# Manifold Matching using Shortest-Path Distance and Joint Neighborhood Selection

Cencheng Shen[a,b], Joshua T. Vogelstein[a,c], Carey E. Priebe[a,d,**]

[a]*Center for Imaging Science, Johns Hopkins University*
[b]*Department of Statistics, Temple University*
[c]*Department of Biomedical Engineering and Institute for Computational Medicine, Johns Hopkins University*
[d]*Department of Applied Mathematics and Statistics, Johns Hopkins University*

## ABSTRACT

Matching datasets of multiple modalities has become an important task in data analysis. Existing methods often rely on the embedding and transformation of each single modality without utilizing any correspondence information, which often results in sub-optimal matching performance. In this paper, we propose a nonlinear manifold matching algorithm using shortest-path distance and joint neighborhood selection. Specifically, a joint nearest-neighbor graph is built for all modalities. Then the shortest-path distance within each modality is calculated from the joint neighborhood graph, followed by embedding into and matching in a common low-dimensional Euclidean space. Compared to existing algorithms, our approach exhibits superior performance for matching disparate datasets of multiple modalities.

## 1. Introduction

The abundance of data in the modern age has made it crucial to effectively deal with large amounts of high-dimensional data. For the purpose of data analysis, it is imperative to apply dimension reduction to embed data into a low-dimensional space for subsequent analysis. Traditional linear embedding techniques have solid theoretical foundations and are widely used, e.g., principal component analysis (PCA) [1, 2] and multidimensional scaling (MDS) [3, 4, 5] for datasets of a single modality, and canonical correlation analysis (CCA) [6, 7] for datasets of multiple modalities.

However, real datasets often exhibit nonlinear geometry, discovering which can be advantageous for subsequent inference. Many manifold learning algorithms have been proposed to learn the intrinsic low-dimensional structure of nonlinear datasets, including Isomap [8, 9], locally linear embedding (LLE) [10, 11], Hessian LLE [12], Laplacian eigenmaps [13, 14], local tangent space alignment (LTSA) [15, 16], among many others. Most of them start with the assumption that the data are locally linear, explore the local geometry via the nearest-neighbor graph

of the sample data, transform the data using the neighborhood graph, and eventually learn the low-dimensional manifold by optimizing some objective function. These nonlinear embedding algorithms usually serve as a preliminary feature extraction step that enables subsequent inference. They have been used successfully in object recognition and image processing.

In this paper, we consider the manifold matching task for datasets of multiple modalities, which is traditionally modeled by multiple dependent random variables. Conventional methods for identifying the relationship among multiple random variables are still very popular in theory and practice, such as canonical correlation [6, 17, 18] and Procrustes transformation [19, 20, 21, 22]. However, it has become a much more challenging task to match real datasets of multiple modalities from disparate sources due to their complex dependency structures, such as the same document in different languages, an image and its descriptions, or networks of the same actors on different social websites.

There have been many recent endeavors regarding data fusion and manifold matching [23, 24, 25, 26, 27, 28, 29]. Similar to dimension reduction for datasets of a single modality, manifold matching can serve as a feature extraction step to explore datasets of multiple modalities, and has also been shown to help subsequent inference in object recognition [30], information retrieval [31], and transfer learning [32]. Furthermore, the match-

---
**Corresponding author

*e-mail:* cshen6@jhu.edu (Cencheng Shen), jovo@jhu.edu (Joshua T. Vogelstein), cep@jhu.edu (Carey E. Priebe)

ing task is important on its own and has been applied to explore multiple graphs and networks [33, 34, 35]. One such application is seeded graph matching, where two large networks are collected but only a percentage of training vertices have known correspondence. Then the remaining vertices need to be properly matched to uncover potential correspondence and benefit later inference.

Due to the success of nonlinear embedding algorithms for datasets of a single modality, it is often perceived that these algorithms can be directly combined into the matching framework to improve the matching performance when one or more modalities are nonlinear. A naïve procedure is to pick one nonlinear algorithm, apply it to each modality separately, and match the embedded modalities. But such a simplistic procedure does not always guarantee a good matching performance, since many nonlinear embedding algorithms only preserve the local geometry up to some affine transformation [36]. Furthermore, using nonlinear transformations separately can even deteriorate the matching performance when compared to using simple linear transformations, as shown in our numerical simulations.

To tackle the problem, we propose a manifold matching algorithm using shortest-path distance and joint neighborhood selection. By utilizing a robust distance measure that approximates the geodesic distance, and effectively combining the correspondence information into the embedding step, the proposed algorithm can significantly improve the matching quality from disparate data sources, compared to directly take linear or nonlinear embeddings for matching. All code and data are made publicly available [1].

## 2. Manifold Matching

In this section, the matching framework and evaluation criteria are first introduced. Next we present the main algorithm, followed by relevant implementation details. Additional discussions are offered on issues that can affect the matching performance.

### 2.1. The Matching Framework

Suppose $n$ objects are measured under two different sources. Then $X_l = \{x_{il}\} \in \Xi_l$ for $l = 1, 2$ are the actual datasets that are observed / collected, with $x_{i1} \sim x_{i2}$ for each $i$ ($\sim$ means the two observations are matched in the context). Thus $X_1$ and $X_2$ are the two different views / modalities of the same underlying data. This setting is extendable to datasets of more than two modalities, but for ease of presentation we focus mainly on the matching task of two modalities.

$\Xi_1$ and $\Xi_2$ are potentially very different from each other, such as a flat manifold and its nonlinear transformation, an image and its description, or texts under different languages. A typical example is the social network, where many users have accounts on Youtube, Facebook, Twitter, etc. People sometimes post different contents and connect with different groups on each network site, such that data analysis of better quality is only possible when multiple accounts of the same person are combined. Some users already linked their accounts from different places, or unique user information are filled (like actual name, occupation), certain accounts can be automatically matched, providing a set of matched training data; but all the other accounts need to be matched by machine (as manual match is too expensive for millions of accounts), presenting a set of testing data from each website.

We assume $x_{il} \in \Xi_l$ is endowed with a distance measure $\Delta_l$ such that $\Delta_l(i, j) = dist(x_{il}, x_{jl})$. To match multiple modalities, we find two mappings $\rho_l : \Xi_l \rightarrow \mathbb{R}^d, l = 1, 2$ such that the mapped data $\hat{X}_l = \{\rho_l(x_{il})\}$ are matched into a common low-dimensional Euclidean space $\mathbb{R}^d$. A simple example of $\rho_l$ is MDS (e.g., classical MDS first doubly centers the distance matrices, followed by eigen-decomposition and keeping the top $d$ eigenvalues and eigenvectors to yield the embedding) followed by CCA (find two orthogonal $d \times d$ transformation matrices for each data set to maximize their correlation), which is a linear embedding and matching procedure.

Once the mappings are learned from the training data, the learned mappings $\rho_l$ can be applied to match any new observations $y_1 \in \Xi_1$ and $y_2 \in \Xi_2$ of unknown correspondence, i.e., compute $\hat{y}_l = \rho_l(y_l) \in \mathbb{R}^d$, and declare $y_1$ and $y_2$ as matched if and only if $\hat{y}_1$ is sufficiently close to $\hat{y}_2$ in the common space. Ideally, a good matching procedure should be able to correctly identify the correspondence of the new observations, i.e., if the testing observations are truly matched in the context, the mapped points should be very close to each other in $\mathbb{R}^d$. If the testing observations are not matched, the mapped points should be far away from each other.

To evaluate a given matching algorithm, a natural criterion is the matching ratio used in seeded graph matching [33]. Assume that there exist multiple testing observations in each space; and for each testing observation $y_1$ in $\Xi_1$, there is a unique testing observation $y_2 \in \Xi_2$ such that $y_1 \sim y_2$. Then they are correctly matched if and only if $\hat{y}_2$ is the nearest neighbor of $\hat{y}_1$ among all other testing data from $\Xi_2$, and vice versa. The matching ratio equals the percentage of correct matchings, and a higher matching ratio indicates a better matching algorithm.

The matching ratio based on nearest neighbor is often conservative, and can be a very small number when matching disparate real datasets. In practice, it is often more interesting to consider all neighbors within a small threshold, or rank multiple neighbors up to a limit. To that end, the testing power of the statistical hypothesis $H_0 : y_1 \sim y_2$ considered in [27] is another suitable criterion, which directly takes the Euclidean distance $\|\hat{y}_1 - \hat{y}_2\|$ as the test statistic. To estimate the testing power for given data, we first split all observations into matched training data pairs, matched testing data pairs, and unmatched testing data pairs. After learning $\rho_l$ from the matched training data and applying them to all testing data, the test statistic under the null hypothesis can be estimated from the matched testing pairs, and the test statistic under the alternative hypothesis can be estimated from the unmatched testing pairs. The testing power

---

[1] https://github.com/cshen6/MMSJ

at any type 1 error level is directly estimated from the empirical distributions of the test statistic, and a higher testing power indicates a better manifold matching algorithm.

We used both the testing power and the matching ratio for evaluation in the numerical experiments, and in most cases they yield the same interpretation regarding which method has a better matching performance. Note that if the critical value at a given type 1 error level is used as a distance threshold, the testing power equals the probability that the distance between the matched pair is no larger than the distance threshold. Since the matching ratio only considers the nearest neighbor of the matched pair, the testing power is never smaller than the matching ratio.

## 2.2. Main Algorithms

Our methodology is henceforth referred to as MMSJ. Algorithm 1 serves to learn the matching transformations from the matched training data, while algorithm 2 maps any testing observation onto the learned manifolds.

Given the distance matrices $\Delta_l$ for the training data $\{X_l, l = 1, 2\}$, we first construct an $n \times n$ binary graph $G$ by k-nearest-neighbor using the sum of normalized distance matrices $\sum_{l=1}^{2} \frac{\Delta_l}{\|\Delta_l\|_F}$, i.e., $G(i, j) = 1$ if and only if $\sum_l \frac{\Delta_l(x_{il}, x_{jl})}{\|\Delta_l\|_F}$ is among the smallest $k$ elements in the set $\{\sum_l \frac{\Delta_l(x_{il}, x_{ql})}{\|\Delta_l\|_F}, q = 1, \ldots, n\}$.

Next, for each modality $X_l$, we calculate the shortest-path distance matrix $\Delta_l^G$ based on the normalized $\Delta_l$ and the joint graph $G$, i.e., solve the shortest-path problem using the weighted graph $\frac{\Delta_l \circ G}{\|\Delta_l\|_F}$, where $\circ$ denotes the Hadamard product. Then we apply MDS to embed $\Delta_l^G$ into $\mathbb{R}^d$ for each $l$, followed by the Procrustes matching to yield the matched data $\hat{X}_l$, i.e., the Procrustes matching finds a $d \times d$ rotation matrix by

$$P = \arg \min_{P'P=I} \|P\tilde{X}_1 - \tilde{X}_2\|_F^2,$$

and sets $\hat{X}_1 = P\tilde{X}_1$ and $\hat{X}_2 = \tilde{X}_2$, where $\tilde{X}_l$ denotes the embedded data by MDS.

Then each testing observation $y_1 \in \Xi_1$ is mapped as follows: Given the distance between testing and training $\Delta_1(y_1, X_1)$ and the shortest-path distances for the training data $\Delta_1^G$, we first approximate the shortest-path distances $\Delta_1^G(y_1, X_1)$ by the respective nearest neighbors of the testing data within each modality. Then the testing data $y_1$ are embedded by MDS out-of-sampling (OOS) technique into $\mathbb{R}^d$ to yield $\tilde{y}_1$, followed by the Procrustes matching (i.e., $\hat{y}_1 = P\tilde{y}_1$ or $\hat{y}_2 = \tilde{y}_2$). Similarly for any $y_2 \in \Xi_2$. Note that MMSJ merely requires the training observations from different modalities to be corresponded and of same size, but the testing observations from different modalities can be of different and arbitrary size, because they are separately mapped within each modality and of unknown correspondence. In the numerical experiments we opt to set the testing data to be of the same size for convenience of presentation and evaluation.

To better visualize the process, we summarize the main algorithm in the flowchart of Figure 1.

## 2.3. Implementation Details

In this subsection, we elaborate on various implementation details of MMSJ.

---

**Algorithm 1** Manifold Matching using Shortest-Path Distance and Joint Neighborhood Selection (MMSJ)

---

**Require:** The distance matrices $\Delta_l$ for the matched datasets $\{X_l, l = 1, 2\}$, the neighborhood choice $k$, and the dimension choice $d$.
**Ensure:** The mapped datasets $\{\hat{X}_l \in \mathbb{R}^{d \times n}, l = 1, 2\}$, the shortest-path distance $\Delta_l^G$, and the learned Procrustes transformation $P$.
1: **function** MMSJ($\Delta_1, \Delta_2, k, d$)
2:     **for** $i, j := 1, \ldots, n$ **do** $G_{ij} \leftarrow \sum_l \frac{\Delta_l(x_{il}, x_{jl})}{\|\Delta_l\|_F}$ **end for**
3:     $G = \text{RANK}(G)$      ▷ rank distances within each row
4:     **for** $i, j := 1, \ldots, n$ **do** $G_{ij} \leftarrow I(G_{ij} \leq k)$ **end for**
5:     **for** $l := 1, 2$ **do**
6:         $\Delta_l^G = \text{SHORTESTPATH}(\frac{\Delta_l \circ G}{\|\Delta_l\|_F})$
7:         $\tilde{X}_l = \text{MDS}(\Delta_l^G, d)$      ▷ embedding into $\mathbb{R}^d$
8:     **end for**
9:     $[U, S, V] = \text{SVD}(\tilde{X}_2^T \tilde{X}_1)$
10:     $P \leftarrow UV^T$      ▷ Procrustes matching
11:     $\hat{X}_1 = P\tilde{X}_1$
12:     $\hat{X}_2 = \tilde{X}_2$
13: **end function**

---

**Algorithm 2** MMSJ on Testing Data

---

**Require:** The distance vectors $\Delta_l(y_l, X_l)$ (either $l = 1$ or $2$), the shortest-path distance matrices $\Delta_l$ and the mapped data $\hat{X}_l$, the learned Procrustes transformation $P$, and the neighborhood choice $k$.
**Ensure:** The mapped testing observation $\hat{y}_l$.
1: **function** MMSJ2($\Delta_l(y_l, X_l), \Delta_l^G, \hat{X}_l, P, k$)
2:     $G_l = \text{RANK}(\Delta_l(y_l, X_l))$
3:     $\Delta_l^G(y_l, X_l) = \text{SHORTESTPATH}([\Delta_l^G | \Delta_l(y_l, X_l) \circ G_l])$
4:     $\tilde{y}_l = \text{MDS-OOS}(\hat{X}_l, \Delta_l^G(y_l, X_l))$
5:     **if** $l = 1$ **then**
6:         $\hat{y}_1 = P\tilde{y}_1$
7:     **else**
8:         $\hat{y}_2 = \tilde{y}_2$
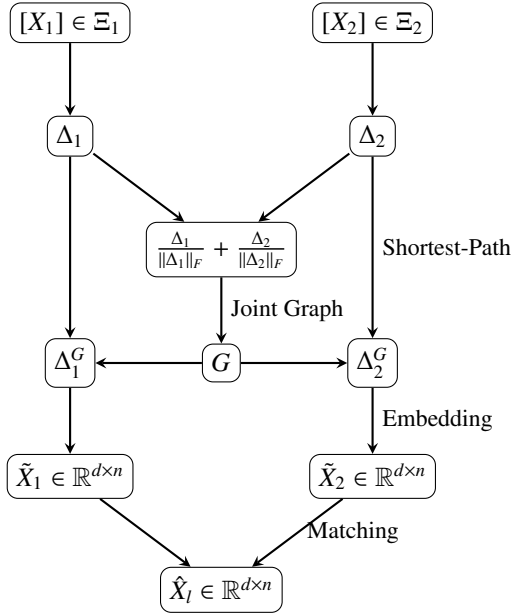9:     **end if**
10: **end function**

Fig. 1: Flowchart for Algorithm 1

MMSJ starts with two distance matrices rather than the sample observations directly, which means it is directly applicable to multiple modalities of different feature dimensions, as long as a distance metric can be defined for each modality. Although there is no limitation on applying the algorithm once the distances are given, the actual matching performance is clearly dependent on the choice of the metric. The most common choice is the Euclidean distance, or $L^p$ metrics in general. Other similarity or dissimilarity measures may be more appropriate in certain domains, such as the cosine distance for text data (see Section 3.2), or suitable kernels for structured data [37].

The joint neighborhood graph ensures consistent neighborhood selection when at least one of the modalities are nonlinear or noisy, and is intuitively better than two separate neighborhood graphs for matching. Alternatively, one may use a weighted sum of distances or a rank-based method to derive the joint neighborhood graph instead. Note that joint neighborhood requires the distance matrices of the training data to be properly scaled in advance, but is clearly not applicable to the testing data with unknown correspondence.

Using the joint neighborhood graph allows the resulting shortest-path distance to utilize the dependency structure of the training data. Computationally, the shortest-path distance matrix can be effectively implemented by Floyd's algorithm or Dijkstra's algorithm [8], which can be further sped up by choosing a small set of landmark points [9, 38]. Theoretically, the shortest-path distance can recover the geodesic distance of isometric manifolds with high probability under certain sampling conditions [39, 9]. When embedding the testing data, we essentially treat the training data as landmark points and only compute the shortest-path distances from the testing data to the training data.

Embedding the shortest-path distances into the Euclidean space followed by matching is a standard procedure. Alternatively, one may match the embeddings by CCA or joint

MDS, as discussed in [27, 40]. The advantages of MMSJ mostly lie in joint neighborhood and shortest-path distance; in fact, MMSJ always exhibits significant improvement, no matter which matching method to use. Thus we mainly consider the Procrustes matching for ease of presentation in the paper. For the testing data, they are embedded by out-of-sample MDS, which is a standard technique for MDS and kernel PCA [41, 38, 42] and more efficient than re-embedding all training and testing data. After all testing data are mapped onto the manifolds by the learned Procrustes matching, we may evaluate the matching performance as described in Section 2.1.

In terms of computation speed, suppose $n$ is the sample size of training pairs, and $n'$ is the size of all testing data. The running time complexity of MMSJ is $O(n^2 + nn')$, assuming the distance matrices are already given and the shortest-path step uses the fast landmark approximation. The only overhead is the distance matrix construction, which takes an additional $O(n^2 d + nn' d)$, where $d$ denotes the maximal feature dimension among all modalities. Therefore MMSJ is computationally efficient for high-dimensional data or data with large amount of testing observations.

To compare with MMSJ, we use the standard procedure that embed each modality separately by MDS / Isomap / LLE / LTSA, followed by Procrustes matching. Note that MDS / Isomap / LLE can all operate directly on a distance matrix, but some nonlinear algorithms like LTSA have to start with the Euclidean data rather than a distance measure. Thus, if only the distance matrices are available, the distance matrices are pre-embedded into a Euclidean space $\mathbb{R}^{d'}$ with $d' \geq d$ by MDS, followed by embedding into $\mathbb{R}^d$ via LTSA and matching by Procrustes.

### 2.4. Discussions

In this section, we offer further discussions on factors that can affect the matching performance.

In general, other than which matching method to apply, the matching performance is further dependent on how match-able the actual data sets are. If the observations from different modalities are strongly linearly correlated (a simple example is two almost identical databases with trivial manual errors), the matching performance will likely be perfect for any matching method; if the two modalities are related via either a very complex or non-existent transformation (such that they appear to be almost independent, e.g., weather and stock price on the same day), there is no hope to recover any meaningful matching regardless of what method to use. However, if there are certain local information that are shared by different modalities (e.g., the Swiss roll example in Section 3), that is the situation that MMSJ may improve over other methods by extracting useful local correspondence via the joint graph and shortest-path distance.

Assuming the given modalities can be reasonably combined for matching, the size and selection of the training observations are critical for all matching methods (or generally any supervised learning task): If the given training data are representative of the underlying manifold, then it is like to identify the correct matching for any number of testing observations; but

if the training data only reveal part of geometry, the matching performance will take a hit as testing observations increase. Of course, given the training data, whether the underlying manifold can be correctly recovered is also dependent on the embedding method. So alternatively, the advantage of MMSJ can be viewed as requiring much less training data to capture the manifolds for matching, e.g., Figure 3(A) can be interpreted as that MMSJ only takes around 1000 observations to uncover the intrinsic manifolds for perfect matching, while all other benchmarks require much larger sample size.

Another performance factor is the out-of-sampling technique. It has been popularly used as a fast approximation for distance-based embedding, and often called the Nystrom approximation [43] in numerical analysis. Clearly, different sampled landmark points will yield different embeddings of the out-of-sample observations, so the matching performance can be affected if the landmark points are not chosen appropriately. It turns out that the role of landmark points is similar to the role of the training data: if the in-sample landmark data approximately represent the whole manifold, then the out-of-sample embedded points are generally faithful to the original geometry; but if the in-sample data fail to reflect the manifold, then the out-of-sample embedded observations will have a larger discrepancy. Therefore, in order to minimize the effect of OOS in the matching task, the suggested strategy is to take all training data as the in-sample observations and embed the testing data via OOS, which is applied to MMSJ and all benchmarks in the experiments.

## 3. Numerical Experiments

In this section, we demonstrate the numerical advantages of MMSJ. Overall, we observed that our method is significantly better than all the benchmarks (MDS, Isomap, LLE, and LTSA) in matching ratio and testing power under various simulated settings and real experiments.

### 3.1. Swiss Roll Simulation

The Swiss roll data from [8] is a 3D dataset representing a nonlinear manifold, which is intrinsically generated by points on a 2D linear manifold. Figure 2 shows the 3D Swiss roll data with 5000 points in colors, along with its 2D embeddings by MDS, Isomap, and LLE. Clearly, MDS fails to recognize the nonlinear geometry while both Isomap and LLE succeed. However, the LLE embedding has a distorted geometry, while the Isomap embedding is more faithful to the underlying 2D linear manifold.

For the first simulation, we match the 3D Swiss roll with its underlying 2D linear manifold at varying training data size. A total of $n$ points from the 3D Swiss roll are randomly generated to construct the first modality $X_1$, and the corresponding points on the underlying 2D linear manifold are taken as the second modality $X_2$. Thus $X_1$ and $X_2$ are matched training data with distinct geometries. Once the training data are matched, we embed and apply the learned mappings to new testing observations $y_1$ and $y_2$ in each space.
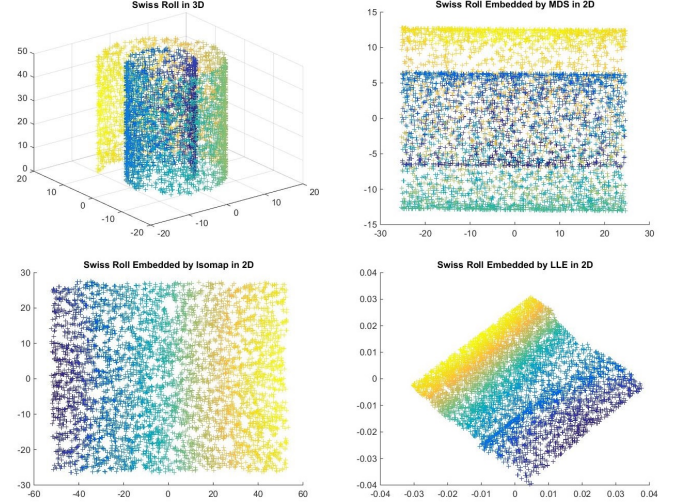


Fig. 2: The 3D Swiss roll dataset (top left), its 2D embedded data by MDS (top right), 2D embedding by Isomap at neighborhood size $k = 10$ (bottom left), and 2D embedding by LLE at $k = 10$ (bottom right).

We set the neighborhood size as $k = 10$, the dimension choice as $d = 2$, and generate $n' = 100$ testing pairs to compute the matching ratio. We repeat the above process for 100 Monte-Carlo replicates, and show the average matching ratio in Figure 3(A) with respect to increasing training data at $n = 50, 100, \ldots, 1000$. The MMSJ algorithm exhibits a significant advantage over all other algorithms: it enjoys a better matching ratio from small sample size onwards, and achieves almost perfect matching as sample size grows.

Next, we check the robustness of the manifold matching algorithms against noise, by adding white noise to the linear modality $X_2$ and fixing the training data size to $n = 1000$. The noise is independently and identically distributed as $Normal(0, \epsilon I_{2 \times 2})$, and the same testing procedure is applied to compute the matching ratio at increasing noise levels. The results are plotted in Figure 3(B) with respect to $\epsilon = 0, 1, 2, \ldots, 10$. The MMSJ algorithm is clearly better than all the benchmarks as the noise level increases.

For the third simulation, we consider an outlier scenario that randomly permutes a portion of the training data. For $\epsilon = 0, 0.01, \ldots, 0.1$, we randomly pick $\epsilon n$ training data from $X_2$ and permute their indices, such that those training data are no longer matched with the corresponding observations from $X_1$, i.e., there exists $\epsilon n$ outliers in the training data. Fixing $n = 1000$, we apply the same testing procedure and plot the matching ratio in Figure 3(C) with respect to the outlier percentage $\epsilon$. The MMSJ algorithm again has better matching ratio throughout increasing $\epsilon$; and all methods exhibit insignificant matching ratio when the outlier percentage $\epsilon$ reaches beyond 0.1, implying that the matching task may benefit significantly from excluding outliers prior to matching.

### 3.2. Wikipedia Articles Experiments

In this section, we apply the manifold matching algorithm to match disparate features of Wikipedia articles. The raw data contains 1382 pairs of articles from Wikipedia English and the
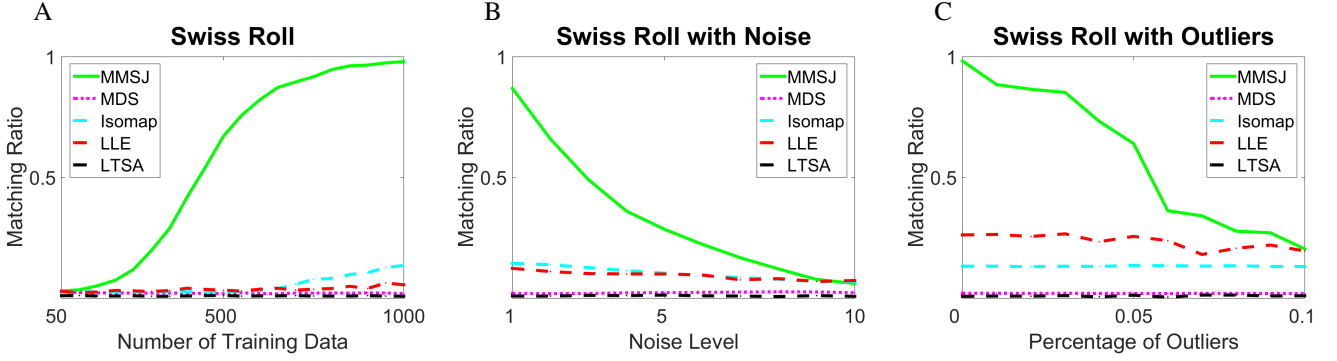
Fig. 3: Matching Ratio of 3D Swiss Roll versus its 2D Underlying Linear Manifold. (A) Matching Ratio with respect to Increasing Size of Training Data. (B) Matching Ratio with respect to Increasing Noise at $n = 1000$. (C) Matching Ratio with respect to Growing Number of Outliers at $n = 1000$.

corresponding French translations, within the 2-neighborhood of the English article "Algebraic Geometry". On Wikipedia, the same articles of different languages are almost never the exact translations of each other, because they are very likely written by different people and their contents may differ in many ways.

For the English articles and their French translations, a text feature and a graph feature are collected separately under each language. For the texts of each article, we use latent semantic indexing (LSI) (i.e., first construct a term-document matrix to describe the occurrences of terms in documents, then apply the low-rank approximation to the term matrix to 100 dimensions by singular value decomposition, see [44] for details) followed by cosine dissimilarity to construct two dissimilarity matrices $TE$ and $TF$ (representing the English texts and French texts). For the networks, two shortest-path distance matrices $GE$ and $GF$ (representing the English graph and French graph) are calculated based on the Internet hyperlinks of the articles under each language setting, with any path distance larger than 4 imputed to be 6 to avoid infinite distances and scaling issues.

Therefore, there exist four different modalities for pairs of Wikipedia articles on the same topic, making $TE$, $TF$, $GE$, and $GF$ matched in the context. Furthermore, as the text matrices are derived by cosine similarity while the graph matrices are based on the shortest-path distance with imputation, the former probably have nonlinear geometries while the latter are linear from the view of our matching algorithm.

For each Monte-Carlo replicate, we randomly pick $n = 500$ pairs of training observations, 100 pairs of testing matched observations, and 100 pairs of testing unmatched observations for evaluation. The parameters are set as $k = 20$, $d = 10$, $d' = 50$ (for LTSA only), and the manifold matching algorithms are applied for every possible combination of matching two modalities. We perform a total of 100 Monte-Carlo replicates. The mean matching ratio is reported in Table 1, and the estimated testing power is presented in Table 2 at type 1 error level 0.05.

Clearly, MMSJ achieves the best performance throughout all combinations. From the tables and figures, we further observe that without using shortest-path distance or joint neighborhood, separate nonlinear embeddings from LLE or LTSA are worse than the linear MDS embeddings in matching. Isomap does fairly well in the testing power, as it also uses shortest-path distance, but it can still be occasionally similar or slightly inferior

to MDS in the matching ratio. Our proposed MMSJ algorithm is consistently the best manifold matching algorithm in both the testing power and the matching ratio.

For the next experiment, we demonstrate that MMSJ algorithm is also robust against misspecification of parameters. The first two panels of Figure 4 plot the MMSJ and Isomap testing powers (the best two algorithms in our matching experiments) for matching $(TE, GE)$, against different choices of $d \in [2, 30]$ and $k \in [10, 30]$. Comparing the two panels not only shows that MMSJ is always better than Isomap in matching, but it also attains comparable testing power in a wide range of parameter choices. The same robustness holds for MMSJ under all other matching combinations.

Table 1: Wikipedia Documents Matching Ratio

| Modalities | MMSJ | MDS | Isomap | LLE | LTSA |
|---|---|---|---|---|---|
| $(TE, TF)$ | **0.2942** | 0.2546 | 0.2003 | 0.1265 | 0.0491 |
| $(TE, GE)$ | **0.1209** | 0.0675 | 0.0866 | 0.0143 | 0.0260 |
| $(TF, GF)$ | **0.0624** | 0.0419 | 0.0522 | 0.0134 | 0.0144 |
| $(GE, GF)$ | **0.1347** | 0.1280 | 0.1081 | 0.0157 | 0.0236 |
| $(TE, GF)$ | **0.0677** | 0.0429 | 0.0560 | 0.0132 | 0.0138 |
| $(TF, GE)$ | **0.0946** | 0.0545 | 0.0698 | 0.0132 | 0.0238 |

Table 2: Wikipedia Documents Testing Power at Type 1 Error Level 0.05

| Modalities | MMSJ | MDS | Isomap | LLE | LTSA |
|---|---|---|---|---|---|
| $(TE, TF)$ | **0.8124** | 0.4974 | 0.7476 | 0.3594 | 0.1930 |
| $(TE, GE)$ | **0.5184** | 0.2563 | 0.4255 | 0.0948 | 0.1116 |
| $(TF, GF)$ | **0.2782** | 0.1128 | 0.1877 | 0.0903 | 0.1028 |
| $(GE, GF)$ | **0.3108** | 0.2141 | 0.2485 | 0.0961 | 0.1063 |
| $(TE, GF)$ | **0.3199** | 0.1130 | 0.2141 | 0.0923 | 0.1021 |
| $(TF, GE)$ | **0.4464** | 0.2114 | 0.3595 | 0.0943 | 0.1064 |

### 3.3. Brain Structural Networks

In this section, we assess the matching performance via brain structural networks. There are a total of $n = 109$ subjects, each with diffusion weighted magnetic resonance imaging (MRI) data. For the raw brain imaging data, we derived two different modalities. For each scan, (i) process diffusion and structural MRI data via MIGRAIN, a pipeline for estimating brain
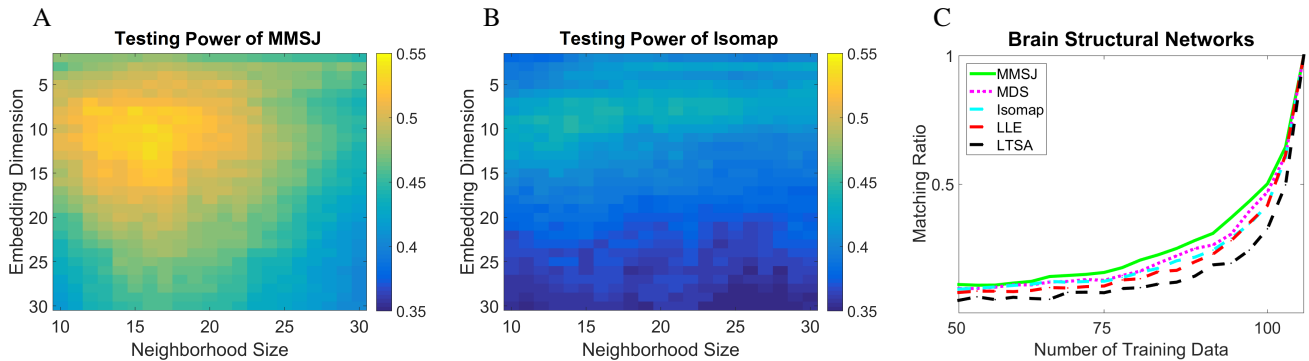
Fig. 4: (A) The Testing Power of MMSJ for Matching Wikipedia English Text and English Graph with respect to Different Dimension Choices and Neighborhood Sizes at Type 1 Error Level 0.05. (B) Same as (A) but for The Testing Power of Isomap. (C) Matching Ratio of Brain Structural Networks with respect to Increasing Size of Training Data.

networks from diffusion data [45], (ii) compute the distance between brain networks using the semi-parametric graph test statistic [46, 47], then embed each graph into two dimensions and align the embeddings via a Procrustes analysis. The Euclidean distance is used on both modalities.

Therefore the first modality seems like a more faithful representation of the brain imaging, while the second modality is inherently a graph representation of the brain structure. Although these two modalities are merely different transformations of the same raw data, they are distinct in many aspects such that there is no guarantee that one can recover their underlying geometry or succeed the matching task via machine learning algorithms.

For each Monte-Carlo replicate, we randomly pick $n$ pairs of matched observations for training, with all remaining sample observations for testing. The parameters are set as $k = 7$, $d = 2$, $d' = 2$, with a total of 100 Monte-Carlo replicates. The mean matching ratio is shown in Figure 4(C) with respect to increasing size of training data. It is clear that MMSJ is the best matching method among all algorithms, and all matching ratios improve significantly as the training data size increases relative to the testing data size.

## 4. Concluding Remarks

In summary, we propose a nonlinear manifold matching algorithm using shortest-path distance and joint neighborhood selection. The algorithm is straightforward to implement, efficient to run, and achieves superior matching performance. It is able to significantly improve the testing power and matching ratio for multiple modalities of distinct geometries, and is robust against noise, outliers, and model misspecification. Our experiments indicate that the shortest-path distance and joint neighborhood selection are two key catalysts behind the improvement of the matching performance.

There are a number of potential extensions of this work. First, pursuing theoretical aspects of the manifold matching task is a very challenging but rewarding task: so far there is a very limited number of literatures even for manifold learning of a single modality, and no nonlinear transformation can always recover the linear manifold under a wide range of geometries. On the other hand, the task of matching multiple

modalities is unique on its own. As a first step towards better theoretical understanding, we successfully proved in [48] that testing dependence via local correlations (which makes use of joint neighborhood graph and local distance in a similar manner) can successfully detect almost all relationships as sample size grows large, which shall shed more lights into the consistency of the matching task and may further advance the MMSJ algorithm.

Second, MMSJ requires a pair of metrics (or distances) for each modality. In this work we assume such metrics are predefined by domain knowledge, or use the Euclidean distance otherwise. If an optimal metric can be reasonably selected for each modality, it is likely to further boost the performance of MMSJ. From another point of view, if we are given high-dimensional or structured data (say graphs or images) to match, the algorithm may further benefit from first carrying out an appropriate feature selection down to certain landmark features [49, 50, 51], then use MMSJ. This quest is a valuable future direction to work on.

## References

[1] I. T. Jolliffe, *Principal Component Analysis*. Springer, 2nd ed., 2002.

[2] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society, Series B*, vol. 61, pp. 611–622, 1999.

[3] W. Torgerson, *Multidimensional Scaling: I. Theory and method*. Psychometrika, 1952.

[4] I. Borg and P. Groenen, *Modern Multidimensional Scaling: theory and applications*. Springer-Verlag, 2005.

[5] T. Cox and M. Cox, *Multidimensional Scaling*. Chapman and Hall, 2001.

[6] H. Hotelling, "Relations between two sets of variates," *Biometrika*, vol. 28, pp. 321–377, 1936.

[7] F. R. Bach and M. I. Jordan, "A probabilistic interpretation of canonical correlation analysis," tech. rep., Department of Statistics, UC Berkeley, 2005.

[8] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimension reduction," *Science*, vol. 290, pp. 2319–2323, 2000.

[9] V. de Silva and J. B. Tenenbaum, "Global versus local methods in nonlinear dimensionality reduction," *Advances in Neural Informaiton Processing Systems*, vol. 15, pp. 721–728, 2003.

[10] L. K. Saul and S. T. Roweis, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, 2000.

[11] S. T. Roweis and L. K. Saul, "Think globally, fit locally: Unsupervised learning of low dimensional manifolds," *Journal of Machine Learning Research*, vol. 4, pp. 119–155, 2003.

[12] D. Donoho and C. Grimes, "Hessian eigenmaps: New locally linear embedding techniques for high-dimensional data," in *Proceedings of the National Academy of Arts and Sciences*, vol. 100, pp. 5591–5596, 2003.

[13] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[14] X. He, S. Yan, Y. Hu, P. Niyogi, and H. Zhang, "Face recognition using Laplacianfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 328–340, 2005.

[15] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimensionality reduction via tangent space alignment," *SIAM Journal on Scientific Computing*, vol. 26, no. 1, pp. 313–338, 2004.

[16] Z. Zhang, J. Wang, and H. Zha, "Adaptive manifold learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 2, pp. 253–265, 2012.

[17] J. R. Kettenring, "Canonical analysis of several sets of variables," *Biometrika*, vol. 58, no. 3, pp. 433–451, 1971.

[18] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural Computation*, vol. 16, no. 12, pp. 2639–2664, 2004.

[19] R. Sibson, "Studies in the robustness of multidimensional scaling: Procrustes statistics," *Journal of the Royal Statistical Society. Series B*, vol. 40, no. 2, pp. 234–238, 1978.

[20] R. Sibson, "Studies in the robustness of multidimensional scaling: Perturbation analysis of classical scaling," *Journal of the Royal Statistical Society. Series B*, vol. 41, no. 2, pp. 217–229, 1979.

[21] Y. Goldberg and Y. Ritov, "Local Procrustes for manifold embedding: a measure of embedding quality and embedding algorithms," *Machine learning*, vol. 77, no. 1, pp. 1–25, 2009.

[22] J. C. Gower and G. B. Dijksterhuis, *Procrustes Problems*. Oxford University Press, 2004.

[23] S. Lafon, Y. Keller, and R. Coifman, "Data fusion and multi-cue data matching by difusion maps," *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1784–1797, 2006.

[24] C. Wang and S. Mahadevan, "Manifold alignment using Procrustes analysis," in *Proceedings of the 25th International Conference on Machine Learning*, 2008.

[25] C. Wang, B. Liu, H. Vu, and S. Mahadevan, "Sparse manifold alignment," in *Technical Report, UMass Computer Science UM-2012-030*, 2012.

[26] A. Sharma, A. Kumar, H. D. III, and D. Jacobs, "Generalized multiview analysis: A discriminative latent space," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[27] C. E. Priebe, D. J. Marchette, Z. Ma, and S. Adali, "Manifold matching: Joint optimization of fidelity and commensurability," *Brazilian Journal of Probability and Statistics*, vol. 27, no. 3, pp. 377–400, 2013.

[28] M. Sun, C. E. Priebe, and M. Tang, "Generalized canonical correlation analysis for disparate data fusion," *Pattern Recognition Letters*, vol. 34, no. 2, pp. 194–200, 2013.

[29] C. Shen, M. Sun, M. Tang, and C. E. Priebe, "Generalized canonical correlation analysis for classification," *Journal of Multivariate Analysis*, vol. 130, pp. 310–322, 2014.

[30] T.-K. Kim, J. Kittler, and R. Cipolla, "Discriminative learning and recognition of image set classes using canonical correlations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1005–1018, 2007.

[31] M. Sun and C. E. Priebe, "Efficiency investigation of manifold matching for text document classification," *Pattern Recognition Letters*, vol. 34, no. 11, pp. 1263–1269, 2013.

[32] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[33] V. Lyzinski, D. Fishkind, and C. E. Priebe, "Seeded graph matching for correlated Erdos-Renyi graphs," *Journal of Machine Learning Research*, vol. 15, pp. 3513–3540, 2014.

[34] J. Vogelstein, J. Conroy, V. Lyzinski, L. Podrazik, S. Kratzer, E. Harley, D. Fishkind, R. Vogelstein, and C. Priebe, "Fast approximate quadratic programming for graph matching," *PLOS ONE*, vol. 10, no. 4, p. e0121002, 2015.

[35] V. Lyzinski, D. Fishkind, M. Fiori, J. T. Vogelstein, C. E. Priebe, and G. Sapiro, "Graph matching: Relax at your own risk," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 60–73, 2016.

[36] Y. Goldberg and Y. Ritov, "Manifold learning: the price of normalization," *Journal of Machine learning research*, vol. 9, pp. 1909–1939, 2008.

[37] T. Hofmann, B. Scholkopf, and A. Smola, "Kernel methods in machine learning," *The Annals of Statistics*, vol. 36, no. 3, pp. 1171–1220, 2008.

[38] Y. Bengio, J. F. Paiement, and P. Vincent, "Out-of-sample extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering," in *Advances in Neural Information Processing Systems*, pp. 177–184, MIT Press, 2003.

[39] M. Bernstein, V. de Silva, J. C. Langford, and J. B. Tenenbaum, "Graph approximations to geodesics on embedded manifolds," 2000.

[40] D. Fishkind, C. Shen, Y. Park, and C. E. Priebe, "On the incommensurability phenomenon," *Journal of Classification*, vol. 33, no. 2, pp. 185–209, 2016.

[41] B. Scholkopf, A. Smola, and K. Muller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, pp. 1299–1319, 1998.

[42] M. W. Trosset and C. E. Priebe, "The out-of-sample problem for classical multidimensional scaling," *Computational Statistics and Data Analysis*, vol. 52, no. 10, pp. 4635–4642, 2008.

[43] P. Drineas and M. Mahoney, "On the Nystrom method for approximating a Gram matrix for improved kernel-based learning," *Journal of Machine Learning Research*, vol. 6, p. 21532175, 2005.

[44] S. Deerwester, S. Dumais, T. Landauer, G. Furnas, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society of Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[45] W. Gray Roncal, Z. H. Koterba, D. Mhembere, D. M. Kleissas, J. T. Vogelstein, R. Burns, A. R. Bowles, D. K. Donavos, S. Ryman, R. E. Jung, L. Wu, V. D. Calhoun, and R. J. Vogelstein, "MIGRAINE: MRI graph reliability analysis and inference for connectomics," *Global Conference on Signal and Information Processing*, 2013.

[46] D. L. Sussman, M. Tang, D. E. Fishkind, and C. E. Priebe, "A consistent dot product embedding for stochastic blockmodel graphs," *Journal of the American Statistical Association*, vol. 107, no. 499, pp. 1119–1128, 2013.

[47] M. Tang, A. Athreya, D. L. Sussman, V. Lyzinski, Y. Park, and C. E. Priebe, "A semiparametric two-sample hypothesis testing problem for random graphs," *Journal of Computational & Graphical Statistics*, 2016.

[48] C. Shen, C. E. Priebe, M. Maggioni, and J. T. Vogelstein, "Discovering relationships across disparate data modalities," *https://arxiv.org/abs/1609.05148*, 2017.

[49] J. Jiang, S. Zheng, A. Toga, and Z. Tu, "Learning based coarse-to-fine image registration," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[50] D. Conte, P. Foggia, C. Sansone, and M. Vento, "Thirty years of graph matching in pattern recognition," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 18, no. 3, pp. 265–298, 2004.

[51] M. Fiori, P. Sprechmann, J. Vogelstein, P. Mus, and G. Sapiro, "Robust multimodal graph matching: Sparse coding meets graph matching," in *Advances in Neural Information Processing Systems*, pp. 127–135, 2013.