

Dissimilarity learning from Minimum Spanning Tree for Clustering

Sung Jin Hwang

Advisors: Professors Damelin and Hero

Abstract

In machine learning literatures, clustering is to find a natural segmentation, or clusters of samples. There are many suggested algorithms to solve clustering problem, all of them with some limitations. Many of these limitations, however, come from the dissimilarity measure used.

To achieve better clustering performance, more natural clustering dissimilarity measure between sample points need to be discovered. L^2 distance is a very common choice for dissimilarity measure in machine learning applications but this distance fails to take advantage of other samples. Recent developments of ISOMAP or Eigenmap techniques suggest that graph approach is useful to establish a natural clustering dissimilarity measure.

The research will concentrate on development of new dissimilarity measure for clustering based on minimum spanning tree (MST). The basic idea is that Prim's algorithm, a greedy algorithm that solves MST problem, suggests a natural way to grow neighborhood from a sample point in the sense that it finds the nearest neighbor at each iteration. This leads us to the concept of signature of a sample point. The research studies the effect of new dissimilarity measures based on MST on clustering algorithms, especially k -means.

Contents

1	Introduction	4
1.1	Clustering	4
1.2	Dimensionality Reduction	5
2	Main Idea	6
2.1	Neighborhood Estimation	7
2.2	Dissimilarity Based on Neighborhood Estimation	8
3	Analysis	11
4	Experiments	12
5	Discussion and Future Works	14
6	References	15
A	Computation of MST	15
B	Triangle Inequality	16

1 Introduction

1.1 Clustering

In many areas of pattern recognition or machine learning, perhaps classification or supervised learning is more familiar problem. In most cases, the theories of classification require certain amount of training data and help from experts of specific applications with prior information beyond the scope of learning theories to provide certain answers to the training data. For example, to test whether a patient is affected by certain disease, classification frameworks ask to train themselves with data from nontrivial number of patients and it must be known whether each patient has the disease determined by experts, here probably medical doctors. In real applications, however, it is often the case such expert help is not available or there is no clue for sample how sample points can be classified. Unsupervised learning or clustering asks to solve classification problems with no training sample. Since lack of training sample occurs in many real applications, there are many applications for clustering techniques and they include computer vision, document clustering, and bioinformatics[?].

Like other machine learning applications, there are mainly two branches for attempts for clustering problem. One is generative approach in which every sample points are considered as a realization of a random variable from some mixture of probability distributions. Well-known frameworks which belong to this branch include EM algorithm. Though generative approach is a natural solution when there exists adequate prior information, the more preferable branch in clustering is discriminative approach, where distances between sample points and geometric structures the sample points consist is more emphasized. Among discriminative approaches, perhaps the most popular and successful method so far is k -means algorithm.

Despite its success, k -means algorithm has its weaknesses and one of them is that the algorithm creates convex and blob-like clusters. This is due to the fact that the algorithm assumes a centroid or a representative point for each cluster and determines the cluster a sample point belongs to by comparing the distances to these centroids. However, though often k -means algorithm is regarded to work with L^2 norm, it works generally with other kind of norms. One of the reasons that L^2 norm is often chosen for k -means algorithm is that use of other well-known norms like L^1 norm does not bring noticeable improvement in most of the cases, which suggests the need of deliberate ways to learn more proper notion of dissimilarity for given sample. One of the reasons is that commonly used norms have local property in that norms do not take advantage of the sample set, which is often the only information given.

Recently there have been some successful and motivating researches in spectral clustering addressing this dissimilarity learning problem; Shi and Malik[?], Ng et al.[?], etc. For example, Ng et al.[?] introduces spectral clustering method which maps sample points into \mathbb{R}^k with coordinates from eigenvectors of graph Laplacian[?] of sample. This effectively re-defines dissimilarity between sample points in terms of L^2 norm in \mathbb{R}^k . The authors of algorithms illustrate some

success to discover nonconvex clusters in the original space.

This example turns our attention to dimensionality reduction techniques in that they not only reduce the dimensions of data but can also assign new dissimilarities between sample points.

1.2 Dimensionality Reduction

Classical examples of dimensionality reduction are principal component analysis (PCA) and multidimensional scaling (MDS). PCA is a projection technique that maximizes variance or norm of each sample point and tries to preserve the dot product between sample points. MDS, or classical MDS is similar but it finds lower dimension coordinates such that pairwise distances between sample points are best approximated. These two traditional examples in common try to reduce the dimension with respect to the given inner product or norm, which may not be reliable.

There has been attempts to generalize PCA and one probably most successful one is kernel PCA[?]. Classical PCA computes eigen-decomposition of Gram matrix of dot products but in general dot product can be replaced with any other types of inner product, and equivalently, if there exists a real *kernel* function that generates Gram matrix of sample and if the matrix is positive definite, all mechanisms of PCA work.

Some other dimensionality reduction techniques include ISOMAP[?], locally linear embedding (LLE)[?], Eigenmap[?], etc. ISOMAP is a generalization of MDS in that it uses shortest path distance on nearest-neighbor graph of sample points rather than L^2 distance and induces lower dimensional coordinates which approximates the shortest path distances. Since ISOMAP specifies every pairwise distance between sample points to be mapped, ISOMAP is known to find the global geometric structure.

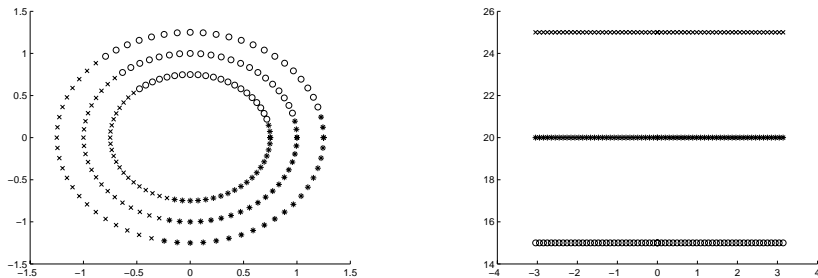
Eigenmap[?] is a technique that computes graph Laplacian of nearest-neighbor graph of sample points and assigns eigenvectors of graph Laplacian as coordinates of sample points. The authors show that graph Laplacian is an approximation of Laplace Beltrami operator and it has properties desirable for embedding. Though Eigenmap does not force to use kernelized approach, practically kernel functions are used for Eigenmap applications to determine weights on graph Laplacian matrix. More recent development of diffusion map[?] can be viewed as interpretation of Eigenmap in terms of Markov chain.

These techniques, with their strengths, have shortcomings as well. For example, kernelized approaches, even though kernel function can be any positive definite function, the most popular choice is Gaussian kernel, $k(x, y) = \exp(-|x - y|^2/t)$, for data which is expected to have some (hidden) geometric structure. The problem is with the parameter t , the kernel width. Though it is known that the kernel width parameter can greatly affect kernelized techniques, there has been no good theoretical framework to determine t and this leads to some heuristics and trial-and-error.

The dimensionality techniques introduced, including Eigenmap which is used in [?], are not class-sensitive. These are general methods to discover geometric

structure and they do not take into consideration to separate clusters. In this paper, we try to establish a new dissimilarity measure between sample points from different perspective based on neighborhood estimation specifically aimed for clustering.

2 Main Idea



(a) Three circles clustered with k -means algorithm with L^2 distance. Each 'o', 'x', and '*' stand for different clusters. They are not well clustered.

(b) Three circles in polar coordinate with some scaling. Now k -means algorithm with L^2 distance discovers three circles.

Figure 1: three circles example

One example illustrating the importance of dissimilarity choice is figure 1, where we have three concentric circles to be clustered. Standard k -means algorithm with L^2 norm, which is commonly used in clustering problems, yields data labels shown in figure 1(a), and the algorithm fails to detect the circles. The reason is that L^2 norm does not catch the structures of data and some sample points from different circles are measured to be closer while some sample points from the same circles are measured to be dissimilar. However, from elementary calculus we know polar coordinate will be more natural and useful for concentric circles. Figure 1(b) shows k -means result of three circles on polar coordinate¹ using L^2 distance and the algorithm now detects three circles.

We have seen that change of coordinates and dissimilarity measure can affect and improve the clustering algorithms. However, the previous example was a very simple one. The dimensionality of the sample was low so that it could be easily visualized, and a well known coordinate, polar coordinate, was a natural choice for the sample. Of course most of the real problems are not so simple and we need more powerful method that does not require human perception.

In the following sections, \mathcal{X}_n will be the set of n sample points, *input space* will the space where \mathcal{X}_n belongs to², and d will be the metric in the input space.

¹Some scaling has been applied as well.

²It is often some finite dimensional Euclidean space \mathbb{R}^m .

When the elements of \mathcal{X}_n are treated as random variables for analysis, they are considered to be independent.

2.1 Neighborhood Estimation

Now we have seen that coordinates and dissimilarity measure choice can be critical to clustering problem, so our goal is to make a new dissimilarity measure so that dissimilarity is low for pairs of points in the same cluster and high for pairs of points in different clusters, and we need to construct it from information given from the input space and the sample.

It is evident that we need at least some minimal assumptions since, for example, if the data extraction or data sampling is random so that the cluster structure is completely ignored then we are not likely to recover the clusters. One reasonable assumption is that the sampling procedure or sampling function³ is in some sense continuous such that some neighborhood structure is preserved. In other words, we may assume that for every sample point, there exists some connected neighborhood, possibly with respect to the given metric, such that its inverse image is contained within a cluster.

This nearest neighbor assumption is widely exploited in many applications. For example, ISOMAP uses shortest paths on nearest-neighbor graph to estimate geodesic distance between sample points. This can be justified since when k is constant and n goes to infinity, the distance to k -th nearest neighbor goes to zero in probability when k is fixed and this implies that the line segments from a sample point to its some nearest neighbors asymptotically become tangential to the local geometric structure. Eigenmap[?] also uses graph Laplacian on nearest neighbor graph. Recent application with more emphasis on neighborhood structure can be found in Hadsell et al.[?]

There are two major variants in nearest-neighbor rule: ϵ -nearest neighbor and k -nearest neighbor. For a sample point $x_i \in \mathcal{X}_n$, let $N_\epsilon(x_i)$ be the ϵ -nearest neighbor and let $N_k(x_i)$ be the k -nearest neighbor of x_i , then they are defined as

$$N_\epsilon(x_i) = \{x_j \in \mathcal{X}_n : 0 < d(x_j, x_i) < \epsilon\}$$

$$N_k(x_i) = \{x_j \in \mathcal{X}_n : 0 < |\{x_k \in \mathcal{X}_n : d(x_k, x_i) < d(x_j, x_i)\}| < k + 1\}$$

These nearest neighbor rules, however, are under some arguable assumptions from practical point of view. Note that it is more likely that a neighborhood to have its inverse image contained within a cluster when the neighborhood is small. Asymptotically, for any $\epsilon > 0$, a neighborhood centered in the probability density support includes infinitely many sample points. In practice, however, small ϵ will make the neighborhood contain no sample point. For k -nearest neighbor, when k is small, the graphs constructed by connecting neighbors tend

³In this paper, sampling function will mean the process of mapping sample points from unknown abstract space to some input space. An example is image scanning. Then here, the unknown abstract space will be the set of photographs and the input space will be some finite dimensional space of which dimension is equal to the number of pixels.

to break into many small components such that between most of sample points their relationship cannot be determined. Therefore there is a tradeoff on neighborhood size.

Though this tradeoff cannot be eliminated, there is room for improvement. From here, we take deviation and suggest an alternative way to grow the neighborhood. From the preceding argument, for a sample point $x_i \in \mathcal{X}_n$, 1-nearest neighbor of x_i , say x_j , is the most likely sample point that would share some commonalities, that is in the same cluster. Then to enlarge the neighborhood further, we use currently available information on the neighborhood of x_i ; x_j is the most likely sample point to be included in the neighborhood and we may find the nearest neighbor of a subset $\{x_i, x_j\}$.

Then dissimilarity between (disjoint) subsets of sample points must be defined to discuss the nearest neighbor of $\{x_i, x_j\}$. This topic has been dealt extensively in hierarchical clustering literatures[?] and this dissimilarity measure between subsets of sample points is usually referred as linkage. Among various linkages that have been suggested so far, we choose single linkage:

$$d_{SL}(A, B) \equiv \inf\{d(x, y) : x \in A, y \in B\}$$

There are mainly two reasons for choosing single linkage. First, strategy here is to place confidence to the pairs of sample points with least dissimilarity in input space, and single linkage is the one which best follows this scheme. For example, complete linkage which is defined as $d_{CL}(A, B) \equiv \sup\{d(x, y) : x \in A, y \in B\}$ places weights to the pair of sample points with maximum distance, which is not coherent with our line of thought. Second, one of the goal here is to discover complicated geometric structure but most of the common linkage types except for single linkage have tendency to create convex clusters and this behavior is against our purpose.

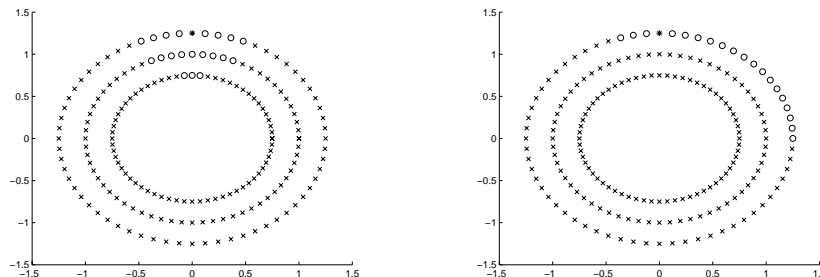
This procedure can be summarized as follows:

$$\begin{aligned} N_1(x_i) &= \{x_i\} \\ N_k(x_i) &= N_{k-1}(x_i) \cup \{\arg \min_{x_j \in \mathcal{X}_n \setminus N_{k-1}(x_i)} d_{SL}(\{x_j\}, N_{k-1}(x_i))\} \end{aligned}$$

Figure 2 shows comparison between k -nearest neighbor and neighborhood estimation with $\{N_k(x_i)\}_{k=1}^n$. x_i here is the point at the top of the outer circle and figure 2(a) shows k -nearest neighbor of x_i and it can be observed that sample points from other circles are included in this neighborhood. Meanwhile figure 2(b) shows $N_{k+1}(x_i) \setminus N_1(x_i)$ and this neighborhood does not include sample points from other circles.

2.2 Dissimilarity Based on Neighborhood Estimation

The simplest way to cluster from neighborhood estimations $\{N_k\}_{i,k=1}^n$ will be to construct neighborhood graph with proper k and use components as clusters. However, this method shares many shortcomings of partitioning by nearest neighbors; for example, the components do not split in reliable way. Moreover,



(a) Three circles example. The points with 'o' mark are 20-nearest neighbors of the point with '*' mark.

(b) The points with 'o' mark are elements of $N_i^{21} \setminus N_i^1$ and the point with '*' mark is the element of N_i^1 . Due to many pairs of equal distance, the ties are broken arbitrarily.

Figure 2: neighborhood comparison on three circles example

N_k as well loses its confidence as k increases since larger the k is, more likely there exist some links of which distance is large, hence $\{N_k\}$ is less reliable⁴. For example, $N_n(x_i) = \mathcal{X}_n$ for all i and it can be easily expected that as k increases, there is less useful information in $N_k(x_i)$. Therefore rather than to determine cluster structure directly from some N_k 's, our approach to establish new dissimilarity measure from the set of increasing chains $\{\{N_k(x_i)\}_{k=1}^n\}_{i=1}^n$.

One already suggested is to use the collision time between chains, or *dual-rooted hitting time*[?] defined as

$$h(x_i, x_j) = \min\{t : N_t(x_i) \cap N_t(x_j) \neq \emptyset\}.$$

Grikschat et al.[?] includes some successful unsupervised and semi-supervised learning results. However, it is also true that this dual-rooted hitting time suffers from computation issue since there is yet no efficient algorithm to compute the hitting time between all the chains, and there is need for some more efficient means to extract information from the chains.

One useful insight is that since $\mathcal{X}_n = \{N_k(x_i)\}_{k=1}^n$ for all i , a family of functions $\{\phi_i\}_{i=1}^n$, which will be mentioned as *Prim signatures*, can be defined such that

$$\begin{aligned} \phi_i : \mathcal{X}_n &\rightarrow \{1, 2, \dots, n\} \\ \phi_i(x_j) = k &\Leftrightarrow N_k(x_i) \setminus N_{k-1}(x_i) = \{x_j\}. \end{aligned}$$

Note that dual-rooted hitting time can be defined in this Prim signature framework: $h(X_i, X_j) = \min_{x_k \in \mathcal{X}_n} \max\{\phi_i(x_k), \phi_j(x_k)\}$. If we treat Prim signatures to be a natural descriptor here, then dissimilarities other than dual-rooted hitting time on Prim signatures can be of interest. One interesting view on Prim signatures is to interpret them as rank functions or permutations since

⁴More details will be discussed in analysis section

they are all bijections. Some useful statistics from rank theory are[?]

Kendall's coefficient:

$$\tau = \binom{n}{2}^{-1} \sum_{1 \leq k < l \leq n} \text{sign}(\phi_i(x_k) - \phi_i(x_l)) \text{sign}(\phi_j(x_k) - \phi_j(x_l))$$

Spearman's coefficient:

$$\rho = \frac{12}{n(n^2 - 1)} \sum_{k=1}^n \left(\phi_i(x_k) - \frac{n+1}{2} \right) \left(\phi_j(x_k) - \frac{n+1}{2} \right)$$

From Kendall's τ , corresponding distance can be defined: [?]

$$I(\phi_i, \phi_j) = |\{(i, j) : 1 \leq k, l \leq n, \phi_i(x_k) < \phi_i(x_l), \phi_j(x_k) > \phi_j(x_l)\}|.$$

Also from Spearman's coefficient, the corresponding distance is L^2 distance, and to avoid confusion, let S be the L^2 distance defined on Prim signatures[?]. Note that since Prim signatures are on the $(n - 1)$ -dimensional hypersphere S^{n-1} in \mathbb{R}^n , a very natural choice for distance is arc distance $\cos^{-1}(\langle \phi_i, \phi_j \rangle)^5$, which is geodesic distance on S^{n-1} . Elementary calculus shows that arc distance and distance S have monotonically increasing relationship on S^{n-1} and this relationship is approximately linear in close range.

This arc distance is also related to the distance I . To see this intuitively, first consider ϕ_i 's as n -dimensional vectors. The distance $I(\phi_i, \phi_j)$ is equivalent to the least number of *adjacent swaps* on ϕ_i^{-1} to ϕ_j^{-1} [?]⁶. Then each adjacent swap is a hop on all $n!$ permutation vectors where these hops occur between the vectors whose L^1 norm is 2, on S^{n-1} . Then the trace of adjacent swaps from ϕ_i^{-1} to ϕ_j^{-1} with least number of swaps will correspond to the shortest path on $(n - 1)$ -nearest neighbor graph of $n!$ nodes on S^{n-1} and asymptotically this shortest path distance will converge to the geodesic distance on S^{n-1} , the arc distance.

It should be mentioned that these are entirely based on the ranks of sample points and explicit values of distances in the input space are discarded. Dimensionality techniques vary their decisions on this matter. For example, since ISOMAP uses shortest path distance from the input space, ISOMAP does use explicit values from d and this is why ISOMAP assumes isometry between the input space and the embedded structure. On the other hand, applications like dimensionality reduction by learning invariant mapping(DrLIM)[?] consider only neighborhood relationship and discard the explicit values from d . And applications like Eigenmap leaves this decision to the users. In fact, when input space is not reliable, more active use of d may be undesirable. For instance, consider 1-nearest neighbor of concentric circles as in the previous three circles example. When the metric in input space is used, the sample points on circles with larger radii will be loosely connected and the sample points on the circles with smaller radii will be tightly connected, but this is not always desirable.

⁵Proper normalization assumed for ϕ_i and ϕ_j .

⁶Note that the term *adjacent* is well-defined since the domain of ϕ_i^{-1} is $\{1, 2, \dots, n\}$.

3 Analysis

In the previous section, single linkage concept from hierarchical clustering was used to construct $\{N_k(x_i)\}_{k=1}^n$. Then it is natural to ask its connection to hierarchical clustering with single linkage since vast amount of analysis exists for hierarchical clustering. A good starting point is that single linkage agglomerative hierarchical clustering(SLAHL) discovers minimum spanning tree(MST) structure [?]⁷, and simple comparison reveals that SLAHL is identical to Kruskal's algorithm[?], a classic algorithm to find MST, while the iteration steps for neighborhood estimation explained in the previous section is known as Prim's algorithm[?] in graph theory literatures. Therefore our neighborhood estimation shares many properties with SLAHL and they differ in the direction of approach: local greedy optimization and global optimization.

One of theories in hierarchical clustering that helps us in analysis is the consistency analysis of single linkage in the terms of high-density clusters[?]. High-density clusters are maximally connected sets of probability density greater than given threshold α . Then consider nearest neighbor density estimator

$$\hat{f}_n(x) = \frac{C}{\min_{x_i \in \mathcal{X}_n} d^m(x, x_i)}$$

where m is the dimension of the input space and C is a constant. Then this density estimation is inversely proportional to the least volume of sphere centered at x that contains a sample point. If spheres are centered at sample points with radius r such that $\alpha = Cr^{-m}$, then the union of these spheres is the level set of level α and the components are the high-density clusters with respect to \hat{f}_n , and SLAHL discovers these high-density clusters at some steps.

Note that when $|e_k(x_i)| = d_{SL}(N_k(x_i) \setminus N_{k-1}(x_i), N_{k-1}(x_i))$ and $\arg \max_{l=1,2,\dots,k+1} |e_l(x_i)| = k+1$, then $N_k(x_i)$ is a cluster discovered by SLAHL at some level since spheres of certain volume centered at elements of $N_k(x_i)$ are maximally connected. This indicates that each $N_k(x_i)$ is high-density region around x_i . For example, let infimum of \hat{f}_n on $N_k(x_i)$ is greater than α and there exists a sample point x_j outside $N_k(x_i)$ such that infimum of \hat{f}_n on $N_k(x_i) \cup \{x_j\}$ is still greater than α . If we happen to add a sample point to $N_{k+1}(x_i)$ such that the infimum is less than α , then high-density cluster of α that contains x_i is not detected and this violates the analysis of SLAHL.

Now we establish a hypothesis: $N_k(x_i)$ is a subset of the input space such that it contains k sample points, which includes x_i , and the infimum of nearest neighbor density estimation on paths that connects sample points in $N_k(x_i)$ is maximal. Note that this hypothesis implies greedy maximization of probability density and it is analogous to the fact that Prim's algorithm, a greedy algorithm, finds the global minimum, MST.

To prove the hypothesis, the concept of Voronoi partition is useful. Voronoi partition divides the input space into n subsets or cells such that the nearest sample point of all points in i -th cell is x_i . Then by definition, in each cell

⁷It finds MST structure of a complete graph of which node set is \mathcal{X}_n .

the density estimation is completely determined by x_i . Hence in turn, infimum of density estimation of a cell occurs at some boundary point. Therefore, if d satisfies triangle inequality, a path that connects two adjacent cells, i -th cell and j -th cell, has its infimum at $(x_i + x_j)/2$, and it is $2^m C/d^m(x_i, x_j)$, which is inversely proportional to $d^m(x_i, x_j)$. If the cells are not adjacent, then it means there exists some x_k such that $d(x_i, x_k) < d(x_i, x_j)$. Hence the hypothesis follows.

There are some issues that remain. One is that N_k 's cannot be directly used for clustering since nearest neighbor density estimation, \hat{f}_n , is not consistent. Though it is not *fully* consistent, Hartigan[?] shows that single linkage is *fractionally* consistent in that clusters at some level contains some positive fraction of true high-density clusters and approximately contain them. Note that k -nearest neighbor estimation with $k \rightarrow \infty$ and $k/n \rightarrow 0$ is consistent. Hence there is a tradeoff since when $k > 1$, efficient algorithms to compute MST is no longer useful to estimate neighborhood.

4 Experiments

The experiments here mainly compares k -means algorithm with L^2 norm in the input space, spectral method introduced in Ng et al.[?]⁸, and distance S on Prim signatures.

We start with three circles example we have used repeatedly. Figure 3 has clustering results on three circles example and two moons dataset which is a common toy example. Figure 3(a) is a duplicate copy of figure 1(a) and it is just to help comparison. As can be seen from figure 3(b), spectral clustering fails to detect concentric circles but as expected dissimilarity on Prim signature does reveal the structure in figure 3(c). Similar happens for two moons dataset as well. Plain k -means algorithm cannot detect two crescent-like structures and spectral clustering partially does but it is not perfect. Prim signature searches neighbors with single linkage and it does detect the geometric structure.

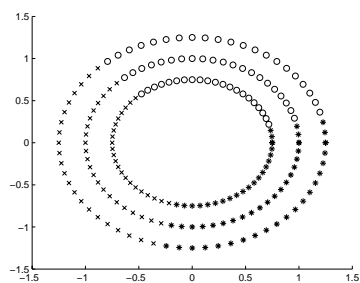
Next, we test IRIS data from UCI machine learning repository⁹. This four-dimensional data has 150 sample points and three clusters. Since four-dimensional data has difficulty to visualize, we use Rand index to compare the performance. Rand index between two labels I and J is defined as

$$R(I, J) = \frac{A + D}{A + B + C + D}$$

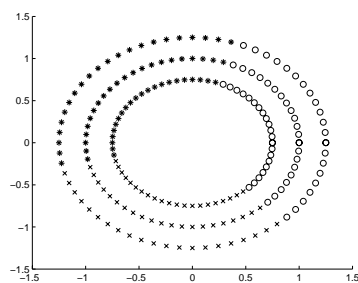
where A is the number of pairs of sample points which are in the same cluster by both I and J , B is the number of pairs which are in the same cluster by I but not by J , C is the number of pairs which are in different clusters by I but not by J , and D is the number of pairs which are in different clusters by both I and J . Performance of algorithms can be measured by computing Rand index against the actual label.

⁸Note that the technique introduced in [?] uses method similar to Eigenmap then use k -means algorithm to actually find labels.

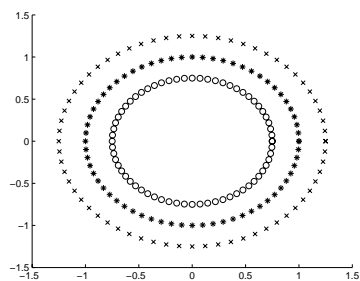
⁹<http://www.ics.uci.edu/mllearn/MLRepository.html>



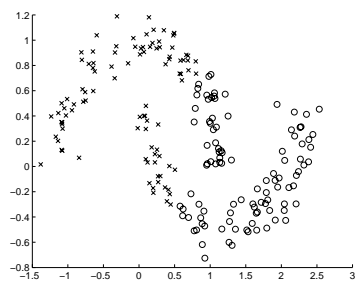
(a) Three circles clustering by k -means.



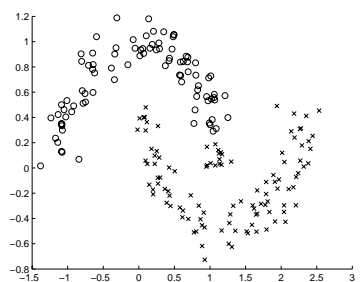
(b) Three circles clustering by spectral method.



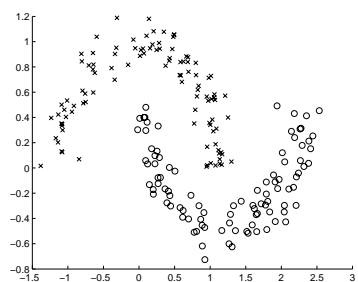
(c) Three circles clustering by Prim signature.



(d) Two moons clustering by L^2 norm.



(e) Two moons clustering by spectral method.



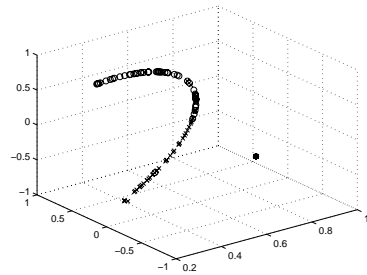
(f) Two moons clustering by Prim signature.

Figure 3: clustering results on three circles and two moons

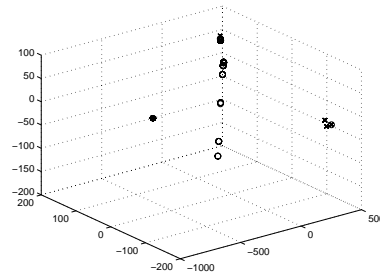
L^2 norm	spectral method	Prim signature
.8797	.8859	.9495

Table 1: Rand indices on IRIS

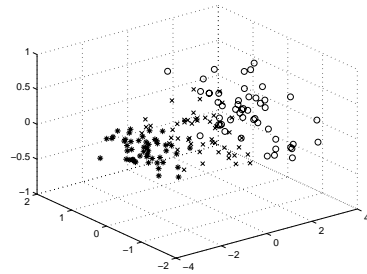
Table 1 shows the Rand index evaluation for three cases. Figure 4 visualizes how approximately k -means algorithm view IRIS dataset with various methods. Figure 4(a) shows how spectral method maps sample points. We can see that one cluster is very well separated and in fact one cluster in this dataset is known to be perfectly separated by linear classifier. However, the other two clusters are not well separated but they are mixed on an arc. Note that spectral method introduced in [?] maps sample points into \mathbb{R}^3 so figure 4 is exact. On the other hand, figure 4(b) well separates clusters and few sample points are misclustered. Note that to visualize Prim signature, MDS has been applied, and k -means run on this reduced data still achieves the same performance as in table 1. Figure 4(c) is a reference figure.



(a) IRIS data visualization by spectral method.



(b) IRIS data visualization by Prim signature. MDS was applied to visualize the data.



(c) IRIS data visualization by MDS on input space.

Figure 4: Clustering results on IRIS data. The labels in this figure are actual labels not results of algorithms.

5 Discussion and Future Works

One of issues mentioned is consistency problem. It is almost straightforward to apply k -nearest neighbor rule with $k > 1$ to our neighborhood estimation steps.

However, this increases computation time and many algorithms that find MST do not easily generalize. Hence improvement on consistency is required with suppression on the increase of computation cost.

Besides consistency, currently we encode each sample point x_i as $\{N_k(x_i)\}_{k=1}^n$, or Prim signatures. This scheme has several weaknesses. First, it is equivalent to n -dimensional vector and this dimension is often too high for data processing. Second, some information is obviously redundant. For example, once we know $\{N_k(x_i)\}_{k=1}^{n-1}$, since $N_n(x_i)$ is always equal to \mathcal{X}_n , $\phi_i^{-1}(n)$ is already known. Moreover, we already expect that $N_k(x_i)$ is not a good neighborhood estimation when k is near to n . Therefore it must be true that there is more efficient and still useful representation than our current encoding. In fact, k -means algorithm works very well in conjunction with MDS for Prim signatures.

In addition, there is still no good approach and justification how to extract information from Prim signatures in most effective way. There are some candidates mentioned in this paper: dual-rooted hitting time, arc distance, distance I , distance S , etc. And there are more candidates not mentioned yet; for example, KL divergence. However, they all do not have sufficient theoretic justification.

Some more practical issues still remain. One of them is out-of-sample extension. Most of dimensionality reduction techniques introduced here has computation issue in the sense that their computational complexities do not scale well with the number of sample points. The main reason is that these techniques compute their mappings only on given sample points but not on the whole input space. The concept of out-of-sample extension is to make approximations of the mapping based on the actual mappings computed on the sample points. Since our method is computationally expensive, out-of-sample extension will be helpful but not yet studied. Linear out-of-sample extensions for existing techniques has been addressed in Bengio et al.[?]

Lastly, though we do have some connection to existing theories like hierarchical clustering, it will be desirable to find some theoretic connections between our method to other recent popular methods like Eigenmap or LLE to study more general framework.

6 References

A Computation of MST

For many of concepts in this paper based on MST, one of the drawbacks is the computational burden. In fact, directly running Prim's algorithm on the sample for each sample point will not be a feasible means. To see this, Prim's algorithm has time complexity of $O(m \log n)$, where m is the number of edges of graph, and in our case, this time complexity is equivalent to $O(n^2 \log n)$. Even if we specialize Prim's algorithm for complete graphs¹⁰, it is still $O(n^2)$. Running this

¹⁰Most of implementations of Prim's algorithm utilize some heap structure to implement priority queue[?]. However, when given graph is known to be complete, heap structure only

for each sample point, the procedure of computing Prim signatures will have time complexity $O(n\hat{n}^2) = O(n^3)$.

Fortunately, MST has been interest to many researchers in many areas and efficient algorithm to find MST has been extensively studied. Here, we introduce an algorithm which is theoretically least computational so far, best to our knowledge. It is the technique introduced in Karger et al.[?], which proposes a randomized algorithm which has expected time complexity of $O(m)$. The main idea of [?] is to generate a tree or forest, which need not have the minimum spanning property, then find a sparser subgraph using the cycle property of MST. Since the cycle property does not eliminate the possibility of any MSTs, this idea satisfies our requirements, and the overall time complexity of procedure of computing Prim signatures can be improved down to $O(n^2 \log n)$.

One practical problem in [?] is the edge verification procedure by the cycle property. There has been suggestions in [?] and [?] but they were rather proofs of existence of linear-time procedure than practical algorithms. However, more recent work of Katriel et al.[?] suggests a practical implementable algorithm and we used this practical algorithm for our computer simulations.

Other possible way to improve the speed of overall process may be to use MST itself as the subgraph which contains MST. In other words, after the first invocation of Prim's algorithm, we feed the MST obtained from the first invocation to the following invocations. In fact, when the given graph has unique MST, MST itself is the best choice for this kind of subgraph. Furthermore, if sample points are assumed to be from some probability density function, all the pairs of sample points in \mathbb{R}^d have distinct distances almost surely, and the consequence is that \mathcal{X}_n has a unique MST.

In practice, however, many real datasets do have more than one MST. In addition, due to the finite precision in computer simulations, it is very common for a dataset of moderate size or larger generated from a distribution to have more than one MST. It is also true that verification of the assumption that sample points are from some probability density function is not a simple task. Then the validity to restrict the process in arbitrary way remains questionable.

B Triangle Inequality

There exists some kind of triangle inequality for neighborhood estimation:

$$\phi_i(x_k) \leq \phi_i(x_j) + \phi_j(x_k) \text{ for all } i, j, \text{ and } k. \quad (1)$$

This property can be used to prove some simple distances like symmetric distance ($d(x_i, x_j) = \phi_i(x_j) + \phi_j(x_i) - 2$) are actually metrics.

The proof relies on the properties of MST and assumes all pairs of sample points have distinct distances, and hence uniqueness of MST is also implied and the term MST is unambiguous without specifying the root of Prim's algorithm.

complicates the procedure without any benefit, hence needs to be removed. The priority queue is used to keep time complexity lower than $O(n^2)$ when the number of edge is $o(n^2)$.

One useful lemma is the following[?]. Let $x_i^k = \phi_i^{-1}(k)$. On MST, the maximum edge weight between x_i^k and x_i^l with $k < l$ is $\max_{m \in \{k+1, k+2, \dots, l\}} |e_i^m|$, where $|e_i^m| = d_{SL}(x_i^m, N_{m-1}(x_i))$.

Now we prove that for any $j \leq k$ and when $x_j = x_i^j$, the following holds:

$$\phi_J(x_i^k) < \phi_J(x_i^{k+1}) \quad (2)$$

First, note that the lemma shows that the maximum edge weight between x_i^k and x_i^{k+1} is $|e_i^{k+1}|$, and by the uniqueness of the edge weights, the edge is e_i^{k+1} . Let $e_i^{k+1} = (x_i^b, x_i^{k+1})$, of course $b < k+1$. Suppose inequality 2 is not true. Then in the context of x_i^j , the heaviest edge between x_i^j and x_i^k is $\max_{m \in \{\phi_J(x_i^{k+1})+1, \dots, \phi_J(x_i^k)\}} |e_J^m|$. By uniqueness of the edge weights, this implies $\phi_J(x_i^{k+1}) < \phi_J(x_i^b) \leq \phi_J(x_i^k)$.

Note that at each iteration of Prim's algorithm, the algorithm maintains a connected subgraph. Hence in the context of x_i , the path between x_i^b and x_i^j must not have x_i^{k+1} but in the context of x_i^j , it does. Therefore we have contradiction and the inequality 2 must be true.

Then inequality 1 can be easily proven. Inequality 2 means that for a run of Prim's algorithm rooted at x_i^j to reach x_i^k , the procedure must include all of x_i^m for $m \in \{j+1, j+2, \dots, m\}$. Hence inequality 1 is true.