Pattern Recognition Letters
journal homepage: www.elsevier.com

Manifold Matching using Shortest-Path Distance and Joint Neighborhood Selection

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

[a]*Department of Statistics, Temple University*
[b]*Center for Imaging Science, Johns Hopkins 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

Exploring and matching data sets of multiple modalities has become an important task in data analysis. Most existing matching methods rely on embedding and transformation techniques of single data set without fully utilizing the matched-ness, which often yield sub-optimal performance for the inference task. In this paper we propose a new nonlinear manifold matching algorithm using shortest-path distance and joint neighborhood selection. Specifically, a joint graph is built based on the correspondence information between the multiple modalities. Then the shortest-path distance within each data set is calculated from the joint neighborhood graph, followed by embedding into and matching in a common low-dimensional Euclidean space. Compared to existing popular algorithms, our approach exhibits superior performance for matching disparate data sets.

© 2016 Elsevier Ltd. All rights reserved.

1. Introduction

In today's world, it is becoming increasingly important to deal effectively with large amounts of high-dimensional data. For the purpose of data analysis, it is imperative to consider dimension reduction and embed the 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) [16, 36] and multidimensional scaling (MDS) [37, 5, 6] for a single data set, and canonical correlation analysis (CCA) [15, 1] for multiple data sets.

However, real data may exhibit nonlinear geometry, and discovering the underlying non-linearity can be beneficial for subsequent inference. Many manifold learning algorithms have been proposed to learn the intrinsic low-dimensional structure of nonlinear data, including Isomap [35, 32], locally linear embedding (LLE) [25, 24], Hessian LLE [8], Laplacian eigenmaps [2, 14], local tangent space alignment (LTSA) [43, 42], among many others. 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 two or more data sets from disparate sources. Classical methods for identifying the relationship among multiple random variables are still very popular in theory and practice, such as canonical correlation [15, 17, 13] and Procrustes transformation [30, 31, 11, 12]. However, it has become a much more challenging task to match real data sets collected from distinct sources, 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 [19, 41, 40, 27, 23]. Similar to dimension reduction of a single data set, manifold matching can serve as a feature extraction step to explore multiple data sets, and has also been shown to help subsequent inference in object recognition, information retrieval, and transfer learning [18, 22, 33, 34, 29]. Furthermore, the matching task is important on its own and has been applied to explore multiple graphs and networks [21, 39, 20]. One such application is seeded graph matching, where two large networks are collected but only a percentage of training vertices have known correspondence. Thus, the remaining vertices need to be matched properly to uncover the potential correspondence.

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

Due to the success of nonlinear embedding algorithms for a single data set, it is often perceived that these algorithms can be combined into the matching framework to improve the matching performance when handling nonlinear data. A naive procedure is to pick one nonlinear algorithm, apply it to each data set separately, and match the embedded data sets. 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 [10]. Furthermore, we will show in the numerical experiments that a direct matching of separate nonlinear embeddings can even deteriorate the matching performance when compared to linear embeddings.

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, our proposed algorithm can significantly improve the matching quality from disparate data sources, compared to matching linear embeddings or separate nonlinear embeddings.

The paper is organized as follows: In Section 2.1 we describe the matching task and the evaluation criteria in mathematical notations. The proposed nonlinear manifold matching algorithm is presented in Section 2.2, with various implementation issues discussed in Section 2.3. In Section 3, we illustrate the advantages of our algorithm via numerical simulations and real data experiments, using the simulated Swiss roll simulation and the Wikipedia document data sets with text and graph features. Finally, we summarize the results in Section 4. All codes and data are available on our website [1].

2. Manifold Matching

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 data sets that are observed/collected, with $x_{i1} \sim x_{i2}$ for each i ($\sim$ means the two observations are matched in the context). Alternatively, X_1 and X_2 can be thought of as two different views of the same underlying data. This setting is extendable to more than two data sets, but for ease of presentation we focus mainly on the matching task of two data sets.

Ξ_1 and Ξ_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. We assume $x_{il} \in \Xi_l$ is endowed with a distance measure Δ_l such that $\Delta_l(i, j) = dist(x_{il}, x_{jl})$. To match data sets from different spaces, we find two mappings $\rho_l : \Xi_l \to \mathbb{R}^d, l = 1, 2$ 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 ρ_l can be MDS or PCA followed by CCA, which is a linear embedding and matching procedure.

Once the mappings are learned, for any new observations $y_1 \in \Xi_1$ and $y_2 \in \Xi_2$ of unknown correspondence, the learned mappings ρ_l can be applied to match the testing observations in the low-dimensional Euclidean space, i.e., $\hat{y}_l = \rho_l(y_l) \in \mathbb{R}^d$. 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 the common Euclidean space. 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 [21]. Suppose sufficient training observations are given to learn the mappings, and there exist some testing observations of unknown correspondence in each space. Assume that for each testing observation y_1 in Ξ_1, there is another 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 mapped testing data from Ξ_2. The matching ratio represents the percentage of correct matching of all testing observations, and thus a higher 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 data sets. 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 statistical testing power of the hypothesis $H_0 : y_1 \sim y_2$ considered in [23] is another suitable criterion, which 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 ρ_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 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 use both the testing power and the matching ratio for evaluation in the numerical experiments. 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 Algorithm

The main algorithm is shown in algorithm 1, henceforth referred to as MMSJ. Once the manifolds are learned from matched training data by MMSJ, new testing data of unknown correspondence are embedded onto the manifolds by algorithm 2.

To better visualize the algorithm, we also summarize it by the following flowchart.

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

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

Input: The matched data sets $\{X_l, l = 1, 2\}$ of sample size n with distance matrices Δ_l, a neighborhood parameter k, and a dimension choice d.

Output: The mapped data sets $\{\hat{X}_l \in \mathbb{R}^{d\times n}, l = 1, 2\}$, and the learned Procrustes transformation P.

Step 1: Construct an $n \times n$ graph adjacency 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 $\{\sum_l \frac{\Delta_l(x_{il},x_{ql})}{\|\Delta_l\|_F}, q = 1, \ldots, n\}$.

Step 2: For each data set X_l, calculate the corresponding shortest-path distance matrix Δ_l^G from the graph G. This can be implemented as follows: for each i, j, first initialize

$$\Delta_1^G(i, j) = \begin{cases} \Delta_1(i, j), & \text{if } G(i, j) = 1, \\ \infty, & \text{otherwise ;} \end{cases}$$

then iterate through $q = 1, \ldots, n$ and set

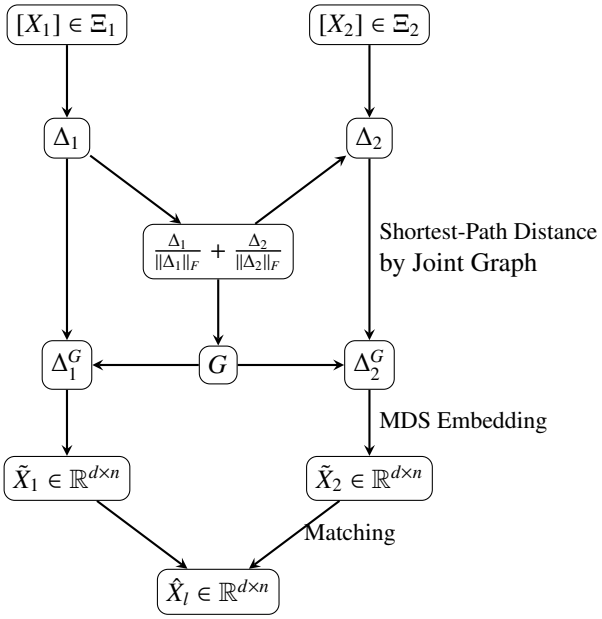
$$\Delta_1^G(i, j) = \min\{\Delta_1^G(i, j), \Delta_1^G(i, q) + \Delta_1^G(q, j)\};$$

the final Δ_1^G is the shortest-path distance matrix of X_1 by joint neighborhood selection. Repeat the above for Δ_2 yields Δ_2^G.

Step 3: Embed the normalized shortest-path distance matrix $\frac{\Delta_l^G}{\|\Delta_l^G\|_F}$ into $\mathbb{R}^d$ by MDS, then use the Procrustes matching to yield the matched data $\hat{X}_l$. Namely, 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.



Algorithm 2 is similar to algorithm 1, except joint neighborhood is not used for testing observations of unknown correspondence; and algorithm 2 is applicable to testing data of arbitrary size, but for simplicity we present it for one testing observation from each data source.

Algorithm 2 Embed Testing Data based on MMSJ

Input: Same as algorithm 1, plus two testing observations $y_1 \in \Xi_1$ and $y_2 \in \Xi_2$.

Output: $\hat{y}_l \in \mathbb{R}^d$ for $l = 1, 2$.

Step 1: Apply algorithm 1 to the training observations, which produces $\hat{X}_l \in \mathbb{R}^{d\times n}$ for $l = 1, 2$, and the learned Procrustes transformation P.

Step 2: For $l = 1, 2$ respectively, find the k-nearest-neighbor of y_l within X_l, and calculate the shortest-path distance vector from the testing observation to the training data.

Step 3: Normalize and embed y_l by MDS into $\mathbb{R}^d$ to yield $\tilde{y}_l$. Then set $\hat{y}_1 = P\tilde{y}_1$ and $\hat{y}_2 = P\tilde{y}_2$.

2.3. Discussions

In this subsection, we discuss some implementation details of MMSJ, and the benchmarks we compare with.

The joint neighborhood graph ensures consistent neighborhood selection in the case of noisy or nonlinear data sets, and is intuitively better than two separate neighborhood graphs for later matching when the correspondence are known. Alternatively, one may use a weighted sum of distances or rank-based method to derive the joint neighborhood graph instead. Note that it is necessary for the distance matrices to be properly scaled so that the joint neighborhood selection is meaningful.

The shortest-path distance can recover the geodesic distance of isometric manifolds with high probability under certain sampling conditions [4, 32]. When used together with joint neighborhood, the shortest-path distance makes use of the correspondence information. Computationally, the shortest-path distance matrix can be effectively implemented by Floyd's algorithm

or Dijkstra's algorithm [35], which can be further sped up by choosing a small set of landmark points [32, 3].

Embedding the shortest-path distance matrices followed by matching is a standard procedure. Alternatively, one may match the embeddings by CCA or joint MDS, as discussed in [23, 9]. 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.

Once the manifolds are learned by MMSJ from the training data, any testing data can be mapped to the learned manifolds, i.e., the shortest-path distance is computed for the testing data, followed by embedding into $\mathbb{R}^d$ by MDS and applying the learned Procrustes transformation. Note that joint neighborhood cannot be used for the testing data, as there is no known correspondence; moreover, rather than re-embed all training and testing data, the testing data can be quickly embedded by out-of-sample MDS, which is a standard technique for MDS and kernel PCA [26, 3, 38]. After the testing data are mapped onto the learned manifolds, we may test the matched-ness of any two testing observations from different data sources as in section 2.1.

To compare with MMSJ, we use the common procedure that embed each data 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, MDS is first used to embed the distance matrices into a Euclidean space $\mathbb{R}^{d'}$ with $d' \geq d$, followed by LTSA to embed into $\mathbb{R}^d$, then Procrustes matching.

3. Numerical Experiments

In this section we demonstrate the numerical advantages of the proposed manifold matching algorithm, with MDS, Isomap, LLE, and LTSA as the benchmarks. Overall, we observe that our algorithm is significantly better than all the benchmarks in matching ratio and testing power.

3.1. Swiss Roll Simulation

The Swiss roll data from [35] is a 3D data set representing a nonlinear manifold, but intrinsically generated by points on a 2D linear manifold. Figure 1 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 similar to the underlying 2D linear manifold.

For the first simulation, we match the 3D Swiss roll with its underlying 2D linear manifold. A total of $n = 1000$ points from the 3D Swiss roll are randomly generated to construct the first data set X_1, and the corresponding points on the underlying 2D linear manifold are taken as the second data set X_2. Thus X_1 and X_2 are matched training data with distinct geometries. Once

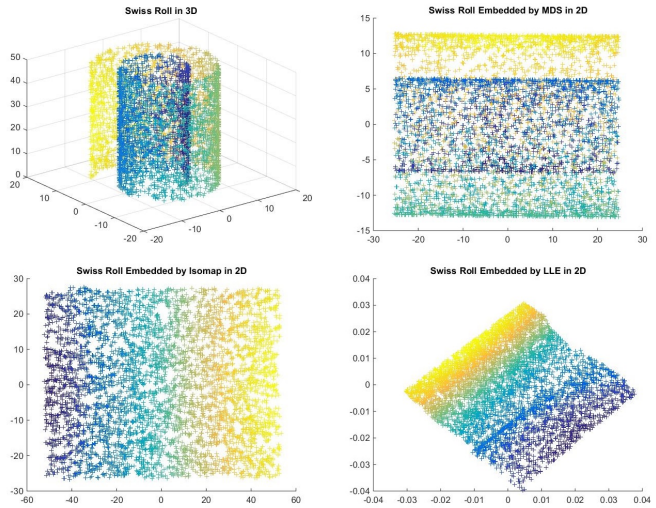


Fig. 1: The 3D Swiss roll data set (top left), its 2D embedded data by MDS (top right), 2D embedding by Isomap at neighborhood size $k = 10$ (bottom left), and 2D embedding by LLE at $k = 10$ (bottom right).

the training data are matched, we embed and apply the learned mappings to new testing observations y_1 and y_2 in each space.

We set the neighborhood size as $k = 10$ and the dimension choice as $d = 2$, and use 100 matched testing pairs and 100 unmatched testing pairs to calculate the test statistic under the null and the alternative hypotheses. Repeat the above for 100 Monte-Carlo replicates. The mean testing powers with respect to the increasing type 1 error level are plotted in Figure 2(a). The matching ratio is estimated based on the same 100 matched testing pairs from 100 MC replicates, which is shown in the first row of Table 1. The proposed MMSJ is clearly much better than all the benchmarks in both the testing power and the matching ratio. MDS and LTSA have the worst matching performance, while Isomap and LLE perform reasonably good.

Next, we check the robustness of the manifold matching algorithms against noise, by adding white noise to the linear data set X_2. The noise is independently and identically distributed as $Normal(0, \epsilon I_{2\times 2})$, and the same testing procedure is applied to estimate the testing powers at each noise level. The powers are plotted in Figure 2(b) with respect to the increasing noise level $\epsilon = 0, 1, 2, \ldots, 10$ at type 1 error level 0.05. The proposed MMSJ algorithm is still better than all the benchmarks as the noise level increases.

For the third simulation, we match two linear data sets: instead of the 3D Swiss roll, we use the 2D LLE embedding as X_1; thus X_1 and X_2 are both linear with some differences. Using the same procedure and parameters as the first simulation, the testing power is plotted in Figure 2(d) and the matching ratio is shown in the second row of Table 1. In this case MMSJ, MDS, and Isomap have similar performances with LLE and LTSA being slightly worse, which indicates that the shortest-path distance is more robust when matching noisy linear manifolds. Note that the matching ratio and testing power here are lower than those in the first simulation, because the LLE embedding has a distorted linear geometry.

Table 1: Swiss Roll Matching Ratio

Data Pair	MMSJ	MDS	Isomap	LLE	LTSA
(3D Swiss Roll, 2D Linear Manifold)	**0.9787**	0.0200	0.1324	0.2123	0.0697
(2D LLE Embedding, 2D Linear Manifold)	**0.1771**	0.1386	0.1412	0.1248	0.0828

3.2. Wikipedia Articles Experiments

In the real data experiments, 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 its 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) [7] 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 data representations for the same Wikipedia articles, 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 data sets. After 100 Monte-Carlo replicates, the mean matching ratio is reported in Table 2, the estimated testing power is presented in Table 3 at type 1 error level 0.05, and the power curves for some matching combinations are plotted in Figure 3.

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 similar or slightly inferior to MDS in the matching ratio occasionally. Our proposed MMSJ algorithm is consistently the best manifold matching algorithm in both the testing power and the matching ratio throughout.

We should point out that the matching performance depends on the parameters k and d, and the testing power or matching ratio of each algorithm can be slightly different by changing the parameters. For example, the MDS testing power is 0.49 for matching (TE, TF) at $d = 10$, which can be improved to 0.7 by varying d. MMSJ has the best power of 0.82, which can be increased to 0.9 by cross-validating the parameters as well. So far we have used fixed parameter choices to offer meaningful comparisons throughout all algorithms and matching combinations, but our MMSJ algorithm is in fact robust against misspecification of parameters. As an example, in Figure 4 we show the MMSJ and Isomap testing powers for matching (TE, GE) (the best two algorithms in our matching experiments) against different choices of d and k, for which d ranges from 2 to 30 and k ranges from 10 to 30. It is clear that MMSJ is always better than Isomap in matching and attains close-to-optimal testing power in a large range of parameter choices. Furthermore, the best MMSJ testing power is 0.55 while the best Isomap testing power is 0.45. The same robustness holds for MMSJ under all other matching combinations.

Table 2: Wikipedia Features Matching Ratio

Data Pair	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	0.236
(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 3: Wikipedia Features Testing Power at Type 1 Error Level 0.05

Data Pair	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

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, and achieves superior and robust performance. It is able to significantly improve the testing power and matching ratio when matching data of distinct geometries, and is robust against noise and model selection. Our experiments indicate that the shortest-path distance and joint neighborhood selection are two key catalysts behind the improvement of the matching performance.

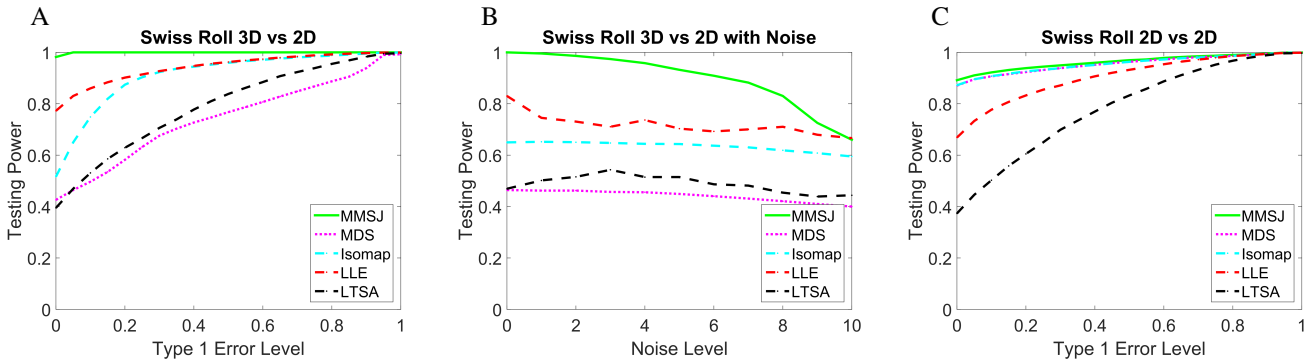


Fig. 2: Testing Powers of Swiss Roll Data Sets with respect to Increasing Type 1 Error Level. (A) Testing Power of 3D Swiss Roll versus its 2D Underlying Linear Manifold. (B) Testing Power of 3D Swiss Roll versus its 2D Underlying Linear Manifold with Increasing Noise. (C) Testing Power of 2D LLE Embedding of Swiss Roll versus its 2D Underlying Linear Manifold.

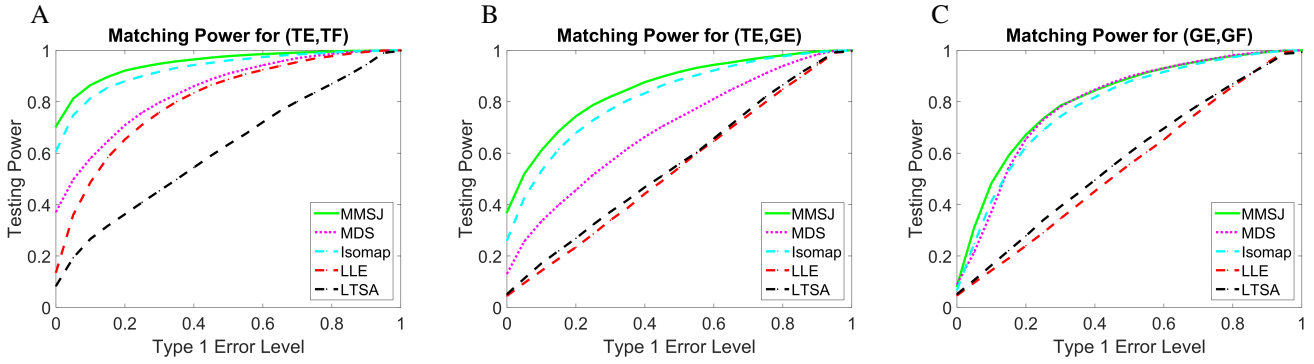


Fig. 3: Testing Powers of Wikipedia Data Sets with respect to Increasing Type 1 Error Level. (A) Testing Power of Wikipedia English Text versus French Text. (B) Testing Power of Wikipedia English Text versus English Graph. (C) Testing Power of Wikipedia English Graph versus French Graph.

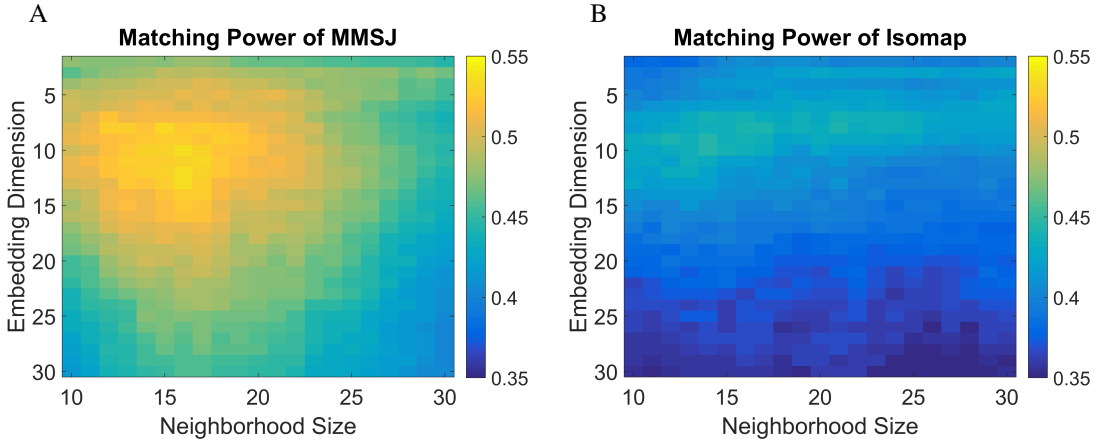


Fig. 4: Testing Power of Wikipedia English Text and English Graph with respect to Different Dimension Choices and Neighborhood Sizes at Type 1 Error Level 0.05. (A) The testing power of MMSJ. (B) The testing power of Isomap.

Furthermore, the improvement in manifold matching motivates our follow-up work [28] regarding dependency discovery by local scale, which makes use of the joint neighborhood and local distances in a similar manner.

Acknowledgment

This work is partially supported by National Security Science and Engineering Faculty Fellowship (NSSEFF), Johns Hopkins University Human Language Technology Center of Excellence (JHU HLT COE), and the XDATA program of the Defense Advanced Research Projects Agency (DARPA) administered through Air Force Research Laboratory contract FA8750-12-2-0303. This work is also supported by the Defense Advanced Research Projects Agency (DARPA) SIMPLEX program through SPAWAR contract N66001-15-C-4041 and DARPA GRAPHS N66001-14-1-4028.

References

[1] Bach, F.R., Jordan, M.I., 2005. A Probabilistic Interpretation of Canonical Correlation Analysis. Technical Report. Department of Statistics, UC Berkeley.

[2] Belkin, M., Niyogi, P., 2003. Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation 15, 1373–1396.

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

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

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

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

[7] Deerwester, S., Dumais, S., Landauer, T., Furnas, G., Harshman, R., 1990. Indexing by latent semantic analysis. Journal of the American Society of Information Science 41, 391–407.

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

[9] Fishkind, D., Shen, C., Park, Y., Priebe, C.E., 2016. On the incommensurability phenomenon. Journal of Classification accepted.

[10] Goldberg, Y., Ritov, Y., 2008. Manifold learning: the price of normalization. Journal of Machine learning research 9, 1909–1939.

[11] Goldberg, Y., Ritov, Y., 2009. Local Procrustes for manifold embedding: a measure of embedding quality and embedding algorithms. Machine learning 77, 1–25.

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

[13] Hardoon, D.R., Szedmak, S., Shawe-Taylor, J., 2004. Canonical correlation analysis: An overview with application to learning methods. Neural Computation 16, 2639–2664.

[14] He, X., Yan, S., Hu, Y., Niyogi, P., Zhang, H., 2005. Face recognition using Laplacianfaces. IEEE Transactions on Pattern Analysis and Machine Intelligence 27, 328–340.

[15] Hotelling, H., 1936. Relations between two sets of variates. Biometrika 28, 321–377.

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

[17] Kettenring, J.R., 1971. Canonical analysis of several sets of variables. Biometrika 58, 433–451.

[18] Kim, T.K., Kittler, J., Cipolla, R., 2007. Discriminative learning and recognition of image set classes using canonical correlations. IEEE Transactions on Pattern Analysis and Machine Intelligence 29, 1005–1018.

[19] Lafon, S., Keller, Y., Coifman, R., 2006. Data fusion and multi-cue data matching by difusion maps. IEEE transactions on Pattern Analysis and Machine Intelligence 28, 1784–1797.

[20] Lyzinski, V., Fishkind, D., Fiori, M., Vogelstein, J.T., Priebe, C.E., Sapiro, G., 2016. Graph matching: Relax at your own risk. IEEE Transactions on Pattern Analysis and Machine Intelligence 38, 60–73.

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

[22] Pan, S.J., Yang, Q., 2010. A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering 22, 1345–1359.

[23] Priebe, C.E., Marchette, D.J., Ma, Z., Adali, S., 2013. Manifold matching: Joint optimization of fidelity and commensurability. Brazilian Journal of Probability and Statistics 27, 377–400.

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

[25] Saul, L.K., Roweis, S.T., 2000. Nonlinear dimensionality reduction by locally linear embedding. Science 290, 2323–2326.

[26] Scholkopf, B., Smola, A., Muller, K., 1998. Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation 10, 1299–1319.

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

[28] Shen, C., Priebe, C.E., Maggioni, M., Vogelstein, J.T., 2016. Dependence discovery from multimodal data via multiscale graph correlation. Submitted .

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

[30] Sibson, R., 1978. Studies in the robustness of multidimensional scaling: Procrustes statistics. Journal of the Royal Statistical Society. Series B 40, 234–238.

[31] Sibson, R., 1979. Studies in the robustness of multidimensional scaling: Perturbation analysis of classical scaling. Journal of the Royal Statistical Society. Series B 41, 217–229.

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

[33] Sun, M., Priebe, C.E., 2013. Efficiency investigation of manifold matching for text document classification. Pattern Recognition Letters 34, 1263–1269.

[34] Sun, M., Priebe, C.E., Tang, M., 2013. Generalized canonical correlation analysis for disparate data fusion. Pattern Recognition Letters 34, 194–200.

[35] Tenenbaum, J.B., de Silva, V., Langford, J.C., 2000. A global geometric framework for nonlinear dimension reduction. Science 290, 2319–2323.

[36] Tipping, M.E., Bishop, C.M., 1999. Probabilistic principal component analysis. Journal of the Royal Statistical Society, Series B 61, 611–622.

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

[38] Trosset, M.W., Priebe, C.E., 2008. The out-of-sample problem for classical multidimensional scaling. Computational Statistics and Data Analysis 52, 4635–4642.

[39] Vogelstein, J., Conroy, J., Lyzinski, V., Podrazik, L., Kratzer, S., Harley, E., Fishkind, D., Vogelstein, R., Priebe, C., 2015. Fast approximate quadratic programming for graph matching. PLOS ONE 10, e0121002.

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

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

[42] Zhang, Z., Wang, J., Zha, H., 2012. Adaptive manifold learning. IEEE Transactions on Pattern Analysis and Machine Intelligence 34, 253–265.

[43] Zhang, Z., Zha, H., 2004. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. SIAM Journal on Scientific Computing 26, 313–338.