GRAPH CONSTRUCTION FOR MANIFOLD DISCOVERY

A Dissertation Outline Presented

by

CJ CAREY

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2015

School of Computer Science

GRAPH CONSTRUCTION FOR MANIFOLD DISCOVERY

A Dissertation Outline Presented

by

CJ CAREY

Approved as to style and content by:

Sridhar Mahadevan, Chair

Don Towsley, Member

Ben Marlin, Member

M. Darby Dyar, Member

Lori Clarke, Chair
School of Computer Science

ABSTRACT

GRAPH CONSTRUCTION FOR MANIFOLD DISCOVERY

MAY 2015

CJ CAREY

B.Sc., WASHINGTON UNIVERSITY IN ST. LOUIS

M.Sc., WASHINGTON UNIVERSITY IN ST. LOUIS

Directed by: Professor Sridhar Mahadevan

Manifold discovery is the essential first component of the manifold learning pipeline, and graph construction is the primary means of discovering manifold structure. Thus, it is important to develop accurate, efficient, and reliable algorithms for constructing graphs. To do so, I propose several new directions of study based on varied assumptions about the structure of both the data and downstream task. I will also develop a generic benchmark suite for evaluating the manifold approximation capability of graphs, which I will use as the basis for the experimental justification for the new methods I introduce.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

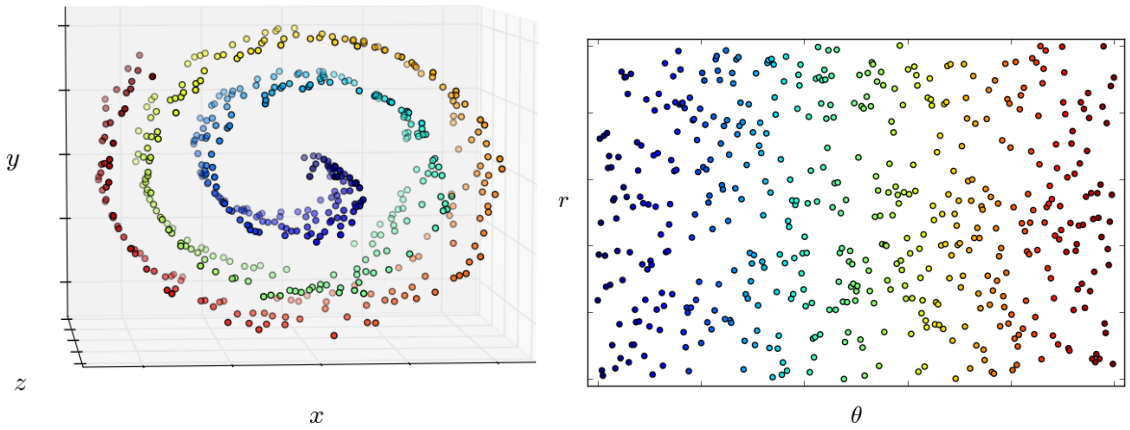
CHAPTER 1

INTRODUCTION

As machine learning practitioners tackle increasingly large and complex data sets, better data representations are necessary to improve task performance while reducing computational burden. Manifold learning techniques, which aim to uncover the underlying geometry of a data set, have proven to be especially useful in combating the *curse of dimensionality* [29]. For instance, the PageRank algorithm that powers the Google search engine operates on the manifold of linked web sites [32]. Manifold learning has also been very successful in the realms of data visualization and transfer learning.

These methods operate under the *manifold hypothesis*, which assumes that input data lie on a lower-dimensional manifold. In this context, a manifold is defined as a topological space that is smooth, compact, and locally Euclidean [36]. These properties allow any manifold to be viewed as a set of overlapping *charts*, each of which is a linear subspace. The dimensionalities of these charts are collectively referred to as the *intrinsic dimension* of the manifold, and typically this intrinsic dimension is much smaller than the dimensionality of the original data. A simple example of a manifold embedded in a higher dimensional space, is shown in Figure 1.1. By performing traditional machine learning methods in manifold-space rather than the original space, we can reduce computational cost and improve the generalizability of our models.

In many real-life applications, the underlying manifold structure of a given dataset is unknown. In these cases, we must first apply *manifold discovery* to learn the struc-



(a) Original coordinate space.

(b) Manifold coordinate space.

Figure 1.1: Illustration of the "swiss roll" manifold. The manifold is illustrated by sampling uniformly at random from the (r, θ) manifold space, then projecting those points to (x, y, z) coordinates as $(\theta \sin \theta, \theta \cos \theta, r)$. In both plots, points are colored by their θ value.

ture of one or more manifolds based on the data at hand. Manifolds are continuous objects, so in practice we work with a discrete representation in the form of a graph. In this representation, each chart is defined by a vertex and the vertices with which it shares edges. The manifold discovery problem can thus be reduced to the problem of *graph construction*: given a set of vertices, add edges such that the resulting graph best approxmates the true structure of the manifold. This task is made tractable by exploiting the smoothness, compactness, and local linearity properties of manifolds.

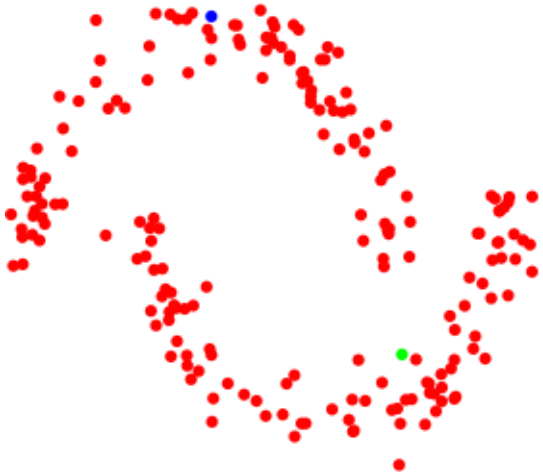
Historically, the label "manifold learning" has been applied to methods that operate on known manifolds as well as manifold discovery algorithms. For the purposes of this proposal, we define the *manifold learning pipeline* as follows:

1. Given a data set X, discover M, the underlying manifold structure.

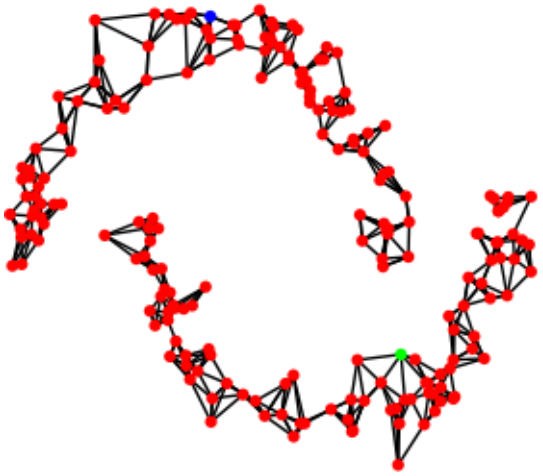
2. Using M and potentially some auxiliary information Z, perform the desired learning task.

3. Optionally, define a mapping for new data X' onto M to generalize the model to new instances.

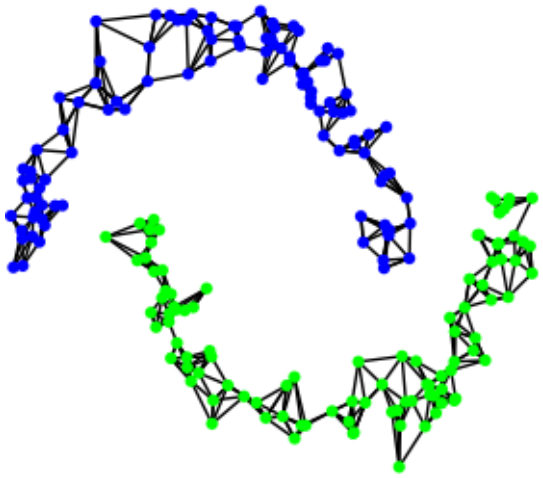
Figure 1.2 provides an illustration of this pipeline on a simple task. Note that the manifold discovery step is critical to the success of the pipeline, as errors in the estimated manifold structure can propagate to errors in the downstream task.



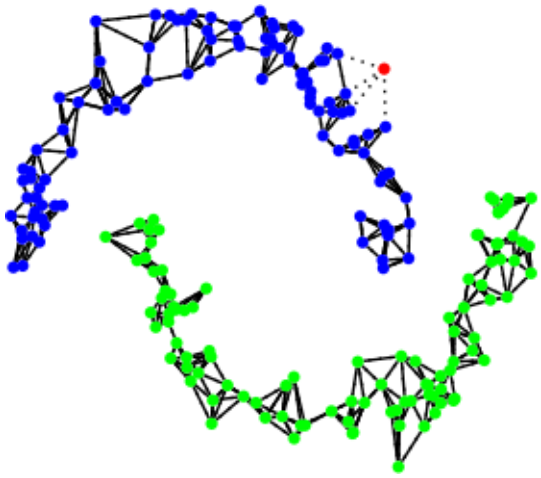
(a) The "two moons" dataset in original coordinates, with one labeled example per moon (in green and blue) and 198 unlabeled examples (in red). The task is to color each red point either green or blue, and thus separate the two moons. Using the terminology from the manifold learning pipeline,X is the set of coordinates for all points and Z is the label information.



(b) Step 1: Manifold discovery via graph construction. Each point is connected to its four nearest neighbors using a Euclidean distance metric, creating a graph with two connected components.



(c) Step 2: Manifold learning using the graph constructed in step 1 and the auxilliary labels, Z. Using an edge propagation algorithm, each labeled point gives its color to each of its neighboring points. This process is repeated until all points are labeled, correctly separating the two moons.



(d) Step 3: Generalizing to new data. When a new point is added to the dataset (in red), it takes its label from the labels of its four nearest neighbors from the existing dataset. This point will be colored blue, as all four of its neighbors (dotted lines) are blue.

Figure 1.2: The manifold learning pipeline, as applied to the task of labeling instances of the "two moons" dataset. Many options are available for each step of the pipeline; for clarity of illustration, we have selected simple methods that visualize well.

CHAPTER 2

RELATED WORK

In this chapter, we review existing manifold learning and discovery methods using the notation from the manifold learning pipeline defined in the previous chapter. In section 2.1, we discuss methods for machine learning on manifolds. In section 2.2, we discuss graph construction algorithms for several categories of manifold discovery problems.

2.1 Manifold Learning

Recall step two of the manifold learning pipeline:

> Using M and potentially some auxiliary information Z, perform the desired learning task.

This section clusters manifold learning tasks into five groups and reviews existing methods for performing them. In each case, we assume that M is given (typically in the form of a graph), along with various forms of additional information Z.

Embedding / Dimension Reduction

Dimension reduction is the task of finding a transformation of $X \in \mathbb{R}^{n \times d_o}$ from its original space to lower-dimensional coordinates $Y \in \mathbb{R}^{n \times d_e}$ such that $d_e \leq d_o$. Early work in this area drew on statistical measures of information content to find linear embeddings that preserve "useful" properties of the input data. These methods include Principal Components Analysis [23] and multidimensional scaling [26], both of which formed the basis for later methods that incorporate constraints to preserve manifold structure.

State of the art results in dimensionality reduction tasks popularized manifold learning in the early 2000s, paving the way for a dramatic increase in the study of manifold methods. These "manifold embedding" methods solve for an optimal Y while respecting the structure of M. Notably, the manifold structure constraints allow for nonlinear embedding transformations. This is typically formulated as an unsupervised task, so no auxiliary data are required (e.g., $Z = \emptyset$).

Algorithms for manifold embedding can be clustered into one of three families:

- **Geodesic family**: Geodesic distances along M are estimated, then Y is constructed such that $\text{pdist}(Y) \approx \text{geodesic}(M)$, where $\text{pdist}(\cdot)$ is a pairwise distance function. Methods in this family include Self-Organizing Maps [24], Curvilinear Component Analysis [12], and Isomap [42].

- **Laplacian family**: The graph Laplacian L of a discrete approximation of M is computed, then Y is given by the first d_e eigenvectors of L. Methods in this family include Laplacian Eigenmaps [3] and Diffusion Maps [10].

- **Locally Linear family**: Linear reconstruction weights W are found using M, then Y is found by minimizing $\|Y - WY\|_F$. Methods in this family include Locally Linear Embedding [35], Local Tangent Space Alignment [56], Hessian Locally Linear Embedding [13], Sparse Manifold Clustering and Embedding [14], and Low Rank Embedding [27].

Each of these methods operates on a discrete graph representation of M, though most attempt to be agnostic to the specific type of graph used. While some methods are more robust than others, researchers have found that the quality of the graph approximation is critical to each method's performance [2, 53].

The loss functions in these methods are typically solved in the form of a generalized eigendecomposition problem, which dominates the running-time of the algorithm. In

straightforward applications, this is a $O(n^3)$ operation, though extensions have been proposed to reduce this computational burden [39, 28].

Semisupervised learning

Broadly, semisupervised learning encompasses all learning tasks with partial label information. These tasks are often formulated as classification or clustering problems, using the set of labeled instances L_{known} to extrapolate labels for the unknown instances, $L_{unknown}$. We require that both L_{known} and $L_{unknown}$ are drawn from the same underlying distribution. The addition of the unlabeled instances allows learning algorithms to more accurately generalize and avoid overfitting.

In the manifold learning context, we make an additional assumption that the underlying distribution has some manifold structure M. In this setting, also known as "manifold regularization" [4], we define $Z = L_{known}$. The addition of manifold structure allows for "label propagation", in which labeled instances influence the labels of their unlabeled neighbors in manifold space [58, 57]. More sophisticated algorithms, such as Linear Neighborhood Propagation [47] and Laplacian Assignment [9] are built on this core idea.

Reinforcement learning

Reinforcement learning (RL) is the technique of solving Markov decision processes, in which an agent observes its environment, performs actions, and receives reward signals [41]. Much of the effort involved in reinforcement learning is concerned with learning a trustworthy value function, which maps states and actions to a real-valued estimate of "value", typically based on expected future rewards. As the dimension of the state-action space increases, the domain of the value function increases, and value function estimation becomes more difficult. In response, RL researchers have investigated the use of low-dimensional feature spaces in which the value function can

be approximated more easily. Much of the early work in feature learning for RL used a fixed set of nonlinear basis functions, such as radial or Fourier bases [25].

In a manifold learning context, feature representations for reinforcement learning can be constructed using manifold structure. Unlike the previously described manifold learning settings, in reinforcement learning the full set of instances X is not available at the outset. Instead, X is generated by the agent moving through state space as a result of a sequential decision process. This setting can provide additional clues to the underlying M structure, with the auxiliary data Z containing information about reward signals and state-action pairs.

Methods designed for manifold-based feature learning in reinforcement learning include Proto-Value Functions [30] and its directed variants [22]. These methods are limited by the requirement that the entire state space needs to have been explored to build an accurate representation of M, which is a prerequisite for constructing the value function approximation. Work on methods that simultaneously perform manifold discovery, feature construction, and policy learning is still an active reseasrch area.

Transfer learning

Transfer learning is the collective term for tasks in which information about one domain is used to bootstrap learning in another domain. This cross-domain learning strategy is attractive because it has the potential to speed the learning process for new problems dramatically, while also providing structure for underspecified problems [33]. For example, the "self-taught learning" framework uses a large unsupervised dataset to learn structure, then performs a specific learning task using a smaller labeled dataset [34].

One task in this category that has especially benefited from manifold-based approaches is data set alignment, a semi-supervised problem in which correspondences

are learned between multiple data sets. Manifold alignment is a class of techniques that solves this alignment problem when the input data sets are assumed to share a common underlying structure. These techniques find a shared latent space in which the disparate input data can be compared directly.

Manifold alignment was introduced as a semi-supervised, nonlinear extension of canonical correlation analysis (CCA) [1] that aimed to preserve both local geometry and inter-set correspondences [46]. Many methods for manifold-based transfer learning derive from this framework, including Semisupervised Manifold Alignment [19], Manifold Alignment Preserving Global Geometry [45], Manifold Warping [44], and Low Rank Alignment [6].

Metric learning

The final category of machine learning tasks we will consider is metric learning, the problem of defining a function that compares two instances in a semantically relevant way [54]. Specifically, we learn the function $d(x_i, x_j) \geq 0$ where $x \in X$, such that the distance d is smaller when two instances are more similar for some problem-specific notion of similarity.

Much of the work on metric learning has followed the framework of generalized Mahalanobis distance, which defines a specific distance $d_A(x_i, x_j) = \sqrt{(x_i - x_j)^\top A (x_i - x_j)}$ in terms of a matrix A. When $A = I$, this is Euclidean (L_2) distance. Traditional Mahalanobis distance uses $A = \mathrm{cov}(X)^{-1}$, and several methods have been proposed for finding an optimal A given semi-supervised label information, including Large Margin Nearest Neighbors [51] and Information Theoretic Metric Learning [11].

Manifold learning can be viewed as a kind of metric learning because the distances used are geodesics along the manifold. In this case, standard distance metrics will accurately capture local interactions, but underestimate the true manifold distance between non-local instances. Unlike generalized Mahalanobis distances, manifold dis-

tances may be nonlinear functions; this brings increased ability to match the problem semantics but also complicates the metric's generalizability. The dimension reduction methods reviewed above may be viewed as manifold-based metric learning methods as well, but an explicit embedding is not required for the discovery of metrics.

2.2 Manifold Discovery via Graph Construction

Recall step one of the manifold learning pipeline:

> Given a data set X, discover M, the underlying manifold structure from which X derives.

This section discusses representations for M, then reviews methods for graph construction and evaluation in the presence of one or more unknown manifolds.

Explicit vs. Implicit Graph Representations

An explicit graph can be viewed as a set of vertices and edges $G = (V, E)$, where the edges may be directed and/or weighted. In practice, it is often useful to represent these graphs as adjacency matrices W, where W_{ij} is the weight of the edge between vertices i and j. Most of the methods presented in Section 2.1 operate on explicit graphs, which are assumed to be sparse such that $|E| \propto |V|$.

Other methods use a dense pairwise similarity matrix rather than finding a set of edges for an explicit graph. This can be viewed as the adjacency matrix of a complete graph, with $|E| \propto |V|^2$. These methods include Low Rank Embeddings [27], t-SNE [43], Affinity Propagation [16], Low Rank Alignment [6], and Spectral Clustering [38]. In these cases, we view the graph construction problem as a complete graph weighting problem, and many of the graph construction methods presented below can be applied with minimal modification.

Unsupervised (Classical) Graph Construction

Traditional approaches to graph construction have treated the problem as an unsupervised task that relies on varied assumptions about the input data and the manifold(s) they lie on. These assumptions, which enable enhanced performance at the cost of flexibility, are a useful way to categorize families of existing methods.

The simplest assumption is that neighbors in original space are neighbors on the manifold, and this property underlies the k-nearest neighbors, ϵ-close neighbors, and b-matching methods [21]. The Perturbed/Disjoint Minimum Spanning Tree algorithms assume that the minimum spanning tree of the input data lies on the manifold [55]. Other methods assume that edges in the original space should lie on locally-linear subspaces, including Non-Local Manifold Tangent Learning [5] and Manifold Spanning Graphs [8]. Finally, some methods assume that the manifold is sampled non-uniformly, including Multi-Class Manifold Learning [52] and Robust Locally Linear Embedding [18].

Semisupervised Graph Construction

In many real-world applications, some edges of the desired graph are already known and the task is to find additional edges to complete the graph. These given edges may come from structural properties (i.e., trajectories, lattices, and trees), expert-provided labels, or constraints specific to the downstream task. For example, reinforcement learning and other sequential learning problems naturally generate data in the form of trajectories with explicit edges between time-adjacent states. This semi-supervised setting casts the edge assignment problem as a binary classification task in the unweighted case and a regression task in the weighted case.

Existing work in this category is relatively sparse. Huang and Jebara introduced a method for graph construction given a degree distribution [20]. Bengio and Monperrus

presented Non-Local Manifold Tangent Learning [5], in which a function is learned to predict edge connectivity given a k-nearest neighbors graph.

Graph Quality

Regardless of construction algorithm, it is important to be able to evaluate how well the constructed graph approximates the true underlying manifold. This evaluation may derive from the performance of downstream manifold learning applications, or from graph-theoretic properties of the graph itself. The former measure is directly applicable but expensive to compute, and is thus rarely used for improving graph quality "in the loop". The latter is less strongly linked to the specific task at hand, but often provides enough information to tune the graph construction algorithm's free parameters.

These graph-theoretic properties include *sparsity*, *connectivity*, *eccentricity*, and *centrality*. The sparsity of a graph is the ratio of actual edges to possible edges, $\frac{|E|}{|V|^2}$. A connected component is set of vertices with at least one path between all pairs in the set. The eccentricity of a vertex is the length of the longest shortest path between v and any other vertex in the graph, and the diameter of a graph is the maximum eccentricity over all vertices. Betweenness centrality is a measure of how common a given edge or vertex is among all shortest paths in a graph [15]. For weighted graphs, this is defined as

$$b\left(x\right) = \sum_{s \in V} \sum_{t \in V} \frac{\sigma_{st}\left(x\right)}{\sigma_{st}},$$

where x is an element of the graph (i.e., a vertex or an edge), σ_{st} is the total weight of all shortest paths between vertices s and t, and $\sigma_{st}\left(x\right)$ is the weight of only those $s - t$ shortest paths that include x.

Each of these measures can provide insights to the degree to which the graph approximates the underlying manifold. For instance, a single manifold should be approximated by a graph containing a single connected component, and a uniformly-

sampled manifold should be approximated by a graphs with uniform betweenness centrality.

Many graph construction algorithms target one or more of these properties. For example, the k-nearest neighbors algorithm exerts direct influence on the resulting graph's sparsity, but provides no guarantees about its other properties. Preliminary work has also shown that it is possible to tune existing graph construction algorithms by monitoring the changes to these properties while adjusting the algorithm's hyperparameters.

Multiple Manifold Discovery

In the preceding sections of this document, we have assumed that there exists only one underlying manifold that can explain our data concisely. In many practical applications, however, it is more accurate to assume that the data are generated by a mixture of manifolds. Each point can be assigned to exactly one manifold, though the manifolds may overlap or intersect.

With this new flexibility, it is possible to apply manifold learning algorithms to a much wider range of settings, but the difficulty of the manifold discovery process increases as well. As with traditional manifold discovery, existing methods of multiple manifold discovery can be grouped by the assumptions they rely on, in decreasing order of required information:

- The intrinsic dimension of each manifold is known: k-Manifolds [40].
- The maximum intrinsic dimension is known: Local and Structural Consistency [49], mumCluster [48], and Spectral Multi-Manifold Clustering [50].
- Only the number of manifolds is known: k-Planes [7], Spectral Clustering [38, 31].

These methods label each input point according the manifold it lies on, but do not necessarily build representations for each manifold. This reduces the multi-manifold discovery problem to a set of traditional manifold discovery tasks, for which single-manifold graph construction methods may be applied to each manifold independently.

CHAPTER 3

PROPOSED RESEARCH

The preceding chapters introduced the field of manifold learning and discussed the importance of manifold discovery. The review of existing work covered a wide array of methods for discovering, representing, and working with manifolds. In this chapter, I propose a course of study that expands upon this work.

3.1 Unsupervised Manifold Discovery

Unsupervised graph construction has a relatively large body of prior work, but room still exists for further innovation. Due to their unsupervised nature, these methods must exploit assumptions about the data and its generating process. Building on this notion, I propose several new research directions to pursue.

The first such proposal is *Active Manifold Discovery*. Rather than working with a fixed set of instances, we allow the algorithm to ask for samples from the manifold. This model has connections to reinforcement learning, directed exploration, and active learning. In this setting, the graph construction algorithm must maintain estimates of local manifold approximation error, which can be derived from measures of local intrinsic dimension. A successful method of this variety should be able to build a graph with fewer vertices than existing methods, without sacrificing manifold representation accuracy.

The next type of method I propose to explore is *Trajectory Reconstruction*. Given unordered instances that we assume to have been generated by one or more sequential

processes, the task is to recover trajectories along the manifold. This would build on the existing work in bandwidth minimization and sparse matrix reordering problems.

The final unsupervised method I propose is *Multi-Scale Graph Construction*. In this setting, we construct graphs at varying scales from coarse- to fine-grained, using the previously-computed graph to inform construction of the next graph. This method would build upon the semisupervised graph construction methods in the next section, because each coarse graph can be viewed as supervision for the next finer-scale graph.

3.2 Semisupervised Manifold Discovery

My course of study will also examine semisupervised graph construction algorithms, an area that has not received much attention. The first method I propose to explore is *Maximum Likelihood Graph Construction*, which, given a representative subset of edges, computes edge features and adds new edges in a maximum-likelihood style. The next method is *Trajectory Stapling*, which adds inter-trajectory edges given a set of trajectories. This could be achieved by computing reachability cones, or something equivalent that accounts for the directionality of edges. Finally, I will work on *Out-of-Sample Graph Extension*, which would add new edges to an existing graph, given a previously unseen set of points. This could draw from my proposed work on computing edge features, or build on Huang and Jebara's degree distribution ideas.

3.3 Multiple Manifold Discovery

By combining the manifold labeling components of the existing work with a graph construction algorithm, it may be possible to perform multi-manifold discovery in a single stage. I propose a new method for jointly learning manifold assignments and structure that iteratively assigns edges such that each connected component of the resulting graph is a good approximation of its manifold. This setting could be

described as an equilibration problem, with each connected component competing for edges to optimize its individual fitness function.

3.4 Task-Specific Manifold Discovery

Given a manifold learning task among those presented in Section 2.1, we can potentially make stronger assumptions about the structure of the input data. This allows for the design of tailor-made graph construction algorithms that work better than the generic ones from Section 2.2.

Graph Construction for Manifold Alignment

As described in section 2.1, we have multiple data sets $X^{(i)}$ that are assumed to lie on a common manifold M. Traditionally, M is found by building intra-set graph representations $W^{(i)}$ independently, then combining all the graphs together and adding edges using given inter-set correspondences. This has convenient parallelism properties, but does not exploit all available information.

I propose a novel method that makes use of the extra information available in manifold alignment graph construction problems, using the given inter-set correspondences to build better intra-set graphs. Correspondences can be used to define relative geodesic distance constraints, such that the path distance between two correspondence vertices in graph $W^{(i)}$ is proportional to the path distance between the corresponding vertices in graph $W^{(j)}$.

Graph Construction for Reinforcement Learning

In a reinforcement learning setting (section 2.1), several new constraints allow for improved graph construction. Edge directionality is semantically important, directed trajectories are easy to obtain, and the Markov assumption allows us to make intelligent edge assignments. On the other hand, efficient use of the available data is more

important, as the process of manifold discovery (and associated feature learning) is subject to the classic exploration vs. exploitation trade-off.

Building on my existing set of proposed work, I propose two extension methods that exploit the reinforcement learning domain-specific information described above. The first adapts the "trajectory stapling" method from section 3.2 by adding the Markov assumption to the set of constraints on new edges. This would improve upon the graph construction ideas from Johns and Mahadevan's directed Laplacian work [22], especially for continuous state-spaces with non-linear dynamics. The second method extends the "active manifold discovery" algorithm from section 3.1 by directing the learning agent to explore states that aid in graph construction. These new trajectories can be actively sampled from regions of the manifold with lower connectivity, increasing confidence in the manifold approximation with fewer observed states than a random exploration policy.

Graph Construction for Metric Learning

As described in section 2.1, manifold learning problems can often be viewed as nonlinear metric learning. To this end, graph construction algorithms can be tailored to metric learning tasks in which one or more weak similarity measures are known. Such an algorithm could construct local neighborhoods using the available, untrustworthy similarity measures, learning a manifold structure that can be used to define a global distance metric.

For example, consider the case of learning a distance metric for comparing pairs of Raman spectra (Figure 3.1). Each spectrum is an ordered set of (wavelength, intensity) tuples, which can often be simplified to a vector of intensity values when all spectra are sampled at the same wavelengths. A useful distance measure between spectra needs to account for the locations and shapes of peaks without placing undue emphasis on differences in peak amplitude and non-overlapping segments. For this reason,

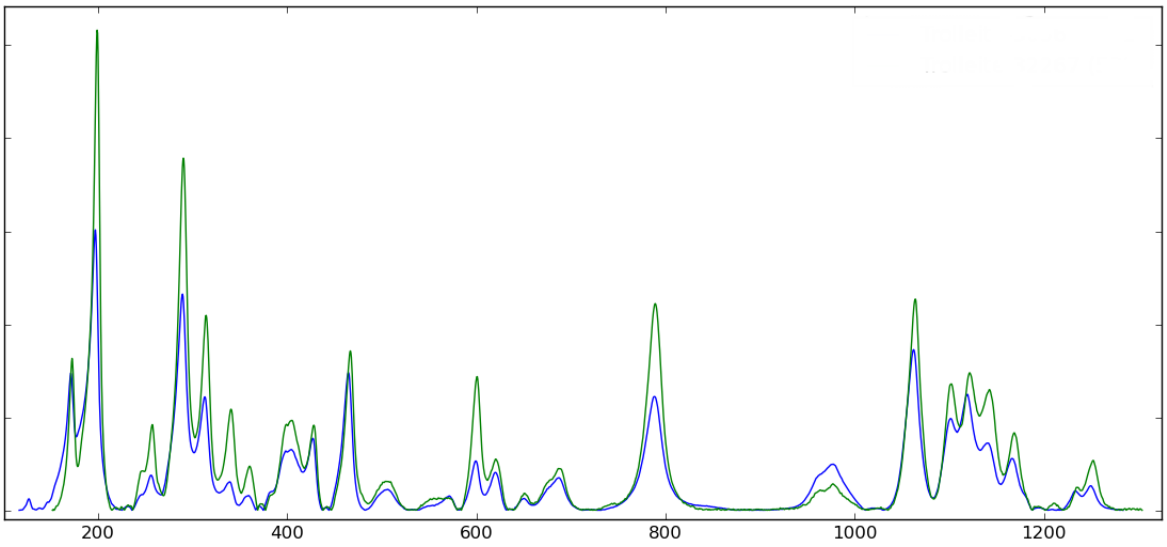


Figure 3.1: Example of a pair of Raman spectra. The two lines represent two spectra of the mineral species trolleite, with many matching peaks of varied heights. A good distance measure for Raman spectra should consider these to be very similar.

traditional metrics like Euclidean distance are not sufficient. This metric learning problem is of critical importance for geoscientists performing mineral identification, where new spectra are compared against libraries of known samples.

Some supervision for this problem as applied to the mineral identification task is available in the form of the hierarchical Dana classification system [17]. Each mineral is assigned, in increasing order of specificity, to a class, type, group, and species. This hierarchy can be used as a weak similarity measure, and can be combined with cosine similarity to inform the construction of a graph that approximates the manifold of Raman spectra. Manifold methods are especially applicable here because they are coordinate-free: spectra are ordered sequences of two-dimensional points, which makes traditional coordinate-based representations difficult to construct.

CHAPTER 4

EVALUATION

4.1 Graph-Based Manifold Approximation Benchmark

Studies of graph representations of manifolds require some objective measure of fitness for justification purposes. This fitness is typically defined in the context of a downstream manifold learning task, but both the task and dataset vary among published studies of graph construction methods. As a result, graph construction algorithms are often difficult to compare in a holistic way.

In my thesis, I will design and use a generic graph evaluation framework that will characterize the utility of a given graph construction algorithm across several types of downstream learning tasks and several types of input data. The result of this work will be a common grounding on which new and existing graph construction algorithms can be objectively compared.

4.2 Preliminary Results

Manifold Approximation

While the abovementioned benchmark is still a work in progress, my previous work has yielded firm results for problems with known manifold structure, such as the "swiss roll". Figure 4.1 demonstrates the hyperparameter sensitivity of two classic graph construction algorithms on a swiss roll. Figure 4.2 compares the performance of several graph construction algorithms over many randomized swiss rolls, using a correctness criterion based on true distances in the known underlying manifold.

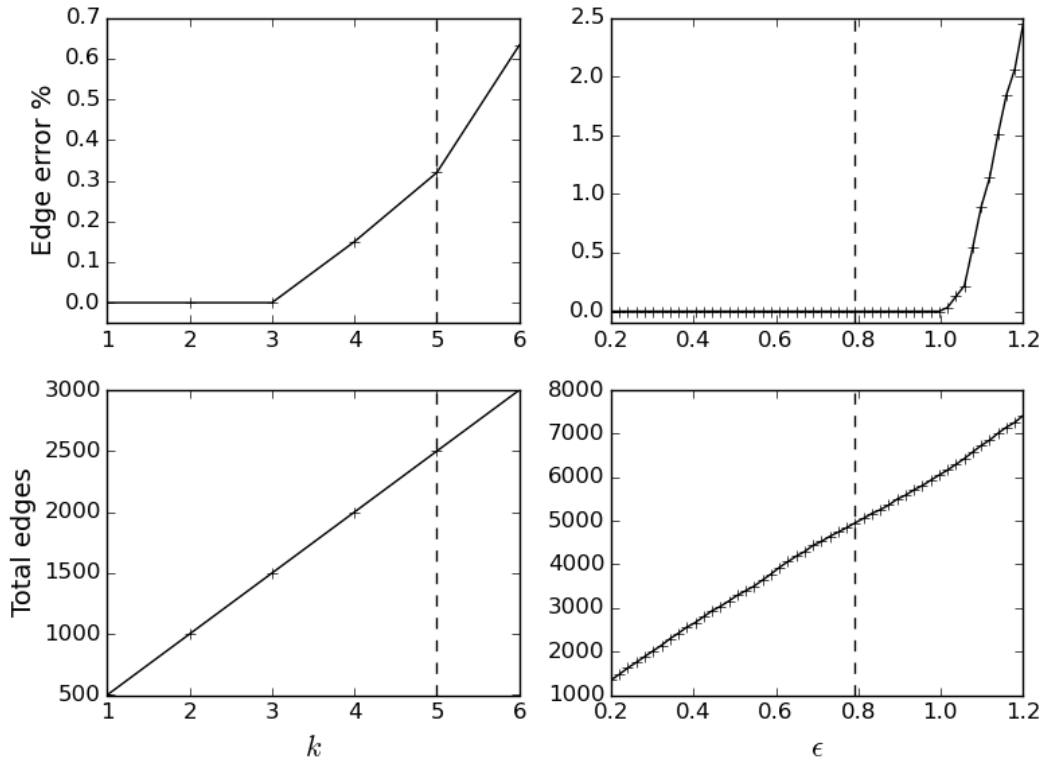
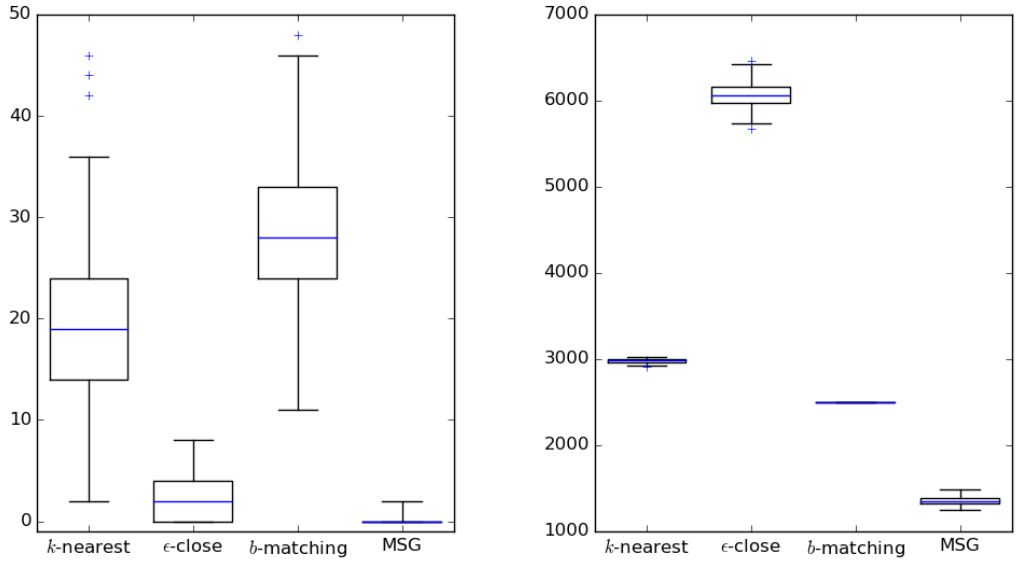


Figure 4.1: Sensitivity of k-nearest (left) and ϵ-close (right) algorithms. The dashed vertical lines represent the first hyperparameter value producing one connected component. Edge error percentage is calculated as a ratio of the number of short-circuiting edges to the total number of edges.



(a) Incorrect edges

(b) Total edges

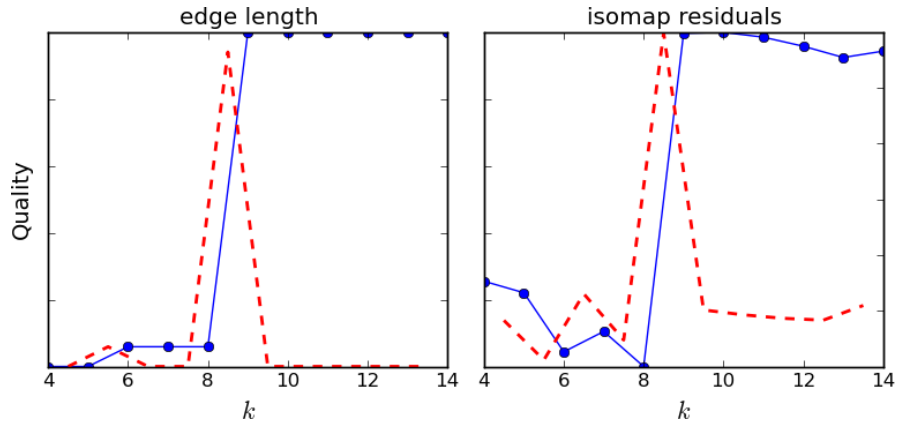
Figure 4.2: Summarized performance over 200 random swiss rolls, each with 500 points. Hyperparameters k, ϵ, and b were chosen to produce one connected component for the first swiss roll, then the same parameters were used for all future examples. The MSG algorithm produces graphs with almost zero incorrect edges, forming a single connected component with a modest number of total edges. The ϵ-close algorithm also produces a small number of bad edges, but tends to overconnect the graph. The k-nearest and b-matching algorithms produce a reasonable number of total edges, but many of these are incorrect. The high variance of their error rates also indicates sensitivity to noise.

In an as-yet unpublished paper, I investigated the use of graph theoretic measures (section 2.2) as a way to evaluate the quality of a graph's manifold approximation. Figure 4.3 demonstrates the efficacy of this approach using another dataset with a known manifold structure. The algorithm for choosing an optimal graph based on these heuristics is still an open problem, but these preliminary results indicate that it is possible to evaluate graph-based manifold discovery without requiring downstream processing.

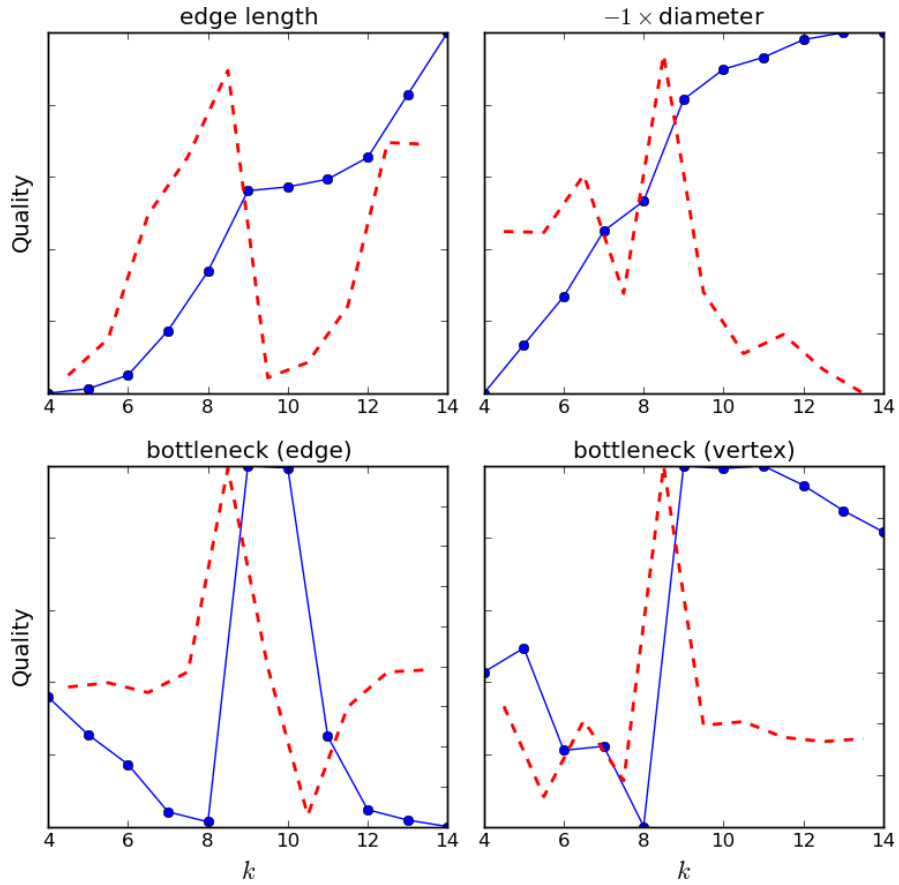
Manifold Learning

In a 2012 paper on Manifold Warping [44], Hoa Vu and I extended the manifold alignment framework [46] to the special case of sequential datasets. We applied Dynamic Time Warping [37], a classic sequence alignment algorithm, as a regularization step between successive iterations of traditional manifold alignment. This reduces the need for much of the supervision for the alignment task, creating reasonable inter-set correspondences by exploiting the trajectory structure of the data.

In a 2015 paper called Aligning Mixed Manifolds [6], Thomas Boucher and I extended the method of Low Rank Embedding [27] to the manifold alignment problem. This new method has the benefit of easy application to datasets containing multiple manifolds, and we demonstrated that this improves alignment performance on complex tasks like cross-language sentence retrieval and calibration transfer for spectroscopy.



(a) Ground truth measures: edge length in manifold space, and the residual error between the manifold coordinates and the Isomap embedding.



(b) Heuristic measures: edge length in original coordinates, graph diameter, and two variants of "bottlenecking" based on betweenness centrality.

Figure 4.3: Graph evaluation using an ensemble of heuristics, applied to the synthetic Isomap faces dataset [42]. The solid blue line indicates the value of each measure at varied parameter settings for k-nearest neighbor graph construction. The dashed red line indicates the slope of each blue segment for clarity. Note how the large jump in ground-truth manifold approximation error corresponds to large jumps in each of the graph-theoretic quality heuristics.

CHAPTER 5

CONCLUSION

Manifold discovery is an exciting problem with important applications to the wide domain of manifold learning. Existing algorithms for manifold discovery often rely on simplistic graph construction methods, which can prevent downstream learning steps from achieving the desired performance.

In this document, I lay the groundwork for a comphensive study of graph construction techniques that will be applied to the manifold discovery problem, and explain the need for new algorithms that fully exploit the available information using reasonable assumptions.

5.1 Original Work

In the previous sections, I discussed potential research directions building on the existing literature and my own original work. The publications resulting from my original work in this area are:

- **Manifold Warping** (Vu, Carey, and Mahadevan, AAAI 2012) [44]: We present a novel framework for aligning two sequentially-ordered data sets, taking advantage of a shared low-dimensional manifold representation. Our approach combines traditional manifold alignment and dynamic time warping algorithms using alternating projections. We also show that the previously-proposed canonical time warping algorithm is a special case of our approach.

- **Manifold Spanning Graphs** (Carey and Mahadevan, AAAI 2014) [8]: We demonstrate the hyperparameter sensitivity of existing graph construction meth-

ods, then present a new algorithm for unsupervised graph construction based on minimal assumptions about the input data and its manifold structure.

- **Aligning Mixed Manifolds** (Boucher, Carey, Mahadevan, and Dyar, AAAI 2015) [6]: We propose a novel manifold alignment algorithm, low rank alignment (LRA), that uses a low rank representation instead of nearest neighbor graph construction to embed and align data sets drawn from mixtures of manifolds. LRA does not require the tuning of a sensitive nearest neighbor hyperparameter or prior knowledge of the number of manifolds, both of which are common drawbacks with existing techniques.

- **Graph Evaluation for Manifold Discovery** (Carey, unpublished): We introduce a set of easily-computed graph quality heuristics that provide a way to evaluate graph construction hyperparameters without ground truth information about the underlying manifold. We then present a simple algorithm for automatic hyperparameter tuning based on an ensemble of these heuristics.

Over the course of my thesis work, I will explore methods for manifold discovery in many settings. In section 3.1, I described several methods for improving unsupervised graph construction, and in section 3.2, I extended that work to the semi-supervised case. In section 3.3, I proposed methods for graph construction in the presence of multiple manifolds.

In addition to these generic graph construction methods, I will also develop task-specific manifold discovery algorithms. In section 3.4, I proposed novel methods for performing transfer learning, reinforcement learning, and metric learning with specialized graph construction.

To demonstrate the impact of the aforementioned methods, I will create a generic evaluation framework for graph construction algorithms (4.1) that will allow for direct, objective comparisons of graph quality on a variety of downstream tasks. This

component of my thesis will provide the experimental justification for my proposed methods, and will aid future researchers in choosing the best algorithm for their tasks.

5.2 Thesis Statement

Graph construction that accounts for both manifold structure and task structure significantly improves the performance of manifold learning applications.

5.3 Timeline

The following is a timeline for the completion of my proposed work.

Spring 2015: Thesis proposal.

Fall 2015: Work on evaluation framework and proposed methods.

Spring 2016: Work on proposed methods.

Summer/Fall 2016: Thesis writing and defense.

BIBLIOGRAPHY

[1] Anderson, T.W. *An introduction to multivariate statistical analysis.* Wiley series in probability and mathematical statistics. Probability and mathematical statistics. Wiley-Interscience, 2003.

[2] Balasubramanian, Mukund, and Schwartz, Eric L. The isomap algorithm and topological stability. *Science 295*, 5552 (2002), 7–7.

[3] Belkin, M., and Niyogi, P. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in neural information processing systems 14* (2001), 585–591.

[4] Belkin, Mikhail, Niyogi, Partha, and Sindhwani, Vikas. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *The Journal of Machine Learning Research 7* (2006), 2399–2434.

[5] Bengio, Yoshua, and Monperrus, Martin. Non-local manifold tangent learning. *Advances in Neural Information Processing Systems 17* (2005), 129–136.

[6] Boucher, Thomas, Carey, CJ, Mahadevan, Sridhar, and Dyar, M Darby. Aligning mixed manifolds. *Twenty-Ninth AAAI Conference on Artificial Intelligence* (2015).

[7] Bradley, Paul S, and Mangasarian, Olvi L. k-plane clustering. *Journal of Global Optimization 16*, 1 (2000), 23–32.

[8] Carey, CJ, and Mahadevan, Sridhar. Manifold spanning graphs. In *Twenty-Eighth AAAI Conference on Artificial Intelligence* (2014).

[9] Carreira-Perpinán, Miguel A, and Wang, Weiran. Lass: A simple assignment model with laplacian smoothing. *arXiv preprint arXiv:1405.5960* (2014).

[10] Coifman, Ronald R, and Lafon, Stéphane. Diffusion maps. *Appl. Comput. Harmon. Anal 21* (2006), 5–30.

[11] Davis, Jason V, Kulis, Brian, Jain, Prateek, Sra, Suvrit, and Dhillon, Inderjit S. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning* (2007), ACM, pp. 209–216.

[12] Demartines, Pierre, and Hérault, Jeanny. Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets. *Neural Networks, IEEE Transactions on 8*, 1 (1997), 148–154.

[13] Donoho, David L, and Grimes, Carrie. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences 100*, 10 (2003), 5591–5596.

[14] Elhamifar, Ehsan, and Vidal, René. Sparse manifold clustering and embedding. In *Advances in neural information processing systems* (2011), pp. 55–63.

[15] Freeman, Linton C. A set of measures of centrality based on betweenness. *Sociometry* (1977), 35–41.

[16] Frey, Brendan J, and Dueck, Delbert. Clustering by passing messages between data points. *science 315*, 5814 (2007), 972–976.

[17] Gaines, Richard V, Dana, James Dwight, and Dana, Edward Salisbury. *Dana's new mineralogy: the system of mineralogy of James Dwight Dana and Edward Salisbury Dana*. Wiley, 1997.

[18] Hadid, Abdenour, and Pietikäinen, Matti. Efficient locally linear embeddings of imperfect manifolds. In *Machine learning and data mining in pattern recognition*. Springer, 2003, pp. 188–201.

[19] Ham, Jihun, Lee, Daniel, and Saul, Lawrence. Semisupervised alignment of manifolds. In *Proceedings of the Annual Conference on Uncertainty in Artificial Intelligence, Z. Ghahramani and R. Cowell, Eds* (2005), vol. 10, pp. 120–127.

[20] Huang, Bert, and Jebara, Tony. Maximum likelihood graph structure estimation with degree distributions. In *Analyzing Graphs: Theory and Applications, NIPS Workshop* (2008), vol. 14.

[21] Jebara, Tony, Wang, Jun, and Chang, Shih-Fu. Graph construction and b-matching for semi-supervised learning. In *Proceedings of the 26th Annual International Conference on Machine Learning* (2009), ACM, pp. 441–448.

[22] Johns, Jeff, and Mahadevan, Sridhar. Constructing basis functions from directed graphs for value function approximation. In *Proceedings of the 24th international conference on Machine learning* (2007), ACM, pp. 385–392.

[23] Jolliffe, I.T. *Principal component analysis*. Springer series in statistics. Springer-Verlag, 2002.

[24] Kohonen, Teuvo. The self-organizing map. *Proceedings of the IEEE 78*, 9 (1990), 1464–1480.

[25] Konidaris, George, Osentoski, Sarah, and Thomas, Philip. Value function approximation in reinforcement learning using the fourier basis. In *Twenty-Fifth AAAI Conference on Artificial Intelligence* (2011).

[26] Kruskal, Joseph B., and Wish, Myron. *Multidimensional scaling*, vol. 11. Sage, 1978.

[27] Liu, R., Hao, R., and Su, Z. Mixture of manifolds clustering via low rank embedding. *Journal of Information & Computational Science 8* (2011), 725–737.

[28] Liu, Wei, He, Junfeng, and Chang, Shih-Fu. Large graph construction for scalable semi-supervised learning. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)* (2010), pp. 679–686.

[29] Ma, Y., and Fu, Y. *Manifold Learning Theory and Applications*. CRC Press, 2011.

[30] Mahadevan, Sridhar, and Maggioni, Mauro. Proto-value functions: A laplacian framework for learning representation and control in markov decision processes. *Journal of Machine Learning Research 8*, 2169-2231 (2007), 16.

[31] Ng, Andrew Y, Jordan, Michael I, Weiss, Yair, et al. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems 2* (2002), 849–856.

[32] Page, Lawrence, Brin, Sergey, Motwani, Rajeev, and Winograd, Terry. The PageRank citation ranking: Bringing order to the web. Tech. rep., Stanford InfoLab, 1999.

[33] Pan, Sinno Jialin, and Yang, Qiang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on 22*, 10 (2010), 1345–1359.

[34] Raina, Rajat, Battle, Alexis, Lee, Honglak, Packer, Benjamin, and Ng, Andrew Y. Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the 24th international conference on Machine learning* (2007), ACM, pp. 759–766.

[35] Roweis, S.T., and Saul, L.K. Nonlinear dimensionality reduction by locally linear embedding. *Science 290*, 2323–232 (2000).

[36] Rowland, Todd. Manifold. From MathWorld—A Wolfram Web Resource. Last visited on March 30, 2015.

[37] Sakoe, H., and Chiba, S. Dynamic programming algorithm optimization for spoken word recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on 26*, 1 (1978), 43–49.

[38] Shi, Jianbo, and Malik, Jitendra. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 22*, 8 (2000), 888–905.

[39] Silva, Vin D, and Tenenbaum, Joshua B. Global versus local methods in nonlinear dimensionality reduction. In *Advances in neural information processing systems* (2002), pp. 705–712.

[40] Souvenir, R., and Pless, R. Manifold clustering. In *Tenth IEEE International Conference on Computer Vision* (2005), vol. 1, IEEE, pp. 648–653.

[41] Sutton, Richard S, and Barto, Andrew G. *Introduction to reinforcement learning*. MIT Press, 1998.

[42] Tenenbaum, J. B., Silva, V. De, and Langford, J. C. A global geometric framework for nonlinear dimensionality reduction. *Science* (2000).

[43] Van der Maaten, Laurens, and Hinton, Geoffrey. Visualizing data using t-sne. *Journal of Machine Learning Research 9*, 2579-2605 (2008), 85.

[44] Vu, H.T., Carey, CJ, and Mahadevan, Sridhar. Manifold warping: Manifold alignment over time. In *AAAI* (2012).

[45] Wang, C., and Mahadevan, S. Manifold alignment preserving global geometry. *The 23rd International Joint conference on Artificial Intelligence* (2013).

[46] Wang, Chang, and Mahadevan, Sridhar. A general framework for manifold alignment. In *AAAI Fall Symposium on Manifold Learning and its Applications* (2009), pp. 53–58.

[47] Wang, Fei, and Zhang, Changshui. Label propagation through linear neighborhoods. *Knowledge and Data Engineering, IEEE Transactions on 20*, 1 (2008), 55–67.

[48] Wang, Y., Jiang, Y., Wu, Y., and Zhou, Z.H. Multi-manifold clustering. In *PRICAI 2010: Trends in Artificial Intelligence*. Springer, 2010, pp. 280–291.

[49] Wang, Yong, Jiang, Yuan, Wu, Yi, and Zhou, Zhi-Hua. Local and structural consistency for multi-manifold clustering. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence* (2011), vol. 22, Citeseer, p. 1559.

[50] Wang, Yong, Jiang, Yuan, Wu, Yi, and Zhou, Zhi-Hua. Spectral clustering on multiple manifolds. *Neural Networks, IEEE Transactions on 22*, 7 (2011), 1149–1161.

[51] Weinberger, Kilian Q, and Saul, Lawrence K. Distance metric learning for large margin nearest neighbor classification. *The Journal of Machine Learning Research 10* (2009), 207–244.

[52] Wu, Yiming, and Chan, Kap Luk. An extended isomap algorithm for learning multi-class manifold. In *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on* (2004), vol. 6, IEEE, pp. 3429–3433.

[53] Yang, Bo, and Chen, Songcan. Sample-dependent graph construction with application to dimensionality reduction. *Neurocomputing 74*, 1 (2010), 301–314.

[54] Yang, Liu, and Jin, Rong. Distance metric learning: A comprehensive survey. *Michigan State Universiy 2* (2006).

[55] Zemel, Richard S, and Carreira-Perpiñán, Miguel Á. Proximity graphs for clustering and manifold learning. In *Advances in neural information processing systems* (2004), pp. 225–232.

[56] Zhang, Zhenyue, and Zha, Hongyuan. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM Journal on Scientific Computing 26*, 1 (2005), 313–338.

[57] Zhou, Dengyong, Bousquet, Olivier, Lal, Thomas Navin, Weston, Jason, and Schölkopf, Bernhard. Learning with local and global consistency. *Advances in neural information processing systems 16*, 16 (2004), 321–328.

[58] Zhu, Xiaojin, and Ghahramani, Zoubin. Learning from labeled and unlabeled data with label propagation. Tech. rep., CMU CALD, 2002.