Co-embedding of Structurally Missing Data by Locally Linear Alignment

Takehisa Yairi

Research Center for Advanced Science and Technology, University of Tokyo
yairi@space.rcast.u-tokyo.ac.jp

Abstract. This paper proposes a "co-embedding" method to embed the row and column vectors of an observation matrix data whose large portion is structurally missing into low-dimensional latent spaces simultaneously. A remarkable characteristic of this method is that the co-embedding is efficiently obtained via eigendecomposition of a matrix, unlike the conventional methods which require iterative estimation of missing values and suffer from local optima. Besides, we extend the unsupervised co-embedding method to a semi-supervised version, which is reduced to a system of linear equations. In an experimental study, we apply the proposed method to two kinds of tasks – (1) Structure from Motion (SFM) and (2) Simultaneous Localization and Mapping (SLAM).

1 Introduction

Recently, the dimensionality reduction and matrix factorization techniques have been regarded as a significant machine learning tool for feature extraction and data compression, as both the size and dimensionality of data in most application are continuing to increase rapidly.

A non-trivial issue in applying these techniques to actual problems is how to deal with *missing* data elements, as the real-world data, e.g., medical testing data, food preference questionnaire data, purchase records, etc. usually contains missing parts. If the missing portion is relatively small, *ad hoc* treatment such as filling the missing elements with constant values and inferring them from similar data is acceptable. A more sophisticated approach commonly used is to alternately estimate the missing values and conduct dimensionality reduction or matrix factorization until convergence. The method is known as EM (expectation maximization) algorithm in machine learning. However, if the missing portion is very large and has some structural pattern, these conventional approaches are expected to fail.

Consider the following situation for an example. An observer is wandering around the town, carrying a wireless device (such as tablet PC). The device is assumed to be capable of recording approximate relative directions to all detected wireless access points (APs). If the device could always communicate with all APs in the town wherever it is, the observation data could be represented as a complete matrix, whose (i, j)-th element is the relative direction to the j-th AP from the i-th observation position. Unfortunately, however, most of the elements are missing, because the wireless communication range is limited and affected by occlusion. Besides, the pattern of missing data is not random but structured, as whether a measurement is present or absent is dependent on the

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 419–430, 2012.

spatial relationship between the observer and AP. The conventional approaches are not suitable for this kind of missing data.

In this paper, we propose the locally linear alignment co-embedding (LLACoE) that embeds both row and column vectors of a matrix-form observation data with largely and structurally missing elements into low-dimensional latent spaces respectively. A key idea is that a measurement $\boldsymbol{y}_{i,j}$ can be approximated by some linear projection of the state vector of j-th object $\boldsymbol{z}_j$ onto the subspace determined by the observer's state $\boldsymbol{x}_i$. A remarkable feature of LLACoE is that it does not require iterative computation to estimate the missing values, but is efficiently solved by eigendecomposition or a system of linear equations.

2 Related Works

Dimensionality reduction is a major topic of machine learning, as well as classification, regression and clustering. Especially, in the last decade, non-linear dimensionality reduction (a.k.a. manifold learning) methods such as Isomap[7] and LLE[3] have been developed and become popular. In addition, matrix factorization or low-rank matrix approximation techniques such as singular value decomposition (SVD) and non-negative factorization (NMF) have been widely used in a variety of datamining applications.

A practical difficulty is that the real-world data is not only huge and high-dimensional, but also often incomplete due to various reasons. The simplest way of dealing with such incomplete data is to fill the missing parts with some proper constant values, typically by zero. This approach will be reasonable enough, when the values are "missing" because they are out of measurement ranges. However, the applicability of this method is obviously limited, because not all measurement data have such a property. Besides, it is sometimes nontrivial to find a proper constant value, even when it is applicable.

A more sophisticated and popular approach is to estimate the missing values and conduct dimensionality reduction or matrix factorization alternately until it converges. In computer vision (CV), PCAMD (PCA with missing data) methods[5] such as alternate least squares (ALS) and Wiberg's algorithm[1] have been utilized for the structure from motion (SFM), in which a 3-dimensional surface model of target object is estimated from a sequence of 2-dimensional images. In machine learning (ML), this kind of iterative algorithm is generally formalized as the EM algorithm. In fact, it was shown that PPCA (probabilistic PCA) with EM algorithm can deal with incomplete data[4]. Also in NMF, some iterative algorithms that alternately estimating missing values and factorizing a matrix into two low-rank ones have been recently developed[6]. While this iterative estimation approach works fine if the missing part is relatively small, the convergence property and solution quality become drastically worse as the missing portion becomes larger. Besides, even if the missing data has some pattern or structure which contains information of latent low-dimensional spaces, it does not have any mechanism to utilize the information. In summary, these conventional approaches implicitly assume small and randomly generated missing elements.

In contrast, our method utilizes only existing elements of the matrix data, which means it is not necessary to fill the absent elements with constants, nor to estimate them alternately. In addition, it takes advantage of the pattern of missing data, based on the idea that existing (i.e. not missing) elements are roughly linear to their corresponding latent vectors.

3 Problem Definition

In this paper, we deal with a $M \times N$ data matrix $\boldsymbol{Y} = [\boldsymbol{y}_{i,j}]_{i=1,\dots,M,j=1,\dots,N}$. It should be noted that (i,j)-th element $\boldsymbol{y}_{i,j}$ is a D-dimensional vector in general.[1] As $\boldsymbol{Y}$ contains missing elements, we introduce a set of Boolean indicator variables $\{q_{i,j}\}$ to specify whether each element is existing or missing. That is to say,

$$q_{i,j} = \begin{cases} 0 & (\text{if}(i,j)\text{-th element } \boldsymbol{y}_{i,j} \text{ is missing}) \\ 1 & (\text{otherwise}) \end{cases} \quad (1)$$

Now we pursue two goals at the same time:

1. Obtain a set of n-dimensional row latent vectors $\boldsymbol{X} = [\boldsymbol{x}_1, \dots, \boldsymbol{x}_M]^\top$ by reducing the dimension of $\boldsymbol{Y}$'s row vectors.
2. Obtain a set of m-dimensional column latent vectors $\boldsymbol{Z} = [\boldsymbol{z}_1, \dots, \boldsymbol{z}_N]^\top$ by reducing the dimension of $\boldsymbol{Y}$'s column vectors.

where, $n << N \cdot D$ and $m << M \cdot D$. It should be noted that our purpose is not to approximate or reconstruct $\boldsymbol{Y}$ by the product of $\boldsymbol{X}$ and $\boldsymbol{Z}^\top$, but to embed the row and column vectors of $\boldsymbol{Y}$ to low dimensional latent spaces respectively. It can be called as simultaneous dimensionality reduction or *co-embedding*.

We can give another view to this problem. First, assume that a measurement $\boldsymbol{y}_{i,j}$ is generated by an *unknown* function of an observer's latent state $\boldsymbol{x}_i \in \mathcal{R}^n$ and an item's latent state $\boldsymbol{z}_j \in \mathcal{R}^m$, i.e.,

$$\boldsymbol{y}_{i,j} = g(\boldsymbol{x}_i, \boldsymbol{z}_j) + \boldsymbol{e}_{i,j} \quad (2)$$

where $\boldsymbol{e}_{i,j}$ is the noise. Our goal is to estimate sets of $\{\boldsymbol{x}_i\}$ $(i = 1, \dots, M)$ and $\{\boldsymbol{z}_j\}$ $(j = 1, \dots, N)$, when a *partial* set of $\{\boldsymbol{y}_{i,j}\}$ is given. Note that function g itself is not necessarily estimated.

Now we make an assumption that the presence of an observation $\boldsymbol{y}_{i,j}$ has a locality as to $\boldsymbol{z}_j$. Roughly speaking, this assumption states "if there exists i such that $q_{i,j} = q_{i,j'} = 1$, then $\boldsymbol{z}_j$ and $\boldsymbol{z}_{j'}$ are close to each other".

While this assumption seems to be very restrictive, there are many problems which hold this property in fact. For example, in the case of mobile wireless device and access points mentioned in section 1, this assumption is expected to be valid because the device at a position $\boldsymbol{x}_i$ can communicate only with APs in its neighborhood. It is also the case with SLAM (simultaneous localization and mapping) problem[8] in mobile robotics, where $\boldsymbol{x}_i$ is the robot's pose and $\boldsymbol{z}_j$ is the j-th landmark's position. Another example is the SFM (structure from motion) problem in computer vision, where $\boldsymbol{x}_i$ is the relative spatial relationship between the camera and target object, $\boldsymbol{z}_j$ is the j-th visual feature's

[1] Although $\boldsymbol{Y}$ should be regarded as a $M \times N \times D$ tensor in this sense, we treat it as a matrix whose element is a vector because it makes us understand the subsequent discussion more easily.

3D coordinates in the body frame, and $\boldsymbol{y}_{i,j}$ is its 2D coordinates on the camera screen. Obviously, if j-th and j'-th features are observed at the same time, they are expected to close to each other.

The assumption may be valid even in collaborative filtering. If we consider the Netflix rating data set, $\boldsymbol{x}_i$ is the preference of the i-th user, and $\boldsymbol{z}_j$ is the j-th movie. If a user watched two movies, they are likely to be in the same genre.

4 Locally Linear Alignment Co-embedding

In this section, we introduce the proposed method named LLACoE (locally linear alignment co-embedding).

4.1 Basic Idea

We consider the above assumption "if there exists i such that $q_{i,j} = q_{i,j'} = 1$, then $\boldsymbol{z}_j$ and $\boldsymbol{z}_{j'}$ are close to each other" holds. Then, if $q_{i,j} = 1$ or $\boldsymbol{y}_{i,j}$ is not missing, a linear approximation below is possible in its neighborhood, i.e.,

$$\boldsymbol{y}_{i,j} = g(\boldsymbol{x}_i, \boldsymbol{z}_j) + \boldsymbol{e}_{i,j} \approx \boldsymbol{G}(\boldsymbol{x}_i)[\boldsymbol{z}_j^\top, 1]^\top = \boldsymbol{G}(\boldsymbol{x}_i)\tilde{\boldsymbol{z}}_j \tag{3}$$

where $\boldsymbol{G}(\boldsymbol{x}_i)$ stands for a projection matrix determined by $\boldsymbol{x}_i$, and $\tilde{\boldsymbol{z}}_j$ is a homogeneous coordinates of $\boldsymbol{z}_j$.

Assume that $\boldsymbol{x}_i$ implies observer's latent state at time i, while $\boldsymbol{z}_j$ implies j-th object's state or position. Then the above approximation states that when observer's state is $\boldsymbol{x}_i$, its observation data is formed by linear projections of all observable objects $j \in \mathcal{V}_i$ into the observation subspace $\boldsymbol{G}(\boldsymbol{x}_i)$ determined by $\boldsymbol{x}_i$. In other words, each observation data at a time can be regarded as linear projections of a piece (fragment) of the whole world's state into a low-dimensional perception space.

Now, our first goal is to reconstruct the latent states of all objects, i.e., $\{\boldsymbol{z}_j\}$ by *aligning* the pieces of observation data. Intuitively, it is similar to jigsaw puzzles or reconstruction of fragmentary fossils. Since the alignment operation of each piece reflects the observer's state, $\boldsymbol{x}_i$ is also expected to be reconstructed. In the remaining of this section, we will explain how to realize this rough idea.

4.2 Unsupervised Locally Linear Alignment Co-embedding

First we consider reconstructing the column latent vectors $\{\boldsymbol{z}_j\}$. The assumption in the previous section means that $\boldsymbol{z}_j$ is approximately linear (more strictly, affine) to $\boldsymbol{y}_{i,j}$ if $q_{i,j} = 1$. We use this local linearity property in a reverse way. That is to say, we think of approximating $\boldsymbol{z}_j$ by an affine transformation of $\boldsymbol{y}_{i,j}$ when $q_{i,j} = 1$:

$$\hat{\boldsymbol{z}}_{i,j} \equiv \boldsymbol{T}_i[\boldsymbol{y}_{i,j}^\top, 1]^\top = \boldsymbol{T}_i\tilde{\boldsymbol{y}}_{i,j} \tag{4}$$

where $\boldsymbol{T}_i$ is an alignment transformation matrix common for $\boldsymbol{y}_{i,j}$ ($j = 1, \ldots, N$) as long as $q_{i,j} = 1$. $\tilde{\boldsymbol{y}}_{i,j}$ is the homogeneous coordinates of $\boldsymbol{y}_{i,j}$. It would be reasonable to decide the final estimate of $\boldsymbol{z}_j$ by averaging all the temporary estimates as:

$$\hat{\boldsymbol{z}}_j = \frac{\sum_{i=1}^{M} q_{i,j}\hat{\boldsymbol{z}}_{i,j}}{\sum_{i=1}^{M} q_{i,j}} = \sum_{i=1}^{M} \tilde{q}_{i,j}\hat{\boldsymbol{z}}_{i,j} \tag{5}$$

where, $\tilde{q}_{i,j} = q_{i,j}/\sum_{i=1}^{M} q_{i,j}$ is the normalized observability indicator.

Now our main concern is how we can obtain the optimal set of alignment matrices $\{\boldsymbol{T}_i\}$ $(i = 1, \ldots, M)$. A reasonable way is to choose them so that $\{\hat{\boldsymbol{z}}_{i,j}\}$ $(i = 1, \ldots, M)$– the estimates of $\boldsymbol{z}_j$ for all i coincide with each other. This idea can be realized by minimizing the following cost function Φ_{aln} with respect to $\{\boldsymbol{T}_i\}$:

$$\Phi_{aln} = \frac{1}{2}\sum_{j=1}^{N}\sum_{i \neq i'} \tilde{q}_{i,j}\tilde{q}_{i',j}\|\hat{\boldsymbol{z}}_{i,j} - \hat{\boldsymbol{z}}_{i',j}\|^2 \tag{6}$$

Although we omit the detailed derivation here, by introducing some auxiliary matrices and vectors such as $\boldsymbol{v}_j = [\tilde{q}_{1,j}\tilde{\boldsymbol{y}}_{1,j}^\top, \ldots, \tilde{q}_{M,j}\tilde{\boldsymbol{y}}_{M,j}^\top]$, $\boldsymbol{V} = [\boldsymbol{v}_1^\top, \ldots, \boldsymbol{v}_N^\top]^\top$, $\boldsymbol{D}_i = \sum_{j=1}^{N} \tilde{q}_{i,j}\tilde{\boldsymbol{y}}_{i,j}\tilde{\boldsymbol{y}}_{i,j}^\top$, $\boldsymbol{D} = diag(\boldsymbol{D}_1, \ldots, \boldsymbol{D}_M)$, $\boldsymbol{T} = [\boldsymbol{T}_1, \ldots, \boldsymbol{T}_M]^\top$, Eq.(6) can be rewritten as:

$$\Phi_{aln}(\boldsymbol{T}) = Tr(\boldsymbol{T}^\top(\boldsymbol{D} - \boldsymbol{V}^\top\boldsymbol{V})\boldsymbol{T}) \tag{7}$$

Note that this is a trace of a matrix quadratic form of $\boldsymbol{T}$, and that $\boldsymbol{Z} = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N]^\top$ can be obtained as $\boldsymbol{Z} = \boldsymbol{V}\boldsymbol{T}$.

As the minimization of Φ_{aln} has a trivial solution $\boldsymbol{T} = \boldsymbol{0}$ if there are no constraints, we impose a constraint :

$$\boldsymbol{Z}^\top\boldsymbol{Z} = \boldsymbol{T}^\top(\boldsymbol{V}^\top\boldsymbol{V})\boldsymbol{T} = \boldsymbol{I} \tag{8}$$

The solution of this constrained minimization is obtained as $\boldsymbol{T}_{opt} = [\boldsymbol{u}_2, \ldots, \boldsymbol{u}_{m+1}]$ where $\boldsymbol{u}_2$, $\boldsymbol{u}_{m+1}$ are the second smallest and $(m+1)$-smallest eigenvectors of the generalized eigenvalue problem:

$$(\boldsymbol{D} - \boldsymbol{V}^\top\boldsymbol{V})\boldsymbol{u} = \lambda(\boldsymbol{V}^\top\boldsymbol{V})\boldsymbol{u} \tag{9}$$

Then we obtain $\hat{\boldsymbol{Z}} = \boldsymbol{V}\boldsymbol{T}_{opt}$.

Next we consider reconstructing the row latent vectors $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M]^\top$. As each alignment transformation matrix $\boldsymbol{T}_i$ obtained in the previous step is supposed to characterize the corresponding row latent vector, estimates of $\{\boldsymbol{x}_i\}$ are obtained by reducing the dimension of $vec(\boldsymbol{T}_i)$ to n, where $vec(\boldsymbol{T}_i)$ is a column vector obtained by reshaping the elements of matrix $\boldsymbol{T}_i$. Note that $\{vec(\boldsymbol{T}_i)\}$ contain no missing elements, unlike the original observation matrix $\boldsymbol{Y}$. We employed the simple SVD for the dimensionality reduction this time, while other advanced non-linear methods are also applicable.

The above cost function and the solution of column latent vectors $\{\boldsymbol{z}_j\}$ originate from Verbeek and Roweis's method for non-linear PCA and CCA[9]. However, they did not deal with the missing elements nor simultaneous dimensionality reduction of column and row vectors. Therefore, our method is different from theirs.

4.3 Regularization

In actual applications, we can often improve the estimation results by introducing task-specific regularization terms into the original cost function. Especially, when the row latent vector $\boldsymbol{x}_i$ corresponds to the observer's state at time i, each pair of $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i+1}$ and corresponding pair of alignment matrices $\boldsymbol{T}_i$ and $\boldsymbol{T}_{i+1}$ are expected to be close to each other. This soft constraint can be realized by introducing a regularization term for smoothing successive rows of $\boldsymbol{X}$ expressed as,

$$\Phi_{smo} = Tr(\boldsymbol{T}^\top(\boldsymbol{S}^\top\boldsymbol{S})\boldsymbol{T}) \tag{10}$$

where $\boldsymbol{S}$ is a matrix that computes the differences of pairs of successive elements in $\boldsymbol{T}$. We minimize the weighted sum of cost functions $\Phi_{aln} + \alpha_{smo} \cdot \Phi_{smo}$ instead of Φ_{aln} under the same constraint.

4.4 Semi-supervised Co-embedding

In some application domains, a semi-supervised problem setting where the partial label information about row and column latent vectors are available beforehand is more natural. For example, in the case of wireless device and access points story, it is no wonder that exact positions of observer are partially available by GPS. LLACoE can be extended to a semi-supervised version in a straightforward way.

We denote the labeled data of j-th column latent vector $\boldsymbol{z}_j$ as $\boldsymbol{z}_j^*$. We also define a Boolean variable δ_j to indicate whether the label information is available or not. That is to say,

$$\boldsymbol{z}_j^* = \ \boldsymbol{z}_j \text{ (if } \delta_j = 1), \qquad \boldsymbol{0} \text{ (if } \delta_j = 0) \tag{11}$$

Then we define the cost function for the label information as:

$$\Phi_{zlb} \equiv \sum_{j=1}^{N} \delta_j \|\hat{\boldsymbol{z}}_j - \boldsymbol{z}_j^*\|^2 \tag{12}$$

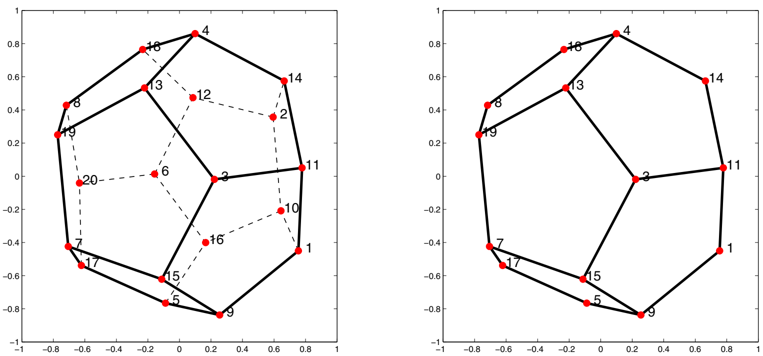
By defining $\boldsymbol{Z}^* = [\boldsymbol{z}_1^*, \ldots, \boldsymbol{z}_N^*]^\top$ and $\boldsymbol{J}_z = diag(\delta_1, \ldots, \delta_N)$, Eq.12 can be re-written as,

$$\Phi_{zlb} = Tr((\boldsymbol{VT} - \boldsymbol{Z}^*)^\top \boldsymbol{J}_z(\boldsymbol{VT} - \boldsymbol{Z}^*)) \tag{13}$$

The whole cost function $\Phi_{sem}(\boldsymbol{T}) = \Phi_{aln} + \alpha_{smo} \cdot \Phi_{smo} + \alpha_{zlb} \cdot \Phi_{zlb}$ can be easily minimized by solving a system of linear equations:

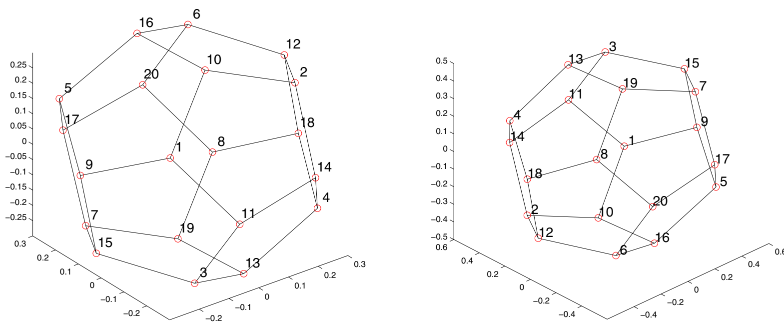
$$\boldsymbol{T}_{opt} = (\boldsymbol{D} + \boldsymbol{V}^\top(\alpha_{zlb}\boldsymbol{J}_z - \boldsymbol{I})\boldsymbol{V} + \alpha_{smo}\boldsymbol{S}^\top\boldsymbol{S})^{-1}(\alpha_{zlb}\boldsymbol{V}^\top\boldsymbol{J}_z\boldsymbol{Z}^*) \tag{14}$$

Introducing the label information of row latent vectors $\{\boldsymbol{x}_i^*\}$ is similar to the above discussion, but much simpler. It is a general semi-supervised regression problem, where $\{vec(\hat{\boldsymbol{T}}_i)\}$ are input vectors. While there are many advanced methods for the semi-supervised regression, this time we solved it simply by the ridge regression or least-squares linear regression with Tiknov regularization.



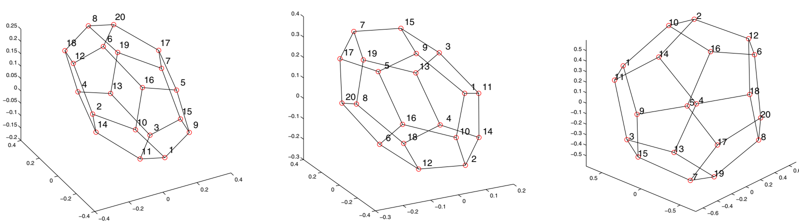
(a) Complete Observation (b) Incomplete Observation

Fig. 1. Examples of (a) complete and (b) incomplete observation in Exp.1. Numbers indicate vertices's' IDs



(a) SVD (b) (Unsupervised) LLACoE

Fig. 2. Reconstructed 3D model of dodecahedron with complete observation data



(a) ALS (18.9 sec) (b) Wiberg (16.4 sec) (c) LLACoE (0.17 sec)

Fig. 3. Reconstructed 3D model of dodecahedron with incomplete observation data with computational time. All algorithms are implemented in Matlab and conducted by a Dell Precision T1500.

5 Experiment

5.1 Experiment 1: Structure from Motion Task

First, we applied the proposed co-embedding method to the structure from motion (SFM) task in computer vision domain, and compared it with conventional methods.

Assume that we look at a dodecahedron from a randomly chosen direction, identify all visible vertices, then obtain their 2-dimensional coordinates on camera image as the observation data $[\boldsymbol{y}_{i,1}, \ldots, \boldsymbol{y}_{i,N}]$, where $N = 20$ because a dodecahedron has 20 vertices. We repeat this procedure for $M = 100$ times, and obtain the observation data $\boldsymbol{Y}$. The goal of this task is to reconstruct a 3-dimensional model of dodecahedron, or estimate 3-D coordinates $\{\boldsymbol{z}_j\}$ of 20 vertices in the body frame.

For comparison, we first conducted this experiment under the condition that all vertices are always visible, i.e., $\boldsymbol{Y}$ has no missing elements (Fig.1 (a)). In this case, ordinary SVD is applicable. In fact, a perfect 3-D model is reconstructed by SVD as Fig.2 (a). Unsupervised version of the proposed method (LLACoE) also succeeds in reconstructing it as Fig.2 (b).

Next we impose the practical condition that observation elements of occluded vertices are lost (Fig.1 (b)). As a result, approx. 30 % of $\boldsymbol{Y}$'s elements are missing. In this case, we cannot use the ordinary SVD anymore, because filling the missing elements with some constants is obviously inappropriate. So we applied two PCAMD methods, i.e., alternate least squares (ALS) algorithm and Wiberg's algorithm[1]. The resultant models are shown in Fig.3 (a)-(c). Although all three methods reconstructed the model successfully, LLACoE is much faster than others because it does not need iterations.

5.2 Experiment 2: Mapping and Localization for Wireless Devices

Next we applied LLCoE to a simultaneous localization and mapping (SLAM) problem with wireless devices in a simulated environment.

In this task, we assume that 564 access points (APs) are distributed in a virtual campus, and a walking observer with a wireless client device records the relative positions of detected APs periodically. Fig.4 illustrates the simulated environment (research campus) and the ground truth map of APs. Some APs' IDs are indicated for later evaluation. Fig.5 illustrates the ground truth trajectory of the observer and observation points. Number of observation points is 310. In this task, the row latent vector $\boldsymbol{x}_i$ ($i = 1, \ldots, 310$) is the observer's state (i.e., position and heading direction), whereas the column latent vector $\boldsymbol{z}_j$ ($j = 1, \ldots, 564$) is each AP's position. Observation data $\boldsymbol{y}_i, j$ is computed from a very noisy bearing and range information. For example, Fig.6 (a) and (b) are a ground truth map and a observed relative positions of detected APs at one time. We generated the observation data with:

$$Pr(q_{i,j} = 1) = \frac{1}{1 + \exp(0.15 \cdot (d_{i,j} - 50))} \tag{15}$$

where $d_{i,j}$ is the distance between i-th observation point and j-th AP. As a result, the ratio of missing elements in $\boldsymbol{Y}$ becomes approx. 97 %. Fig.7 shows the distribution of missing (gray) and existing (white) elements in $\boldsymbol{Y}$.

Fig. 4. Simulated environment with 564 AP positions

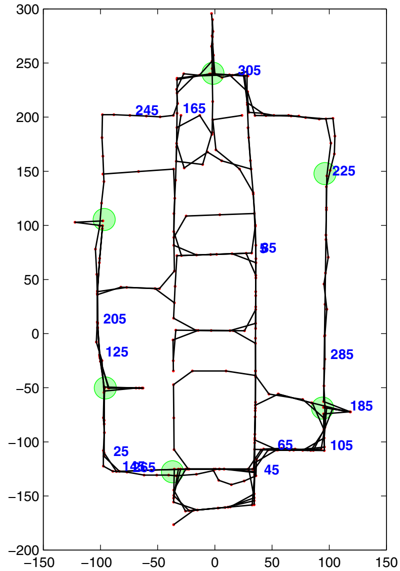
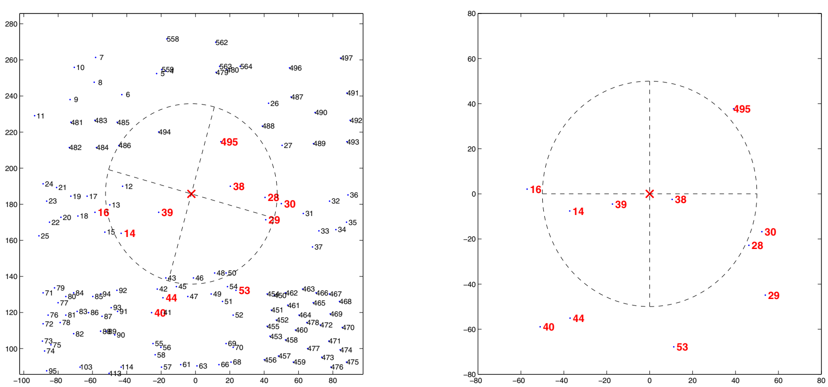


Fig. 5. Ground truth trajectory of observer



(a) Ground truth

(b) Observed

Fig. 6. Example of ground truth submap (a) and observed data (b). Observation is noisy and distant APs are missing. Circles show the approximate communicatable ranges.

Unsupervised Localization and Mapping. First we applied the unsupervised version of LLACoE to estimate $\boldsymbol{X}$ and $\boldsymbol{Z}$ from $\boldsymbol{Y}$ without the smoothing regularization. Although the map of APs ($\boldsymbol{Z}$) in Fig.8(a) is largely distorted, we can see the approximate relative relationships with neighbors are reconstructed to some extent. On the other hand, the trajectory of observer ($\boldsymbol{X}$) in Fig.9(a) is reconstructed very well.

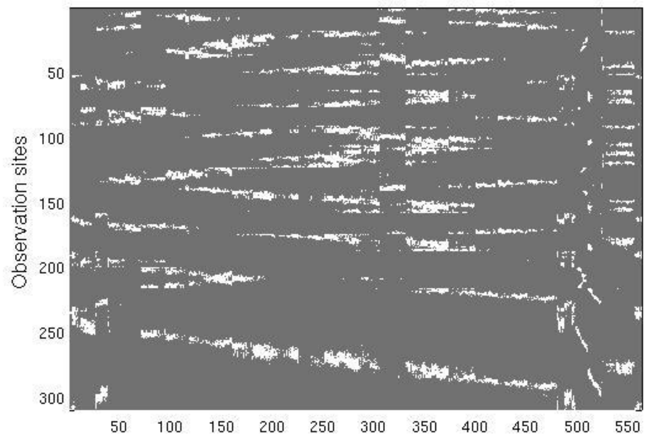
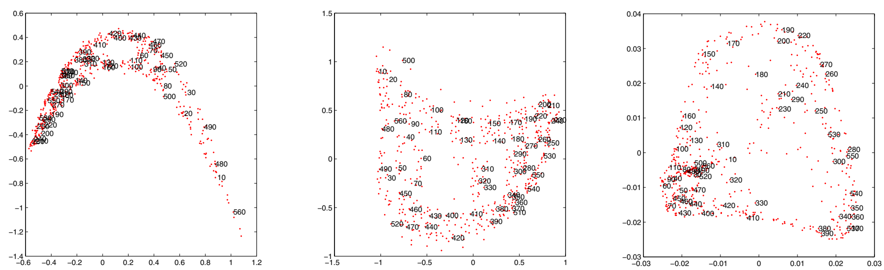
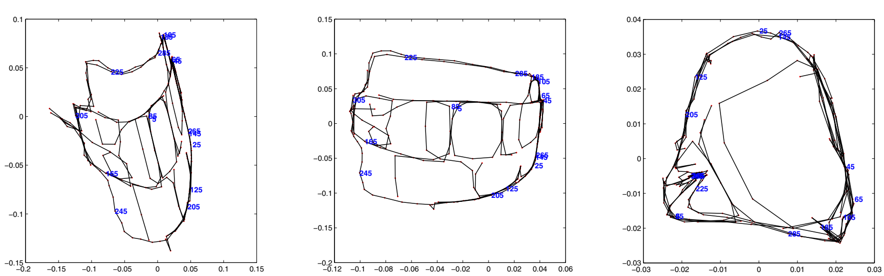


Fig. 7. Distribution of missing (gray) and existing (white) elements of observation data $\boldsymbol{Y}$. About 97% is missing.



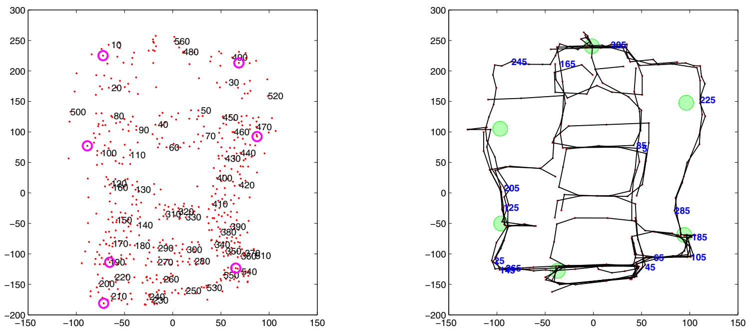
(a) LLACoE w/o smoothing (b) LLACoE with smoothing (c) LSI

Fig. 8. Reconstructed maps by unsupervised methods



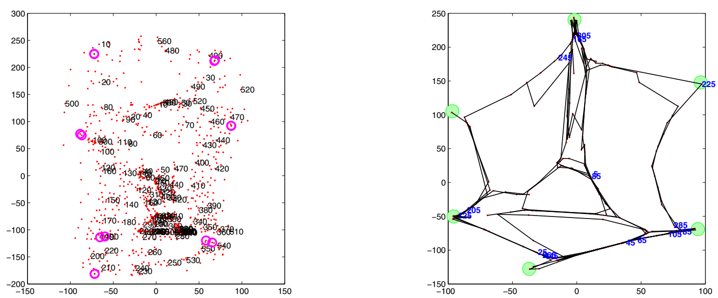
(a) LLACoE w/o smoothing (b) LLACoE with smoothing (c) LSI

Fig. 9. Reconstructed trajectories by unsupervised methods



(a) Map (b) Trajectory

Fig. 10. Estimated map and trajectory by semi-supervised LLACoE



(a) Map (b) Trajectory

Fig. 11. Estimated map and trajectory by co-localization[2]

Then we added the smoothing regularization term Φ_{smo} described in section 4.3. We set the weight parameter value as $\alpha_{smo} = 0.2$ here. The results are shown in Fig.8(b) and Fig.9(b). We can see that the reconstruction of $\boldsymbol{X}$ (map of APs) is much improved.

For comparison, we applied latent semantic indexing (LSI) method as in [2] to the data. To do so, we converted the range measurements into signal strengths by a monotonically decreasing function. The results are much worse than those of LLACoE as shown in Fig.8(c) and Fig.9(c).

Semi-supervised Localization and Mapping. We also tested the semi-supervised version of LLACoE in this experiment. We gave exact positions of 7 APs as the label information $\boldsymbol{z}_j^*$, which are emphasized by circles in Fig.4. Partial label information of observation points $\boldsymbol{x}_i$ were also provided within the "areas" indicated in Fig.5.

Fig.10 (a) and (b) show the obtained map and trajectory, respectively. Owing to the label information, the absolute accuracy of estimated positions is much improved.

For comparison, we applied Pan's co-localization algorithm based on graph regularization [2] to the range measurements. The resultant map and trajectory are shown in Fig.11 (a) and (b). Unfortunately, it completely failed in this experiment.

6 Conclusion

In this paper, we proposed a co-embedding method to embed the row and column vectors of an observation matrix data whose large portion is structurally missing into low-dimensional latent spaces simultaneously. The proposed method outperforms the conventional methods based on EM algorithm and ALS in computational cost and stability, because it is solved by eigendecomposition of a symmetric matrix. We also a semi-supervised version of the proposed co-embedding method, which is solved by a system of linear equations. In the experiment, we evaluated the method on two kinds of tasks, and compared it with other methods. In future, we are going to apply this method to a variety of problems.

References

1. Okatani, T., Deguchi, K.: On the wiberg algorithm for matrix factorization in the presence of missing components. International Journal of Computer Vision 72(3), 329–337 (2007)
2. Pan, J., Yang, Q.: Co-localization from labeled and unlabeled data using graph laplacian. In: Proceedings of IJCAI 2007, pp. 2166–2171 (2007)
3. Roweis, S., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding. Science 290, 2323–2326 (2000)
4. Roweis, S.: Em algorithms for pca and spca. In: in Advances in Neural Information Processing Systems, pp. 626–632 (1998)
5. Shum, H.Y., Ikeuchi, K., Reddy, R.: Principal component analysis with missing data and its application to polyhedral object modeling. IEEE Trans. Pattern Anal. Mach. Intell. 17(9), 854–867 (1995)
6. Sindhwani, V., Bucak, S.S., Hu, J., Mojsilovic, A.: One-class matrix completion with low-density factorizations. In: Proceedings of the 2010 IEEE International Conference on Data Mining, pp. 1055–1060 (2010)
7. Tenenbaum, J.B., Silva, V.D., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science 290, 2319–2323 (2000)
8. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. MIT Press (2005)
9. Verbeek, J., Roweis, S.T., Vlassis, N.: Non-linear cca and pca by alignment of local models. In: Procs. of NIPS (2004)