\,T}\boldsymbol{\ell}_p^k & x_{p,q}-\boldsymbol{\ell}_p^{k\,T}\boldsymbol{\ell}_q^k \\ x_{p,q}-\boldsymbol{\ell}_q^{k\,T}\boldsymbol{\ell}_p^k & 1-\boldsymbol{\ell}_q^{k\,T}\boldsymbol{\ell}_q^k \end{pmatrix}. \quad (10)$$

Then, the solution of $x_{p,q}$ is given by the following theorem.

**Theorem 2:** The admissible value interval of a missing element $x_{p,q}$ in $A_{p,q}^{k+2}$ is as follows.

$$\hat{x}_{p,q}^{(k)} - \Delta x_{p,q}^{(k)} \le x_{p,q} \le \hat{x}_{p,q}^{(k)} + \Delta x_{p,q}^{(k)},$$

where

$$\hat{x}_{p,q}^{(k)} = \boldsymbol{\ell}_p^{k\,T}\boldsymbol{\ell}_q^k,$$
$$\Delta x_{p,q}^{(k)} = \sqrt{\left(1-\boldsymbol{\ell}_p^{k\,T}\boldsymbol{\ell}_p^k\right)}\sqrt{\left(1-\boldsymbol{\ell}_q^{k\,T}\boldsymbol{\ell}_q^k\right)}.$$

□

**Proof 2:** We consider the determinant of $A_{p,q}^{k+2}$. Due to the triangularity of $L_{p,q}'^{k+2}$ in Eq. (9), $\det(L_{p,q}'^{k+2}) = \det(L^k)\det(\mathrm{R}_{p,q})$ holds. Based on this fact and Eq. (9),

$$\begin{aligned}\det(A_{p,q}^{k+2}) &= \det(L_{p,q}'^{k+2})\det(L_{p,q}'^{k+2\,T}) \\ &= \det(L^k)\det(\mathrm{R}_{p,q})\det(L^{k\,T})\det(\mathrm{R}_{p,q}^T) \\ &= \det(L^kL^{k\,T})\det(\mathrm{R}_{p,q}\mathrm{R}_{p,q}^T).\end{aligned}$$

From $A_{p,q}^{k+2} \ge 0$, $A^k = L^kL^{k\,T}$ and the assumption $A^k \succ 0$, $\det(A_{p,q}^{k+2}) \ge 0$ and $\det(A^k) = \det(L^kL^{k\,T}) > 0$. Thus, the last part in the r.h.s. of the former equation, *i.e.*, $\det(R_{p,q}R_{p,q}{}^{T})$, must be nonnegative. According to this result and Eq. (10), we obtain the following quadratic inequality.

$$\begin{aligned}&\det(R_{p,q}R_{p,q}{}^{T}) \\ &\quad= \left(1-\boldsymbol{\ell}_p^{k\,T}\boldsymbol{\ell}_p^k\right)\left(1-\boldsymbol{\ell}_q^{k\,T}\boldsymbol{\ell}_q^k\right) - \left(x_{p,q}-\boldsymbol{\ell}_p^{k\,T}\boldsymbol{\ell}_q^k\right)^2 \ge 0,\end{aligned}$$

and thus,

$$\begin{aligned}&x_{p,q}^2 - 2\boldsymbol{\ell}_p^{k\,T}\boldsymbol{\ell}_q^k x_{p,q} + \left(\boldsymbol{\ell}_p^{k\,T}\boldsymbol{\ell}_q^k\right)^2 \\ &\quad - \boldsymbol{\ell}_p^{k\,T}\boldsymbol{\ell}_p^k\boldsymbol{\ell}_q^{k\,T}\boldsymbol{\ell}_q^k + \boldsymbol{\ell}_p^{k\,T}\boldsymbol{\ell}_p^k + \boldsymbol{\ell}_q^{k\,T}\boldsymbol{\ell}_q^k - 1 \le 0.\end{aligned}$$

This inequality derives the following interval of $x_{p,q}$.

$$\begin{aligned}\hat{x}_{p,q}^{(k)} &= \boldsymbol{\ell}_p^{k\,T}\boldsymbol{\ell}_q^k, \\ \Delta x_{p,q}^{(k)} &= \sqrt{\left(\boldsymbol{\ell}_p^{k\,T}\boldsymbol{\ell}_q^k\right)^2 - \left(\left(\boldsymbol{\ell}_p^{k\,T}\boldsymbol{\ell}_q^k\right)^2 - \boldsymbol{\ell}_p^{k\,T}\boldsymbol{\ell}_p^k\boldsymbol{\ell}_q^{k\,T}\boldsymbol{\ell}_q^k + \boldsymbol{\ell}_p^{k\,T}\boldsymbol{\ell}_p^k + \boldsymbol{\ell}_q^{k\,T}\boldsymbol{\ell}_q^k - 1\right)} \\ &= \sqrt{\left(1-\boldsymbol{\ell}_p^{k\,T}\boldsymbol{\ell}_p^k\right)}\sqrt{\left(1-\boldsymbol{\ell}_q^{k\,T}\boldsymbol{\ell}_q^k\right)}.\end{aligned}$$

■

Then, the solution of $x_{p,q}$ is given by the following corollary.

**Corollary 1:** The admissible value interval of the missing element $x_{p,q}$ in $A_{p,q}^{k+2}$ is as follows.

$$\hat{x}_{p,q}^{(k)} - \Delta x_{p,q}^{(k)} \le x_{p,q} \le \hat{x}_{p,q}^{(k)} + \Delta x_{p,q}^{(k)}$$

where $\hat{x}_{p,q}^{(k)} = \boldsymbol{\ell}_p^{k\,T}\boldsymbol{\ell}_q^k$, $\Delta x_{p,q}^{(k)} = \sqrt{e_p^{(k)}}\sqrt{e_q^{(k)}}$. □

**Proof 3:** $\hat{x}_{p,q}^{(k)} = \boldsymbol{\ell}_p^{k\,T}\boldsymbol{\ell}_q^k$ is trivial from Theorem 2. Similarly, from Theorem 2 and $e_p^{(k)}$, $e_q^{(k)}$ given by Eq. (7), $\Delta x_{p,q}^{(k)} = \sqrt{e_p^{(k)}}\sqrt{e_q^{(k)}}$. ■

These theorem and corollary ensure that the estimation of the relation and its admissible error bound are always factorized into an individual object $p$, *i.e.*, $\boldsymbol{\ell}_p^k$ and $\sqrt{e_p^{(k)}}$ respectively as far as $k \le h$, *i.e.*, $X^{n-h}$ is a principal submatrix of $Z^{n-k}$, is maintained. Once $\boldsymbol{\ell}_p^k$ and $\sqrt{e_p^{(k)}}$ for all $p \in OB^{n-k}$ are computed from $B^{n-k}$, we can easily estimate the relation $\hat{x}_{p,q}^{(k)}$ and its error bound $\Delta x_{p,q}^{(k)}$ between any $p$ and $q$ in $OB$. Because Theorem 2 and Corollary 1 is a necessary condition to make a local matrix $A_{p,q}^{k+2} \succeq 0$, $A \succ 0$ and the maximum $\det(A)$ are not ensured. Thus, our PSD Estimation is different from PSD Completion.

The factorized forms of the solution and its admissible error bound are derived from $\boldsymbol{\ell}_p^k$ of each object $p$. As $\boldsymbol{\ell}_p^k$ for every $p \in OB^{n-k}$ is efficiently deduced by Incomplete Cholesky Decomposition of $A(k)$ represented by Eq. (3)-(7), PSD Estimation is performed under the same computational scheme with PSD Approximation. On the progression of Incomplete Cholesky Decomposition under the condition of $k \le h$, the object $p_{max} \in OB^{n-k}$ providing the maximum residual $e_{p_{max}}^{(k)} = \max_{p \in OB^{n-k}} e_p^{(k)}$ is selected for the next Pivot similarly to Pivoting of PSD Approximation. If $e_{p_{max}}^{(k)} = 0$, $e_p^{(k)} = 0$ for all $p \in OB^{n-k}$ since $e_p^{(k)} \ge 0$ from Lemma 1. Then, Corollary 1 implies that the similarities of all missing elements have been estimated without any error, and thus any further estimation is not needed. This fact ensures $e_{p_{max}}^{(k)} > 0$ whenever any further estimation is needed, and hence $A_p^{k+1} \succ 0$ and $\det(A_p^{k+1}) > 0$ hold by Lemma 1. Its principal submatrix $A^k$ also shares the same property, *i.e.*, $A^k \succ 0$ and $\det(A^k) > 0$. This supports the assumption $A^k \succ 0$ of Lemma 1.

### 5.2 Row Reduction

Given a residual threshold value $r_{th}$ ($0 \le r_{th} \le 1$) specified by users, if the residual $e_p^{(k)}$ of an object $p \in OB^{n-k}$ has the following relation with $r_{th}$ and the aforementioned maximum residual $e_{p_{max}}^{(k)}$ at the $k$-th step under the condition $k \le h$,

$$\sqrt{e_p^{(k)}}\sqrt{e_{p_{max}}^{(k)}} \le r_{th}, \quad (11)$$

the error boundary, $\Delta x_{p,q}^{(k)}$, of the relation between the object $p$ and any other $q \in OB^{n-k}$ is always less than or equal to $r_{th}$ by Corollary 1. The object $p$ satisfying this condition is considered to have sufficiently accurate estimations with any other objects.

At the end of the $k$-th step of Incomplete Cholesky Decomposition, $\boldsymbol{\ell}_p^{k\,T}$ and $e_p^{(k)}$ of all objects $p \in OB^{n-k}$ are computed by Eq. (5) and Eq. (7). Then, the following completed set $OC^k$ of the objects is derived.

$$OC^k = \{p | \forall p \in OB^{n-k}, \text{Eq. (11) is satisfied.}\}.$$

These objects are removed from $OB^{n-k}$ as follows without moving to $OA^k$.

$$OB^{n-k} \leftarrow OB^{n-k} - OC^k.$$

By this operation, the set of the rows of $L(k)$ in Eq. (3) and Eq. (4) corresponding to $OC^k$ is removed before the $k+1$-th step of Incomplete Cholesky Decomposition is conducted, and the size of $L(k)$ reduces. This stepwise reduction of the rows speeds up Incomplete Cholesky Decomposition as $k$ increases, and guarantees that every residuals of objects in $OB^{n-k}$ are greater than 0 before $k+1$-th step of Incomplete Cholesky Decomposision. We call this computation scheme "*Row Reduction*". When $OB^{n-k} = \phi$ or $k = h$ is achieved, Incomplete Cholesky Decomposition is terminated. Accordingly, every object $p \in OB$ except the pivot is stored in some $OC^k$ with its factorized estimation and error bound at the end of this algorithm.

### 5.3 Complexity Analysis

The efficiency of Row Reduction strongly affects the computational complexity of the entire PSD Estimation, since they reduce the number of rows to be manipulated in every process of the estimation. If any reduction is not applied, the number of the rows remains as $O(n)$ at every step of Incomplete Cholesky Decomposition. Since the complexity of Incomplete Cholesky Decomposition for each element at $i$-th step is $O(i-1)$ $(i = 1, \ldots, k)$ due to Eq. (5) and Eq. (6) containing some summations over $i-1$ terms at maximum, the total complexity of Incomplete Cholesky Decomposition up to the $k$-th step is $O(n \times \sum_{i=1}^{k}(i-1)) = O(k^2 n)$ in this worst case. If the reduction significantly eliminates the rows within a small number of the steps, the number of the rows at the initial step is $O(n)$, and that can become $O(k-i+1)$ at $i$-th step $(i = 2, \ldots, k)$ at the smallest. Because the complexity of Incomplete Cholesky Decomposition for each element at $i$-th step is $O(i-1)$ as mentioned above, the total complexity of Incomplete Cholesky Decomposition is $O(n \times 0 + \sum_{i=2}^{k}\{(k-i+1) \times (i-1)\}) = O(k^3)$ in total under the best condition. In summary, the computational complexity of Incomplete Cholesky Decomposition applying Row Reduction is $O(k^3) \sim O(k^2 n)$ in total when the scheme is terminated at the $k$-th step. Though the worst case is comparable to the conventional PSD Approximation, the ordinary case in which the rows are efficiently reduced shows less computational complexity.

Besides, the derivation of elements for all pairs of objects in $OB^{n-k}$ is $O(k \times (n-k)^2) = O(kn^2)$ under the consideration that the computation cost of each element is proportional to the size of ${\boldsymbol{\ell}_p^k}^T$ due to Corollary 1. As the matrix $A$ does not involve any independent objects more than its rank, all residuals become 0 if the rank of $A$ is less or equal to $h$, and $k$ achieves the rank. Accordingly, the efficiency of PSD Estimation strongly depends on the rank of the objective relation matrix.

The memory complexity of PSD Estimation with Row Reduction is $O(k^2) \sim O(kn)$, since $L(k)$ is the unique array used in the estimation. When all estimated elements are recorded on memory, obviously $O(n^2)$ space is required.

## 6. Performance Evaluation

A program named "*PERCH* (Psd Estimation by row Reduction based on incomplete CHolesky decomposition)" has been developed based on the proposed principles. Its input parameter is $r_{th}$ for Row Reduction. To assess the practicality of PERCH, we evaluate PERCH by using correlation matrices among $m$ dimensional vectors representing the objects in $OB$, since correlation matrices are very popular among various PSD matrices. The fundamental parameters of the input correlation matrices for both artificial data and real world data are $n$ the size of $OB$, $m$ the intrinsic dimension and $h$ the size of $OA^h$. Artificial data have some extra parameters according to the contents embedded in the data. To further assess the practicality of our proposed PSD Estimation for matrices representing more complex relations, we additionally conducted the performance evaluation for some real world data of correlation matrices and graph diffusion kernel matrices. The performance indices for comparison are the required computation time and the standard error of $|x_{p,q}^{(k)} - \hat{x}_{p,q}^{(k)}|$. The experimental environment was PERCH coded in C language and an IBM AT machine having Intel Core2 CPU with 1.2 GHz clock and 2039MB RAM.

### 6.1 Comparison with PSD Completion, Regression Model, and NNLS

PERCH is compared with a PSD Completion program named *dualcomp* [7], the regression model described in the first section, and NNLS program named *APGL*. We generated five artificial correlation matrices which represent the correlation coefficients among object vectors forming 3 or 10 clusters with or without 10% background noise and uniformly distributed without any clusters respectively under $n = 70$ and $m = 1000$. Since $n < m$, the matrices are principally full-rank. However, the matrices obtained from the object vectors forming 3 or 10 clusters without background noise have lower ranks, because objects in each cluster are in a very close formation so that their correlations are mutually more than 0.9. On the other hand, the rank of the correlation matrix with the noise is presumed to be the number of the clusters plus the number of the noise objects up to $n$. The rank of the correlation matrix generated from uniformly distributed objects is supposed to be $n$ as all the noisy objects are independently distributed. Each single noise object forms a linearly independent component having a minor eigenvalue, and objects in each cluster form a linearly independent component having a large eigenvalue in our high contrast data sets.

Table 1, 2, and 3 show the computation time and the standard error of PERCH, dualcomp, regression model and APGL. In the experiments described in the tables, the pa-

**Table 1** Estimation results for artificial data.
($h$ = 10% of the number of elements in $OB$)

| # Cluster | Noise Type | Method | Comp. Time (sec.) | Error |
|---|---|---|---|---|
| 3 | without noise | PERCH | 3.90E-04 | 1.74E-04 |
| | | dualcomp | 5.57E+03 | 9.42E-06 |
| | | regression | 8.04E+00 | 4.73E-01 |
| | | APGL | 4.44E+00 | 4.95E-02 |
| | with noise | PERCH | 7.33E-04 | 2.69E-03 |
| | | dualcomp | 5.59E+03 | 2.70E-03 |
| | | regression | 8.83E+00 | 6.88E-01 |
| | | APGL | 3.75E+00 | 7.02E-02 |
| 10 | without noise | PERCH | 8.27E-04 | 2.03E-01 |
| | | dualcomp | 5.03E+03 | 2.06E-01 |
| | | regression | 6.06E+00 | 3.15E-01 |
| | | APGL | 1.96E+00 | 2.23E-01 |
| | with noise | PERCH | 9.67E-04 | 2.12E-01 |
| | | dualcomp | 4.72E+03 | 2.12E-01 |
| | | regression | 5.71E+00 | 2.90E-01 |
| | | APGL | 2.16E+00 | 2.21E-01 |
| - | uniformly | PERCH | 1.17E-03 | 3.21E-02 |
| | | dualcomp | 2.89E+03 | 3.21E-02 |
| | | regression | 2.89E+00 | 1.29E-01 |
| | | APGL | 9.87E-01 | 3.62E-01 |

**Table 2** Estimation results for artificial data.
($h$ = 20% of the number of elements in $OB$)

| # Cluster | Noise Type | Method | Comp. Time (sec.) | Error |
|---|---|---|---|---|
| 3 | without noise | PERCH | 3.90E-04 | 1.89E-04 |
| | | dualcomp | 3.43E+04 | 7.59E-06 |
| | | regression | 6.99E+01 | 5.55E-01 |
| | | APGL | 3.04E+00 | 5.47E-03 |
| | with noise | PERCH | 8.43E-04 | 2.18E-03 |
| | | dualcomp | 3.59E+04 | 2.40E-03 |
| | | regression | 1.03E+02 | 5.20E-01 |
| | | APGL | 1.38E+00 | 5.88E-02 |
| 10 | without noise | PERCH | 1.06E-03 | 1.51E-01 |
| | | dualcomp | 3.12E+04 | 1.63E-01 |
| | | regression | 7.22E+01 | 3.19E-01 |
| | | APGL | 1.66E+00 | 7.41E-02 |
| | with noise | PERCH | 1.25E-03 | 1.58E-01 |
| | | dualcomp | 2.97E+04 | 1.68E-01 |
| | | regression | 6.28E+01 | 3.00E-01 |
| | | APGL | 2.16E+00 | 1.21E-01 |
| - | uniformly | PERCH | 2.32E-03 | 3.20E-02 |
| | | dualcomp | 1.77E+04 | 3.20E-02 |
| | | regression | 3.52E+01 | 1.36E-01 |
| | | APGL | 1.43E+00 | 3.29E-01 |

rameter $h$ of known elements provided as $A^h$ and $B^{n-h}$ was set to be 10%, 20%, or 30% of the total number of objects in $OB$, *i.e.*, $h = 0.1n$, $0.2n$ or $0.3n$, and the input parameter $r_{th}$ of PERCH was set at 0.05 which practically provides sufficient accuracy for the evaluation of correlation coefficients. Under every conditions, PERCH almost outperformed the conventional approaches in both computation time and estimation accuracy. This is because PERCH and these artificial experiments are adapted to our problem setting described in the second section, while the conventional approaches are for more generic conditions on the matrix completion.

**Table 3** Estimation results for artificial data.
($h$ = 30% of the number of elements in $OB$)

| # Cluster | Noise Type | Method | Comp. Time (sec.) | Error |
|---|---|---|---|---|
| 3 | without noise | PERCH | 3.75E-04 | 1.57E-04 |
| | | dualcomp | aborted* | - |
| | | regression | 2.07E+02 | 4.91E-01 |
| | | APGL | 1.41E+00 | 4.78E-04 |
| | with noise | PERCH | 6.55E-04 | 2.15E-03 |
| | | dualcomp | aborted* | - |
| | | regression | 4.84E+02 | 5.14E-01 |
| | | APGL | 1.34E+00 | 4.41E-02 |
| 10 | without noise | PERCH | 1.14E-03 | 1.08E-01 |
| | | dualcomp | aborted* | - |
| | | regression | 3.84E+02 | 3.22E-01 |
| | | APGL | 1.64E+00 | 1.10E-02 |
| | with noise | PERCH | 1.48E-03 | 9.48E-02 |
| | | dualcomp | aborted* | - |
| | | regression | 3.21E+02 | 3.04E-01 |
| | | APGL | 2.43E+00 | 7.22E-02 |
| - | uniformly | PERCH | 3.46E-03 | 3.13E-02 |
| | | dualcomp | aborted* | - |
| | | regression | 1.54E+02 | 1.45E-01 |
| | | APGL | 1.65E+00 | 2.96E-01 |

For the computation time, dualcomp requires the impractically much computation time to finish the matrix completion due to its high complexity $O(n^6)$. Though APGL resulted in the shortest computation time in the conventional approaches, PERCH is still more than 1000 times faster than it.

For the estimation accuracy, while dualcomp has better accuracy than that of PERCH if the number of clusters is less, and the noise is absent, it is almost equal to that of PERCH under the other conditions which are practically often. Similarly, PERCH provides more accurate estimation than APGL except the cases of 10 clusters. The number of effective linear equality constraints for APGL is large when the matrix has observed elements forming many large eigenvalued components. These components correspond to the clusters consisting of many objects, and consequently the estimation accuracy of APGL is improved in the cases of 10 clusters and $n = 70$. As this advantage in accuracy of NNLS does not take place for large $n$ as described in the next section, the estimation accuracy of NNLS is sometimes better than that of PERCH only in the cases of large number of clusters and small-scale data sets, for which fast and accurate estimation of missing relations is less important because exhaustive computation is tractable.

### 6.2 Application to Artificial Correlation Data Having Large Sizes

To evaluate the scalability of PERCH, various artificial correlation matrices having large sizes of $n = 300 \sim 100000$ have been generated. Some of them are obtained from the object vectors forming 10 clusters with or without 10% noise, and the others are from uniformly distributed object

vectors. These features are identical with those of the small data sets in the previous subsection. We show the result of PERCH under $m = 1000$, $h = 0.2n$ and $r_{th} = 0.05$. Similar tendencies of the results have been observed under the other parameter settings. We also applied these data sets to conventional approaches dualcomp, the regression model, and APGL for reference. However, dualcomp did not work due to the memory overflow, because it consumes much memory to manage very high dimensional vectors proportional to the objective matrix size. The regression model took intractable computation time due to its very high computational complexity $O(n^5)$ as discussed in the first section.

The computation time of PERCH shown in log-log scale plots of Fig. 1 are between $O(n^2)$ and $O(n^3)$, which is consistent with the analytical result in the subsection 5.3, *i.e.*, $O(kn^2)$ and $k \in [1, n]$. Although the computational complexity of APGL is almost comparable to that of PERCH in Fig. 1, the absolute computation time of PERCH is 10 $\sim$ 1000 times faster than that of APGL.

Table 4 shows the comparison of the estimation error by PERCH and APGL. In any cases, the maximum errors and the standard errors of PERCH are less than those of APGL respectively. The standard error of PERCH is less than the given threshold $r_{th} = 0.05$ (=5.0E-02), while the maximum error, which is given by Eq. (11), is sometimes greater than $r_{th}$ in the case of the data set with noise and the uniformly distributed data set, because the matrices have high ranks due to the independent distribution of the objects, and PERCH could not have enough Pivots under the available data provided by $h = 0.2n$. Given much less residual threshold value $r_{th}$, the maximum errors are guaranteed if matrices have low ranks, and consequently the observed maximum errors are within the given error tolerances. Both standard and maximum errors are very small in the case of the uniformly distributed data set and $n = 10000$, because the number of available Pivots is $h = 0.2n = 2000$ which is larger than the dimension of the object vector $m = 1000$ and sufficiently covers the subspace where the data is distributed.

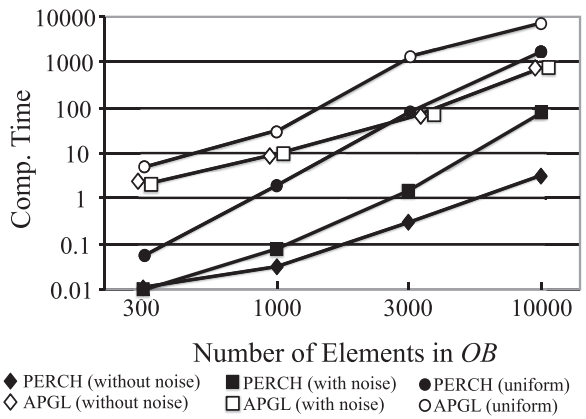


**Fig. 1** Computation time of PERCH and APGL for large artificial data.

## 6.3 Application to Real World Correlation Data

To demonstrate the practicality of PERCH, we applied PERCH to four data sets in UCI Machine Learning Repository [17]. These are *musk*, *isolet*, *spambase* and *ionosphere* in which each object is represented by a numerical vector. Their numbers of objects and vector dimensions are indicated at the bottom of Fig. 2. Unlike the artificial data, the correlations of each data are distributed in a broad range while one or two peaks are observed in the distribution. The

**Table 4** Estimation error of PERCH for large artificial data. ($h = 0.2n$ and # cluster is 10.)

| $n$ | Noise Type | Method | Standard Error | Max Error |
|---|---|---|---|---|
| | without noise | PERCH | 1.72E-04 | 7.19E-04 |
| | | APGL | 1.04E-03 | 5.36E-03 |
| 300 | with noise | PERCH | 2.53E-03 | 9.18E-02 |
| | | APGL | 2.93E-02 | 1.01E+00 |
| | uniformly | PERCH | 3.05E-02 | 1.36E-01 |
| | | APGL | 2.46E-01 | 9.99E-01 |
| | without noise | PERCH | 1.83E-04 | 7.88E-04 |
| | | APGL | 6.60E-04 | 3.14E-03 |
| 1000 | with noise | PERCH | 2.57E-03 | 1.29E-01 |
| | | APGL | 1.06E-02 | 9.97E-01 |
| | uniformly | PERCH | 2.83E-02 | 1.36E-01 |
| | | APGL | 2.35E-01 | 1.00E+00 |
| | without noise | PERCH | 1.74E-04 | 8.78E-04 |
| | | APGL | 5.91E-04 | 2.68E-03 |
| 3000 | with noise | PERCH | 2.46E-03 | 1.38E-01 |
| | | APGL | 6.57E-03 | 9.97E-01 |
| | uniformly | PERCH | 1.99E-02 | 1.04E-01 |
| | | APGL | 8.18E-02 | 7.92E-01 |
| | without noise | PERCH | 1.83E-04 | 1.02E-03 |
| | | APGL | 5.56E-04 | 2.67E-03 |
| 10000 | with noise | PERCH | 2.28E-03 | 1.34E-01 |
| | | APGL | 4.46E-03 | 9.99E-01 |
| | uniformly | PERCH | 5.13E-03 | 3.26E-02 |
| | | APGL | 5.85E-02 | 9.50E-01 |

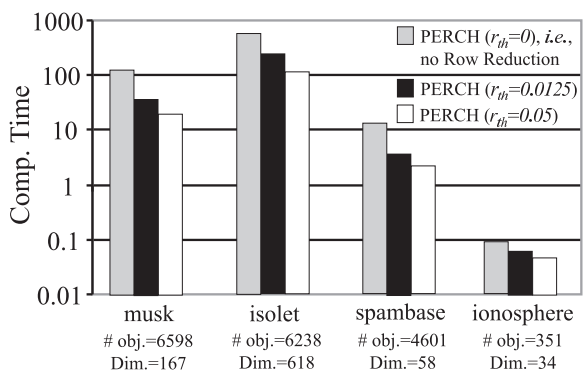


**Fig. 2** Computation time of PERCH for UCI data.

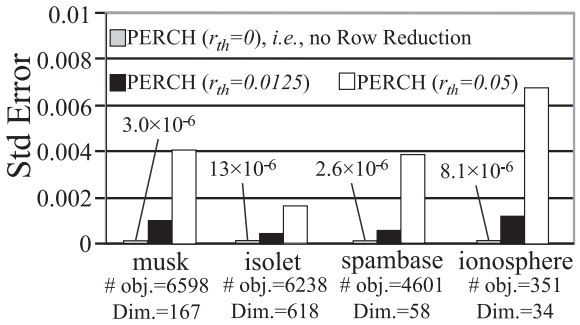


**Fig. 3** Computation error of PERCH for UCI data.

correlation histogram of *musk* has two sharp peaks around 0.3 and 0.7, that of *isolet* has one sharp peak around 0.8, *spambase* has a precipitous peak near 1.0, and *ionosphere* has two moderate peaks around 0.0 and 0.9.

Figure 2 shows the computation time of three PERCH settings of $r_{th} = 0$, 0.0125 and 0.05 under given $h = 0.2n$. This indicates that PERCH can quickly estimate all missing PSD similarities of practical and large data having broad relation distributions. The computation time for $r_{th} = 0$ is longer than those for $r_{th} = 0.0125$ and 0.05 because no Row Reduction is applied and all the available $h = 0.2n$ Pivots are selected when $r_{th} = 0$. Figure 3 depicts the standard error of PERCH under the three settings. The standard error is smaller than $r_{th}$ in every cases for $r_{th} = 0.0125$ and 0.05. The standard error for $r_{th} = 0$, *i.e.* PERCH without Row Reduction, has excess accuracy by consuming longer computation time.

### 6.4 Application to Real World Diffusion Kernel Data

To assess the performance of PERCH for more expensive relation matrices, we applied PERCH to the estimation of the missing elements in three graph diffusion kernel matrices [18]. Given an $n \times n$ Laplacian matrix $L$ of a graph $G(V, E)$ as follows.

$$L = \begin{cases} deg\{v_i\}, & \text{for } i = j \\ -1, & \text{for } i \neq j \text{ and } (v_i, v_j) \in E \\ 0, & \text{otherwise} \end{cases},$$

where $i, j = 1, \cdots, n(= |V|)$, $v_i, v_j \in V$, and $deg\{v_i\}$ is the degree of $v_i$. The graph diffusion kernel matrix of $G$ is defined by the following formula.

$$K(\beta) = \exp(-\beta L) = \lim_{n\to\infty} \left(I - \frac{\beta L}{n}\right)^n,$$

where $\beta > 0$ is a parameter to define the mean diffusion range, and $I$ is an $n \times n$ identity matrix. Each element of this matrix represents the closeness among vertices in the structure of the graph $G$. This can be computed more easily by the following expression,

$$K(\beta) = V^{-1} \exp(-\beta D)V,$$

in which $D$ is diagonal and $V$ is a matrix to diagonalize $L$ as $L = V^{-1}DV$.

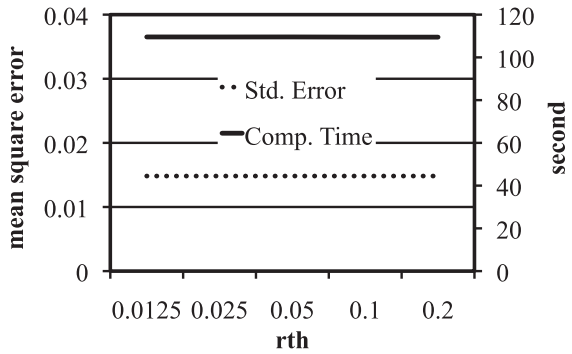


**Fig. 4** Computation time and standard error of PERCH for Tic Tac Toe data. ($n$ = 5634)

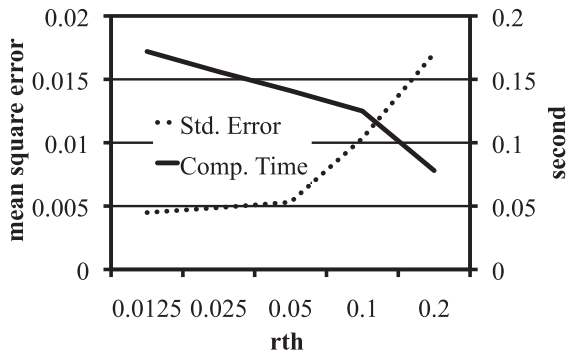


**Fig. 5** Computation time and standard error of PERCH for metabolic pathways data. ($n$ = 668)

We computed the diffusion kernel matrices of three graph data by setting $\beta = 10$ which characterize the moderately local structure of the graphs. The first graph is the entire state transition network of a game named *Tic Tac Toe* which is available in UCI Machine Learning Repository [17]. This represents a large graph containing 5634 vertices where each vertex represents a state of the game, and each edge represents a possible transition path of the game. The second graph is a network representing *metabolic pathways* of the yeast S. Cerevisiae in KEGG/PATHWAY database [19]. Each vertex of this network represents a protein, and each edge indicates that the protein pair works as enzymes that catalyze successive reactions in the pair. This graph contains 668 vertices, and is known to have a hierarchical hub and spoke structure in which edges are sparse. The third is a *protein-protein interactions* network constructed by von Mering et al. [20]. We used the medium confidence network data consisting of 2617 vertices which has been analyzed in many studies of bioinformatics and link prediction [21], [22]. This graph is larger than that of the metabolic pathways but similarly sparse.

Figure 4, 5 and 6 indicate the computation time and standard error of PERCH. PERCH was tested under various residual threshold $r_{th}$ under given $h = 0.2n$ to assess the relation between the computation time and the estimation error bound. The numbers of the known elements used in Pivot-

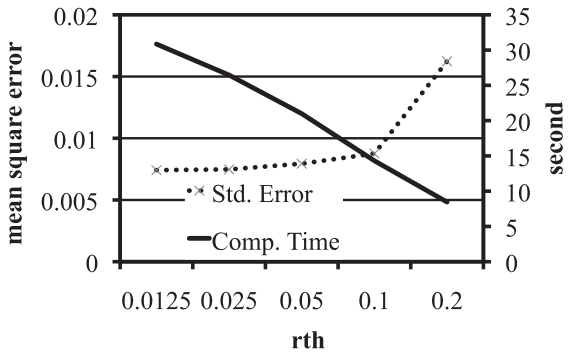


**Fig. 6** Computation time of PERCH and EC for protein-protein interactions data. ($n = 2617$)

ing were 9.7% for *Tic Tac Toe*, 5.2% ∼ 7.6% for *metabolic pathways* and 8.8% ∼ 13.3% for *protein-protein interactions* under this range of $r_{th}$, respectively. By reducing $r_{th}$, *i.e.*, enhancing the accuracy, PERCH becomes to require more computation time and provide smaller standard error in both Fig. 5 and Fig. 6. In contrast, the case of *Tic Tac Toe* depicted in Fig. 4 does not show any significant difference of the computation time and the estimation error under any $r_{th}$. This is due to a particular feature of its kernel matrix reflecting a uniform structure of the state transition network governed by some simple rules of the game. The number of vertices of the state transition network is $n = 5634$, and hence $h = 0.2n = 1126$. On the other hand, the kernel matrix has an eigenvalue equal to 9 having 547 degeneracy degree, and the rests are all 0. This indicates that the kernel matrix has 547 principal components having an identical magnitude only. Accordingly, the maximum residual is greater than any $r_{th}$ in this evaluation before Incomplete Cholesky Decomposition progresses to $k = 547 < h$ to capture all of the principal components, and suddenly becomes zero at $k = 547$. Subsequently the Row Reduction is always terminated under any $r_{th}$ at $k = 547$. Thus the computation time and the estimation error of *Tic Tac Toe* data are independent from $r_{th}$ and constant as shown in Fig. 4.

## 7. Discussion

The experimental results show that PERCH has significant efficiency when the objective PSD matrix has a large size and/or a low rank. As discussed in the third section, PERCH includes Incomplete Cholesky Decomposition applying Row Reduction and the derivation of the unknown elements having $O(k^3) \sim O(k^2 n)$ and $O(kn^2)$ computational complexity respectively. This explains the strong dependency of the efficiency on the matrix size $n$. In addition, the computation time of PERCH is limited by the size of the observed part of the matrix, $h$, because $k \leq h$ always hold. Particularly, if the matrix has a less rank, the number of steps to terminate Incomplete Cholesky Decomposition applying Row Reduction, $k$, becomes smaller. This is also consistent with the experimental result. In contrast, the experiments shows that dualcomp and the regression are not very tractable. This is due to their very high computational complexity which are $O(n^6)$ for dualcomp and $O(n^5)$ for the regression as mentioned in the first section. APGL showed computational complexity comparable to that of PERCH as expected. Increase in computation time for each approach is consistent with each computational complexity respectively. However in absolute practical computation time, PERCH is faster than APGL in orders of magnitude.

The experiments also show that PERCH has significant accuracy when the objective matrix is observed as Eq. (1) and the size of the observed part, $h$, is greater than the rank of the matrix. This is because majority of principal components of the matrix are statistically reflected in the observed part as the size of the observed part $h$ is large enough and the rows and the columns of the matrix are randomly included in its observed part.

The main reason of the advantage of PERCH in our problem setting is that it ties each missing element with one of the strongest local PSD constraints, while PSD Completion and NNLS globally optimizes objective functions over the entire matrix regardless of the accuracy of the individual missing element. PERCH also outperforms the regression model both in terms of computation time and accuracy because PERCH is highly adapted to our problem setting by leveraging PSD constraints and an assumption that missing elements make up a principal submatrix of an original PSD matrix.

It is reasonable that conventional approaches are outperformed by PERCH as they are intended to solve general problems. PSD Completion completes a matrix representing a chordal graph when it is interpreted to be an adjacency matrix of the graph having non-zeros at known elements and zeros at missing elements [5]. A graph is chordal if any minimal cycles in the graph have at most three vertices. On the other hand, NNLS is suitable for solving large-scale matrix completion problems when the solution matrix has low-rank. PERCH is applicable to a matrix representing a complete split graph which is a subclass of the chordal graph, thus, the generality is more limited in the order of NNLS, PSD Completion and PERCH. Although PERCH is specialized to the particular problem setting described in the Sect. 2, it is shown that PERCH can quickly and accurately estimate the missing elements in a PSD matrix compared to the conventional approaches.

## 8. Demonstrative Application

This section demonstrates an application example of PERCH to indicate its potential practicality in the real world use. We apply PERCH to a problem to enhance the activity of user communities by promoting user interaction in Social networking service (SNS). PERCH finds user pairs likely to become friends, and these users are mutually introduced by a SNS automated service. Such a SNS service is expected to increase the entire SNS activity.

Epinions.com, a platform on the Web for evaluations and personalized recommendations on commercial products

by its users, is used as the target SNS [23]. The users can recommend some products to their friends in the platform, and thus the platform has log data on the friendships of these users. The data represents friendships among 165592 users in the SNS where the average number of friends per user is 3.8. We define core users as the users who have more than 10 times the average number of friends, *i.e.*, specifically 40 friends. Total number of the core users included in this SNS log data is 922. These core users are considered to be key persons in this network of product recommendations, and the increase of the proper friendships among the core users is expected to enhance the activity of this SNS. In this application, PERCH is used to estimate the core user pairs which have high potential to become new friends. The friendships among the core users are represented by an adjacency matrix where an element is 1 if the user pair corresponding to the row and column of the matrix has a friendship, otherwise 0. Note that PERCH was used to estimate the current unknown elements in a partially observed matrix so far in this paper. In this section, however, all the current friendship information is already known, *i.e.*, all the elements of the adjacency matrix are given, and PERCH is used to *forecast* the future friendships, *i.e.*, the future elements in the matrix. The detailed approach is explained below.

To apply PERCH to this problem, a core user set $OB$ is randomly halved into two sets $OA^h$ and $OB^{n-h}$ where $n = 922$ and $h = 461$. PERCH estimates core user pairs which are likely to become friends in $OB^{n-h}$. Here we set the following assumptions for the future friendship forecast.

(1) The friendships among all core users in $OB$ follows an identical statistical distribution.
(2) The statistical distribution of the friendships is steady under a stable condition for some period.

PERCH estimates the friendships in $OB^{n-h}$ from the rest in $OB$. Because of the assumption (1), the actual friendships in $OB^{n-h}$ are expected to match to the friendships in $OB^{n-h}$ estimated from the rest in $OB$ under the assumption (2). In other words, the user pairs in $OB^{n-h}$ are considered to have high potential to build friendships in the future if they are currently not friends but in the estimation result. As an adjacency matrix is not a PSD matrix, PERCH is not directly applicable. Alternatively, PERCH applies to the corresponding normalized signless Laplacian matrix which is always PSD. The partial adjacency matrix representing the friendships within $OA^h$ and between $OA^h$ and $OB^{n-h}$ is transformed into the corresponding partial signless Laplacian matrix where each diagonal element is a summation of all elements in the row (the column) of the original adjacency matrix, and each off-diagonal is the element of the original adjacency matrix. This partial signless Laplacian matrix is further transformed into a partial normalized signless Laplacian matrix by diagonal normalization, and is applied to PERCH for the estimation of the rest part reflecting the potential friendships in $OB^{n-h}$. Because this transformation is invertible, the objective adjacency matrix including the elements to represent the potential friendships in $OB^{n-h}$ is easily computed from the estimated normalized signless Laplacian matrix. However, the estimated elements representing the potential friendships in $OB^{n-h}$ are not binary values in most cases due to the nature of the PSD estimation. Accordingly, we apply a discretization of each estimated element by introducing a common threshold value $f_{th}$ where each element is discretized into 1 if its value is greater than or equal to $f_{th}$ otherwise it is discretized into 0. A pair of the core members in $OB^{n-h}$, which corresponding elements in the estimated and discretized adjacency matrix and the original adjacency matrix are 1 and 0 respectively, is judged to have a future friendship. Precision is a rate of the currently existing friendships among the estimated potential friendships in $OB^{n-h}$, and recall is a rate of the estimated potential friendships among the currently existing friendships in $OB^{n-h}$ in our problem setting. The future friendship forecast must achieve a certain recall of its estimation to efficiently capture the future friendships while maintaining its precision to ensure the estimation reliability. Accordingly, the threshold $f_{th}$ was set at 0.15 which achieves the best recall of the estimation under keeping its precision greater than 0.5 through preliminary numerical experiments. Under this setting, the precision was 0.55 and the recall was 0.19. The residual threshold $r_{th}$ required in the row reduction of PERCH is set for 0 to provide the maximum accuracy of the PSD estimation. The computation time of the PSD estimation by PERCH under this parameter setting is around 3.24 seconds in average that is sufficiently fast for the real application.

The random halving of $OB$, the conversion of the partial adjacency matrix to its partial normalized signless Laplacian and the PSD estimation by PERCH described in the former paragraph are repeated, and an ensemble of the estimated results is taken to finally provide the accurate forecast on the future friendships among all core members in $OB$. The number of the repeats is set for 20. By this procedure, 5 estimations on each pair of the core members is obtained in the average over the 20 repeats. This is because the probability that a pair of particular two members is included in $OB^{n-h}$is $1/4$ under the random halving. For taking the ensemble, a bagging technique is applied [24]. The potential friendship of a core member pair is forecasted if more than or equal to $s_{th} = 2$ estimations predict the potential friendship among the multiple estimations in the bagging. The precision and the recall of this bagging decision are expected to be 0.94 and 0.23 respectively based on the aforementioned precision 0.55 and recall 0.19 of every single estimation and an assumption of mutual independence of the individual estimations. Although these expected numbers are merely upper bounds of the precision and the recall under the assumption on the ideal mutual independence, we apply $s_{th} = 2$ since the high precision with the moderately large recall are desirable to serve the reliable introduction of potential friends to many core members.

Table 5 shows the forecast result by PERCH and the bagging technique. This method found 2295 potential friend pairs and 60% of them have already been real friend pairs as

**Table 5** SNS friendship forecast result.
($f_{th} = 0.15$, $r_{th} = 0$ and $h = 50\%$ of the elements in $OB$)

| | |
|---|---|
| # forecasted potential friend pairs | 2295 |
| # forecasted potential friend pairs in the non-friend pairs | 889 |
| Precision in the estimation of the potential friendships | 0.60 |
| Recall in the estimation of the potential friendships | 0.20 |

shown in the table. Because the rest 40%, *i.e.*, 889 pairs, are the potential friend pairs which have not had any friendship yet, invitations by the automated friend introduction should be sent to these potential friend pairs. After the invitations are sent, 60% of them, *i.e.*, 533 pairs, are expected to become new friends according to the accuracy of the forecast. Although the forecast does not exhaustively find all the potential friendships because of the low recall, the gain of new core user friend pairs, *i.e.*, 533, which is 7.6% of the current friend pairs between the core users, will increase the communication activities and the associated product recommendations between the users in proportion to the increase rate of the friendships. The result indicates that this application of PERCH enables high quality automated friend introduction for the SNS service, and the practicality of PERCH was demonstrated for the service such as the SNS automated friend introduction.

## 9. Conclusion

We have formalized a problem to estimate the missing part of a PSD matrix, proposed novel techniques named PSD Estimation and Row Reduction for both accurate and efficient estimation of a PSD matrix, and developed PERCH introducing these techniques. PERCH requires conditions that the matrix is PSD and the missing elements compose a principal submatrix of the original PSD matrix. PERCH is adapted to our problem setting using the above conditions and showed superior performance in both efficiency and accuracy in practical problems in comparison with the conventional approaches of dualcomp, regression and APGL. The conditions where PERCH has significant efficiency is that the objective PSD matrix has a large size and/or a low rank, while the condition where PERCH has significant accuracy is that a part of the objective matrix is randomly observed, and the size of the observed part is greater than the rank of the matrix. Particularly, PERCH enhances the scalability to estimate large PSD matrices as its computational complexity is $O(k^3 + kn^2) \sim O(k^2n + kn^2)$ while those of the conventional approaches are $O(n^3) \sim O(n^6)$. In our performance evaluation for real world data sets, PERCH showed tractable performance, whereas the other conventional approaches required computation time intractable or longer than that of PERCH. Furthermore, we demonstrated the practicality of PERCH by applying it to a SNS automatic friend introduction example. Our proposed techniques can be extended and applied to various PSD matrices and problem settings.

## References

[1] E. Dietzenbacher, M.L. Lahr, and W.W. Leontief, Wassilly leontief and input-output economics, Cambridge University Press, 2004.
[2] Y. Noda, "World trade matrix revised: by asian international input-output table 24 sectors," Report of Institute of Developing Economics Japan External Trade Organization (IDE-JETRO), no.84, 2003.
[3] H. Taira, Y. Fan, K. Yoshiya, and H. Miyagi, "A method of constructing pairwise comparison matrix in decisionmaking," IEEE International Conference on Systems, Man, and Cybernetics, vol.4, no.14-17, pp.2511–2516, 1996.
[4] M. Laurent, "Cuts, matrix completions and graph rigidity," Mathematical Programming, vol.79, pp.255–283, 1997.
[5] R. Grone, C.R. Johnson, E.M. Sa, and H. Wolkowicz, "Positive definite completions of partial hermitian matrices," Liner Algebra and its Applications, Elsevier Science, vol.58, pp.109–124, 1984.
[6] C.R. Johnson, B. Kroschel, and H. Wolkowicz, "An interior point method for approximate positive semidefinite completions," Computational Optimization and Applications, vol.9, no.2, pp.175–190, Kluwer Academic Publishers, 1998.
[7] H. Wolkowicz, "dualcomp," Henry Wolkowicz's Research, Faculty of Mathematics, University of Waterloo, http://orion.math.uwaterloo.ca/%7Ehwolkowicz/henry/software/MatrixApprox.shtml, 2007.
[8] T. Graepel, "Kernel matrix completion by semidefinite programming," Artificial Neural Networks, Proc. ICANN 2002: International Conference on Artificial Neural Networks, Springer Verlag, pp.687–693, 2002.
[9] K. Tsuda, S. Akaho, and K. Asai, "The *em* algorithm for kernel matrix completion with auxiliary data," J. Machine Learning Research, vol.4, pp.67–81, 2003.
[10] D. Williams and L. Carin, "Analytical kernel matrix completion with incomplete multi-view data," Proc. Workshop on Learning with Multiple Views at the 22nd International Conference on Machine Learning (ICML2005), pp.80–86, 2005.
[11] B. Kaltenbacher, "Regularization by truncated cholesky factorization: A comparison of four different approaches," J. Complexity, vol.23, pp.225–244, 2007.
[12] S. Fine and K. Scheinberg, "Predictive low-rank decomposition for kernel methods," J. Machine Learning Research, vol.2, pp.243–264, 2001.
[13] F.R. Bach and M.I. Jordan, "Predictive low-rank decomposition for kernel methods," Proc. ICML: 22nd International Conference on Machine Learning, 2005.
[14] K. Tanabe, "Norm-reducing pivoting strategy for the cholesky method and an incomplete decomposition," The Institute of Statistical Mathematics Cooperative Research Report, vol.4, pp.41–48, 1987.
[15] C.K.I. Williams and M. Seeger, "The effect of the input density distribution on kernel-based classifiers," Proc. ICML: 17th International Conference on Machine Learning, 2000.
[16] R. Bhatia, Positive definite matrices, Princeton Series in Applied Mathematics, Princeton University Press, Princeton, New Jersey, 2007.
[17] C.B.D. Newman, S. Hettich, and C. Merz, "Uci repository of machine learning databases," University of California, Irvine, Dept. Information and Computer Sciences, http://mlearn.ics.uci.edu/MLRepository.html, 1998.
[18] R.I. Kondor and J. Lafferty, "Diffusion kernels on graphs and other discrete input spaces," Proc. 19th International Conference on Machine Learning, pp.315–322, 2002.
[19] M. Kanehisa, G. Susumu, K. Shuichi, O. Yasushi, and H. Masahiro, "The KEGG resources for deciphering the genome," Nucleic Acids Research, vol.32, pp.D277–D280, 2004.
[20] C. von Mering, R. Krause, B. Snel, M. Cornell, S.G. Oliver, S.

Fields, and P. Bork, "Comparative assessment of largescale data sets of protein-protein interactions," Nature, vol.417, pp.399–403, 2002.
[21] K. Tsuda and W.S. Noble, "Learning kernels from biological networks by maximizing entropy," Bioinformatics, vol.20, no.1, pp.i326–i333, 2004.
[22] H. Kashima and N. Abe, "A parameterized probabilistic model of network evolution for supervised link prediction," Proc. 6th IEEE International Conference on Data Mining (ICDM2006), pp.340–349, 2006.
[23] Epinions.com, "a Shopping.com company," http://www99.epinions.com/about/, accessed Jan. 14. 2010.
[24] L. Breiman, "Bagging predictors, technical report no.421," Department of Statistics University of California Berkeley, 1994.
[25] K.C. Toh and S.W. Yun, "An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems," Pacific J. Optimization, no.6, pp.615–640, 2010.
[26] K.C. Toh and S.W. Yun, "NNLS version 0 – a MATLAB software for nuclear norm regularized linear least squares problems based on an accelerated proximal gradient method," Home page: TOH Kim Chuan, Department of Mathematics, National University of Singapore, http://www.math.nus.edu.sg/˜mattohkc/NNLS.html.

**Hiroshi Kuwajima** has graduated from the master course of Graduate School of Osaka University, Japan in 2008. He is an engineer at DENSO CORPORATION. At DENSO CORPORATION, he works on the software platform development of car electric control units (ECU). DENSO CORPORATION, 1-1 Showa-cho Kariya, Aichi, 448-8661 Japan.

**Takashi Washio** received the Ph.D. degree in nuclear engineering from Tohoku University, Japan, in 1983, on the topic of process plant diagnosis based on qualitative reasoning. He is a professor in the Institute of Scientific and Industrial Research (ISIR), Osaka University. At ISIR, he works on the study of scientific discovery, graph mining, and high-dimensional data mining. He received the best paper award from the Atomic Energy Society of Japan in 1996, the best paper award from the Japanese Society for Artificial Intelligence in 2001, and the Journal Award of Computer Aided Chemistry in 2002, Contribution Award from the Japanese Society for Artificial Intelligence in 2009. He is a member of the IEEE Computer Society. Department of Advanced Reasoning, The Institute of Science and Industrial Research, Osaka University, 8-1 Mihogaoka Ibaraki, Osaka, 567-0047 Japan.

**Ee-Peng Lim** is currently a tenured professor at the School of Information Systems of the Singapore Management University (SMU). He received Ph.D. from the University of Minnesota, Minneapolis in 1994 and B.Sc. in Computer Science from National University of Singapore. His research interests include social network/web mining, information integration, and digital libraries. He is currently an Associate Editor of the ACM Transactions on Information Systems (TOIS), Information Processing and Management (IPM), Social Network Analysis and Mining, Journal of Web Engineering (JWE), IEEE Intelligent Systems, International Journal of Digital Libraries (IJDL) and International Journal of Data Warehousing and Mining (IJDWM). He is a member of the ACM Publications Board. He is a member of the Steering Committee of the International Conference on Asian Digital Libraries (ICADL). He is also a member of the Steering Committee of the Pacific Asia Conference on Knowledge Discovery and Data Mining (PAKDD). School of Information Systems, Singapore Management University, 80 Stamford Road, 178902 Singapore.