Stony Brook University

A Scalable Physics-based Data Modeling Framework to Unsupervised High-Dimensional Data Mining

A Dissertation Presented
by
Hao Huang

to
The Graduate School
in Partial Fulfillment of the
Requirements
for the Degree of
Doctor of Philosophy
in
Computer Science
Stony Brook University

December 2014

Stony Brook University
The Graduate School

Hao Huang

We, the dissertation committee for the above candidate for the
Doctor of Philosophy degree, hereby recommend
acceptance of this dissertation.

Hong Qin, Dissertation Advisor
Professor, Department of Computer Science, SBU

Luis E. Ortiz, Chairperson of Defense
Assistant Professor, Department of Computer Science, SBU

Dantong Yu, Dissertation Co-Advisor
Adjunct Professor, Department of Electrical and Computer Engineering, SBU

Wei Zhu
Professor, Department of Applied Mathematics & Statistics, SBU

Jing Hua, Outside Member
Professor, Department of Computer Science, Wayne State University

This dissertation is accepted by the Graduate School

Charles Taber
Dean of the Graduate School

Abstract of the dissertation

A Scalable Physics-based Data Modeling Framework to Unsupervised High-Dimensional Data Mining

by

Hao Huang

Doctor of Philosophy

in

Computer Science

Stony Brook University

2014

Abstract:

Today's modeling and analysis of high-dimensional data is either based on human expertise to hand-craft a set of task-specific data, which suffers significantly from the ever-increasing complexity and the unknown patterns of the new data; or is based on simple data-driven approaches which tend to lose the fundamentally physical insights of real world datasets. Therefore, it is very difficult with today's modeling practice to efficiently, effectively, and unsupervisedly detect reliable patterns and information in high-dimensional data. In this dissertation, we developed a scalable data modeling framework that utilizes modern theoretical physics for unsupervised high-dimensional data analysis and mining. Not only does it have a solid theoretical background, but it is capable of handling different tasks with different capability (clustering, anomaly detection and feature selections, etc.). This framework also has probabilistic interpretation that avoids the sensitivity from scaling parameter tuning or noise appearance in real world applications. Furthermore, we presented a fast approximated approach to make such a framework applicable on large-scale datasets with high efficiency and effectiveness.

During my dissertation research, we made the following salient contributions:

1. We proposed a diffusion-based **Aggregated Heat Kernel (AHK)** to improve the clustering stability, and a **Local Density Affinity Transformation (L-DAT)** to correct the bias originated from different cluster densities. Our proposed framework integrates these two techniques systematically. As a result, it not only provides an advanced noise-resisting and density-aware spectral mapping to the original datasets, but also demonstrates the clustering stability during the process of tuning the scaling parameters.

2. We devised a **Local Anomaly Descriptor (LAD)** that faithfully reveals the intrinsic neighborhood density to detect anomalies. LAD bridges global and local properties, which makes it self-adaptive with different samples' neighborhood. To offer better stability of local density measurement on scaling parameter tuning, we formulated a **Fermi Density Descriptor (FDD)**. FDD steadily distinguishes anomalies from normal instances with most of the scaling parameter settings. We also quantified and examined the effect of different Laplacian normalizations with the purpose of detecting anomalies.

3. We developed a robust feature selection algorithm, called **Noise-Resistant Unsupervised Feature Selection (NRFS)**. It measures multi-perspective correlation that reflects the importance of features with respect to noise-resistant instance representatives and different global trends from spectral decomposition. In this way, the model concisely captures a wide variety of local patterns, and selects representative features with high quality.

4. We mitigated the space and time complexity of spectral embedding in order to apply the above techniques to real-world large data mining, by proposing a **Diverse Power Iteration Embedding (DPIE)**. We tested DPIE on various applications (e.g., clustering, anomaly detection and feature selection). The experimental results showed that our proposed DPIE is more effective than popular spectral approximation methods, and even obtains the similar quality of classic spectral embedding derived from a classic eigen-decompositions. Moreover, DPIE is extremely fast on big data applications.

Finally, we provided a brief introduction of our on-going work and future research directions. By elaborating our developed works within the proposed framework, we showed that our scalable physic-based unsupervised data modeling is potent and promising for large-scale and high-dimensional data analysis, data mining, and knowledge discovery. It is a rich and fruitful area for research in terms of both theory and applications.

For my beloved family, advisors, friends and colleagues.

Contents

List of Algorithms

List of Tables

List of Figures

Acknowledgements

First and above all, I praise God, the almighty for providing me this opportunity and granting me the capability to proceed successfully.

I feel very blessed to have the opportunity to study at Stony Brook University. During these years, I received tremendous help and support from many individuals, to whom I want to express my sincere thanks here.

I would like to express my deepest gratitude to my esteemed advisor, Prof. Hong Qin, for accepting me as a Ph.D student. Your warm encouragement, thoughtful guidance and critical comments, provided me an excellent atmosphere for doing research.

My unfathomable thanks also go to my co-advisor, Prof. Dantong Yu, for the trust, insightful advices, and your support during the whole period of the study, and especially for your patience and guidance during the writing process.

Special thanks to my other committee members. Prof. Wei Zhu served in my committees from research proficiency exam to dissertation defense. I appreciate your constructive suggestions and comments very much. I would also like to thank Prof. Luis E. Ortiz and Prof. Jing Hua for serving in my defense committee, and providing your valuable advices and support. Especially Prof. Jing Hua came all the way from Detroit to Stony Brook to serve as the external member of my committee.

I would like to extend my special thanks to Dr. Shinjae Yoo. As an excellent colleague and a dear friend, you helped my papers and dissertation with a lot of corrections and insights.

During my Ph.D study, I feel honored to work with a collection of colleagues at Stony Brook University and Brookhaven National Laboratory. This includes Shuchu Han, Zhenzhou Peng, Jin Xu, Yufei Ren, Tan Li, Li Shi, Shun Yao, Cheng Chang, Dr. Yan Li and Dr. Dong Huang, etc. I want to show my grateful thanks

to all of you, for all the joyful gathering and sincere support. I could not imagine a better work environment than I've found with all of you.

And finally, great thanks to my dear wife, parents and parents-in-law, without your support and encouragement, I couldn't have finished this work. And what I can just say is: You make me feel so blessed to have you in my life, and thank you, for everything.

Publications

Published Papers:

- **Hao Huang**, Shinjae Yoo, Dantong Yu, and Hong Qin, "Physics-based Anomaly Detection Defined on Manifold Space", ACM Transactions on Knowledge Discovery from Data 2014, 9(2) (ACM TKDD 2014).

- **Hao Huang**, Shinjae Yoo, Dantong Yu and Hong Qin, "Diverse Power Iteration Embeddings and Its Applications", accepted by IEEE International Conference on Data Mining 2014 (IEEE ICDM 2014, **acceptance rate**: $\mathbf{9.7}\%$).

- **Hao Huang**, Shinjae Yoo, Dantong Yu and Hong Qin, "Noise-Resistant Unsupervised Feature Selection via Multi-Perspective Correlations", accepted by IEEE International Conference on Data Mining 2014 (IEEE ICDM 2014, **acceptance rate**: $\mathbf{9.7}\%$).

- **Hao Huang**, Shinjae Yoo, K. Kaznatcheev, K. G. Yager, F. Lu, Dantong Yu, O. Gang, A. Fluerasu and Hong Qin, "Diffusion-based Clustering Analysis of Coherent X-ray Scattering Patterns of Self-assembled Nanoparticles", in Proceedings of the 29th Symposium on Applied Computing 2014 (ACM SAC 2014, Data Mining track, **acceptance rate**: $\mathbf{23.2}\%$).

- **Hao Huang**, J. Xu, Z. Peng, Shinjae Yoo, Dantong Yu, D. Huang and Hong Qin, "Cloud Motion Estimation for Short Term Solar Irradiation Prediction", in Proceedings of IEEE International Conference on Smart Grid Communications 2013 (IEEE SmartGridComm 2013).

- **Hao Huang**, Hong Qin, Shinjae Yoo and Dantong Yu, "A New Anomaly Detection Algorithm based on Quantum Mechanics", in Proceedings of IEEE

International Conference on Data Mining 2012 (IEEE ICDM 2012, **acceptance rate**: $\mathbf{19.9}\%$).

- **Hao Huang**, Hong Qin, Shinjae Yoo and Dantong Yu, "Local Anomaly Descriptor: A Robust Unsupervised Algorithm for Anomaly Detection based on Diffusion Space", in Proceedings of ACM Conference on Information and Knowledge Management 2012 (ACM CIKM 2012, **acceptance rate**: $\mathbf{13.4}\%$).

- **Hao Huang**, Shinjae Yoo, Dantong Yu, D. Huang and Hong Qin, "Correlation and Local Feature based Cloud Motion Estimation", in KDD Multimedia Data Mining workshop 2012 (KDD/MDM 2012).

- **Hao Huang**, Shinjae Yoo, Hong Qin and Dantong Yu, "A Robust Clustering Algorithm based on Aggregated Heat Kernel Mapping", in Proceedings of IEEE International Conference on Data Mining 2011 (IEEE ICDM 2011, **acceptance rate**: $\mathbf{12.2}\%$).

- **Hao Huang**, Shinjae Yoo, Dantong Yu, D. Huang and Hong Qin, "Cloud Motion Detection for Short Term Solar Power Prediction", in ICML Workshop on Machine Learning for Global Challenges 2011.

Submitted Papers:

- **Hao Huang**, Shinjae Yoo, Dantong Yu and Hong Qin, "Density-aware Clustering based on Aggregated Heat Kernel and Its Transformation", submitted to ACM Transactions on Knowledge Discovery from Data, under second round revision.

- Shuchu Han, **Hao Huang**, Hong Qin and Dantong Yu, "Locality-Preserving L1-Graph and Its Application in Clustering", submitted to the 30th ACM/SIGAPP Symposium On Applied Computing 2015 (ACM SAC 2015).

Chapter 1

Introduction

1.1 Problem Statement and Motivations

The last decade has brought a large amount of high-dimensional data collected in business and scientific area. The related databases and information sources are available through advanced devices covering different dynamic domains: atmosphere, medicine, biology, and social network, etc. These data are high-dimensional, large-scale, nonuniformly distributed, and with unprecedented patterns. Such fact arises one of the greatest challenges in data mining and knowledge discovery: to efficiently and unsupervisedly discover unprecedented pattern within the large-scale and high-dimensional datasets, and especially, to **design fast, unsupervised and robust data modeling framework with ability from global pattern mining to local feature detection**.

The importance and difficulty of this unsupervised data modeling is leading to the fundamental study increasing of multi-scale manifold learning, from the basic geometric characteristics such as curvature [105] and geodesics distances [104], to intrinsic topological structure [10] [18] [38] [39] [52] [136], and other particular features [6].

A high-dimensional dataset can be represented as a collection of high-dimensional points in R^m, where m is the number of dimensions or features. Three problems with different perspectives about such kind of dataset have been under spotlight for the last decades: clustering, anomaly detection, and feature selection.

Clustering, or cluster analysis, is the unsupervised task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters). In other word, clustering is to automate the extraction of global representative patterns, and to label each object accordingly. On the other hand, **anomaly detection** (or outlier detection) is the identification of items, events or observations which do not conform to an expected pattern or other items in a dataset. Typically, the anomaly detection can be translated to the problem of deciding which data points show special or suspicious distribution patterns compared with the other points. **Feature selection**, also known as attribute selection, is the process of selecting a subset of relevant features for use in model construction. Its central assumption is that the data contains many redundant or irrelevant features. Redundant features are those which provide no more information than the currently selected features, and irrelevant features provide no useful information in any context.

Towards these three directions respectively, researchers in recent years have explored various techniques. However, given less and even no prior knowledge of the unseen patterns and noise appearance in real world applications, researches about unsupervised and robust data modeling considering clustering, anomaly detection and feature selection in the same framework are rather rare. On the other hand, **high efficiency and effectiveness** of any proposed methodology and framework are both indispensable and essential on real world large-scale applications.

Our work is inspired by a physics-based diffusion theory, since it has built-in multi-scale property and strong probabilistic interpretation. Diffusion theory has been one of the research hot topics in the last decade, because it has robustness against data perturbation and the curse of high dimensionality. Diffusion distance is based on Markov matrix, which is a stochastic representation of random walk on a graph [103]. It can consider up to t steps out of all the possible paths bridging any two points, which makes it much more comprehensive than other measurements such as geodesic distance [27]. In other word, diffusion distance has a potential to be more stable to data perturbation via a family of diffusion maps [28].

One promising option among all the diffusion techniques is heat diffusion [56] [23] [68]. Heat diffusion describes the exchange of thermal energy between physical systems by dissipating heat. Heat equation, a formula describing the process

of heat diffusion, can be naturally used to build multi-scale representation on manifolds and graphs. One advantage of heat diffusion is that it integrates both the topological information of manifold and the multi-scale property of scatter data together. Another advantage is that it provides robust representation of high-dimensional datasets by computing the average probability from the diffusion process. In the next few sections, we explore the utility of heat diffusion on data analysis.

1.2 Research Challenges

Data modeling is a process that lies in between data collection and subsequent analysis tasks such as model optimization [119]. Due to different applications, it can be classified into varying types. Different researches precede the applications in conventional machine learning domain, such as stochastic state space planning [119], clustering [109], anomaly detection [19] and feature selection [44] [123]. In this dissertation, the overall theme of our research is to efficiently, stably and unsupervisedly model data within different perspectives and applications. Challenges that our research faced include the following aspects.

- **Unsupervised Data Modeling for Diverse Applications**
  Unsupervised data modeling raised in the domains of not only machine learning, but also computer vision and medical imaging. With more and more massive scientific datasets having unprecedented patterns from new scientific domains, we need to detect the useful and new patterns without any prior knowledge. According to different applications, information would be in demand on different perspectives. However, with the exiting techniques, the problems of unsupervised data modeling that can handle different applications can be quite difficult: when we concentrate on global data distribution with the purpose of clustering, the embedded structure must be invariant to local data perturbation by covering comprehensive global information; local anomaly detection, on the other hand, should describe local density which is intrinsic and informative on its adaptive or even limited neighborhood. In a common way, both of the global and local patterns are determined only by visible neighborhood while avoiding negative effects such as noise appearance, over

diffusion (with too large scaling parameters), or losing intra-connection inside clusters (with too small scaling parameters). In our research, we introduced an unsupervised framework for high-dimensional data modeling, where basis functions emerge from the dilatory actions of a diffusion operator on the graph, such as random walk, and functions over parameter space are progressively remapped into low-rank-frequency atoms, with parameters to control different scopes.

- **Robust Data Modeling and Analysis**
  Data modeling and analysis is of little use if the results are radically-different when the scaling parameters of algorithms are slightly modified or even with very little noise perturbation. We call such susceptibility the sensitivity of algorithms, and one of the most desirable properties of data modeling is robustness. In our research, we introduced a few robust algorithms with the following advantages: (1) not sensitive to any small change of parameters; (2) not sensitive to data perturbation; (3) non-degraded performance even with significant noise level or less-correct parameter settings; and (4) competitive and comparable results when comparing with those less-robust algorithms without any data perturbation and with correct parameter settings. With these robustness properties, we can reliably analyze data and conduct other data-driven tasks in succeeding analysis steps. The robustness property is equally significant for domain experts who do not have strong machine learning background as they become much more comfortable in utilizing robust algorithms. In short, we propose robust algorithms with comparably stable performance under noise perturbation and parameter tuning.

- **Density-sensitive Clustering Algorithms**
  Many clustering algorithms assume uniform or similar density distribution among different clusters [100], and this assumption does not always reflect the real world data distribution. To alleviate this assumption, previous works used the local density information by incorporating a local scaling parameter into the kernel functions [156] [160] or approximating the local density [153] [30]. However, problems still exist because either the parameters are sensitive to heterogeneous density distributions [156] [30], or such algorithms

could not precisely reflect local density [160] [153]. In our work, we proposed a novel affinity transformation to correct the bias from different density distribution effects across clusters in affinity matrix.

- **Universally Applicable Algorithms on Various Kernel Functions**
  Most of the current existing algorithms which target on the above problems could only be applicable to a small number of applications, e.g. some measurement only works on Euclidean space. In our research we designed a generally applicable framework by proposing methods only based on general affinity matrix instead of certain space or kernels, and such methods work well with any type of kernel functions. Such a feature distinguishes our work from other approaches in handling more diverse and complex real world problems.

- **Multi-Perspective Feature Selection**
  Feature importance is usually more about a "local" conception than a "global" one. To obtain a better representative feature subsets, the feature impact to different low-embeddings or spectrums need to be considered [35]. Besides, the view of instances is also indispensable since some features may only have strong correlation with certain instances with respect to certain spectrums. Therefore it is necessary to design a feature selection algorithm built upon multi-perspective correlations. Our proposed algorithm selects features under local context instead of global context, therefore it has a local view from both the instance and feature perspectives, and measure their local correlations with the global spectrums. Therefore, it provides a more informative feature selection strategy.

- **Efficient Approximation of Spectral Embeddings**
  One foundation of our research framework is spectral embedding construction, which is one of the most effective dimension reduction algorithms in machine learning and data mining [100]. However, its associated high complexity in both time $O(n^3)$ and space $O(n^2)$ prevents it from practical utilization in many large-scale real-world applications. Many researches have developed a few approximate spectral embeddings which are more efficient,

but meanwhile far less effective. In this research we introduced an efficient and effective approximation to spectral embedding. So that our proposed techniques can be easily applied to the large-scale real-world applications with desired quality.

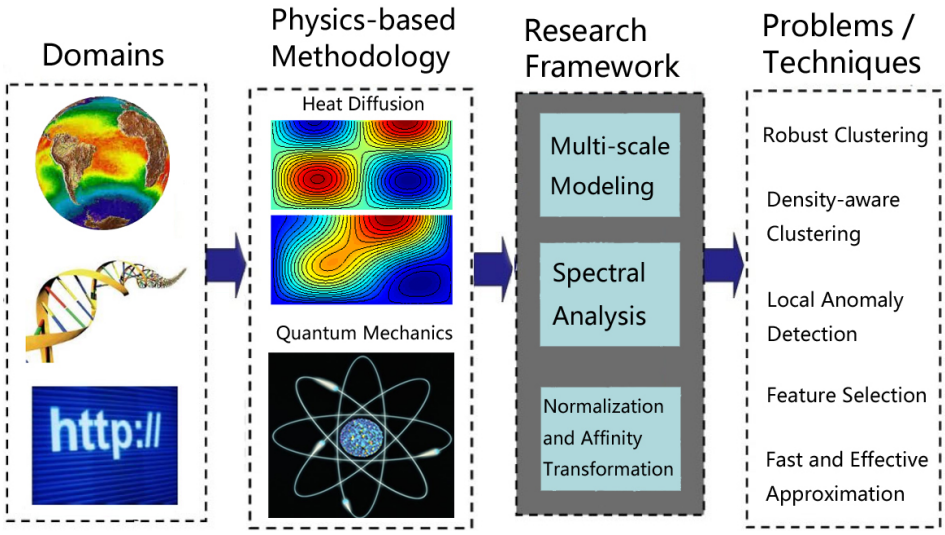


Figure 1.1: Our proposed physics-based unsupervised data modeling framework and the derived techniques.

1.3 Research Contributions

In this dissertation, we present a scalable physics-based data modeling framework (see Figure 1.1) for unsupervised high-dimensional applications. Particularly speaking, our contributions in this research include:

- **Density-Aware Clustering based on Aggregated Heat Kernel and Its Transformation**
  We proposed a diffusion-based Aggregated Heat Kernel (AHK) to improve the clustering stability, and a Local Density Affinity Transformation (LDAT) to correct the bias originating from different cluster densities. AHK aggregately models the heat diffusion traces along all the time scales, so it ensures

robustness during clustering process, while LDAT probabilistically reveals local density of each instance and suppresses the local density bias in the affinity matrix. Our proposed framework integrates these two techniques systematically. As a result, not only does it provide a noise-resisting and density-aware spectral mapping to the original dataset, but also demonstrates the stability during the process of tuning the scaling parameters (which usually control the range of neighborhood). Furthermore, our framework works well with the majority of similarity kernels, which ensures its applicability to many types of data and problem domains. The systematic experiments on different applications showed that our proposed algorithms outperform state-of-the-art clustering algorithms, and achieve robust clustering performance with respect to tuning the scaling parameter and handling various levels of noise. This work is documented in Chapter 3. Our publications related to this work include:

  - **Hao Huang**, Shinjae Yoo, Hong Qin and Dantong Yu, "A Robust Clustering Algorithm based on Aggregated Heat Kernel Mapping", in Proceedings of IEEE International Conference on Data Mining 2011 (IEEE ICDM 2011, **acceptance rate**: $\mathbf{12.2}\%$).
  - **Hao Huang**, Shinjae Yoo, K. Kaznatcheev, K. G. Yager, F. Lu, Dantong Yu, O. Gang, A. Fluerasu and Hong Qin, "Diffusion-based Clustering Analysis of Coherent X-ray Scattering Patterns of Self-assembled Nanoparticles", in Proceedings of the 29th Symposium on Applied Computing 2014 (ACM SAC 2014, Data Mining track, **acceptance rate**: $\mathbf{23.2}\%$).
  - **Hao Huang**, Shinjae Yoo, Dantong Yu and Hong Qin, "Density-aware Clustering based on Aggregated Heat Kernel and Its Transformation", submitted to ACM Transactions on Knowledge Discovery from Data, under second round revision.

- **Physics-based Anomaly Detection Defined on Manifold Space**
  We proposed two unsupervised anomaly detection algorithms through different theoretical physics domain. Building upon the embedding manifold derived from heat diffusion, we devised Local Anomaly Descriptor (LAD)

which faithfully reveals the intrinsic neighborhood density. It uses a scale-dependent umbrella operator to bridge global and local properties, which makes LAD more comprehensive within an adaptive scope of neighborhood. To offer more stability of local density measurement on scaling parameter tuning, we formulated Fermi Density Descriptor (FDD) which measures the probability of a fermion particle being at a specific location, that corresponds to the neighborhood density. By choosing the stable energy distribution function, FDD steadily distinguishes anomalies from normal instances with most of the scaling parameter settings. To further enhance the efficacy of our proposed algorithms, we explored the utility of Anisotropic Gaussian Kernel (AGK) which offers better manifold-aware affinity information. We also quantified and examined the effect of different Laplacian normalizations for the purpose of anomaly detection. Comprehensive experiments on both synthetic and benchmark datasets verified that our proposed algorithms outperform the existing anomaly detection algorithms. This work is recorded in Chapter 4. The related publications include:

  - **Hao Huang**, Hong Qin, Shinjae Yoo and Dantong Yu, "Local Anomaly Descriptor: A Robust Unsupervised Algorithm for Anomaly Detection based on Diffusion Space", in Proceedings of ACM Conference on Information and Knowledge Management 2012 (ACM CIKM 2012, **acceptance rate**: $\mathbf{13}.4\%$).
  - **Hao Huang**, Hong Qin, Shinjae Yoo and Dantong Yu, "A New Anomaly Detection Algorithm based on Quantum Mechanics", in Proceedings of IEEE International Conference on Data Mining 2012 (IEEE ICDM 2012, **acceptance rate**: $\mathbf{19}.9\%$).
  - **Hao Huang**, Shinjae Yoo, Dantong Yu, and Hong Qin, "Physics-based Anomaly Detection Defined on Manifold Space", ACM Transactions on Knowledge Discovery from Data 2014, 9(2) (ACM TKDD 2014).

- **Noise-Resistant Unsupervised Feature Selection via Multi-Perspective Correlations**

  We designed an advanced feature selection strategy, called Noise-Resistant

Feature Selection (NRFS), based on multi-perspective correlation computation that is effective and robust to both noisy observations and features. By selecting representative instances via density distribution statistics, we reduced the occurrence of the noisy observations. For each feature, we computed its local correlation with each instance and each global spectrum (or trend) of data to find the most informative features. Noisy features tend to have lower local associations with all the the global spectrums and representative instances compared with the informative ones, while the locally informative features show strong associations to at least one global spectrum and some representative instances. We thoroughly considered all the correlation scores from different perspectives to obtain the comprehensive and yet non-redundant feature subset. We introduce this work in Chapter 5. The related publication is:

  - **Hao Huang**, Shinjae Yoo, Dantong Yu and Hong Qin, "Noise-Resistant Unsupervised Feature Selection via Multi-Perspective Correlations", accepted by IEEE International Conference on Data Mining 2014 (IEEE ICDM 2014, **acceptance rate**: $\mathbf{9.7}\%$).

- **Diverse Power Iteration Embeddings and Its Applications**
  To resolve the impracticality of spectral embedding due to its computational complexity, and at the same time maintain its effectiveness, we proposed a novel power-iteration-based method, called Diverse Power Iteration Embeddings (DPIE). DPIE aims to find diverse and yet informative low dimensional embeddings, which is different from the single or very-close embedding vectors from previous power iteration methods. In theory, our proposed DPIE has the same or similar representational power of classic spectral embeddings, so that it can be applicable to various spectral analysis. Compared with the existing spectral embedding approximations, DPIE achieves a similar or even lower time and space computational complexity, but a more desired quality. We systematically evaluated DPIE on a number of important applications (e.g. clustering, anomaly detection, and feature selection). The results confirmed that our proposed DPIE significantly outperforms other existing algorithms in terms of both effectiveness and efficiency. The work is described in

Chapter 6 in details and the related publication is:

- **Hao Huang**, Shinjae Yoo, Dantong Yu and Hong Qin, "Diverse Power Iteration Embeddings and Its Applications", accepted by IEEE International Conference on Data Mining 2014 (IEEE ICDM 2014, **acceptance rate**: $\mathbf{9.7}\%$).

1.4 Dissertation Organization

The remainder of this dissertation is organized in the following fashion. In Chapter 2, we begin with background theory review and analyze their properties. In Chapter 3, 4, 5, 6 we introduce our major works. In Chapter 3, we present robust clustering methods with affinity transformation for heterogeneous density clusters that against scaling parameter tuning and noise sensitivity. Chapter 4 introduces two unsupervised anomaly detection algorithms: one is a heat-diffusion-based anomaly detection, another is a quantum-mechanics-based algorithm with strong probabilistic interpretation. Chapter 5 presents a noise-resistant unsupervised feature selection algorithm based on multi-perspective correlation measurement. In Chapter 6 we describe a spectral embedding approximation, which is both efficient and effective for large-scale datasets. Therefore it makes our framework practical in real world applications. Finally, Chapter 7 summarizes our contributions on the finished work and also discusses our ongoing work and future research directions.

Chapter 2

Background Theory

Our framework is based upon Laplace operators, spectral analysis and heat diffusion. In this chapter we briefly review the basic ideas of related techniques and analyze their properties.

2.1 Graph Laplacians

Laplace operator, when it is applied on spectral analysis and methodology, is usually called graph Laplacian. Here we introduce the classic unnormalized and normalized graph Laplacians on finite weighted graphs.

We denote $X \in R^{n \times m}$ as a dataset with n instances, each instance has m features. Its similarity (or affinity) matrix $W \in R^{n \times n}$ represents the pair-wise likeness of instances considering the whole feature space. The degree matrix $D \in R^{n \times n}$ is defined by $D(i,j) = \sum_{p=1}^{n} W(i,p)$ if $i = j$, and 0 otherwise. Then the unnormalized Laplacian matrix $L_{nn} \in R^{n \times n}$ can be defined as:

$$L_{nn} = D - W, \tag{2.1}$$

which is the difference between the degree matrix D and the similarity matrix W of the graph. The nice properties of L_{nn} has been discussed in [100]. One of the most important ones is that L_{nn} has as many eigenvalues 0 as there are connected components, and the corresponding eigenvectors are the indicator vectors of the connected components.

There are two common ways of normalizing L_{nn} to correct its bias of different density [100][28], one is the symmetric normalized Laplacian matrix $L_{sym} \in R^{n\times n}$, and the other is random walk normalized Laplacian matrix $L_{rw} \in R^{n\times n}$:

$$L_{sym} = D^{-1/2} L_{nn} D^{-1/2}. \tag{2.2}$$

$$L_{rw} = D^{-1} L_{nn}. \tag{2.3}$$

The matrix L_{sym} has the advantage of being symmetric therefore it has a more balance view in the instance neighborhood, while L_{rw} is a stochastic matrix which can be viewed as the transition matrix of a Markov chain on each instance.

To better depict the global distribution, Coifman et.al [28] analyzed these two normalization and proposed a new normalization family. It is shown in [28] that if we assume uniform sampling of data points from a sub-manifold $\mathcal{M}$, the eigenvectors of L_{rw} with $\sigma \to 0$ and $n \to \infty$, tend to approximate Laplace-Beltrami operator on $\mathcal{M}$, which guarantees manifold structure reconstruction. However, in reality, the sampled data points tend to be nonuniform and show skewed density distributions, resulting in poor manifold structure reconstruction. To improve the global distributional sensitivity of traditional normalization, the following two additional normalizations are considered in [28]:

$$L_{fp} = I - D^{-1} W^{'}, \tag{2.4}$$

where $W^{'} = D^{-1/2} W D^{-1/2}$, and

$$L_{lbn} = I - D^{-1} W^{''}, \tag{2.5}$$

where $W^{''} = D^{-1} W D^{-1}$. $L_{fp} \in R^{n\times n}$ is called Fokker-Planck normalization (FP), and $L_{lbn} \in R^{n\times n}$ is called Laplace-Beltrami normalization (LBN). Especially, LBN can remove the influence of the dataset density and recovers manifold structures on $\mathcal{M}$ with the condition of both $\sigma \to 0$ and $n \to \infty$ [28]. In other words, the additional re-normalization of affinity matrix W enables to reconstruct manifold structures better under non-uniform density distribution for the purpose of clustering.

Graph Laplacian has been closely integrated into spectral analysis (Section 2.2). From any of the aforementioned L_{**}, we can obtain the corresponding eigenvectors. The first c ($c \ll m$) non-trivial eigenvectors with the smallest eigenvalues

(except 0) are the most important signal components, which in theory form the manifold structure of X [100]. Denote these c eigenvectors as $\Psi \in R^{n \times c}$. Each row of Ψ is the corresponding coordinates of each original instance in the manifold space, while each column of Ψ (eigenvectors) represents an axis (dimension) in the manifold space. These eigenvectors are orthogonal to each other and together provide the compressed and embedding representation of the low-rank distribution of the dataset.

As far as we know, there is no other research focus on the effect of different Laplacians on clustering and anomaly detection within the same framework. In this dissertation we analyze this problem with the modeling techniques we proposed.

2.2 Spectral Embeddings and Clustering

Spectral analysis already gained increasing popularity in the last decade because of its ability to discover embedding data structure. It has a strong connection with graph cut, i.e., it uses eigenspace to solve a relaxed form of the balanced graph partitioning problem [109]. Its second desirable aspect is that, with nonlinear kernels it can capture the nonlinear structure of data, which is difficult for k-means [60] or other linear clustering algorithms.

ALGORITHM 1: SpectralClustering(X, c)

Input: $X \in R^{n \times m}$ where n is #instances, m is #features, and c is #clusters.
Output: Cluster assignments of n instances.

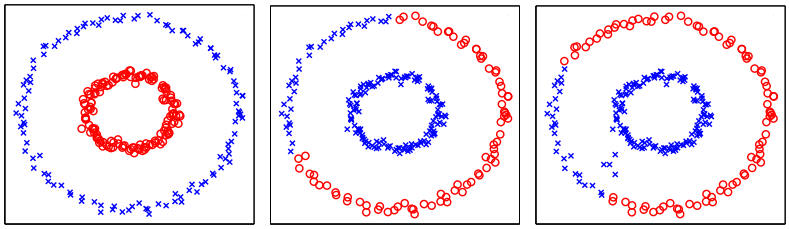
1 Construct the affinity matrix $W \in R^{n \times n}$;
2 Compute the diagonal matrix $D \in R^{n \times n}$ where $D(i,i) = \sum_{j=1}^{n} W(i,j)$ and $D(i,j) = 0$ if $i \neq j$;
3 Apply the graph Laplacian $L \in R^{n \times n}$ using $L_{nn} = D - W$, $L_{rw} = I - D^{-1}W$ or $L_{sym} = I - D^{-1/2}WD^{-1/2}$ where $I \in R^{n \times n}$ is an identity matrix ;
4 Extract the first c nontrivial eigenvectors Ψ of L, $\Psi = \{\psi_1, \psi_2, \ldots, \psi_c\}$;
5 Re-normalize the rows of $\Psi \in R^{n \times c}$ into $Y_i(j) = \psi_i(j)/(\sum_l \psi_i(l)^2)^{1/2}$;
6 Run k-means with c and $Y \in R^{n \times c}$.

Spectral clustering, as shown in Algorithm 1, usually starts with local information encoded in a weighted graph which is constructed from certain similarity

kernels on input data, and clusters according to the global eigenvectors of the corresponding (normalized) affinity matrix. However, it has a few limitations as follows:

- The selection of the scaling parameter (if any) of similarity kernel could affect the clustering results radically (as shown in Figure 2.1(a) and 2.1(b)) because the scaling parameter usually determines each instance's neighborhood scope.
- The clustering results are sensitive to noise. For instance in Figure 2.1(c), with only a few noisy instances, the clustering result is quite different and the optimal range of scaling parameter also varies.
- The reconstructed embedding structures may fail to represent the diversity of density across clusters, which leads to clustering results with a poor quality (as shown in Figure 2.2).

The above problems are partly due to the fact that the similarity kernels (or affinity matrix construction) used in the spectral clustering are sensitive to the parameter scaling and noise appearance [156]. In Section 2.3, we will describe some popularly used similarity kernels. In Section 3.3 we will propose a robust clustering algorithm against parameter and noise sensitivity.

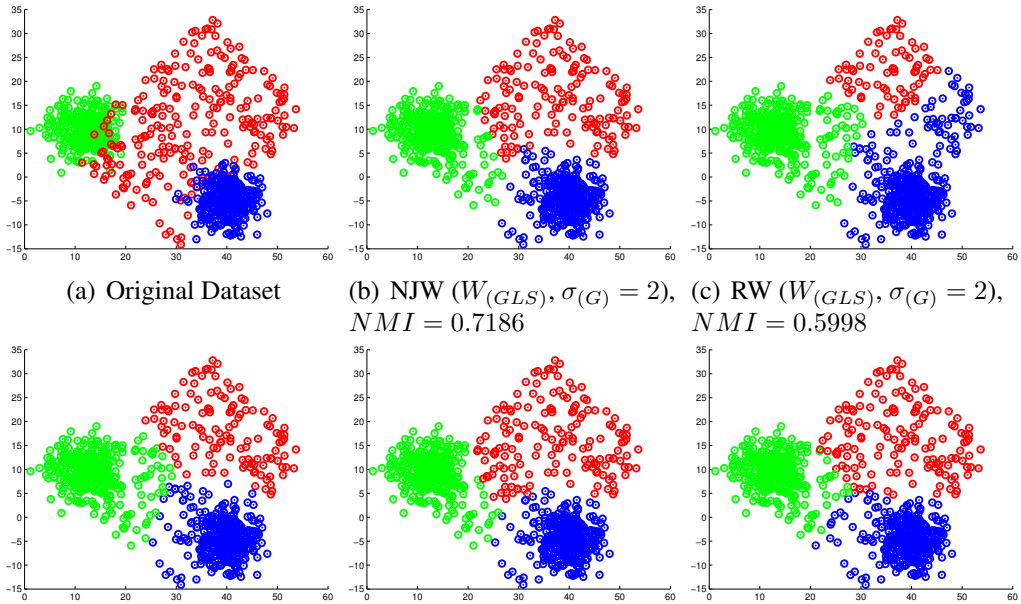


(a) $\sigma = 0.22$ (b) $\sigma = 0.23$ (c) $\sigma = 0.22$ with noises

Figure 2.1: The sensitivity example of NJW [109], one of the traditional spectral clustering algorithms, with respect to different Gaussian scaling parameter σ and noise appearance. The two output clusters are colored with red or blue. A small variation to σ or data points (noise) leads to radically-different results. Such an instability becomes an issue to traditional spectral clustering algorithms.

The second reason for the aforementioned problems is that (normalized) affinity matrix cannot take the local density information into consideration, in particular

for those data points between two clusters with heterogeneous densities. A synthetic example of such problem is demonstrated in Figure 2.2. Section 3.4 will introduce a simple and effective way of correcting the local density bias.



(a) Original Dataset

(b) NJW ($W_{(GLS)}$, $\sigma_{(G)} = 2$), $NMI = 0.7186$

(c) RW ($W_{(GLS)}$, $\sigma_{(G)} = 2$), $NMI = 0.5998$

(d) NN ($W_{(GLS)}$, $\sigma_{(G)} = 2$), $NMI = 0.6731$

(e) ST ($W_{(LCS)}$, $q = 3$ in $\sigma_{(L_{i,p})}$), $NMI = 0.7333$

(f) SCDA ($W_{(GLS)}$, $\sigma_{(G)} = 2$), $NMI = 0.7126$

Figure 2.2: Clustering results of different algorithms on a synthetic dataset with heterogeneous density distributions. Figure 2.2(a) shows the original dataset, where the green and blue clusters with Gaussian distributions have higher density than the red cluster with a uniform distribution. The clustering results of NJW (Figure 2.2(b)), RWC (Figure 2.2(c)) and NN (Figure 2.2(d)) are shown respectively, which are not capable of capturing the density variation. For the localized method, ST (Figure 2.2(e)) has better result since it has a locally adaptive scaling parameter (in Equation 2.9), while SCDA (Figure 2.2(f)) reveals a similar density-awareness as NJW. In short, none of the above methods provides a desirable separation that is aware of both density change and manifold structures across clusters.

2.3 Affinity Matrix Construction

Affinity matrix construction (the construction of W) is the first step of spectral analysis with significant influence to the final results. In practice it is derived from certain similarity kernels. How to choose an appropriate similarity kernel is a critical step in spectral analysis as different types of datasets might have different preference for similarity measurements. There are several popular ways to measure the similarity $W(i,j)$ between any two instances $x(i)$ and $x(j)$. Here we focus on introducing three of them: network connectivity, cosine similarity, and Gaussian kernel.

Network Connectivity. Network datasets such as social networks, computer networks and biological networks, define affinity matrix based on their dataset representations, which are often modeled as undirected graphs. Each edge has a weight to describe the relationship between the two related nodes representing data instances in the dataset. In a network dataset the simplest edge weight is usually defined as the connectivity between two nodes. The simplest network connectivity $W_{(NET)}(i,j)$ can be defined as:

$$W_{(NET)}(i,j) = \begin{cases} 1, & \text{if } x(i) \text{ and } x(j) \text{ are connected,} \\ 0, & \text{if } x(i) \text{ and } x(j) \text{ are unconnected.} \end{cases} \tag{2.6}$$

In addition, we can model network datasets as directed graphs as well. A twitter network with followers and followees is a good example of directed graph.

Cosine Similarity. A popular measurement for text dataset is the cosine angle between two vectors [3]. The cosine similarity is represented using dot product and magnitude as:

$$W_{(COS)}(i,j) = \frac{x(i) \cdot x(j)}{\| x(i) \|_2 \cdot \| x(j) \|_2}. \tag{2.7}$$

For text matching, the vectors $x(i)$ and $x(j)$ are usually the term frequency vectors of the documents. Since term frequency is always positive, the resulting similarity ranges from 0 meaning independence, to 1 meaning exactly the same, and in-between values indicating intermediate similarity. The cosine similarity can be

seen as a method of normalizing length during comparison, with denominator normalizing each vector to compare different text sizes.

Gaussian Kernels. One of the most commonly used similarity measurements in data mining is the Gaussian kernel, of which traditional form is defined as follows:

$$W_{(GLS)}(i,j) = exp(\frac{- \parallel x(i) - x(j) \parallel^2}{2{\sigma_{(G)}}^2}), \tag{2.8}$$

where $\sigma_{(G)}$ controls the width of neighborhood [100] with a globally fixed value. So we call this kernel as the global Gaussian kernel $W_{(GLS)}$. It is widely used because it works well on many datasets with Gaussian distribution. However, its biggest challenge is how to choose the value of $\sigma_{(G)}$, which affects the clustering results significantly [156]. In other words, clustering result is very sensitive of tuning $\sigma_{(G)}$. Besides, $\sigma_{(G)}$ is not adaptive to local density change.

Instead of selecting one globally-fixed parameter $\sigma_{(G)}$, Zelnik-Manor et al. proposed to calculate a local scaling parameter $\sigma_{(L_{i,k})}$ for each data point in their self-tuning spectral clustering algorithm (ST) [156]:

$$W_{(LCS)}(i,j) = exp(\frac{- \parallel x(i) - x(j) \parallel^2}{\sigma_{(L_{i,k})}\sigma_{(L_{j,k})}}), \tag{2.9}$$

where the parameter $\sigma_{(L_{i,k})}$ is the Euclidean distance between $x(i)$ and its k-th nearest neighbor (k-nn). This kernel uses k-nn distance to approximate the local density, which is similar to the idea of Local Outlier Factor (LOF) [14]. Therefore it can adaptively recognize the local density difference to some extent. However it is extremely important to determine the value of k to faithfully reveal the local density, as shown in [69]. On one hand, k cannot be too large to capture the local distribution. On the other hand, an overly small k will lead to statistical error without a sufficient neighborhood scope. This is to say, compared with global Gaussian kernel $W_{(GLS)}$, local kernel $W_{(LCS)}$ shifts the degree of freedom, or sensitivity, from $\sigma_{(G)}$ to k, which is still hard for users to specify.

Particularly, both Equation 2.8 and 2.9 may fail miserably on the boundary area among clusters with different densities. Figure 2.3 shows a synthetic example that both kernels fail to classify the red data point b. Here the blue cluster is relatively denser than the green one. The point b lies between these two clusters

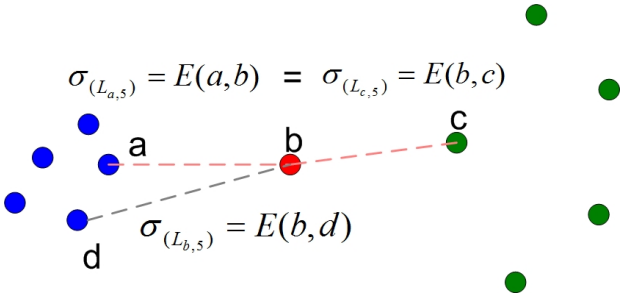


Figure 2.3: Due to the similar density distribution, point b should belong to the green cluster. However, both global and local scaling Gaussian kernels fail to classify point b due to their non-awareness of local density statistics.

with the same Euclidean distance to the closest data point in each cluster (namely $E(b,a) = E(b,c)$, where $E(i,j)$ denotes the Euclidean distance between $x(i)$ and $x(j)$). When the density distribution is considered, point b should belong to the green cluster because its local density is more similar to the density of the green cluster. Therefore, the ideal similarity kernel should return $W(b,a) < W(b,c)$ and $W(a,d) > W(a,b)$.

Nevertheless, the global Gaussian kernel, with a single scaling parameter, only obtains the right result on $W(a,d) > W(a,b)$ because of $E(a,d) < E(a,b)$, while on the other hand returns $W_{(GLS)}(b,a) = W_{(GLS)}(b,c)$ because of $E(b,a) = E(b,c)$. For the local scaling Gaussian kernel, we need to tune k very carefully. If $k = 5$ there is $\sigma_{(L_{a,5})} = E(a,b)$ and $\sigma_{(L_{c,5})} = E(b,c)$, which also leads to $W_{(LCS)}(b,a) = W_{(LCS)}(b,c)$. Only if $k < 5$ it gives $W_{(LCS)}(b,a) < W_{(LCS)}(b,c)$ because of $\sigma_{(L_{a,k})} < \sigma_{(L_{b,k})} \simeq \sigma_{(L_{c,k})}$. However, it will bring another issue: it leads to $W_{(LCS)}(a,d) < W_{(LCS)}(a,b)$ due to $\sigma_{(L_{d,k})} < \sigma_{(L_{b,k})}$ when $k < 5$. The same problem also shows up in Figure 2.2(e): although k is best tuned and ST has additional adjusting steps to further boost the performance [156], it does not improve performance substantially compared with NJW (only about 2% improvement in NMI). In other words, both the global and local scaling Gaussian kernel cannot accurately and stably differentiate the similarity differences with regards to local density.

Recently, Zhang et al. proposed a local density adaptive similarity kernel (SCDA) [160] which is defined as:

$$W_{(DA)}(i,j) = exp(\frac{- \parallel x(i) - x(j) \parallel^2}{2\sigma^2_{(G)}(f_\epsilon(i,j)+1)}), \tag{2.10}$$

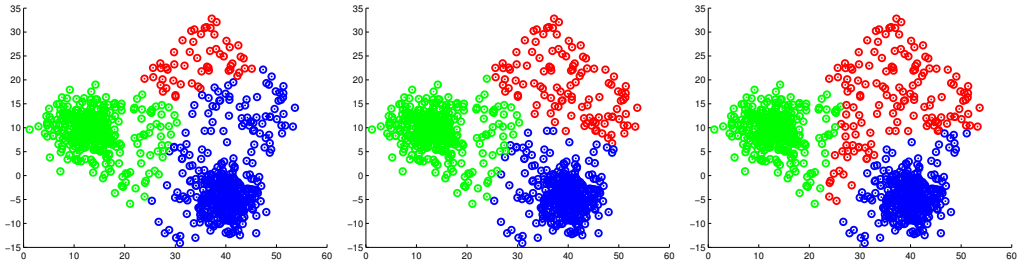
where $f_\epsilon(i,j)$ is the number of instances in the joint region of the ϵ-neighborhoods around instance $x(i)$ and $x(j)$, and ϵ is the specified radius of the sphere neighborhood region. It is claimed that $f_\epsilon(i,j)$ can represent the local density between $x(i)$ and $x(j)$, and therefore $W_{(DA)}$ can be used to distinguish inter-cluster instances. However, the way of choosing ϵ in [160] is by a linear regression with the input parameters such as maximum and variance of all instance pairs in the test dataset. Not surprisingly, it is highly unstable when SCDA is used on the other datasets in an unsupervised way. Correa et al. proposed a similar idea using empty region [30] which also suffers instability by a slight perturbation to the radius of region, especially on the complex high-dimensional datasets, due to the curse of dimensionality. Figure 2.2(f) shows that SCDA performs quite similarly to NJW, and can neither provide a convincing correction to the density bias.

2.4 Diffusion Distance and Diffusion Maps

Embedding reconstruction in spectral clustering (Step 1 to 5 in Algorithm 1) are very sensitive to noise appearance and scaling parameter tuning (if it is built upon the Gaussian kernels). Diffusion maps was proposed by Coifman et al. in [28] to solve these problems.

Diffusion maps (DM) is a Markov-transition-based projection hinged on diffusion process. The non-negativity property of the original affinity matrix W allows to normalize it into a Markov transition matrix $P = D^{-1}W$. The states of the corresponding Markov process are data points, which enables us to analyze it as (positive) random walk. It is straightforward to calculate the transition probability, $p_t(i,j)$ (the probability of transition from $x(i)$ to $x(j)$ after t steps) using entries from P. Thus the diffusion distance $\mathcal{D}_t(i,j)$ between $x(i)$ to $x(j)$ at the time scale of t can be defined as:

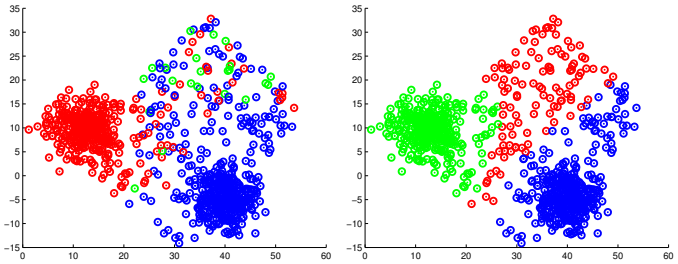
$$\mathcal{D}_t(i,j) = (\sum_k \frac{(p_t(i,k) - p_t(j,k))^2}{\phi_1(k)})^{\frac{1}{2}}, \tag{2.11}$$



(a) Diffusion Maps when $t = 1$, $NMI = 0.5864$

(b) Diffusion Maps when $t = 30$, $NMI = 0.6546$

(c) Diffusion Maps when $t = 50$, $NMI = 0.7056$



(d) Diffusion Maps when $t = 100$, $NMI = 0.6140$

(e) Multiscale Diffusion Maps, $NMI = 0.6486$

Figure 2.4: Clustering results of Diffusion Maps (DM) and Multiscale Diffusion Maps (MDM) on the synthetic dataset in Figure 2.2(a). The global Gaussian kernel is used here with $\sigma_{(G)} = 2$. Figure 2.4(a) to 2.4(d) show the results of DM from $t = 1$ to $t = 100$. Although DM with $t = 50$ obtains better separation in the boundary area among the three clusters, it is hard to guess the best range of t unsupervisedly. MDM, in spite of the elimination of parameter t, easily gets over-diffusion without perception of density change (see Figure 2.4(e)).

where ϕ_1 is the stationary distribution of the positive random walk (trivial left eigenvectors). So the diffusion map at the time scale of t projects the data points to n dimensional eigenspace as:

$$\Psi_t : x \to [\lambda_1^t \psi_1(x), \lambda_2^t \psi_2(x), ..., \lambda_n^t \psi_n(x)], \tag{2.12}$$

where λ_i are eigenvalues and ψ_i are the corresponding right eigenvectors of P [107]. In this way the diffusion distance between $x(i)$ and $x(j)$ becomes:

$$\mathcal{D}_t(i,j) = (\sum_{k=1}^{n} [\lambda_k^{2t} (\psi_k(i) - \psi_k(j))^2])^{\frac{1}{2}}. \tag{2.13}$$

By projecting the data to the diffusion space, 1) the sensitivity to noise is minimized due to the theory of random walk, 2) the effect of scaling parameter σ (if there is any) is reduced. However, another scaling parameter t is still essential because it controls the transitive connectivity. Besides, how to tune the value of t is perplexing because even less clues for tuning it exist than that of σ in the Gaussian kernels. Here we use experiments to reveal its effect, as shown in Figure 2.4(a) to 2.4(d).

In 2009, Richards et al. proposed multiscale diffusion maps (MDM) [122], which considers all possible paths between each instance pair across all the discrete time scales t in the diffusion space. In practice λ_i^t in Equation 2.12 is replaced by:

$$\sum_{t=1}^{\infty} \lambda_i^t = \frac{\lambda_i}{1-\lambda_i}. \tag{2.14}$$

So the multiscale diffusion maps is defined as:

$$\Psi_{(M)} : x \rightarrow [\frac{\lambda_1}{1-\lambda_1}\psi_1(x), \frac{\lambda_2}{1-\lambda_2}\psi_2(x), ..., \frac{\lambda_n}{1-\lambda_n}\psi_n(x)]. \tag{2.15}$$

Multiscale diffusion maps is claimed to be more robust [122] by eliminating the effect of t. The quantity of MDM involves summing over all paths of all discrete time scales connecting $x(i)$ to $x(j)$. As a consequence, this projection should be very robust to noise perturbation in theory, unlike the geodesic distance or Euclidean distance. From the prospect of machine learning, this observation allows us to conclude that this projection is appropriate for designing inference algorithms based on the majority: it takes into account all the evidences relating $x(i)$ to $x(j)$.

Although diffusion maps [28] and multiscale diffusion maps [122] provide more stable descriptions with a strong probabilistic interpretation, and therefore reduce the instability incurred by Gaussian scaling parameters and noise appearance, they still suffer from the lack of density-awareness as shown in Figure 2.4.

2.5 Heat Equation and Heat Kernel

Part of our proposed methodology is inspired by heat diffusion theory [68] with the following reasons:

1. It can provide intrinsic and robust similarity measurement that is aware of manifold structure.

2. It can provide information intimately related to local density.

3. It has many attractive properties such as symmetric, positive semi-definite, and stable under noise appearance.

Laplace operator is closely associated to heat diffusion, connecting geometry of a manifold with the properties of the heat flow. Using the discrete Laplace operator, the heat equation can be simplified, and generalized to matrix operation over spaces with an arbitrary number of dimensions. Due to its intrinsic connection to Markov process, in practice the heat equation is often coupled with random walk graph Laplacian [28], L_{rw} (Equation 2.3), which describes a stochastic process that randomly jumps from vertex to adjacent vertex. Heat equation therefore can be defined by:

$$\frac{\partial H_t}{\partial t} = -L_{rw} H_t, \tag{2.16}$$

where $H_t = e^{-tL_{rw}}$ is the heat kernel on Riemannian manifold $\mathcal{M}$ and t is the time scaling parameter [56]. For $L_{rw} = \psi' \lambda \psi$ (ψ and λ are the eigenvectors and eigenvalues of L_{rw}), the heat kernel can be re-formulated as follows:

$$H_t(i,j) = \sum_{p=1}^{N} [e^{-\lambda_p t} \psi_p(i) \psi_p(j)], \tag{2.17}$$

where λ_p is the p-th eigenvalue and $\psi_p(i)$ is the i-th element in the p-th eigenvector. $H_t(i,j)$ represents the amount of heat being transferred from i to j in time t given a unit heat source at i in the very beginning ($t = 0$). The scaling parameter t in heat kernel is used to control the transitive connectivity: small t makes the loosely-connected graph into slightly stronger connection, while large t makes the graph tend to be more strongly-connected.

Chapter 3

Density-Aware Clustering based on Aggregated Heat Kernel and Its Transformation

3.1 Chapter Introduction

Clustering, the task of discovering natural groupings based on the input data patterns, has been one of the most active research topics in machine learning and knowledge discovery. As a powerful unsupervised data analysis technique, clustering is especially desirable for modeling large datasets because the tedious and often inconsistent manual classification and labeling process can be avoided. While many traditional clustering algorithms have been developed over the past few decades [74] [42], some popular ones that emerged over the last decade generate promising results on various challenging tasks. Among them, spectral clustering [109] [126] [157] [25] [16] [146] demonstrates excellent performance to detect clusters with complex shapes and complicated input space distributions.

3.1.1 Motivations

Despite their earlier success, most of spectral clustering methods still suffer from the following real world challenges:

- The clustering results can be radically different when the scaling parameters of the algorithms are slightly modified or there is some noise perturbation among clusters. We call such a susceptibility **the sensitivity to parameter tuning and noise**.

- Most of these methods tend to assign medium similarity between the boundary instances among clusters with different densities. Therefore **they fail to quantify local density well**, which may result in poor manifold reconstruction and undesirable clustering results.

- Most of the existing density-aware algorithms are only applicable on the Euclidean space. Therefore **their capabilities are significantly constrained in handling today's various types of data**, such as social networks and text datasets.

Robustness is one of the most desirable properties of clustering algorithms, however, here it becomes an essential challenge for spectral clustering. As shown in Figure 2.1, it is a well known problem that the scaling parameter σ of Gaussian kernel (see Equation 2.8 for details) for the affinity matrix has significant impacts on discovering embedded structure because σ determines whether two points are considered similar (neighbors) or not [115]. Although several methods were proposed to address this problem (e.g., [141], [156]), it remains challenging to find a certain range for σ which is optimal to maintain stable yet desirable performance. Another aspect of robustness in spectral clustering is the clustering quality with respect to noise data. As noted in [100], spectral clustering is less sensitive to data perturbation than the popular k-means algorithm. However, given different application domains and/or inappropriate data preprocessing techniques, spectral clustering can still be susceptible to noise [143], which tends to complicate the clustering parameter selection, especially when making use of scaling parameter σ of Gaussian kernel. In summary, since parameter selection can be significantly affected by the noise level of data (as shown in Figure 2.1(c)), we must address robust spectral clustering in terms of parameter selection and noise appearance simultaneously.

In this work, the robustness of clustering algorithms should be measured in the following aspects: (1) not sensitive to small parameter changes; (2) not sensitive to existing noise; (3) stable performance even under a significant noise level

or suboptimal parameter settings; and (4) competitive and comparable results when comparing with those less-robust clustering algorithms without any data perturbation and with correct parameter settings. With these robustness properties, we can reliably analyze data and conduct other data-driven tasks in subsequent analysis steps. The robustness property is equally significant for domain experts who do not have strong machine learning background as they become much more comfortable in utilizing robust algorithms for their domain data analysis. Therefore it is imperative to develop robust clustering algorithms [25].

Another requirement of real world clustering application is to discern the different density distribution among clusters. The traditional spectral clustering algorithms (such as NJW [109] and RWC [126]) assume uniform sampling distribution inside the input dataset to approximate the continuous Laplace operators on Riemannian manifold, and tend to assign medium level affinity on the boundary between low and high density areas. These problems cause the inferior manifold reconstruction especially around cluster boundaries.

As an example, Figure 2.2 shows the clustering results from different graph Laplacians built upon Gaussian kernels. The synthetic dataset in Figure 2.2(a) contains three clusters: the blue and green clusters with a denser Gaussian distribution and the red one with a uniform and sparser distribution within a rectangular area. Figure 2.2(b) to 2.2(d) show the results from three conventional spectral clusterings: NJW [109] with symmetric Laplacian (L_{sym}), RWC [103] with random walk Laplacian (L_{rw}), and NN [100] without Laplacian normalization (L_{nn}) . Since L_{sym} has a more balanced view, NJW performs better than RWC, and demonstrates better density-awareness. However, if we take density distribution into account, all of them fail to separate the clusters appropriately.

Some localized approaches have been focusing on solving these problems, e.g., Self-tuning Spectral Clustering (ST) [156] and Local Density Adaptive Similarity (SCDA)[160]. Nevertheless, they could not effectively capture the local density on the affinity matrix since additional parameters which are very sensitive to heterogeneous density distributions are required. Therefore such approaches may also fail to quantify local density well, and cause undesired clustering results (as shown in Figure 2.2(e) and 2.2(f)). Moreover, these algorithms are built upon Gaussian kernel, and could only be applicable to the applications in the Euclidean space. Their

capabilities are therefore significantly constrained in handling today's big complex data. One example is network dataset, including social networks, computer networks, and biological networks. Network dataset can be naturally represented as affinity matrix themselves because they are already of graph-structure (see details in Equation 2.6). Another example is text data which often uses the cosine kernel to measure similarity (see details in Equation 2.7). Although Gaussian kernel is popularly used in many applications, we also need to handle the aforementioned datasets with diverse characteristics. Therefore, to be more practical and adaptive in different real world situations, local-density-aware clustering algorithms need to work well with any form of similarity kernels.

3.1.2 Contributions

In this research we propose a heat-diffusion-based framework which provides not only competitive average performance, but also robustness to scaling parameter, noise appearance and different density distributions across clusters. Our framework has the following contributions:

1. We derive a robust kernel function by integrating heat kernel along the entire time scale (Section 3.3.1), and combine it with Laplace-Beltrami Normalization (LBN, Section 3.3.3). We call this algorithm as Aggregated Heat Kernel (AHK, Section 3.3.4). As a result, we **provide a robust clustering algorithm while reducing the negative influences on stability by scaling parameter tuning and noise appearance**. We also discuss the connections of AHK to the other popular and related approaches (Section 3.3.2).

2. We design a probability-based local density affinity transformation (LDAT, Section 3.4) that aims to **reduce different density effects across clusters in the affinity matrix**. It is a simple and effective enhancement to local density awareness especially around the cluster boundary area. It is not only based on affinity matrix, **so it works well with any type of similarity kernels**. These features distinguish our proposed framework from other candidate approaches in handling diverse and complex real world problems.

3. Our novel framework (Section 3.5), systematically combining AHK and LDAT

together, delivers robust clustering results in terms of different scaling parameter, noise level and divergent density distribution across different clusters.

4. We thoroughly evaluate the proposed framework with several closely-related baseline algorithms on a number of synthetic and benchmark datasets (Section 3.6). The experimental results confirm that the proposed framework, even under suboptimal parameter settings, outperforms existing approaches for datasets with noise and heterogeneous density distribution, using different similarity kernels.

3.2 Related Works

Towards robustness, researchers have explored various techniques, including robust statistics [11] [72], noise insensitive regression [24] [17], noise resistant transformation [148], and noise robust clustering [33] [57] [88] . However, robust clustering approaches that are adaptive to both parameter tuning and noise sensitivity are rather rare. In fact, as shown in Figure 2.1, scaling parameter and noise perturbation are correlated to each other. Mean shift clustering [29] and noise robust spectral clustering [88] also fail in considering these two simultaneously and systematically. In [20], M-estimation robust statistics is used in a robust path-based similarity measurement which requires no local parameters to be set manually, nonetheless, prior knowledge of data domain is required, which is not our research target here.

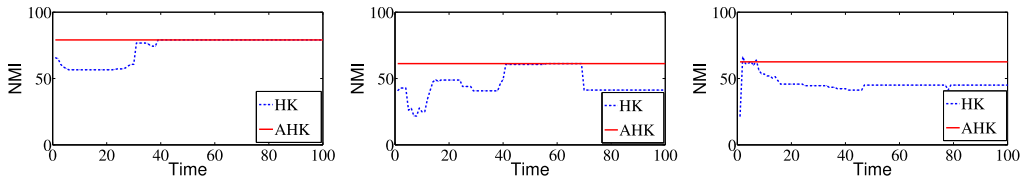
Towards density-driven clustering, some non-spectral clustering algorithms such as DBSCAN [48] [140] and OPTICS [4] start from the estimated density distributions of corresponding nodes. Some researches approached density through updating similarity information, such as shared nearest neighbors (SNN) [75] and [57]. The similarity between two points is confirmed by their shared (or common) nearest neighbors. Later some advanced techniques [46] [131] based on SNN have also been proposed. But their performance suffer significantly from the curse of dimensionality and the sensitivity of neighborhood scaling parameters [46], since their metrics are usually based on Euclidean space. Moreover they cannot cluster datasets well when the density distributions vary significantly [64].

There are some existing graph-based techniques which built upon hierarchical modeling. Chameleon [78] defines affinity from relative inter-connectivity and closeness which are based on a min-cut bisection of clusters. But its computation requires a high computational cost. Recently Graph Degree Linkage [159] was designed with easy implementation and high computational efficiency. However its performance is very sensitive to the perturbation of similarity result from scaling parameter tuning.

In Section 3.3, we propose a new framework that corrects the undesired effects from the aforementioned limitations. It is built upon advanced diffusion space which is stable to scaling parameter tuning and noise perturbation. Also our proposed method is aware of local density change across clusters, therefore it behaves well under non-uniform density distribution. Moreover, compared with the other popular algorithms, our method is more universally applicable since it can use the Gaussian kernels, as well as any other kernel to construct affinity matrix.

3.3 Aggregated Heat Kernel (AHK) and Its Use in Clustering

This subsection proposes a probabilistic clustering method based on heat diffusion theory. The reason we resort to heat diffusion is to minimize the negative influence of both scaling parameter tuning and noise appearance. Since we concentrate on global distribution for data clustering, the embedded structure must be invariant to local perturbation (noise or outliers), and they should be determined only by "visible" neighborhood while avoiding negative effects from changing scaling parameters. The Heat kernel (HK), as the fundamental solution of heat diffusion, offers a statistical description of random walk, so it can be employed to build a diffusion map. Here we integrate spectral clustering and the heat diffusion theory together and show that the integrated approach improves the robustness to both scaling parameter tuning and noise appearance.



(a) $\sigma = 0.15$ (b) $\sigma = 0.16$ (c) $\sigma = 0.28$

Figure 3.1: The sensitivity of heat kernel (HK, Equation 2.17) to time scaling parameter t on Iris dataset clustering (measured by NMI). Experiments in Figure 3.1(a) to 3.1(c) are built upon global Gaussian kernel with different σ. We can see that AHK outperforms HK in most cases and it doesn't require tuning t. We use the random walk Laplacian in this experiment.

3.3.1 Aggregated Heat Kernel

In this subsection, we describe and analyze the construction of Aggregated Heat Kernel (AHK). Similar to the conventional heat kernel in Equation 2.17, AHK is also built upon the eigen-decomposition of (Laplacian-normalized) affinity matrix W. As discussed in Section 2.5, a heat kernel (HK) is multiscale. The function $H_t(i, *)$ is mainly determined by the nearby neighborhood of $x(i)$, and this area grows bigger as t increases. In other words, for a small t, $H_t(i, *)$ only represents local properties of the area around $x(i)$, but a large t can capture the properties from a larger area or even the entire data space. Yet this additional degree of freedom makes it difficult to determine the value of t (Figure 3.1) because we have little clue about how to find the best t value, which is similar to the time scaling parameter in diffusion maps and the scaling parameter σ in Gaussian kernels. In other words, the clustering result become sensitive to the time parameter selection in heat kernel.

We propose a robust kernel function by integrating the entire continuous time scale on heat kernel, and name it as **Aggregated Heat Kernel** (AHK):

$$\mathcal{H}(i,j) = \int_0^\infty H_t(i,j)dt = \sum_{k=1}^{n}[\frac{1}{\lambda_i}\psi_k(i)\psi_k(j)]. \tag{3.1}$$

Specifically, the derivation of this function can be explained in the form of Laplace transform [5]:

$$F(s) = \int_0^\infty e^{-st}f(t)dt, \tag{3.2}$$

where parameter s is a complex number and $f(t)$ is "degeneralized" to a constant function one. Here the Laplace transform is interpreted as a transformation from the time domain, in which inputs and outputs are functions of time (t), to the frequency domain, where the inputs and outputs are the functions of frequency (λ). Therefore this transform provides an alternative functional description that simplifies the process of analyzing the heat system behavior, and synthesizes a new comprehensive system with a set of properties inherited from the original heat kernel:

- Symmetric: $\mathcal{H}(i,j) = \mathcal{H}(j,i)$.
- Semigroup identity: $\mathcal{H}(i,j) = \int_M \mathcal{H}(i,k)\mathcal{H}(j,k)dk$.
- Positive semi-definite: $\sum_{p,q} \mathcal{H}(i,j)c_p c_q \geq 0$, where $c_1, c_2, ..., c_n$ are real numbers.

With the pure and applied probability term, AHK can be explained as an expected value of heat kernel. If we interpret t as a random variable with the probability density function $\mathscr{F}$, then AHK, or the Laplace transform of f is given by the expectation:

$$\mathcal{L}\mathscr{F}(\lambda) = E[e^{-\lambda t}]. \tag{3.3}$$

So AHK, to some degree, is a weighted average of all possible heat diffusion processes across the entire continuous time-domain.

The definition of AHK can also be elucidated by Fredholm theory [41], a theory of integral equations, where the actual function space is determined by the eigenfunctions of the differential operator; that is, by the solutions to $L\psi(i) = \lambda\psi(i)$. The set of eigenvectors ψ here spans a Hilbert space since there is a natural inner product. Therefore the kernel $\mathcal{H}(i,j)$ is a realization of the Fredholm operator or the Fredholm kernel. It follows from the completeness of the basis of the Hilbert space, namely, that one has:

$$\delta(x(i) - x(j)) = \sum_k [\psi_k(i)\psi_k(j)], \tag{3.4}$$

where $\delta(x)$ is the Dirac delta function (a generalized function defined in the real space R, such that its value is zero everywhere except at origin 0) since the eigenvectors ψ associated with L are assumed to be complete and orthogonal to each other.

From Figure 3.1 we observe that in the original heat kernel (HK) the time scaling parameter t is also correlated with Gaussian scaling parameter σ, and t needs to be tuned carefully. Moreover, Figure 3.1 shows that AHK performs better than the original HK on almost all times t regardless the value of σ. Comparatively, AHK is capable of providing more comprehensive and stable probabilistic affinity information.

By the definition of heat diffusion, AHK is naturally associated with the random walk normalization, L_{rw}, but we could also generalize AHK on other Laplacians such as $\mathcal{H}_{sym}$ on symmetric L_{sym} or $\mathcal{H}_{nn}$ on unnormalized L_{nn}. In Section 3.3.3 we will analyze the best Laplacian for constructing AHK.

3.3.2 Connections to AHK

In this subsection we build theoretical connections from AHK to the other existing popular techniques.

Inverse Laplacian. AHK can be viewed as a pseudo inverse or Moor-Penrose inverse [58]. By doing so, we achieve multiscale heat diffusion. Instead of doing pseudo inverse, we could directly inverse graph Laplacian matrix [88] as:

$$(I + \beta L_{sym})^{-1}, \tag{3.5}$$

where β is the positive regularization parameter and I allows us to invert Laplacian matrix always. Note that, [88] used this direct inversion to design noise robust spectral clustering.

Commute Distance. Commute distance $C(i, j)$ between $x(i)$ and $x(j)$ is defined by the expected random walk round trip travel time. AHK is also known as Green's function [120], which is closely related to the commute distance (CD) or resistance distance. The Green's function is a left inverse operator of Laplace operator, $\mathcal{H}_{rw} \cdot L_{rw} = I$. For $\mathcal{H}_{nn}$ constructed on unnormalized L_{nn}, commute distance can be reformulated as:

$$C(i, j) = vol(\mathcal{H}_{nn}(i, i) + \mathcal{H}_{nn}(j, j) - 2\mathcal{H}_{nn}(i, j)), \tag{3.6}$$

where $vol = \sum_{i=1}^{n} D(i,i)$. Just like AHK, commute distance also considers all possible length, paths and their weights, which is more robust than the shortest path or geodesic distance. Note that, commute distance can also be expressed by the random walk L_{rw} or symmetric graph Laplacian L_{sym} [120].

Diffusion Distance. Commute distance is also related to diffusion distance. By integrating Equation 3.1 into the above equation, we get:

$$C(i,j) = vol \sum_{k=2}^{n} [(1/\lambda_k)(\psi_k(i) - \psi_k(j))^2], \tag{3.7}$$

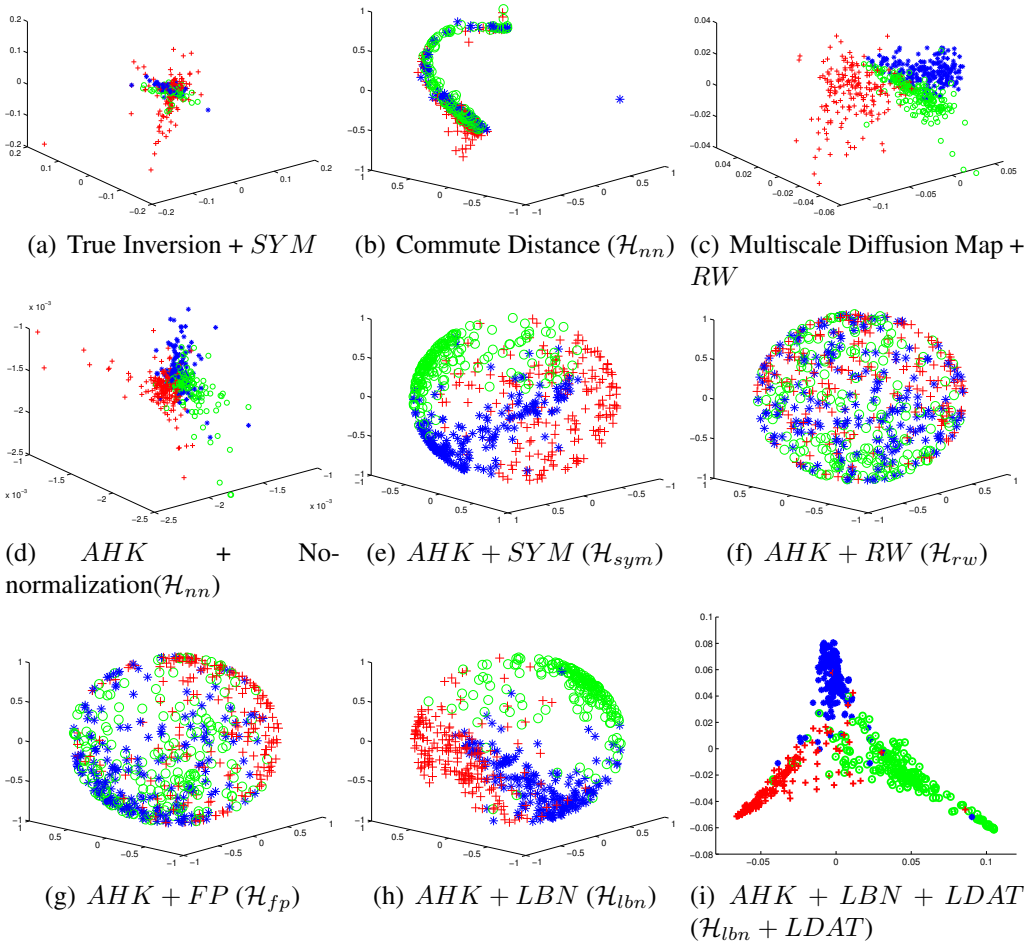
and also multiscale diffusion distance can be defined by:

$$\sum_{t=1}^{\infty} D_t^2(i,j) = \sum_{k=1}^{n} [1/(1-\lambda_k^2)(\psi_k(i) - \psi_k(j))^2]. \tag{3.8}$$

Both commute distance and diffusion distance look similar but they have different eigenvalue weighting and different Laplacian normalization.

Diffusion distance [122] can also be represented by $1/\lambda_i$, which shares the same weighting with $\mathcal{H}$ but it is for distance weighting. If the time summation starts from $t = 1$, then it is exactly the same as the multiscale diffusion distance (MDM) of Equation 2.14. Both of eigenvalue weighting (starting from $t = 0$ or $t = 1$) will show quite similar weighting distribution anyway for $0.5 \leq \lambda \leq 2$, which is common for most of the graph Laplacians.

Laplace Transform and Fourier Transform. As previously analyzed, AHK can be explained as a degeneralized form of Laplace transform [5]. Laplace transform is related to Fourier transform, but whereas the Fourier transform expresses a function or signal as a series of modes of vibration (frequencies), the Laplace transform resolves a function into its moments. Like the Fourier transform, in our derivation of AHK, the Laplace transform is used for solving differential and integral equations. But the equivalence relation between Laplace Transform and Fourier Transform is not valid in our AHK derivation because the region of convergence (ROC) of $F(s)$ in Equation 3.2 contains no imaginary component.



(a) True Inversion + SYM (b) Commute Distance ($\mathcal{H}_{nn}$) (c) Multiscale Diffusion Map + RW

(d) AHK + No-normalization($\mathcal{H}_{nn}$) (e) $AHK + SYM$ ($\mathcal{H}_{sym}$) (f) $AHK + RW$ ($\mathcal{H}_{rw}$)

(g) $AHK + FP$ ($\mathcal{H}_{fp}$) (h) $AHK + LBN$ ($\mathcal{H}_{lbn}$) (i) $AHK + LBN + LDAT$ ($\mathcal{H}_{lbn} + LDAT$)

Figure 3.2: Different ways of manifold reconstruction on 20ngB dataset.

3.3.3 Different Laplacians and Their Comparison

Even though we made proper connections among relative approaches, most of them used different Laplacians without thorough evaluation. Therefore it is not yet clear what is the best graph Laplacian for our proposed $\mathcal{H}$. It is shown in [84] that if we assume uniform sampling of data points from a sub-manifold $\mathcal{M}$, the eigenvectors of L_{rw} with $\sigma \to 0$ and $n \to \infty$ tend to approximate Laplace-Beltrami operator on $\mathcal{M}$, which guarantees manifold reconstruction. However, in reality, the sampling rate of data points tends to be non-uniform and it shows skewed density distributions, resulting in a manifold reconstruction with a poor quality in AHK.

The following two additional normalizations are used to improve the density awareness of Laplacians:

$$W^{(\kappa)} = D^{-\kappa} W D^{-\kappa}, \tag{3.9}$$

$$L^{(\kappa)} = I - {D^{(\kappa)}}^{-1} W^{(\kappa)}, \tag{3.10}$$

where κ is a normalization factor and $D^{(\kappa)}$ is the diagonal matrix with the sum of $W^{(\kappa)}$ row weight.

- If $\kappa = 0$, $L^{(0)} = L_{rw}$ which is exactly the random walk (RW) Laplacians.
- If $\kappa = 1/2$, then it is called Fokker-Planck (FP) diffusion.
- If $\kappa = 1$, it is called Laplace-Beltrami Normalization (LBN).

The relations among the three normalizations are well described in [28]. Depending on κ, LBN can also be reduced to the random walk normalization or Fokker-Planck diffusion. In particular, we focus on LBN because it removes the negative influence of the dataset density and recovers manifold structures on $\mathcal{M}$ with the condition of $\sigma \to 0$ and $n \to \infty$ [28]. In other words, the additional re-normalization of affinity matrix W enables us to reconstruct manifold structures under non-uniform density distribution. Another advantage of LBN is that the consequent clustering results can be less sensitive to noise and scaling parameter tuning.

Figure 3.2 shows the effects of different approaches and Laplacians on 20 newsgroup text data (20ngB) (see Section 3.6 for more details). True inversion (Figure 3.2(a)) and commute distance (Figure 3.2(b)) show the worst results in separating three topics. Although they share the same Laplacian matrix inversion, the results are quite different. Interestingly multiscale diffusion map (MDM, Figure 3.2(c)) shows the best separation among all the non-AHK approaches. In the case of AHK, most of Laplacian approaches (except unnormalized Laplacian) reconstruct the topic distribution as a sphere shape. AHK with unnormalized Laplacian (Figure 3.2(d)) appears to have the ability of separation but the distance among documents are very close to each other compared with other Laplacians. Symmetric Laplacian (Figure 3.2(e)) shows very good separation and sphere shape reconstruction but it is not anisotropic transition. The original random walk (RW) normalization (Figure 3.2(f)) shows the most mixture of three topics but once we add the additional normalization of Equation 3.9, we reconstruct better manifold structures. LBN shows

the best coherent and condensed structure (Figure 3.2(h)) among all different Laplacians. For our future experiments we mainly focus on LBN, but we provide further and more detailed analysis regarding different approaches with different Laplacians in Section 3.6.

3.3.4 Combination of AHK and LBN and Discussion

After investigating the nice properties of AHK and LBN, we now present our robust spectral clustering algorithm that combines these two techniques, and thereby is less sensitive to the scaling parameter selection and noise appearance. For the notational simplicity, we call the integrated algorithm as the AHK Clustering, or AHK directly. Let X be the input dataset of size $n \times m$, where n is the number of data points and m is the number of dimensions (features), our algorithm is detailed in Algorithm 2.

ALGORITHM 2: AHKClustering(X,c,γ)

Input: $X \in R^{n \times m}$ where n is #instances, m is #features, c is #clusters, and γ is an eigenvalue smoothing parameter.

Output: Cluster assignments of n instances.

1 Construct the affinity matrix $W \in R^{n \times n}$;

2 Compute the diagonal matrix $D \in R^{n \times n}$ where $D(i,i) = \sum_{j=1}^{n} W(i,j)$ and $D(i,j) = 0$ if $i \neq j$;

3 Apply Laplace-Beltrami Normalization L_{lbn} using Equation 3.9 and 3.10 with $\kappa = 1$;

4 Extract generalized eigenvectors $\psi(i)$ and corresponding eigenvalues λ_i, $i = 1, 2, ..., n$;

5 Construct $\mathcal{H}_{\text{lbn}}$ matrix with $\psi(i)$ and λ_i using $\mathcal{H}(i,j) = \sum_{k=1}^{n} [\frac{1}{\lambda_i + \gamma} \psi_k(i) \psi_k(j)]$;

6 Extract the first c nontrivial eigenvectors ψ' of $\mathcal{H}_{\text{lbn}}$, $\psi' = \{\psi'_1, \psi'_2, \ldots, \psi'_c\}$;

7 Re-normalize the rows of $\psi' \in R^{n \times c}$ into $Y_i(j) = \psi'_i(j) / (\sum_l \psi'_i(l)^2)^{1/2}$;

8 Run k-means with c and $Y \in R^{n \times c}$.

Algorithm 2 undergoes two times of data warping: In Step 1, the original affinity matrix W is constructed using a similarity kernel as appropriate (please refer to Section 2.3 for the kernel selection) according to the specific data characteristics.

Then we use LBN as Laplacian normalization on W (Step 2 and 3, check Section 3.3.3 for more details) and utilize the derived eigendecomposition to construct AHK (Step 4 and 5, check Section 3.3.1 for more details). After that it comes to the second data warping by extracting the first c nontrivial eigenvectors from the AHK affinity matrix (Step 6). Finally we perform k-means on the normalized eigenvectors and label the projected data points. The smoothing parameter γ in Step 5 is added to avoid the eigenvalue λ from being to small and thereby stabilize the AHK affinity matrix computation. In practice we set $\gamma = 0.001$ by default.

Regarding the computational complexity, eigenvalue decomposition is the most time consuming step and dominates the computation. There are many iterative methods to conduct eigenvalue decomposition (e.g., power iteration [8]), but in general finding the eigenvalues reduces to matrix multiplications by computing a symbolic determinant, in which the running time is $O(n^3 + n^2 log^2 n)$ [112].

It is worth to notice that AHK has the following significant benefits: 1) It is a stronger form of random walk process by taking all possible paths in entire continuous time scales into consideration, therefore it is more robust and less sensitive to noise or artifacts than other regular kernels. 2) To mitigate the biased contribution of the denominator from some extremely small eigenvalues λ, AHK introduces a smoothing term γ to make computation more stable. 3) To relieve the bias to non-uniform density distribution, AHK employs Laplace-Beltrami normalization (LBN) which can recover the Riemannian manifold under skewed density distribution. In other words, AHK enables better and more stable manifold reconstruction, especially under noise, parameter disturbance, and non-uniform density distribution. Therefore in theory it guarantees the strong adjacency (similarity) among intra-cluster instances even under suboptimal conditions. In the Section 3.4, we introduce an affinity transformation to give the clustering algorithm a better insight into the separation between adjacent/overlapping clusters with different density distributions.

3.4 Local Density Affinity Transformation (LDAT)

As we discussed in Section 2.3, there are numerous similarity measurements ranging from network connectivity to Gaussian kernels. Unfortunately, few existing

approaches took local density into consideration. Some exceptions, such as [160] [153] are based on simple approximations of local density that fail to provide the stability against neighborhood perturbation.

In this subsection we propose a ***Local Density Affinity Transformation*** **(L-DAT)** with the following attractive properties: 1) It reveals local density differences for the purpose of correcting density bias; 2) It can be applied on any similarity kernel; 3) It works quite stably with a solid probabilistic interpretation. The entire procedure of LDAT is documented in Algorithm 3.

ALGORITHM 3: LDAT(W, k)

Input: Input affinity matrix $W \in R^{n \times n}$ where n is #instances, and k is the neighborhood size.

Output: LDAT affinity matrix $W^{(\mathrm{LDAT})}$.

1 For each instances, only keep the k-nn affinity information and set all the other as zero in W;
2 Apply a positive random walk normalization P on W (Equation 3.11) ;
3 Construct the reduced $\mathcal{P}$ (Equation 3.13) ;
4 Employ another positive random walk normalization $W^{(\mathrm{LDAT})}$ on the reduced $\mathcal{P}$.

Stage 1 (Step 2 in Algorithm 3). In our research we measure the local density on affinity matrix with a positive random walk normalization as our first step. It provides both probabilistic and local density information by involving degree or volume of each instance, and brings the advantage from the difference between $P(i,j)$ and $P(j,i)$:

$$P(i,j) = \frac{W(i,j)}{\sum_k W(i,k)}, \tag{3.11}$$

where $P(i,j)$ is the transition probability from $x(i)$ to $x(j)$ and $\sum_k W(i,k)$ is the local volume of $x(i)$ if we quantify k in a certain neighborhood ($W(i,k)$ is non-zero if $x(k)$ is inside $x(i)$'s k-nn neighborhood). It means that we only maintain the connections within the k nearest neighborhood (k-nn) and remove other distant affinity information. If W faithfully describes the real affinity information, $\sum_k W(i,k)$, as local volume of $x(i)$, is a simple and effective approximation of $x(i)$'s local density. Intuitively, the larger the local volume is, the denser $x(i)$'s local neighborhood is. In general, $P(i,j)$ is different from $P(j,i)$ if the local density distribution between

instances $x(i)$ and $x(j)$ is different. Intuitive speaking, if the manifold structure and associated data points can be properly recovered by the positive random walk normalized affinity matrix, a data point set with high density before normalization would become even more condense (comparatively) afterward, which can be observed from the blue cluster Figure 3.3(d).

Stage 2 (Step 3 in Algorithm 3). Ideally, the transition probability between two points within the same cluster should be larger than two boundary points across two (neighboring) clusters with different densities. The difference between the transition probability must be captured in order to accurately separate different clusters. For example in Figure 2.3, there should be $P(a,d) \simeq P(d,a) \gg P(a,b)$, and $P(a,b) \ll P(b,a)$ if we consider the local volume difference. In other words, as far as point a is concerned, its affinity to b is relatively smaller compared with its affinity to any other point in the blue cluster. We call the difference between $P(i,j)$ and $P(j,i)$ ***local density bias***.

Our goal is to fix this local density bias by making point b in Figure 2.3 to be further away from point a than from any point in the green cluster, and thereby assimilate b into the green cluster. We achieve this goal by reducing $P(i,j)$ (if $P(i,j) > P(j,i)$) :

$$\mathcal{P}(i,j) = max[P(i,j) - \alpha(P(i,j) - P(j,i)), 0], \qquad \text{if } P(i,j) > P(j,i), \quad (3.12)$$

where $\alpha \in [0, \inf]$ is used to control how much reduction is applied to $\mathcal{P}(i,j)$. When $\alpha = 0$, $\mathcal{P}(i,j)$ is the same as a positive random walk normalization $P(i,j)$. When $\alpha > 0$, the local density bias is taken into account. When $\alpha = 1$, $\mathcal{P}(i,j) = P(j,i)$, which translates into $\mathcal{P}(b,a) = P(a,b) < \mathcal{P}(b,c)$ in Figure 2.3, so that point b can be classified into the green cluster.

Because our goal is to rectify the local density bias, $\alpha = 1$ is a simple and natural choice and our experiments on Figure 2.2(a) dataset also confirmed that when $\alpha = 1$, it shows the best performance, as shown in Figure 3.4. Therefore, Equation 3.12 can be simplified as:

$$\mathcal{P}(i,j) = min(P(i,j), P(j,i)). \quad (3.13)$$

Although this step looks simple, it actually contributes a lot on classifying the boundary points by "assimilating" them to the point set with similar density. Figure

3.3(e) shows the effect of Equation 3.13, where the relative distance between the red point and green cluster become shorter compared with Figure 3.3(d). From the perspective of any blue point, the red one is farther away than the blue species. Intuitively, even though the red point may initially treat blue points as closer neighbors than the green points, the blue points will "push" it away.

Stage 3 (Step 4 in Algorithm 3). After applying Equation 3.13, we employ another positive random walk normalization, which again endows our method with a probabilistic interpretation. Figure 3.3(f) shows the effect of this second normalization on top of Figure 3.3(e): the red point is still far away from blue cluster and close to green cluster.

Superficially speaking, LDAT is similar to SNN (shared nearest neighbors [75] [46] [131]) except for the first random walk normalization in Algorithm 3. But in fact this step is of great importance. If there is no random walk normalization before Equation 3.13, most part of the matrix is still symmetric (considering k is not very small in step 1). In this case Equation 3.13 would not have real impact on the final performance, as shown in the comparison between Figure 3.3(b) and 3.3(c). However in our proposed LDAT, the first random walk normalization between step 1 and step 3 delivers awareness of density difference. Therefore the subsequent reduced $\mathcal{P}$ is capable to correct the bias originating from different cluster densities. Another positive side-effect of the first random walk normalization in LDAT is that it supplies stability with different setting of k, while SNN suffers a lot from such perturbation (Figure 3.10 in Section 3.6).

We now analyze the effect of LDAT in theory, which is closely connected to NCut (normalized cut). Suppose there are only two point sets X and Y in the entire dataset V where $V = X \cup Y$, the corresponding NCut is defined as follows [100]:

$$NCut(X,Y) = \frac{C(X,Y)}{assoc(X,V)} + \frac{C(Y,X)}{assoc(Y,V)}, \tag{3.14}$$

where $C(X,Y) = \sum_{i\in X, j\in Y} W_{ij}$, and $assoc(X,V) = \sum_{i\in X, j\in V} W_{ij}$. If we restraint the connections of each node in k-nn neighborhood, Equation 3.14 can be rewritten as:

$$NCut(X,Y) = \frac{C(X,Y)}{v(X)} + \frac{C(Y,X)}{v(Y)} = P(X|Y) + P(Y|X), \tag{3.15}$$

where $v(X)$ is the summation of volume of all the instances in X, and $P(X|Y)$ is the transition probability from any instance in cluster Y to any instance in cluster X. The minimization of NCut actually seeks a cut through the graph such that a random walk seldom transitions from X into Y or vice versa. However NCut has strong density bias when dealing with datasets with heterogeneous density distributions, which can be proven as follows.

Proposition 1 *Let graph $\mathcal{G}$ be k-nn connected and non bi-partite. And the affinity have been normalized by the positive random walk Laplacian. For two overlapping sets $A, B \subset V$ and $A \cup B = V$, given that A is denser than B, but intra-cluster density is uniform and the number of nodes are very similar. NCut may fail to provide the best cut due to the local density bias.*

Proof: As shown in Figure 3.5(a), define O as the overlapping area, A'' as the area inside A that is close to O, and B'' as the area inside B that is close to O. Assume that under certain k-nn constraint, connections only exist between two adjacent area. In other words, there is no connection between A' and O, A'' and B'', and B' and O. Apparently the overlapping area O has the highest density.

Now let's firstly "zoom-in" to analyze the cutting area. Suppose a cut separate V into two adjacent sets X and Y, where $X \cup Y = V$ and $X \cap Y = \emptyset$. Apparently there is:

$$v(V) = v(X) + v(Y), \tag{3.16}$$

and since the affinity has been normalized by positive random walk normalization, there is:

$$v(X) = |X|, \tag{3.17}$$

where $|X|$ is the number of instance in X. Suppose $|X| \sim |Y| \sim T$, there are:

$$\begin{aligned} NCut(X,Y) &= \frac{C(X,Y)}{v(X)} + \frac{C(Y,X)}{v(Y)} \\ &= \frac{1}{T}(C(X,Y) + C(Y,X)). \end{aligned} \tag{3.18}$$

Now we only focus on the value of $C(X,Y)+C(Y,X)$ under different density distributions. Figure 3.5(b) shows the connection between X and Y. We suppose the simplest 2-nn neighborhood and the average density of X and Y are p and r

respectively. The average density of the boundary area is q. X_b are the points in the boundary area of X close to Y, similarly Y_b are the points in the boundary area of Y close to X. X_c and Y_c are the other points inside X and Y. There is $C(X,Y) = C(X_b,Y_b)$ and $C(Y,X) = C(Y_b,X_b)$. We can assume $C(Y_b,X_b) = q\eta/(q+r)$ and $C(Y_b,Y_c) = r\eta/(q+r)$, and $C(X_b,Y_b) = q\eta/(p+q)$ and $C(X_b,X_c) = p\eta/(p+q)$, where η is a connection factor. $C(X,Y) + C(Y,X)$ will change under different density distributions.

1. If $p \sim r$, q would also have similar value. Then $C(X_b,Y_b) + C(Y_b,X_b) = \eta$;

2. Suppose $p < r$, then we have $p < q < r$ (since $X \cap Y = \emptyset$). There is:

$$C(X_b,Y_b) + C(Y_b,X_b) = \frac{q\eta}{q+r} + \frac{q\eta}{p+q} = \frac{(2q^2+pq+qr)\eta}{q^2+pq+qr+pr}. \quad (3.19)$$

   If $q^2 > pr$, there is $C(X_b,Y_b) + C(Y_b,X_b) > \eta$.

In short, if $|X|$ is similar to $|Y|$, the value of Equation 3.18 is dominated by $C(X_b,Y_b) + C(Y_b,X_b)$. And the minimization of conventional NCut makes it less likely to cut along the boundary where $q^2 > pr$. Since NCut value under the condition of $p \ll r$ or $p \gg r$ is very possible to be greater than that under $p \sim r$, NCut is less likely to cut along L_3, since the two sides have the most different density distribution.

A special case is that A and B are adjacent (A is much more denser than B) but there is (almost) no overlapping area, as shown in Figure 3.6. We make the similar assumption that under certain k-nn constraint, connections only exist between two adjacent area. In other words, there is no connection between A' and B'', and A'' and B'. The same deduction of Equation 3.18 and analysis still holds as in the general case in Figure 3.5(a). Since density on the two sides of $L3$ changes a lot, the conventional NCut tends to cut along $L1$ or $L2$ rather than $L3$. □

Proposition 2 *LDAT alleviates the density bias of NCut through lowering the NCut value.*

Proof: To prove LDAT alleviates the density bias under different density distribution, we suppose $p < r$. In Figure 3.5(b) (after random walk normalization),

$C(Y_b, X_b) < C(X_b, Y_b)$. Step 3 in Algorithm 3 makes $C(X_b, Y_b) \leftarrow C(Y_b, X_b) = q\eta/(q+r)$. After the second random walk normalization (Step 4), there is:

$$\begin{aligned} C(X_b, Y_b) + C(Y_b, X_b) &= \frac{q\eta/(q+r)}{p/(p+q) + q/(q+r)} + \frac{q\eta}{q+r} \\ &< \frac{q\eta/(q+r)}{p/(q+r) + q/(q+r)} + \frac{q\eta}{q+r} = \frac{q\eta}{p+q} + \frac{q\eta}{q+r}, \end{aligned} \tag{3.20}$$

therefore LDAT lowers the NCut value compared with Equation 3.19. Furthermore,

$$\begin{aligned} C(X_b, Y_b) + C(Y_b, X_b) &= \frac{q\eta/(q+r)}{p/(p+q) + q/(q+r)} + \frac{q\eta}{q+r} \\ &= \frac{(2pq^2 + q^3 + pqr + rq^2)\eta + (pq^2 + q^3)\eta}{(2pq^2 + q^3 + pqr + rq^2) + (pqr + pr^2)}, \end{aligned} \tag{3.21}$$

and we have

$$\frac{(pq^2 + q^3)\eta}{pqr + pr^2} = \frac{(pq^2 + q^3)\eta/(pq)}{(pqr + pr^2)/(pq)} = \frac{(q + q^2/p)\eta}{r + r^2/q}. \tag{3.22}$$

Suppose $q = \beta p$ and $r = (\beta + \triangle)p$, where $\beta > 1$ and $\triangle > 0$, the condition that makes Equation 3.21 smaller than η is

$$\begin{aligned} &\frac{q + q^2/p}{r + r^2/q} = \frac{\beta p + \beta^2 p^2/p}{(\beta + \triangle)p + (\beta + \triangle)^2 p^2/(\beta p)} = \frac{\beta + \beta^2}{(\beta + \triangle) + (\beta + \triangle)^2/\beta} < 1 \\ &\Rightarrow \frac{\triangle^2}{\beta} + 3\triangle - (\beta^2 - \beta) > 0 \\ &\Rightarrow \triangle > \frac{(\sqrt{9 + 4(\beta - 1)} - 3)\beta}{2}. \end{aligned} \tag{3.23}$$

1. On the one hand, from the proof of Proposition 1 we know that traditional NCut doesn't cut along L3 when $q^2 > qr$. This condition can be also represented as:

$$\beta^2 > \beta + \triangle. \tag{3.24}$$

Assume that both Equation 3.23 and 3.24 holds, we have:

$$\begin{aligned} &\beta^2 - \beta > \triangle > \frac{(\sqrt{9 + 4(\beta - 1)} - 3)\beta}{2} \\ &\Rightarrow \beta - 1 > \frac{\sqrt{9 + 4(\beta - 1)} - 3}{2} \\ &\Rightarrow \beta > 1, \end{aligned} \tag{3.25}$$

which always holds according to $p < r$.

2. On the other hand, traditional NCut cuts along L3 when $q^2 < qr$. This condition can be also represented as:

$$\beta^2 < \beta + \triangle \Rightarrow \triangle > \beta^2 - \beta. \tag{3.26}$$

We also have the condition $\triangle > \beta^2 - \beta > \frac{(\sqrt{9+4(\beta-1)}-3)\beta}{2}$ holds.

Therefore, after LDAT the value of NCut becomes smaller than η where exists density difference. The effect of LDAT is not so obvious inside each cluster of the area with similar density. However, if $p << r$, $C(X_b, Y_b) + C(Y_b, X_b)$ becomes much smaller than that in the uniform distribution. In short, if $|X| \sim |Y|$, after LDAT $C(X_b, Y_b) + C(Y_b, X_b)$ between areas with the most different density carries the smallest value. So in Figure 3.5(a) it leads to the cutting line to be much closer to $L3$. It means that LDAT alleviates the density bias of NCut. In the special case of Figure 3.6, the cutting line with the most different density distribution on the two sides is $L3$, therefore after LDAT, NCut also tends to cut along $L3$ other than $L1$ and $L2$. □

In Figure 3.7 the positive effect of LDAT is quite evident when applied to the conventional spectral algorithms (RWC). Compared with the simple RWC result in Figure 2.2(c), RWC+LDAT boosted performance more than 25% (Figure 3.7(b)), which will be further verified in Section 3.6. Note that since LDAT is an straightforward transformation of affinity matrix, it can work well with any type of similarity kernels, which gives rise to a novel feature compared with other methods only rely on Euclidean space [160] [153] [30].

However, LDAT has a strong assumption that the affinity matrix W contains sufficient and accurate neighborhood information. Although this requirement can be easily satisfied for those small and simple datasets, it becomes quite challenging when the datasets are large and complex (with high-dimensions). Especially if we only keep k-nn neighborhood for each instance, the constrained information will be highly vulnerable to the change of k given only simple kernel function like Gaussian kernel (see blue curve in Figure 3.8). In our research we perform LDAT on top of AHK, in order to reflect both manifold-aware and density-aware structure. In the next subsection we will describe and further analyze the combined framework.

3.5 The Proposed Framework for Clustering

We have introduced a robust heat-diffusion-based kernel (AHK, Section 3.3) and a local-density-bias-corrected affinity transformation (LDAT, Section 3.4). We now incorporate these two techniques into a systematical framework to provide a more effective and powerful clustering algorithm, as documented in Algorithm 4.

ALGORITHM 4: AHK+LDAT-Clustering(X, c, k)

Input: Input data $X \in R^{n\times m}$ where n is #instances and m is #features, c is #clusters, and k is the neighborhood size.

Output: Cluster assignments of n instances.

Construct the AHK affinity matrix $\mathcal{H}_{\text{lbn}}$ as in Algorithm 2;
Set the diagonal of $\mathcal{H}_{\text{lbn}}$ to be zero to avoid over-diffusion ;
Only keep the k-nn affinity information for each instances ;
Transform the $\mathcal{H}_{\text{lbn}}$ using LDAT as in Algorithm 3 ;
Compute the first c eigenvectors ψ, $\psi = \{\psi_1, \psi_2, \ldots, \psi_c\}$;
Extract the first c nontrivial eigenvectors ψ, $\psi = \{\psi_1, \psi_2, \ldots, \psi_c\}$;
Re-normalize the rows of $\psi \in R^{n\times c}$ into $Y_i(j) = \psi_i(j)/(\sum_l \psi_i(l)^2)^{1/2}$;
Run k-means with c and $Y \in R^{n\times c}$.

Here AHK stably provides sufficient and accurate affinity information of the high-dimensional datasets with complex distribution. Therefore AHK supplies a sturdy platform for the subsequent LDAT. LDAT is also extremely crucial, due to its nice properties of precisely considering the local density and cleanly separating the boundary instances between clusters with diverse densities. It makes corrections and improvements to the traditional NCut-based spectral clustering. By systematically integrating AHK and LDAT, we set up a robust manifold-and-density-aware clustering algorithm. Its running time is $O(n^3 + n^2 log^2 n)$ when the eigen-decomposition method [112] is used.

Although AHK is quite effective in recognizing manifolds even under skewed conditions, it has a negative side-effect of getting over-connection due to the natural properties of random walk. This side-effect might lead to an undesirable uniform affinity since an infinite number of pathes between any two instances tend to draw them deceivably close to each other without proper control. To mitigate this problem, we only use the off-diagonal terms while ignoring the diagonal term of AHK

(Step 2 of Algorithm 4). It avoids the infinite diffusion getting lost on those instances with a lot of connections, as their degrees are enormous [101].

In order to show the positive effect of AHK and LDAT respectively, we perform case study separately.

1. First of all, AHK helps to improve the manifold reconstruction quality of LDAT. Compared with RWC+LDAT, we test AHK+LDAT on the synthetic example of Figure 2.2(a), the result is shown in Figure 3.7(c). It can be observed that AHK can directly helps the subsequent LDAT to gain more cluster-aware separation (by clearly separating the blue and green clusters from the red one), and obtain $7\%+$ performance increment compared with RWC+LDAT. Moreover, AHK+LDAT demonstrates more stable performance than RWC+LDAT as the neighborhood size k changes, which is shown in Figure 3.8.

2. Secondly, to confirm AHK+LDAT's superiority on AHK alone, we test AHK+LDAT on the 20ngB dataset and Figure 3.2(i) shows the corresponding manifold reconstruction. Clearly LDAT helps to make the intra-cluster instances more condense to their cluster center. It boost the performance more than 7% compared with AHK alone.

Supplementary experiments on real benchmark datasets will be presented in Section 3.6 to further support the above discovery.

3.6 Experimental Results and Quantitative Analysis

In this subsection we analyze and verify our proposed AHK, LDAT, and the combination of AHK+LDAT in terms of clustering effectiveness and robustness.

3.6.1 Experimental Setup

Datasets. We verify the effectiveness of our proposed methods by evaluating on eighteen benchmark datasets (Table 3.1) with three types of affinity constructions. Gaussian kernel is used for the first seven datasets from UCI. Network connectivity with undirected edges (all weighted as 1) are applied to the next five datasets

Table 3.1: Statistics of our evaluation datasets.

	Data Set	# inst.	# attr.	# clus.
1	Wine	178	13	3
2	Glass	214	9	6
3	Vehicle	846	18	4
4	Vowel	990	11	11
5	Yeast	1484	8	10
6	Images	2100	19	7
7	Pendigits	3498	16	10
8	Polbooks	105	105	3
9	UMBC	404	404	2
10	MSP	1067	1067	2
11	Citeseer	2114	2114	6
12	Cora	2485	2485	7
13	20ngA	400	400	2
14	20ngB	600	600	3
15	20ngC	800	800	4
16	RCV1-2	1600	29992	2
17	RCV1-3	2400	29992	3
18	RCV1-4	3200	29992	4

ranging from political blogs to scientific paper citation domains. The last six text datasets use cosine similarity: the first three are the subsets of 20 Newsgroup [77] and the last three are the subsets of RVC1 [85]. To reduce sampling bias, we randomly samples from different clusters to make each (sub)datasets 20 times, and record the average clustering performances. 20ngA contains 200 documents from misc.forsale and soc.religion.christian. 20ngB adds 200 documents from talk.politics.guns to 20ngA. 20ngC adds 200 documents from rec.sport.baseball to 20ngB. RCV1-2 contains 1600 documents, among them 800 from C15 and another 800 from ECAT. We add 800 documents from GCAT to RCV1-2 to create RCV1-3. RCV1-4 has 800 more documents from MCAT upon RCV1-3.

Baselines. Eight popularly used clustering algorithms are chosen for comparison: symmetric normalized spectral clustering (NJW) [109] and random walk spectral clustering (RWC) [103] are chosen since they are the classic spectral clustering

algorithms. RWC+SNN is RWC algorithm built upon shared nearest neighbors (SNN) [75] [46] to update similarity. Additionally, we choose Graph Degree Linkage (GDL) [159] as the representative of recent graph-based methods with agglomerative (or hierarchical) modeling. Density-based spatial clustering of applications with noise (DBSCAN) [48] [140] is a density-driven clustering algorithm because it finds a number of clusters starting from the estimated density distribution of corresponding instances. Self-Tuning (ST) spectral clustering [156] and local density adaptive similarity clustering (SCDA) [160] are two locally adaptive clustering methods which adjust classification according to different neighborhood density measurements. We also select the k-nn Diffusion Maps clustering (kDM) [28] as another candidate since it is a diffusion-based algorithm with robustness on noise perturbation.

Evaluation Metrics. Since we have the ground truth of the clustering label information for each dataset, we compare the clustering results against the true labels. We use normalized mutual information (NMI) [132] as the evaluation metric due to its popularity and its information-theoretical interpretation. Suppose $S \in R^{n\times 1}$ is the result label vector for all data instances generated by one particular clustering algorithm and $T \in R^{n\times 1}$ is the true label vector. The NMI score is calculated as follows:

$$NMI(S,T) = \frac{I(S;T)}{\sqrt{H(S) \times H(T)}}, \tag{3.27}$$

where $H(S)$ and $H(T)$ are the entropies, and $I(S;T)$ is the mutual information between S and T. The NMI score is normalized by their entropies and it ranges from zero to one where the larger score indicates the better clustering result.

Parameter Settings. As most of the spectral clustering algorithms assume that the number of clusters c is known a priori, so do our algorithms. Our proposed methods have three parameters: Gaussian scaling factor σ (if Gaussian kernel is used to construct AHK), the size of neighborhood k (to control k-nn connections), and the reduction parameter α in LDAT.

Gaussian scaling factor σ is also used in the other included clustering competitors. To obtain an adaptive parameter and at the same time preserve local density

information, we compute the average distance between each instance to its q-nearest neighbors, and use this value (noted as σ_q) to set the Gaussian scaling parameter. In the remaining subsection, we will test the algorithm performance with different q in the range of $[2, 50]$, with 1 as the step size. When we test the stability of algorithms against the other two parameters k and α (Section 3.6.5 and 3.6.6), we fix $q = 2$ by default.

For a proper neighborhood size k, we set its value as half of the average cluster size $k = n/(2c)$, where n is the number of instances in a dataset and c is the number of clusters. We assume that this is a safe choice for each instance to assemble its true local density. In Section 3.6.5 we will further test the algorithm sensitivity to k in the range of $[10\%, 100\%]$ of $\frac{n}{c}$ with 10% as step size to verify the rationality of $k = 50\%$.

In Section 3.4, we already discussed the effect of α on LDAT. In our general experiments, we use $\alpha = 1$ by default but we will also test RWC+LDAT and AHK+LDAT with different value of $\alpha \in [0, 2]$ in Section 3.6.6.

For DBSCAN experiments, we set Eps, the neighborhood radius, in the same way as we set σ_q. We assign $minPts$, minimal number of instances considered as a cluster, in the range of $[10, min(n/c, 300)]$, and only record the best result among them.

3.6.2 General Comparison with Different Affinity Constructions

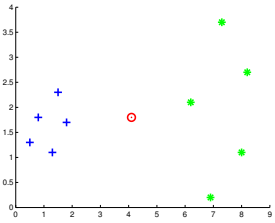
Table 3.2 and 3.3 summarize the clustering performance of seven algorithms: RWC, kDM, ST, SCDA, NJW, DBSCAN, GDL, and our proposed AHK+LDAT. Specifically, for the first seven datasets using Gaussian kernel, we document the best performance across $q \in [2, 50]$ for each algorithm. SCDA and GDL are only defined on Euclidean distance/space and therefore could not work with network connection and cosine similarity.

Generally speaking, AHK+LDAT outperforms those selected algorithms across the three commonly-used affinity construction methods. In Table 3.2 our AHK+LDAT shows the best average score. 1) **Gaussian kernel**: AHK+LDAT obtains 0.4856 average NMI which is 13.09% higher than the second best algorithm

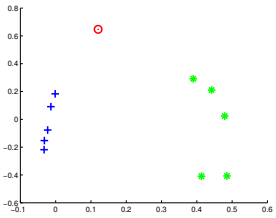
NJW, and 16.14% better than RWC. 2) **Network connectivity**: the average NMI of AHK+LDAT reaches 0.4551, which is 8.18% higher than the second best algorithm (NJW), and 110.89% better than RWC. Table 3.3 shows that for datasets with an increasing number of clusters, our AHK+LDAT on **cosine similarity** has better and more stable performance than the other algorithms. In particular, AHK+LDAT outperforms the second best method kDM by 78.16%.

Our AHK+LDAT either has the best or ranks top three over all the candidate algorithms for each dataset. The only two exceptions are Polbooks and UMBC. However, for UMBC our AHK+LDAT score is more than 95% of the best score from kDM. As for Polbooks, though the NMI score of AHK+LDAT is only about 92% of the best performance, the dataset size is quite small. Intuitively, the density distribution and its variation on such a small dataset is not obvious due to the small sample size v.s. that of a larger one.

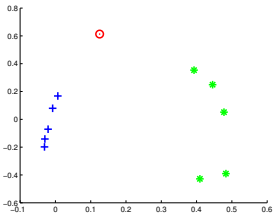
Compared with the other algorithms, DBSCAN fails miserably since it is mainly defined in Euclidean space and suffers from the "curse of dimensionality" and lack of manifold awareness. GDL, ST and SCDA, although based on the theory that supports local density adaptation, are unable to maintain desirable performance across all the datasets, which is mainly caused by their suboptimal local density approximations. Originated from diffusion equations, kDM shows its stability on all the three types of datasets/kernel functions. NJW has comparable performance on Table 3.2 but not Table 3.3, partially due to that it does not have any correction for local density bias.



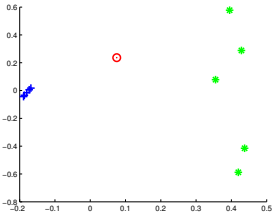
(a) Original dataset (same as Figure 2.3)



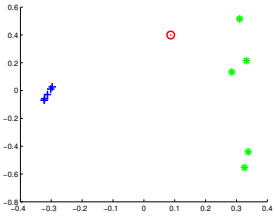
(b) Eigenspace by GAU



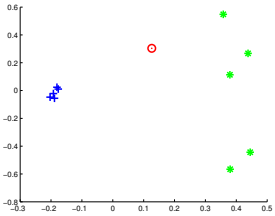
(c) Eigenspace by GAU and SNN



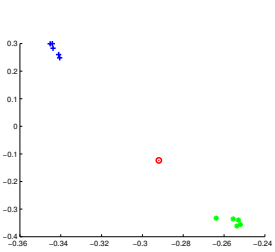
(d) Eigenspace by GAU and Equation 3.11



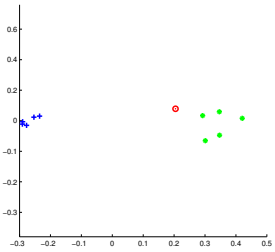
(e) Eigenspace by GAU and Equation 3.11 and 3.13



(f) Eigenspace by GAU and and LDAT



(g) Eigenspace by AHK



(h) Eigenspace by AHK and LDAT

Figure 3.3: 2D Eigenspace derived from the (transformed) affinity matrix of the previous synthetic example (Figure 3.3(a)), with only the first two non-trivial eigenvectors being plotted. Here we only focus on the relative distances. The eigenspace derived from Gaussian similarity (GAU) is shown in Figure 3.3(b), while the one from shared nearest neighbors (SNN) on GAU is shown in Figure 3.3(c). The relative density between blue and green cluster doesn't change much since the projection has no probabilistic transition. Figure 3.3(d) to 3.3(f) show the effect of the three steps in our proposed LDAT built GAU. The blue cluster becomes denser after probabilistic transition. Our proposed AHK in Figure 3.3(g) makes the inner-cluster points even more condense. The combination of AHK+LDAT in Figure 3.3(h) draws the red point into the green cluster.

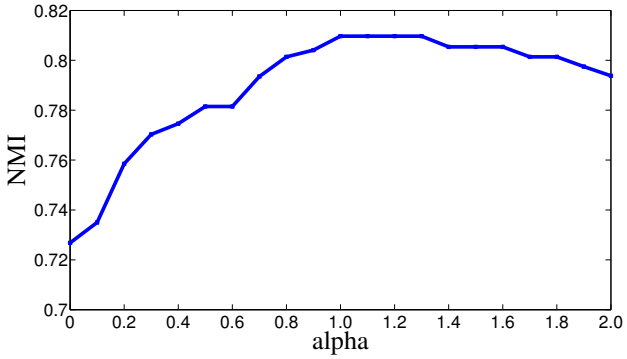
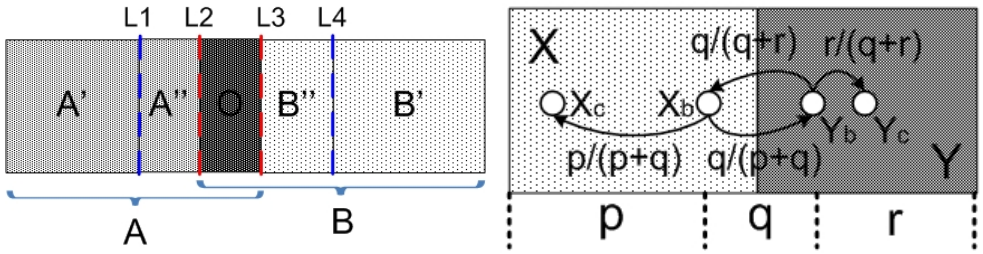


Figure 3.4: RWC+LDAT performance on the dataset of Figure 2.2(a) with reduction factor $\alpha \in [0, 2]$ in Equation 3.12.



(a) (b)

Figure 3.5: In Figure 3.5(a), A and B are two overlapping subsets and A is denser than B. O is the overlapping part and apparently has the highest density. Suppose there is only one cut, L3 would be the best choice to maintain uniform inner-cluster density distribution. Traditional NCut fails to cut along L3 as proven in Proposition 1. But LDAT can correct the density bias of NCut and cut along L3, which is proven in Proposition 2. Figure 3.5(b) shows the connections in the boundary area between two adjacent sets X and Y. The average density of X and Y are p and r respectively. The density of boundary area is q. The change of connection weight before and after LDAT is analyzed in the proof of Proposition 1 and 2.

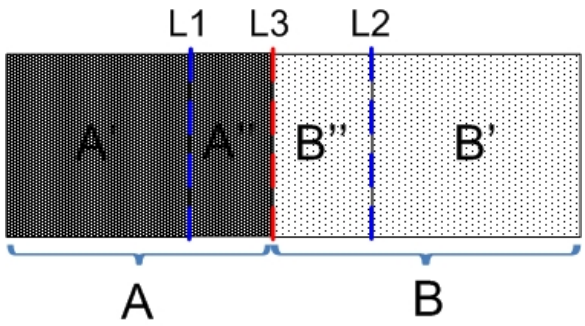
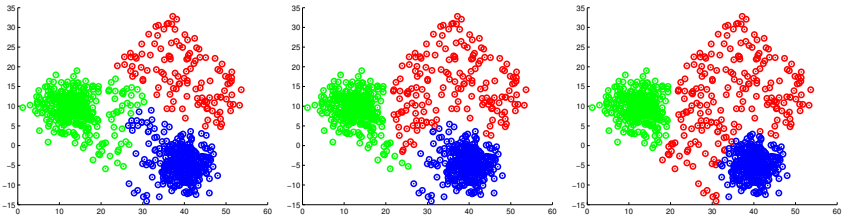


Figure 3.6: Special case: A and B are two adjacent but non-overlapping subsets and A is denser than B. In order to maintain uniform inner-cluster density distribution, L3 is the best cut. Traditional NCut fails to cut along L3 as proven in Proposition 1. But LDAT can correct the density bias of NCut and cut along L3. The effect of LDAT is proven in Proposition 2.



(a) RWC+SNN ($W^{(GLS)}$ with $\sigma_{(G)} = 2$), $NMI = 0.6686$ (b) RWC+LDAT ($W^{(GLS)}$ with $\sigma_{(G)} = 2$), $NMI = 0.7555$ (c) AHK+LDAT ($W^{(GLS)}$ with $\sigma_{(G)} = 2$), **NMI=0.8097**

Figure 3.7: Figure 3.7(a) and 3.7(b) show the effect of shared nearest neighbor (SNN) and our proposed LDAT, both built upon $W^{(GLS)}$ and a positive random walk normalization (RWC). It demonstrates LDAT's advantage of better recognizing density differences among clusters than SNN and other algorithms shown in Figure 2.2 and 2.4. The LDAT built upon AHK, shown in 3.7(c), has the best NMI result through being aware of both density and manifold structures.

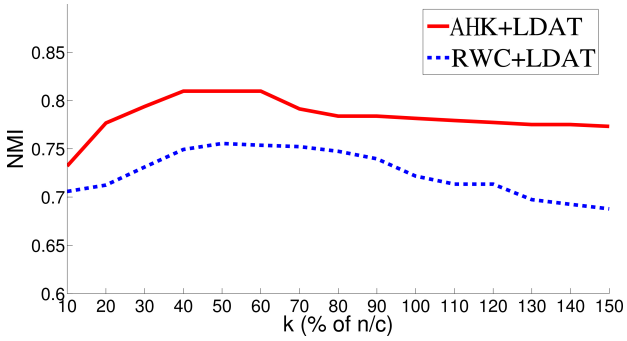


Figure 3.8: LDAT performance on the dataset in Figure 2.2(a) with different neighborhood size k. k is set as the percentage of n/c (n is #instances and c is #clusters). AHK+LDAT has better and more stable performance than RWC+LDAT as k changes.

Table 3.2: Comparison of NMI between AHK+LDAT and the other seven methods on twelve datasets. Experiments with the first seven datasets (from Wine to Pendigits) make use of the Gaussian kernel with σ_q ($q \in [2, 50]$), and the best score across all the q settings is shown for each algorithm and dataset. Experiments with the datasets from Polbooks to Cora use network connectivity. The bold-faced numbers indicate the best method for a particular dataset. The numbers in parentheses are the rankings of the corresponding methods. $AVG^{(GAU)}$ and $AVG^{(NET)}$ are the average NMI of the algorithms using Gaussian kernel and network connectivity respectively.

Dataset	RWC	kDM	ST	SCDA	NJW	DBSCAN	GDL	AHK+LDAT
Wine	0.4355 (6)	0.4421 (4)	**0.4604** (1)	0.4179 (7)	0.4375 (5)	0.2341 (8)	**0.4604** (1)	0.4493 (3)
Glass	0.3615 (8)	0.3858 (5)	0.3813 (6)	0.4054 (3)	0.3686 (7)	**0.4445** (1)	0.4018 (4)	0.4325 (2)
Vehicle	0.1876 (3)	0.1492 (4)	0.0964 (5)	0.0901 (7)	0.1968 (2)	0.0745 (8)	0.0964 (5)	**0.2476** (1)
Vowel	0.4109 (5)	0.4249 (2)	0.4094 (6)	0.3196 (7)	0.4205 (4)	0.0983 (8)	0.4206 (3)	**0.4351** (1)
Yeast	**0.3019** (1)	0.2876 (2)	0.2694 (5)	0.2701 (4)	0.2619 (6)	0.0587 (8)	0.2591 (7)	0.2811 (3)
Images	0.5581 (4)	0.5958 (2)	0.4321 (7)	0.4981 (5)	0.5887 (3)	0.3296 (8)	0.4466 (6)	**0.6746** (1)
Pendigits	0.7131 (4)	0.7129 (5)	0.5972 (7)	0.7056 (6)	0.7315 (3)	0.5046 (8)	0.7858 (2)	**0.8787** (1)
$AVG^{(GAU)}$	0.4241 (4)	0.4283 (3)	0.3780 (7)	0.3867 (6)	0.4294 (2)	0.2492 (8)	0.4101 (5)	**0.4856** (1)
Polbooks	**0.5862** (1)	0.5745 (2)	0.5629 (3)	—	0.5423 (4)	0.4325 (6)	—	0.5402 (5)
UMBC	0.0241 (6)	**0.7488** (1)	0.7375 (2)	—	0.7375 (2)	0.1136 (5)	—	0.7151 (4)
MSP	0.0114 (5)	0.0114 (5)	0.0231 (4)	—	0.0541 (3)	0.0887 (2)	—	**0.2004** (1)
Citeseer	0.3139 (5)	0.3628 (3)	0.3524 (4)	—	0.3728 (2)	0.2073 (6)	—	**0.3871** (1)
Cora	0.1434 (6)	0.2551 (3)	0.2051 (5)	—	0.3966 (2)	0.2103 (4)	—	**0.4325** (1)
$AVG^{(NET)}$	0.2158 (5)	0.3905 (3)	0.3762 (4)	—	0.4207 (2)	0.2103 (6)	—	**0.4551** (1)

Table 3.3: Comparison on text datasets, each of which has an increasing number of clusters. The bold-faced numbers indicate the best method for a particular dataset. The numbers in parentheses are the rankings of the corresponding methods. $AVG^{(COS)}$ is the average performance scores.

Dataset	RWC	kDM	ST	SCDA	NJW	DBSCAN	GDL	AHK+LDAT
20ngA	0.0235 (5)	0.0235 (5)	0.0497 (4)	—	0.1002 (3)	0.3456 (2)	—	**0.6916** (1)
20ngB	0.3321 (5)	0.4625 (2)	0.3435 (4)	—	0.3487 (3)	0.2429 (6)	—	**0.7213** (1)
20ngC	0.2438 (5)	0.3368 (2)	0.2958 (3)	—	0.2927 (4)	0.2058 (6)	—	**0.7076** (1)
RCV1-2	0.2204 (5)	0.2249 (4)	**0.4039** (1)	—	0.0417 (6)	0.3134 (2)	—	0.2341 (3)
RCV1-3	0.2172 (6)	0.5220 (2)	0.4039 (4)	—	0.4134 (3)	0.3523 (5)	—	**0.5467** (1)
RCV1-4	0.4546 (3)	0.3638 (4)	0.2889 (6)	—	0.5136 (2)	0.3478 (5)	—	**0.5438** (1)
$AVG^{(COS)}$	0.2486 (6)	0.3223 (2)	0.2635 (5)	—	0.2851 (4)	0.3013 (3)	—	**0.5742** (1)

3.6.3 Respective Effect of AHK and LDAT

To verify the effect of AHK and LDAT respectively, we document the experiments of RWC, RWC+SNN, RWC+LDAT, AHK, and AHK+LDAT in Table 3.4.

1. For the data experiments using Gaussian kernel, AHK increases NMI by 5.43% compared with RWC, while LDAT boosts up 13.90%. Compared with RWC+SNN, RWC+LDAT raise 3.12%. Therefore here the effect of LDAT is more obvious than that of AHK. Therefore although both using LDAT, AHK+LDAT is only 2% better than RWC+LDAT. And AHK+LDAT outperforms AHK by about 20%.

2. For those network datasets, AHK increases 93.47% while LDAT only raises about 2% over RWC. It means AHK helps a lot here: AHK+LDAT raises 9% compared with AHK only, but on the other hand AHK+LDAT is 107% better than RWC+LDAT.

3. In the experiments of text dataset, AHK enhances performance by 99.68% and LDAT 86.89% when both of them are compared with RWC. AHK+LDAT outperforms RWC+LDAT by 23.59%, and outperforms RWC+SNN by 26.23%. On the other hand, AHK+LDAT also outperforms AHK by about 15.67%.

In short, both AHK and LDAT can improve RWC, but in different aspects: AHK alleviates the clustering sensitivity to the scaling parameter and data perturbation, while LDAT provides more insight to the density change across different clusters. Table 3.4 shows that RWC+LDAT and AHK all increase RWC's performance. When both AHK and LDAT are applied, AHK+LDAT obtains the best performance in general.

Table 3.4: Comparison between RWC, RWC+SNN, RWC+LDAT and AHK, AHK+SNN and AHK+LDAT. Experiments with the first seven datasets (from Wine to Pendigits) make use of the Gaussian kernel with σ_q ($q \in [2, 50]$), and the best score across all the q settings is shown for each algorithm and dataset. Experiments with the datasets from Polbooks to Cora use network connectivity. Cosine kernel is applied on the last six text datasets. The bold-faced numbers indicate the best method for a particular dataset. The numbers in parentheses are the rankings of the corresponding methods. $AVG^{(GAU)}$, $AVG^{(NET)}$ and $AVG^{(COS)}$ are the average NMI of the algorithms using Gaussian kernel, network connectivity and cosine kernel respectively.

Dataset	RWC	RWC+SNN	RWC+LDAT	AHK	AHK+SNN	AHK+LDAT
Wine	0.4355 (5)	0.4375 (3)	0.4375 (3)	0.4244 (6)	0.4417 (2)	**0.4493** (1)
Glass	0.3615 (6)	0.4575 (2)	**0.4618** (1)	0.4266 (4)	0.4067 (5)	0.4325 (3)
Vehicle	0.1876 (6)	0.2031 (4)	0.2173 (3)	0.2262 (2)	0.1880 (5)	**0.2476** (1)
Vowel	0.4109 (4)	0.3725 (6)	0.4321 (3)	**0.4432** (1)	0.4008 (5)	0.4351 (2)
Yeast	**0.3019** (1)	0.2974 (2)	0.2763 (4)	0.2571 (6)	0.2756 (5)	0.2811 (3)
Images	0.5581 (6)	0.6577 (3)	**0.6926** (1)	0.5751 (5)	0.5991 (4)	0.6746 (2)
Pendigits	0.7131 (6)	0.8068 (3)	0.8157 (2)	0.7328 (5)	0.7701 (4)	**0.8787** (1)
$AVG^{(GAU)}$	0.4241 (6)	0.4618 (3)	0.4762 (2)	0.4408 (4)	0.4403 (5)	**0.4856** (1)
Polbooks	**0.5862** (1)	0.5745 (3)	0.5667 (4)	0.5833 (2)	0.5401 (6)	0.5402 (5)
UMBC	0.0241 (5)	**0.7488** (1)	0.0241 (5)	0.5927 (4)	0.5942 (3)	0.7151 (2)
MSP	0.0114 (6)	0.0756 (4)	0.0124 (5)	0.1383 (3)	0.1580 (2)	**0.2004** (1)
Citeseer	0.3139 (5)	0.3644 (3)	0.3324 (4)	0.3697 (2)	0.2873 (6)	**0.3871** (1)
Cora	0.1434 (6)	0.2550 (4)	0.1640 (5)	0.4037 (2)	0.3522 (3)	**0.4325** (1)
$AVG^{(NET)}$	0.2158 (6)	0.4037 (3)	0.2200 (5)	0.4175 (2)	0.3864 (4)	**0.4551** (1)
20ngA	0.0235 (6)	**0.7587** (1)	0.7581 (2)	0.7175 (3)	0.7002 (4)	0.6916 (5)
20ngB	0.3321 (6)	0.4609 (4)	0.4428 (5)	0.6724 (2)	0.6501 (3)	**0.7213** (1)
20ngC	0.2438 (6)	0.2754 (5)	0.3588 (4)	0.4659 (3)	0.4699 (2)	**0.7076** (1)
RCV1-2	0.2204 (5)	0.2221 (4)	0.2249 (2)	0.2234 (3)	0.1290 (6)	**0.2341** (1)
RCV1-3	0.2172 (6)	**0.5487** (1)	0.5327 (3)	0.3876 (5)	0.4982 (4)	0.5467 (2)
RCV1-4	0.4546 (6)	0.4635 (3)	0.4701 (5)	0.5117 (4)	0.5307 (2)	**0.5438** (1)
$AVG^{(COS)}$	0.2486 (6)	0.4549 (5)	0.4646 (4)	0.4964 (2)	0.4964 (2)	**0.5742** (1)

3.6.4 Robustness on Adaptive Scaling Parameter (q)

To systematically demonstrate the superior robustness of AHK and AHK+LDAT across different Gaussian scaling parameter q, we test several algorithms on seven datasets: wine, glass, vehicle, vowel, yeast, image, and pendigits datasets. The test range of q is $[2, 50]$ with one as the step size. Figure 3.9 shows the performance of eight algorithms.

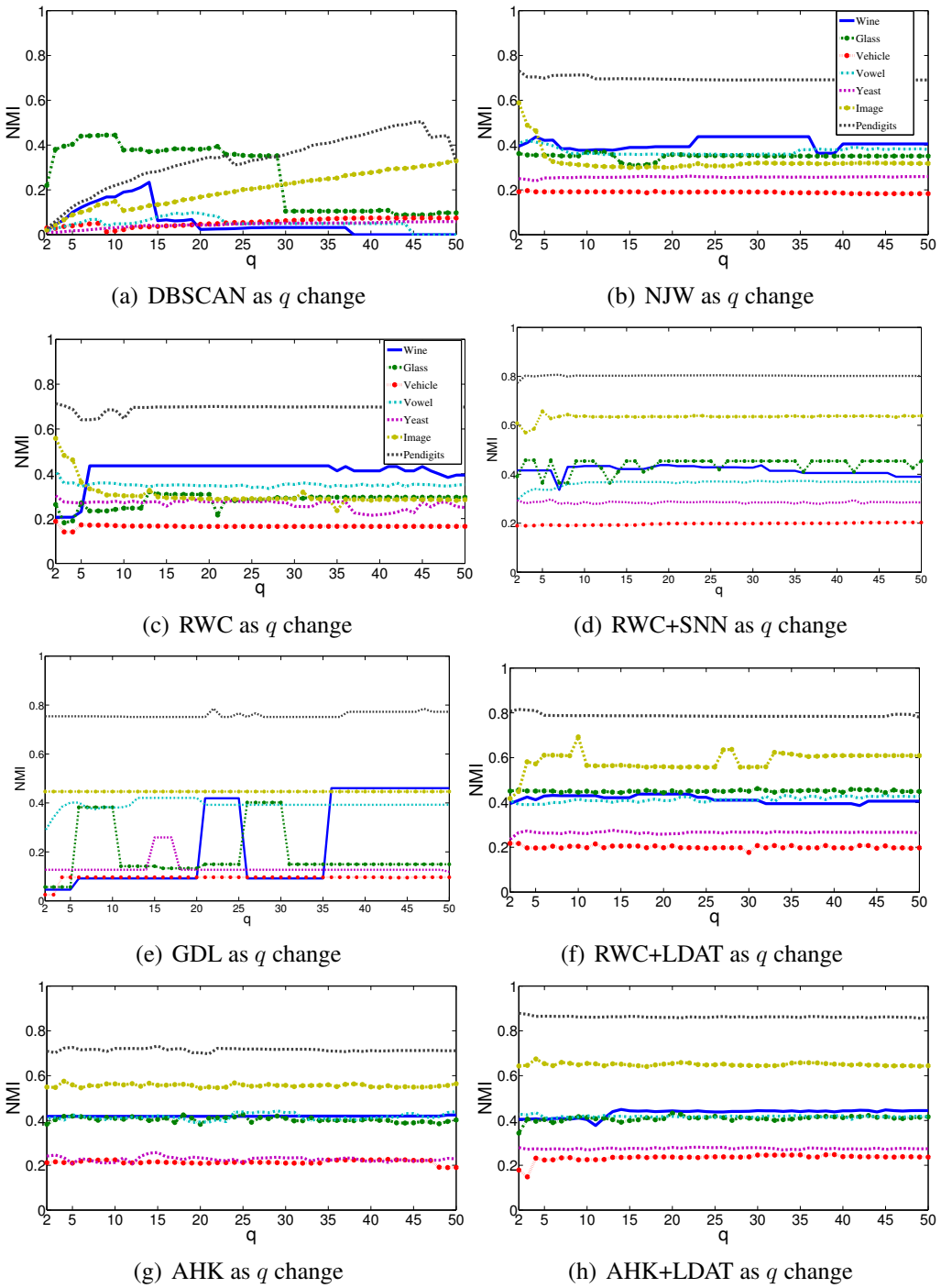
DBSCAN (Figure 3.9(a)) and GDL (Figure 3.9(e)) are extremely unstable and there are little clue about how to tune q in an unsupervised way. With manifold awareness, NJW (Figure 3.9(b)) shows better and more stable clustering performance. Compared with RWC (Figure 3.9(c)), RWC+LDAT (Figure 3.9(f)) achieves higher quality of performances, especially with glass datasets ($41.15\% \uparrow$), yeast ($13.04\% \uparrow$), image ($54.25\% \uparrow$) and pendigits ($7.25\% \uparrow$). On the other hand, compared with RWC+SNN, RWC+LDAT has better performance on wine ($3.03\% \uparrow$), glass ($5.54\% \uparrow$) and vowel ($7.81\% \uparrow$). This confirms that LDAT helps to improve clustering results by providing density awareness in the cluster overlapping area. AHK (Figure 3.9(g)) enhances the clustering stability across q especially on wine, glass and image datasets, but it doesn't necessarily retain the best performance.

Overall, the combination of AHK+LDAT has the best result on average and performs consistently across different values of q. The stability of AHK+LDAT inherently originates from the AHK with LBN, and its outperformance partly comes from LDAT.

3.6.5 Robustness on Neighborhood Size (k)

To reveal the stability with respect to the neighborhood scaling parameter k across datasets, we test $k \in [10\%, 100\%]$ of $\frac{n}{c}$ on kDM, RWC+SNN, RWC+LDAT and AHK+LDAT, the algorithms that built upon k-nn neighborhood. We also report results on wine, glass, vehicle, vowel, yeast, images, and pendigits datasets.

Although it doesn't provide the best performance, kDM (Figure 3.10(c)) indeed has a stable performance across different k since it builds on diffusion map. The peak performance of RWC+LDAT (Figure 3.10(b)) on each dataset mostly surpass the peak performance of kDM because of the power of LDAT. However it fails to sustain stable result across different k. But still RWC+LDAT is more



(a) DBSCAN as q change

(b) NJW as q change

(c) RWC as q change

(d) RWC+SNN as q change

(e) GDL as q change

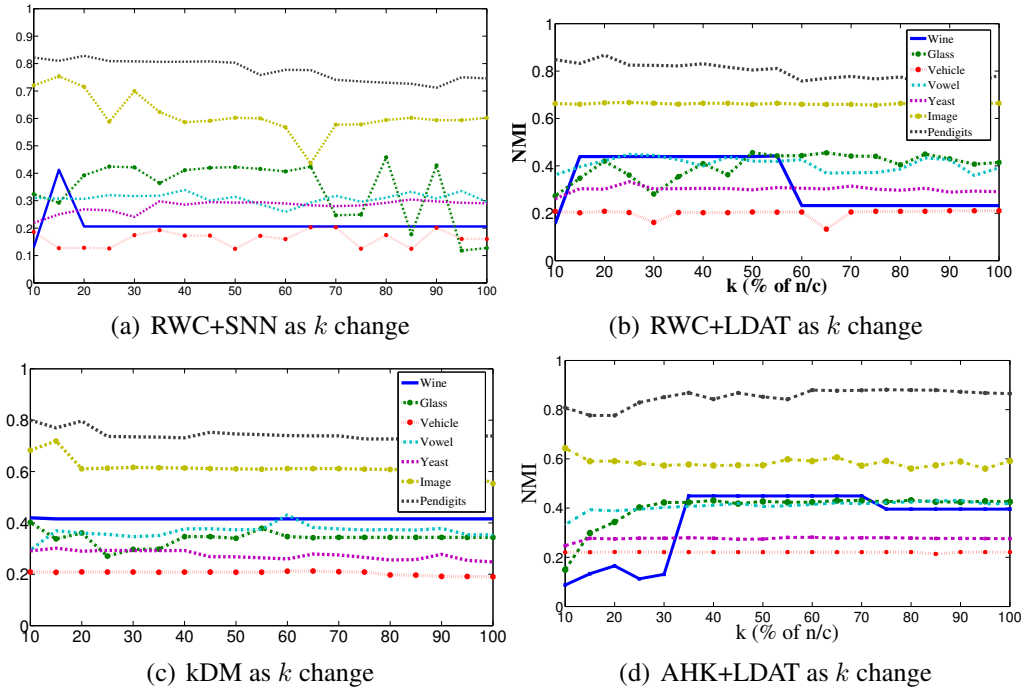
(f) RWC+LDAT as q change

(g) AHK as q change

(h) AHK+LDAT as q change

Figure 3.9: Stability with different adaptive scaling parameter q.

stable than RWC+SNN in that LDAT updates the similarity on a probabilistic interpretation rather than the direct (original) similarity. From the same point of view, AHK+LDAT obviously has better stability than RWC+LDAT since AHK contributes a lot on the manifold-awareness and therefore the whole algorithm has stronger support from statistics. Figure 3.10(d) shows that our AHK+LDAT demonstrates consistently better performance when we choose the neighborhood size $k = 50\% \times \frac{n}{c}$.



(a) RWC+SNN as k change

(b) RWC+LDAT as k change

(c) kDM as k change

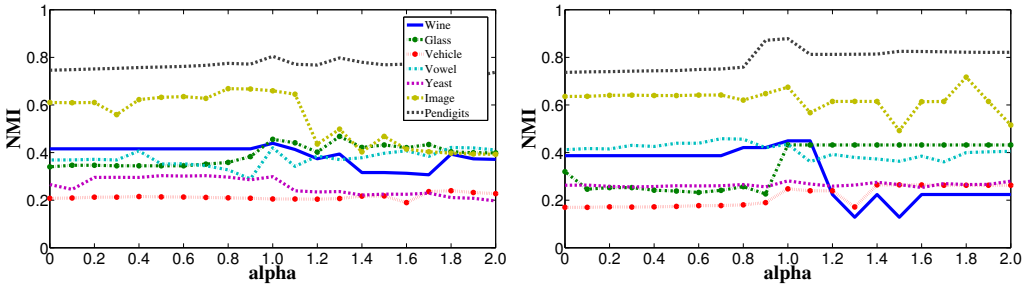
(d) AHK+LDAT as k change

Figure 3.10: Stability under different neighborhood size k.

3.6.6 Robustness on Reduction Degree (α) in LDAT

Figure 3.11 shows the stability of RWC+LDAT and AHK+LDAT when α is being tuned. The integration along all the time scales and LBN provide AHK an advanced random walk process, which leads to better average performance of AHK+LDAT. The only two exceptions are glass and wine, where RWC+LDAT outperforms AHK+LDAT. But if the datasets are getting larger, more sampling points will support a better manifold reconstruction (like image and pendigits).

Generally speaking, $\alpha \in [0.9, 1.1]$ makes LDAT sustainably performs better than RWC (LDAT when $\alpha = 0$) for both RWC+LDAT and AHK+LDAT. Therefore by default we recommend α to be one.



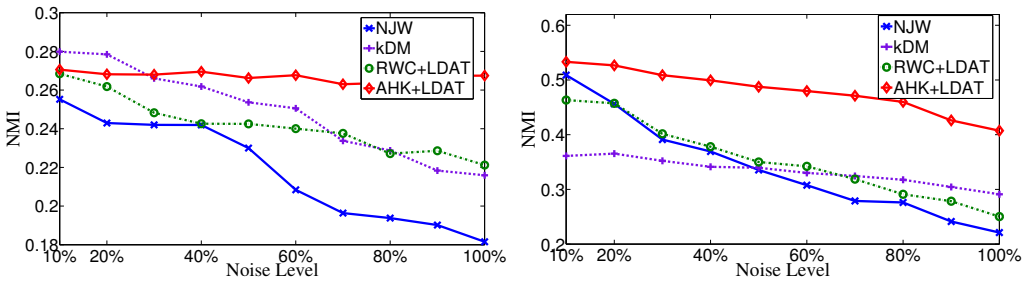
(a) RWC+LDAT as α change (b) AHK+LDAT as α change

Figure 3.11: Stability under different reduction factor α.

3.6.7 Robustness to Different Noise Levels

We conduct experiments on controlled noisy datasets to examine the performance of our algorithms and make comparison with the other algorithms. The yeast and RCV1 datasets are used for this experiment. We added uniformly-distributed noise to create more datasets, each of which has a different percentage of noise, i.e. 0%, 10%, 20%, $\cdots$, 100%. To study the robustness against noise comprehensively and avoid tuning scaling parameter q for the best case scenario, we iteratively evaluate with all possible values for the scaling parameter q and record the average performance result. The experimental results are displayed in Figure 3.12.

Overall, AHK+LDAT shows both robust and better performance across different noise conditions. Although RWC+LDAT shows better performance than NJW for most cases, its effectiveness decreases dramatically and demonstrates a similar trend to that of NJW. Similar to AHK+LDAT, kDM shows stable performance across different noise percentages, occasionally even outperforms AHK+LDAT in the case of yeast dataset with a small percentage of noise.



(a) Yeast with different noise percentages (b) RCV1-4 with different noise percentages

Figure 3.12: Algorithm performance on different noise levels.

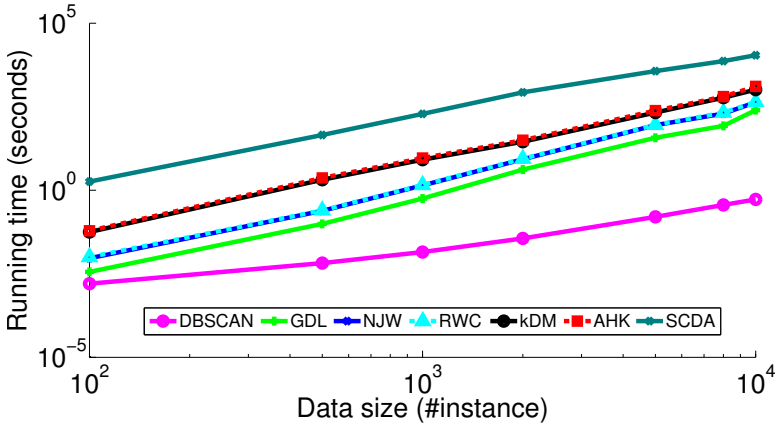


Figure 3.13: Scalability Analysis.

3.6.8 Scalability Analysis

This subsection analyzes the algorithm scalability. The experiment was done on a 2.3 GHz Intel Core $i7$ processor with 8 GB 1600 MHz DDR3 memory.

Figure 3.13 shows that SCDA is the most time-consuming algorithm, which is not surprising since it requires more time to compute the joint region of the ϵ-neighborhoods. On the other hand, DBSCAN is the most efficient algorithm in time with R* tree and a non-matrix based implementation. NJW and RWC are in need of eigen-decomposition, therefore their running time is more than GDL. Our proposed AHK, similar to kDM, also requires second construction of similarity matrix and another eigen-decomposition, so their scalabilities are worse than NJW and RWC, but still better than SCDA.

3.7 Chapter Summary

This chapter presents a novel clustering algorithm that seamlessly integrated two robust and effective techniques, i.e. Aggregated Heat Kernel(AHK) and Local Density Affinity Transformation (LDAT). Consequently, our proposed approach achieved remarkable performance improvements for those datasets with heterogeneous density distributions. Three primary advantages of our work are: (1) Its manifold reconstruction is robust to the scaling parameter tuning and noise appearance; (2) It alleviates local density bias in Normalized Cut; and (3) It functions well with any affinity measurement, and is universally applicable. Our comprehensive experiments validate that the proposed algorithm outperforms the majority of the existing clustering algorithms.

Chapter 4

Physics-based Anomaly Detection Defined on Manifold Space

4.1 Chapter Introduction

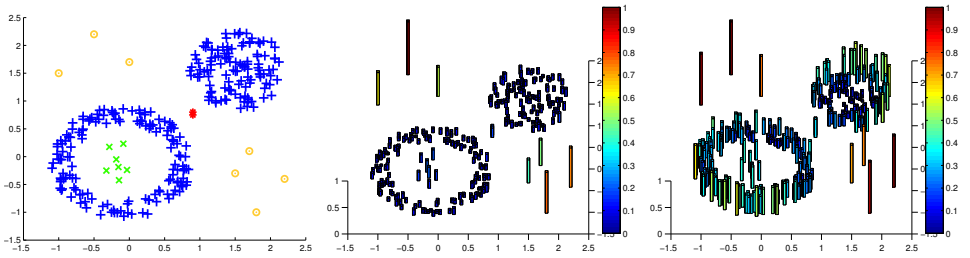
Anomaly detection, or outlier detection, is of great significance to many real world applications [163] [116], such as cancer diagnostics and virus detection. Its primary goal is to distinguish normal instances from a small portion of new or abnormal instances (anomalies) [19] [96] [97]. In many applications, anomalies are sparse and quite diverse, learning with the known anomalies [50] [151] [12] may not be necessarily useful in detecting the unknown ones in previously unseen data [134]. On the other hand, manually labeling known datasets can be extremely time-consuming for real-life applications and sometimes even unpractical to detect new types of rare events. Therefore, the key challenge of anomaly detection still lies in its ability to quantitatively characterize the intrinsic and informative density distribution around every instance in a unsupervised fashion.

In this chapter we propose two different unsupervised anomaly detection algorithms: Local Anomaly Descriptor (LAD), and Fermi Density Descriptor. They measure instance anomalousness based on different physics theory, i.e. heat diffusion and quantum mechanic theory respectively. Compared with the existing algorithms [14] [113] [96] [138] [2], our methods are capable of measuring local density more effectively in that:

- Our methods have solid physics theory background.
- Our methods are based on manifold space, where the distance between anomalies and normal instances would be magnified. It makes anomalies more salient than in the input space.
- Our methods provide a more adaptive scope of neighborhood, which is of great importance to distinguish not only global but also local anomalies from normal instances.
- Our methods are highly desirable to combat scaling parameter tuning sensitivity.

These properties make our algorithms more **informative** and **intrinsic** to detect anomaly.

4.1.1 Related Works



(a) Synthetic Dataset Example (b) LOF Score ($k = 10$) (c) IForest Score ($nt = 100$, $\rho = 4000$)

Figure 4.1: Synthetic dataset is shown in Figure 4.1(a) with normal instances (blue), global anomalies (yellow), and local anomalies (red and green). Figure 4.1(b) LOF score with $k = 10$. 4.1(c) IForest score. The anomalousness are visualized as height bar over all the instances. For each algorithm output, the anomalousness scores are normalized in the range of $[0, 1]$ to have an easy comparison. We can see that both LOF and IForest fail to totally distinguish local anomalies from normal instances.

According to the most classical definition by Hawkins [61], an anomaly is "an observation which deviates so much from the other observations as to arouse suspicion that it was generated by a different mechanism". However, it is far from

trivial to define the quantitative sense of "deviates from the other observations". As Figure 4.1(a) illustrates, global anomalies (in yellow) are those data points with low density in the entire data space. We can also say that these points are with globally low neighborhood density. On the other hand, local anomalies (in red and green) are data points with low local density in a constrained region. We call that these points are with locally low neighborhood density. Profoundly speaking, local anomalies can be thought of as a generalization of global anomalies, as global anomalies will typically also be local anomalies, but not vice versa [34].

In implementation, kNN-based algorithms such as LOF [14], LDOF [158], and LOCI [113] are defined on Euclidean distance. LOF [14], one of the earliest work using kNN distance for anomaly detection, defines anomaly if its distance to its k-th nearest neighbor (kNN) is greatly larger than the distances of its neighbors to their own k-th nearest-neighbors. Recent research [34] extended LOF to high-dimensional dataset by using random projection to reduce dimensions. Two major drawbacks of these approaches are: (1) They tend to miss local anomalies (Figure 4.1(b)) since it is not peculiar that kNN distances of local anomalies are similar to their normal instance neighbors'. (2) It is of extreme importance to determine the value of k to faithfully reveal the instance anomalousness. On one hand, k cannot be too small to avoid statistical error. Specifically, we need to ensure that for each instance, especially those forming micro-cluster of anomalies, it covers a large enough neighborhood that includes more normal instances than anomalies. On the other hand, too large k will lead to overlook some genuine anomalies. In Section 4.8.2 we will show that LOF is unpractical to detect anomalies in benchmark datasets by analyzing its sensitivity of k.

Instead of detecting anomalies based on average neighborhood distance, recent approaches such as IForest [96] [97] and Mass [138] are to separate the anomalies from normal instances with their noteworthy attribute distribution. A representative anomaly definition [96] in these papers states that anomalies should have "attribute-values that are very different from those of normal instances", and at the same time should be "minority consisting of fewer instances". Therefore these approaches have the capacity to handle anomalies with different attribute distribution compared with normal instances. Nonetheless, they may fail to detect local anomalies when their attributes have not-so-different distribution with some normal

instances'. From Figure 4.1(c) we can see that, even though IForest does a good job on global anomaly detection, it fails to distinguish local anomalies (green and red instances in Figure 4.1(a)) from the "boundary" instances in the cluster of normal instances (blue instances in Figure 4.1(a)). The reason is that, these anomaly detectors partition instances mainly based on observable attributes, or more precisely, the attribute distribution in input data space. Therefore it will fail miserably when the anomaly distribution becomes far less discriminative if they share similar attribute range/distribution pattern with parts of the normal instances. In Figure 4.2 we can see that some anomalies have overlapping distribution with normal instances on the first four eigenvectors in ionosphere dataset (a popularly used dataset for anomaly detection [96] [63] [110]). Such overlapping also appears at nonclassical multidimensional scaling (MDS) as well. This case, to a certain degree, shows that the aforementioned problem indeed exists in some real world applications.

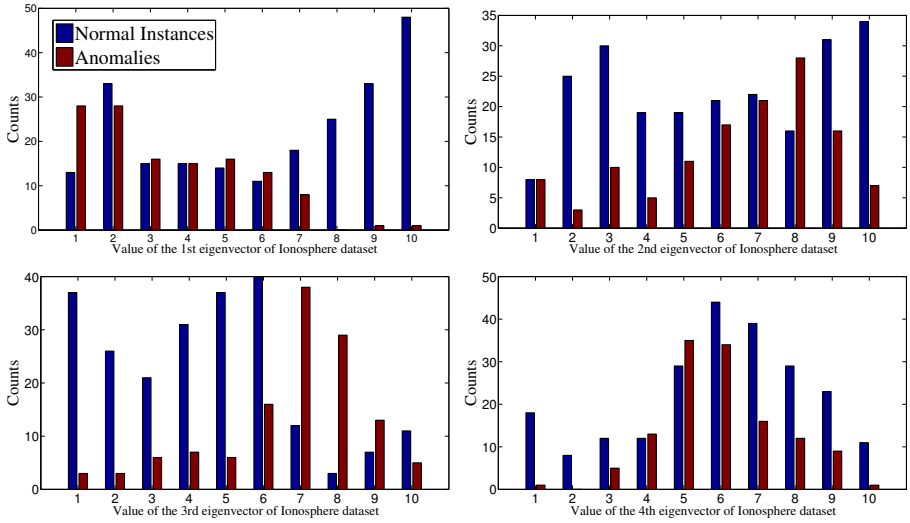


Figure 4.2: Histogram of anomalies (red) and normal instances (blue) on the first four eigenvectors [1] of ionosphere dataset (a popular benchmark dataset for anomaly detection [96] [63] [110]). Some anomalies have overlapped distribution with parts of normal instances and therefore it is nontrivial to separate them simply by difference between attribute distributions.

[1]Since the dataset is high-dimensional, dimension reduction is imperative to provide a concise illustration. Although eigenvectors do not necessarily show full distribution of the input data, they tend to show certain patterns of original dimensions in the input space.

Since the neighborhood density is not as straightforward as pair-wise distance or attribute distribution in the input space, many researches turned to manifold space. In an ideal manifold projection with enlarged distance between anomalous and normal instances, anomaly detection is no longer as hard as that in the input space. A few techniques [2] tried to find an approximation of the data using a combination of attributes that capture the bulk of the variability in the data, and then detect anomalies on the projected space. This kind of approaches is to approximate the manifold subspaces in which the anomalous instances can be easily identified [19]. However, the existing algorithms are based on suboptimal techniques such as isometric feature mapping (ISM) and locally linear embeddings (LLE) [2] which are highly sensitive to density-varying and complex data distribution [84] [142]. Therefore anomaly detection algorithms based on such manifold reconstruction mechanism may fail miserably.

4.1.2 Motivations

Motivated by the aforementioned problems, we refine the definition of anomaly as follows:

Definition: Anomalies are those instances with (1) **locally low neighborhood density** and (2) **small quantity of similar instances** compared with normal instances.

To capture anomalies under such definition, we consider the Laplace operator in physics theory, which has solid foundation and intrinsic relationship with manifold reconstruction. The reason why we resort to manifold space is that normal instances usually lie on low dimensional embedding structures with high density. But the anomalies projected in manifold space tend to deviate from the normal instances which makes them more discriminative. On the other aspect, measurement of anomalousness is highly related to similarity function, in that the anomalousness of an instance is high if it has few of similar neighbors. Laplace operator is a differential operator given by the gradient divergence of a function on Euclidean space. Therefore the Laplace operator, if it is performed on a similarity matrix, is capable of representing the flux density of the gradient flow of the neighborhood similarity.

Consequently, it offers a natural mechanism to express intrinsic neighborhood density information. Furthermore, the Laplace operator occurs in differential equations that describe many physical phenomena, such as the diffusion equation for heat and quantum mechanics. These properties deliver inspiration and solid theoretical foundation to our research in this chapter.

4.1.3 Contributions

This chapter articulates two physics-based unsupervised anomaly detection algorithms with the following contributions:

1. We are the first to quantitatively characterize local density information based on heat diffusion theory (Section 4.2), and develop Local Anomaly Descriptor (LAD, Section 4.4.1). This method has a locally adaptive scope of manifold-aware neighborhood and therefore can very well satisfy the first property of our proposed anomaly definition in Section 4.1.2.

2. In favor of taking the amount of similar instances into account (the second property of the above definition in Section 4.1.2), we integrate scale-dependent umbrella operator (Section 4.4.1) into LAD which can bridge the gap between local and global information.

3. We are the first to explore the use of quantum mechanics theory (Section 4.5) in anomaly detection, and propose Fermi Density Descriptor (FDD, Section 4.6) which supplies rigorous probabilistic explanation for detecting anomalies, and supreme stability to scaling parameter tuning.

4. We firstly analyze different Laplacian normalization effects (Section 4.6.2) with the goal of anomaly detection. Both theoretical proof and quantitative experiments demonstrate that unnormalized Laplacian L_{nn} is the most responsive to local neighborhood density.

5. We explore the use of anisotropic Gaussian kernel (AGK, Section 4.3) which more faithfully approximate the similarity between instances in the ideal manifold space, and therefore can best help in manifold reconstruction with the goal of anomaly detection.

6. We systematically evaluate the proposed algorithms with several closely-related baseline algorithms on a number of benchmark datasets (Section 4.8). Our algorithms show not only better average performance but also more stable results than the other popular algorithms.

4.2 Heat Kernel Signature based on Heat Diffusion

4.2.1 Introduction of Heat Diffusion

Our first proposed algorithm is strongly inspired by heat diffusion theory [68] in that it can provide information intimately related to local density. Heat theory can be interpreted as the transition density function of Brownian motion [133], which is the most fundamental continuous time Markov process. Laplace operator is closely associated to heat diffusion, connecting geometry of a manifold with the properties of the heat flow. Using the discrete Laplace operator, the heat equation can be simplified, and generalized to matrix operation over spaces with an arbitrary number of dimensions. Due to its intrinsic connection to Markov process, in practice the heat equation is often coupled with random walk graph Laplacian [28], L_{rw} (Equation 2.3), which describes a stochastic process that randomly jumps from vertex to adjacent vertex. Heat equation therefore can be defined by:

$$\frac{\partial H_t}{\partial t} = -L_{rw} H_t, \tag{4.1}$$

where $H_t = e^{-tL_{rw}}$ is the heat kernel on Riemannian manifold $\mathcal{M}$ and t is the time scaling parameter [56]. For $L_{rw} = \psi' \lambda \psi$ (ψ and λ are the eigenvectors and eigenvalues of L_{rw}), the heat kernel can be re-formulated as follows:

$$H_t(i,j) = \sum_{p=1}^{N} [e^{-\lambda_p t} \psi_p(i) \psi_p(j)], \tag{4.2}$$

where λ_p is the p-th eigenvalue and $\psi_p(i)$ is the i-th element in the p-th eigenvector. $H_t(i,j)$ represents the amount of heat being transferred from i to j in time t given a unit heat source at i in the very beginning. The scaling parameter t in heat kernel is used to control the transitive connectivity: small t makes the loosely-connected graph into slightly stronger connection, while large t makes the graph tend to be more strongly-connected.

4.2.2 Heat Kernel Signature

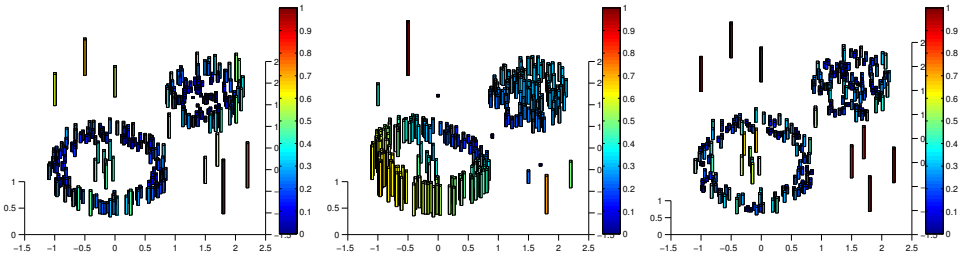
In 2009, Sun et.al [133] proposed a concise form given by the heat kernel from one instance to itself:

$$\mathcal{H}_t(i) = H_t(i,i) = \sum_{p=1}^{N} [e^{-\lambda_p t}(\psi_p(i))^2], \tag{4.3}$$

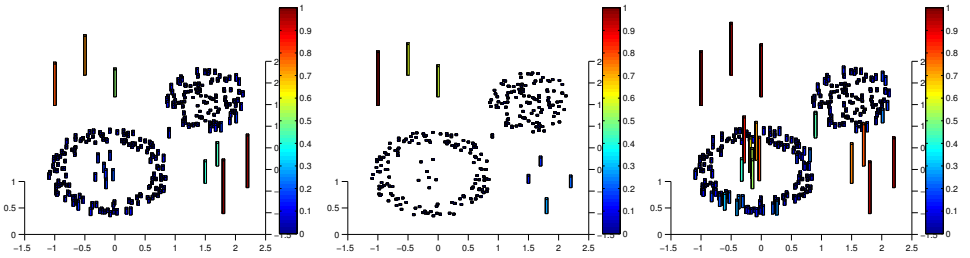
which is called Heat Kernel Signature (HKS). The physical meaning of HKS is the amount of heat each instance keeps within itself in time t. **The property of heat diffusion process states that heat tends to diffuse slower at instances with more sparse neighborhood and faster at instances with denser neighborhood. Therefore HKS can intuitively depict the local density of each instance (the first property in our anomaly definition in Section 4.1.2).** Besides, HKS also has the following properties which make it a very lucrative candidate for local density measurement:

- HKS is intrinsic to the local manifold structure;
- HKS is informative since it contains density information of the whole neighborhood in t scale;
- The stableness of HKS against small perturbation in the neighborhood can be well supported by the probabilistic interpretation of heat diffusion.

However, heat equation is assumed to build on the underlying manifold. But in most applications, the underlying manifold is unknown. In geometric modeling application, HKS is usually built on eigenvectors from Gaussian kernel (GAU, Equation 2.8) on observed space. Although graph Laplacian normalizations [28] based on GAU on observed space can recover manifold structure to certain extent, non-uniformly sampled instances tend to show unpreserved density distribution on the reconstructed manifold. HKS on GAU will fail to reveal local density faithfully in such reconstructions. Figure 4.3(a) and 4.3(b) shows the performance of HKS on anomaly detection with $t = 1$ and $t = 10$ based on GAU and random walk graph Laplacian normalization L_{rw}. When $t = 10$ (Figure 4.3(b)) the heat is extremely easy to dissipate, which blends both local and a few global anomalies into normal instances. Meanwhile many marginal instances of the two normal instance clusters



(a) HKS with GAU($t = 1$) (b) HKS with GAU($t = 10$) (c) LAD with GAU($t = 1$ and $k = 10$)



(d) HKS with AGK($t = 1$) (e) HKS with AGK($t = 10$) (f) LAD with AGK($t = 1$ and $k = 10$)

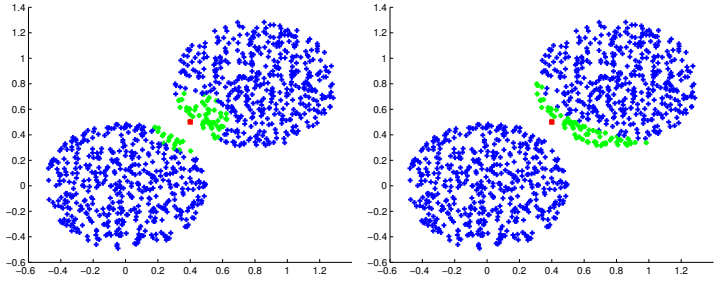
Figure 4.3: HKS and LAD (Local Anomaly Descriptor, Equation 4.10) score with GAU (Gaussian kernel, Equation 2.8) and AGK (Anisotropic Gaussian kernel, Equation 4.7) of the synthetic dataset in Figure 4.1(a). For each algorithm output, the anomalousness scores are normalized in the range of $[0, 1]$ to have an easy comparison. We can see that LAD with AGK is the most aware of both global and local anomalies.

stand out due to the fact that HKS on GAU fails to show manifold-aware properties. When $t = 1$ (Figure 4.3(a)), although the short period of heat dissipation has salient effect on global anomalies, HKS on GAU still fails to distinguish local anomalies from normal instances on the boundary area of normal clusters. Therefore an alternative way is indispensable to build better manifold-aware affinity matrix. One of the most preferable candidates is anisotropic Gaussian kernel (AGK) [127] [128].

4.3 Anisotropic Gaussian Kernel

In this subsection we use anisotropic Gaussian kernel (AGK) [127] to construct HKS in the interest of better manifold reconstruction. In Figure 4.4 we can see the

70 nearest neighbors of red instance when using GAU (Figure 4.4(a)) and AGK (Figure 4.4(b)), which shows that the intra-manifold distances are much shorter than the inter-manifold by using AGK. Figure 4.3(d) and 4.3(e) show that anomaly detection can directly benefit from the use of AGK. In Figure 4.3(e) with $t = 10$, all of the global anomalies are highlighted even though the local anomalies are latent (compared with Figure 4.3(b)). This is because if the manifold is well reconstructed, global anomalies should be separated far away from normal instances even with large t scale. Furthermore, with small scope of $t = 1$ (Figure 4.3(d)), the difference of anomalousness between local anomalies and boundary normal instances are slightly more obvious than Figure 4.3(a), which illustrates that with the support from AGK, HKS is more capable of revealing the density information of the intrinsic manifold structure.



(a) (b)

Figure 4.4: 70 nearest neighbors (in green) of red instance with GAU (Figure 4.4(a)) and AGK (Figure 4.4(b)), which shows that AGK has better manifold-aware property than GAU.

In the rest of this subsection we briefly introduce AGK on the observed space X ($n \times m$ matrix) that approximates the Gaussian kernel on the underlying manifold Y ($n \times d$ matrix, with $d \ll m$). The idea is to approximate the Euclidean distance between instances $y(j)$ in the manifold space Y using covariance matrix $C = JJ^T$ where J is the Jacobian matrix [127] and the instances $x(j) = f(y(j))$ in the observable space X. Let y, ϵ be two instances in the manifold space Y and $x = f(y)$, $\eta = f(\epsilon)$ be their mapping to the observable space X. Let $g : X \to Y$ be the inverse mapping of $f : Y \to X$, that is, $g(f(y)) = y$ and $f(g(x)) = x$, $\forall y \in Y$, $\forall x \in X$. Expanding the functions $y = g(x)$ in a Taylor series at the instance x

gives:

$$\begin{aligned}\epsilon(i) =& y(i) + \sum_j g_j^i(x)(\eta(j) - x(j)) \\ &+ \frac{1}{2}\sum_{kl} g_{kl}^i(x)(\eta(k) - x(k))(\eta(l) - x(l)) + O(\parallel \eta - x \parallel^3),\end{aligned} \tag{4.4}$$

where $g_j^i = \frac{\partial g(i)}{\partial y(j)}$. Therefore, the squared Euclidean distance in manifold space can be approximated by:

$$\begin{aligned}\parallel \epsilon - y \parallel^2 =& \sum_{ijk} g_j^i(x) g_k^i(x)(\eta(j) - x(j))(\eta(k) - x(k)) \\ &+ \frac{1}{2}\sum_{ijkl} g_j^i(x) g_{kl}^i(x)(\eta(j) - x(j))(\eta(k) - x(k))(\eta(l) - x(l)) \\ &+ O(\parallel \eta - x \parallel^4).\end{aligned} \tag{4.5}$$

A similar expansion can be built at instance η and the average of these two equations can be produced as:

$$\begin{aligned}\parallel \epsilon - y \parallel^2 =& \frac{1}{2}(\eta - x)^T[(JJ^T)^{-1}(x) + (JJ^T)^{-1}(\eta)](\eta - x) \\ &+ O(\parallel \eta - x \parallel^4),\end{aligned} \tag{4.6}$$

given that the Jacobian of the inverse g is the inverse of the Jacobian J (a detailed description of calculation can be referred to [127]). So we can construct the anisotropic Gaussian kernel:

$$W^{(AGK)}(i,j) = exp(-\frac{\parallel J^{-1}x(i)(x(i) - x(j)) \parallel^2 + \parallel J^{-1}x(j)(x(j) - x(i)) \parallel^2}{2\sigma^2}), \tag{4.7}$$

where $i, j = 1, ..., n$.

AGK has the desired attributes that it is separable, and its first (nontrivial) eigenfunctions are monotonic functions of the independent parameters [128]. It also has been proved that the eigenvectors of AGK reveal the independent components [127]. HKS, built on such approximation of manifold space, can better capture the embedding structure of data as shown in Figure 4.3(d) and 4.3(e), which is difficult or even impossible to achieve by using GAU or other similar techniques.

4.4 Local Anomaly Descriptor (LAD) and Its Algorithm Framework

4.4.1 Local Anomaly Descriptor

Although HKS on AGK has the capability to offer desirable local density information, it is of importance to select the right time scaling parameter t, which provides a trade-off between the effects of local and global information. However, it is hard to get the "best of both worlds" with single setting for this parameter. Even with better manifold reconstruction, if t is large the heat is still easy to dissipate regardless of normal instances or local anomalies (although not necessarily for global anomalies), which is shown in Figure 4.3(e). This is because with large t scale, the distance between local anomalies and the normal instances around them would still be close. As a result, local anomalies cannot retain their heat. On the other hand, if t is small, the heat diffusion runs for only a short period of time, and the resulting anomalousness capture very local information, but almost carry the same value for instances with similar density inside a very restrained neighborhood, which is the major reason why it sometimes confuses some normal instances with local anomalies. In Figure 4.3(d) we can see HKS on AGK assigns similar scores to the local anomalies and some of the boundary normal instances. Intuitively speaking, HKS on AGK still fails to take the amount of similar instances into account with off-the-sweet-spot t setting.

As a means to handle the above problems, we propose to use umbrella operator [135] [36]. Umbrella operator is an approximation of the Laplace operator measuring the vector from the vertex in question to the barycenter of its neighbors. In practice, umbrella operator U is usually implemented to compute the average difference between a point $x(i)$ and its k nearest neighbors $nb(x(i), k)$:

$$U(i) = \frac{1}{k} \sum_{x(j) \in nb(x(i),k)} (x(j) - x(i)). \tag{4.8}$$

In our research, we need to deliberate on the quantity of similar instances in neighborhood by bridging the gap between global and local properties. If an instance has a lot of close neighbors, the average value of neighborhood should be

very similar to the value of this instance. Therefore we use the scale-dependent (weighted) umbrella operator $\mathcal{U}$:

$$\mathcal{U}(i) = \frac{1}{k} \sum_{x(j) \in nb(x(i),k)} W(i,j)(x(j) - x(i)), \tag{4.9}$$

where $W(i,j)$ is the weight between $x(i)$ and $x(j)$. If we replace $W(i,j)$ with $W^{(AGK)}(i,j)$, then we may use scale-dependent umbrella operator on top of Heat Kernel Signature ($\mathcal{H}$). We call it as Local Anomaly Descriptor (LAD), and define LAD for a point $x(i)$ as follows:

$$\mathcal{L}_t(i) = \mathcal{H}_t(i) - \frac{1}{k} \sum_{x(j) \in nb(x(i),k)} \mathcal{H}_t(j) \times W^{(AGK)}(i,j). \tag{4.10}$$

The geometric meaning of LAD is illustrated in Figure 4.5 where we measure the difference between a single $\mathcal{H}_t(i)$ and its neighborhood's weighted average $\mathcal{H}_t(j) \times W^{(AGK)}(i,j)$ value.

If an instance is globally anomalous, its HKS would be already high enough to discriminate itself to the other instances. While it is locally anomalous, its HKS is likely to be similar to some normal instances' with similarly sparse neighborhood. However, applying the scale-dependent umbrella operator, LAD can serve to recognize the local anomalies from normal instances with expanded horizon of neighborhood and reflection of the amount of similar instances inside. Since local anomaly only has a small amount of neighbors with close HKS, but normal instances, on the other hand, have more such neighbors.

LAD has a very lucrative property in considering the amount of similar instances (the second property in our anomaly definition): since it not only measures a very constrained local area with small t, but also considers the heat diffusion area of the adjacent neighbors. It gives a measurement of an expanded horizon to capture how many similar instances are there inside a large enough neighborhood. If there are lots of similar neighbors (with similar local density), LAD will be quite small since the neighborhood difference of HKS is not large. On the contrary, if the neighbors's HKS are different, LAD score tends to be assigned a greater value. So even though k is not large enough to include the whole appropriate neighborhood, LAD can still capture the information related to the amount of similar instances.

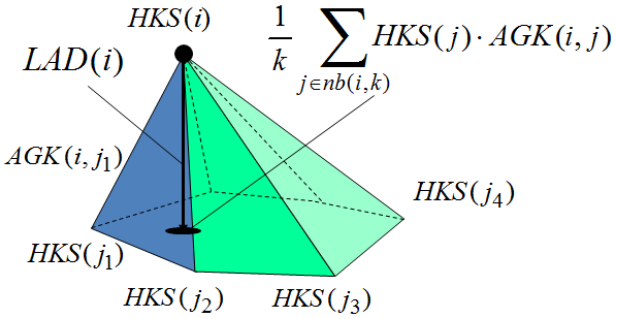


Figure 4.5: Illustration of LAD (Local Anomaly Descriptor, Equation 4.10) which calculates weighted average of neighbor differences. It is one of the ways to take the neighborhood distribution into consideration [135].

The benefits of LAD in comparison with HKS can be seen in Figure 4.3, which shows that our proposed LAD has a penetrating awareness on both global and local anomalies primarily because of the power of scale-dependent umbrella operator on HKS.

Mathematically LAD could also use GAU as the connection weighted function W (or $W^{(GAU)}$), which not only affects the term of subtrahend in Equation 4.10 but also the construction of HKS ($\mathcal{H}_t$). To have a concrete understanding of the effect of AGK, we also compare the different performance between GAU and AGK on LAD. LAD with GAU in Figure 4.3(c), although makes some anomalies more salient, still fails to distinguish some local anomalies and normal instances. But LAD with AGK in Figure 4.3(f) clearly separates all the global and local anomalies from the normal instances. This confirms that as a connection weighting function, AGK is more effective than GAU in that AGK is more aware of the differences between instances in the manifold space.

4.4.2 Algorithmic Framework of LAD

In this subsection we explain LAD framework step by step. Let X be a matrix of size $n \times m$, where n is the number of instances and m is the number of dimensions, our framework is detailed in Algorithm 5. This algorithm undergoes a kind of data warping process by using AGK (Step 1, Section 4.3) and Laplacian

random walk normalization (Step 2, Section 2.1). Then we perform the eigen-decomposition (Step 3) and construct HKS for each instance (Step 4, Section 4.2). Equation 4.10 is used as the last step (Step 6, Section 4.4.1) to compute Local Anomaly Descriptor as the final measurement of anomalousness.

ALGORITHM 5: LocalAnomalyDescriptor(X, σ, t, k)

Input: Input data $X \in R^{n \times m}$, σ the Gaussian scaling parameter, t the time scaling parameter, k the neighborhood size.

Output: LAD score for each instance.

1 Construct anisotropic Gaussian kernel $W^{(AGK)}$ using Equation 4.7 and σ;
2 Construct Laplacian random walk normalization L_{rw} on $W^{(AGK)}$;
3 Compute generalized eigenvectors ψ_p and corresponding eigenvalues λ_p, $p = 1, 2, ..., n.$;
4 Construct Heat Kernel Signature with time scale t using Equation 4.3 ;
5 Compute Local Anomaly Descriptor using Equation 4.10 with Heat Kernel Signature and anisotropic Gaussian kernel in the k nearest neighborhood for each instance ;

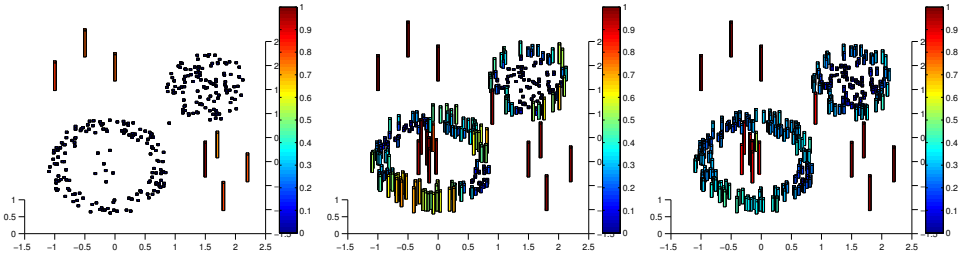
Regarding computational complexity, affinity construction using GAU takes $O(n^2m)$. If using AGK it takes $O(n^2m^2)$. Eigen-decomposition (Step 3) is another time-consuming step. There are many iterative methods to conduct eigenvalue decomposition, but in general finding the eigenvalues reduces to matrix multiplication by computing a symbolic determinant, which gives a running time of $O(n^3 + n^2 log^2 n)$ [112]. An alternative way of estimating the heat kernel $H_t = e^{-tL_{rw}}$ is to use a partial sum of infinite series with:

$$e^{-tL_{rw}} = \sum_{p}^{\infty} \frac{(-tL_{rw})^p}{p!}. \tag{4.11}$$

This method would be especially attractive for small values of t, since only a few terms would be needed to obtain an accurate estimation of $e^{-tL_{rw}}$ [8], which is desired to our LAD calculation since a small amount of t is good enough to reveal the anomalousness.

On the other hand, if we only use a small portion of eigen-system (say the first d eigenvalues and eigenvectors) to compute LAD, the eigendecomposition only takes $O(n^2d)$. We will analyze the performance of this fast version in the experimental Section 4.8.6.

4.4.3 Discussion of LAD



(a) LAD ($t = 100$ and $k = 10$) (b) FDD ($T = 1$) (c) FDD ($T = 0.01$)

Figure 4.6: LAD with large t fails to reveal the local anomalousness [2] (Figure 4.6(a)) due to the over-diffusion. Comparably, FDD acts robustly in measuring anomalousness regardless of small or large scaling parameter (Figure 4.6(b) and 4.6(c)).

As we introduce in the previous subsections, LAD can capture the two properties of our anomaly definition (Section 4.1.2) effectively:

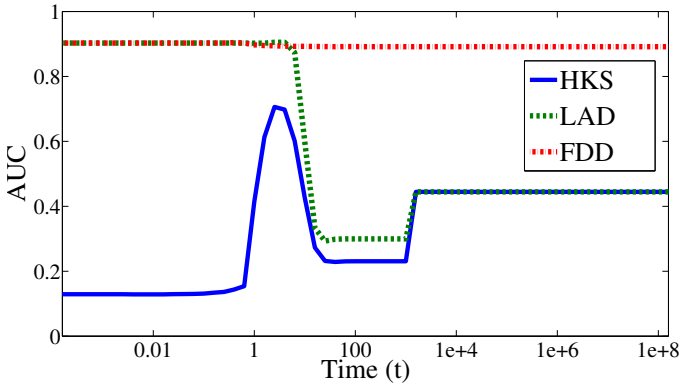


Figure 4.7: Illustration of stability test on ecoli dataset against time scaling parameter (t) tuning. We can see that although LAD (green curve) has better performance and stability than HKS (blue curve) when t is small, it still doesn't make accurate detection when t becomes larger ($t \geq 100$). Our ideal goal is to design an anomaly detection algorithm (red curve) that maintains desirable result regardless of scaling parameter tuning.

[2] For each algorithm output, the anomalousness scores are normalized in the range of $[0, 1]$ to have an easy comparison.

- Local diffusion process calculated by HKS with small t can intuitively depict the local density of each instance (Section 4.2.2).
- Umbrella operator provides a broader view even with too small t, so that it has a lucrative property in considering the amount of similar instances (Section 4.4.1).

Although LAD gets over the instability of HKS to some degree by integrating a scale-dependent umbrella operator, which provides a more broader view of neighborhood distribution, LAD still suffers if the time scaling parameter t goes too large (see the example in Figure 4.6(a)). More completely, Figure 4.7 shows the stability of LAD is below expectation when t is large. The reason is that as the diffusion time gets longer, HKS across all the instances will become all the same. Subsequently there is no difference between HKS of anomalies and their neighbors. This problem of LAD (and of course HKS as well) comes from the essential properties of heat diffusion by natural: once the dissipation time is large, heat will easily get over-diffused.

In the next two subsections we resort to quantum mechanics whose research objects are in a discrete space, which has potential to detect locally low neighborhood density more stably (Figure 4.6(b) and 4.6(c)). In quantum mechanics, particles jump from one quantum state to another, and the waves space is not continuous. The probability of a particle shows up at a certain place is highly related to the local density of this place. In a certain degree, quantum mechanics intuitively focuses on the intrinsic local density distribution, while largely ignoring the extrinsic properties (pair-wise distance, attribute distribution, etc.) of the ambient area of input space.

4.5 Schrödinger Equation and Wave Function in Quantum Mechanics

Besides heat diffusion, another physics concept which is closely related to density measurement is quantum mechanics [55], which also has strong connections to Laplace operator. Quantum mechanics is a mathematical machine for predicting the

behavior of microscopic particles. Anomalous instances can be treated as regions of low density that correspond to the aggregation area of maximal free energy, and such area is easier to trap particles. On the other hand, normal instances indicate high density regions with minima of the free energy in the system, so the probability for particles appearing in such area is low.

The Schrödinger equation is the key equation in quantum mechanics, which describes how the quantum state of a physical system changes with time. One of the most famous examples is the non-relativistic Schrödinger equation for a single particle moving in an electric field. If we ignore the potential energy, it is directly associated with Laplace operator L as follows:

$$\imath\frac{\partial\phi}{\partial t}(x,t) = L\phi(x,t), \tag{4.12}$$

where ϕ is the space-time wave function of the quantum system, $\imath$ is the imaginary unit, x is the position and t is time. The mod square $|\phi(x,t)|^2$ depicts the probability density of a particle at position x at time t, which satisfies:

$$\int |\phi(x,t)|^2 dx = 1. \tag{4.13}$$

Assume the Laplace spectrum has no repeated eigenvalues, and $L = \psi'\lambda\psi$ (ψ and λ are the eigenvectors and eigenvalues of L), the space-time wave function $\phi(x,t)$ can be expressed in the spectral domain as:

$$\phi(x,t) = \sum_{p=1}^{\infty} e^{\imath\lambda_p t}\psi_p(x)f(\lambda_p), \tag{4.14}$$

where $f(\lambda)$ is the energy distribution. This is because in spectral domain, eigenvalue λ is approximately equivalent to energy level E [55], so $f(\lambda)$ can also be rewritten as $f(E)$.

Integrating the mod square of wave function $|\phi(x,t)|^2$ over all time scales, we can get

$$\mathcal{P}(x) = lim_{\mathcal{T}\to\infty}\frac{1}{\mathcal{T}}\int_0^{\mathcal{T}} |\phi(x,t)|^2 dt = \sum_{p=1}^{\infty} f(\lambda_p)^2\psi_p(x)^2. \tag{4.15}$$

The physical meaning of $\mathcal{P}(x)$ is the average possibility for a particle with energy distribution f(λ) found at position x. The property of quantum mechanics

states that due to the fast decaying nature of the evanescent wave, a particle tends to be trapped within the vicinity of region where the strong field enhancement occurs. In high-dimensional dataset, the "tip" regions are those data points with sparse neighborhood. In other words, the particle tends to stay at instances with more sparse neighborhood and rarely shows up at instances with denser neighborhood. Therefore in theory $\mathcal{P}(x)$ can intuitively represent the local density of each instance. In practice, however, there are two key challenges:

- What is the best energy distribution $f(\lambda)$?
- What is the best graph Laplacian L (that directly associates with λ and ψ) ?

Section 4.6.1 will solve the first challenge, while the second challenge will be discussed and conquered in Section 4.6.2.

4.6 Fermi Density Descriptor (FDD) and Its Algorithmic Framework

4.6.1 Energy Distribution Function and Definition of Fermi Density Descriptor

In this subsection we explore what is the best energy distribution function f for Equation 4.15. $f(E)$ ($f(\lambda)$ determines the probability that a particle is in energy state E. It can be viewed as a realization of the ideas of discrete probability in such a case that energy can be treated as a discrete variable. In quantum mechanics there are three distinctly different distribution functions [55], namely Maxwell-Boltzmann distribution (MB), Fermi-Dirac distribution (FD) and Bose-Einstein distribution (BE). Besides quantum mechanics, existing research also explored distributions based on other theoretical assumptions. Section 4.2 already introduce heat dissipation (HD) which was used in [133] to describe the heat diffusion given time t. In 2011, Aubry [7] chose Gaussian distribution (GD) in the logarithmic energy as $f(E)$ to define Wave Kernel Signature. Here we briefly introduce these five distribution functions and analyze their respective performance on anomaly detection.

Maxwell-Boltzmann Distribution (MB).

$$f_{MB}(E) = \frac{1}{e^{E/\kappa T}}. \tag{4.16}$$

MB distribution depends on the energy level E of the single particle state, the absolute temperature T, and the Boltzmann constant κ. In quantum mechanics, the MB distribution usually applies to the particles at a high enough temperature and low enough density where quantum effects can be ignored [55].

Fermi-Dirac Distribution (FD).

$$f_{FD}(E) = \frac{1}{e^{(E-\mu)/\kappa T} + 1}, \tag{4.17}$$

where μ can be obtained from

$$\sum_{E} \frac{1}{e^{(E-\mu)/\kappa T} + 1} = n/2. \tag{4.18}$$

Beside the same parameters as used in Equation 4.16, FD distribution is also conditional on a chemical potential μ, and n the number of electrons in the whole systems. Equation 4.18 represents the number of orbital since only two electrons (with opposite 'spin') can occupy each orbital. In quantum mechanics FD distribution applies to identical particles (fermions) with half-odd-integer spin in a system in thermal equilibrium [55].

Bose-Einstein Distribution (BE).

$$f_{BE}(E) = \frac{1}{e^{(E-\mu)/\kappa T} - 1}, \tag{4.19}$$

where μ can be obtained from

$$\sum_{E} \frac{1}{e^{(E-\mu)/\kappa T} - 1} = n/2. \tag{4.20}$$

The parameters used in BE distribution function have the same physical meaning as those used in Equation 4.16 and 4.17. BE distribution describes the statistical

behavior of integer spin particles (bosons). At low temperatures, bosons can behave very differently than fermions because an unlimited number of them can be collected into the same energy state [55].

Heat Diffusion (HD).

$$f_{HD}(E) = e^{-Et}, \tag{4.21}$$

where t is the time for heat dissipation. Heat diffusion describes how the amount of heat dissipates from a heat source to its neighborhood at time t. Different from the three distributions in quantum mechanics which depict the discrete pattern of particle movement in terms of probability, heat diffusion has a continuous conception in both time and space domains.

Gaussian Distribution (GD).

$$f_{GD}(E) = e^{-\frac{(e-log(E))^2}{2\sigma^2}}. \tag{4.22}$$

It is derived in [7] from a perturbation-theoretical analysis. Under the assumption that the eigenvalues (eigenenergies) of an articulated dataset are log-normally distributed random variables, the authors claimed that it is robust to small data perturbations while being as informative as possible.

Before comparison between the aforementioned different energy functions, we need to clarify a few points. Firstly, since κ is a constant, from now on we will remove κ from the relative formulas (Equation 4.16 $\sim$ 4.20) in the interest of convenience. Secondly, although HD and MB distribution have different physical meaning, they indeed have similar mathematical performance if we simply replace t in Equation 4.21 with $\frac{1}{T}$ in Equation 4.16. In other words, small diffusion time t in heat diffusion has similar effect as large environmental temperature T in MB distribution. Therefore in the following analysis we simply ignore heat diffusion (HD) and only compare the other four distribution functions. Thirdly, although with different physical meanings, for the sake of mathematical convenience, we assign $2\sigma^2$

in GD (Equation 4.22) with the same value as T in the quantum mechanics functions in order to compare the stability of different functions as scaling parameter changes.

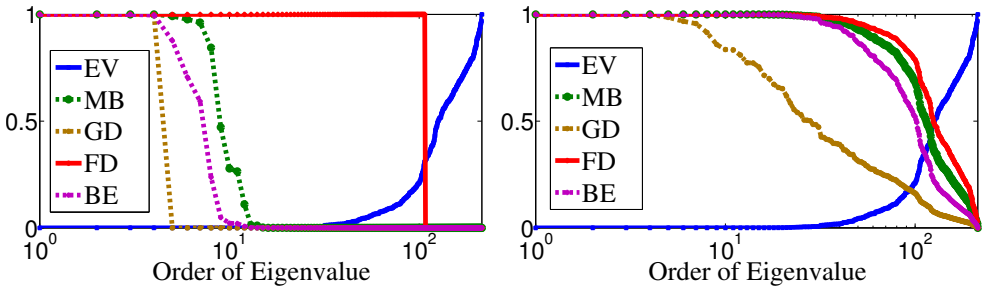


Figure 4.8: Stability comparison between different energy distribution functions on glass dataset. Blue curve is the eigenvalue (EV) ordered by increasing value (decreasing importance since EV are derived from graph Laplacian). Green, red, purple and brown curves are MB, FD, BE and GD distribution, respectively. Figure 4.8(a) shows the performance of four functions when $T = 0.001$, and Figure 4.8(b) shows the performance of four functions when $T = 50$. We can see that FD has the most stable performance as T changes.

Although MB/HD, FD, BE and GD distributions have solid theoretical background, the differences of mathematical performance give rise to very different statistics, especially the stability of outcomes.

1. Among these functions, FD is the most practical one for anomaly detection in terms of performance stability under different parameter (T) setting. In Equation 4.17, two special terms can stabilize the distribution function performance: the constant smoothing term "plus one" and the balancing term μ in the denominator part. The role of the smoothing term is to damp the contribution of the exponential part from being too small, which results from either extremely small λ (E) or large T. The balancing term μ (computed by Equation 4.18) is a parameter controlling the trade-off between small and large λ (E). Besides, it helps to tune a sweet range for λ (E) according to T, since it has a positive side-effect that it can accelerate the attenuation of contribution from those trivial eigenvalues in Equation 4.17.

2. Comparably, MB/HD and GD without any smoothing term or balancing term, are very sensitive to either extremely small λ (E), or large scaling parameter T . Although BE has balancing effect from μ, it actually suffers more from the "minus one" in the denominator part, since it lessens the stability by making the denominator part even smaller.

Figure 4.8 shows the value of different distribution functions across different eigenvalue (energy) of glass dataset (statistic details of glass are in Section 4.8.1). In general, FD distribution tends to assign stable weights regardless how temperature T changes compared with the other energy distribution functions. To have a broader and fair comparison between the effect of different energy distribution functions, we test all the distribution functions on 7 datasets against changing T. The detailed results, which again confirm our findings, are recorded in Section 4.8.5.

Now we integrate the FD distribution function (Equation 4.17) into Equation 4.15, and define **Fermi Density Descriptor (FDD)** at a point $x(i)$ as:

$$\mathcal{F}(i) = \frac{1}{C}\sum_{p=1}^{\infty}(\frac{1}{e^{(\lambda_p-\mu)/T}+1})^2\psi_p(i)^2, \tag{4.23}$$

where $C = \sum_{p=1}^{\infty}(\frac{1}{e^{(\lambda_p-\mu)/T}+1})^2$, μ can be derived from Equation 4.18 where n is set as the number of data instances in practice.

4.6.2 Laplace Operator for FDD

We discuss the best choice of graph Laplacian for FDD in this subsection. In Section 4.5, we have already shown that our proposed Fermi Density Descriptor is derived from Schrödinger equation (Equation 4.12) which is strongly associated with Laplace operator. Laplace operator is intimately related to the "shape" of data, or more precisely, the density distribution of data. More precisely, Laplace operator in Equation 4.12 is aiming to account for the kinetic energy of the particles constituting the system, which depends on the spatial configuration to conserve energy [55]. Using the discrete Laplace operator, or graph Laplacian, the Schrödinger equation can be simplified, and generalized to be matrix operation over space of an arbitrary number of dimensions.

Different graph Laplacian normalizations have been introduced in Section 2.1. Although their effect on clustering has been thoroughly analyzed in [71] and [100], it is still unclear what is the best choice for FDD with the purpose of anomaly detection.

In general, when the data points are sampled from the equilibrium distribution of a stochastic dynamical system, clustering algorithms tend to correct different density bias in order to obtain stable and balanced instance clusters This is quite different from the need of anomaly detection applications when the density of the points is a quantity of interest, and therefore, cannot be ignored [28]. For clustering purpose, we focus on normal instances and want to recover manifold insensitive to the existing anomalies (usually being treated as noise in such applications). In other words, the different density distribution prevents algorithms from the desire clustering result and therefore need to be removed in clustering applications [71]. However, from anomaly detection's point of view, the focus is on the anomalies, and the recovered manifold should be aware of local density variation, therefore in the manifold space the density differences between anomalies and normal instances should be preserved or even magnified with respect to the input space distribution. In a nutshell, we need to find the graph Laplacian that is most reactive to local density distribution with the purpose of anomaly detection.

Theorem 1 *The density impact power for L_{nn}, L_{rw}, L_{sym}, L_{fp} and L_{lbn} normalization are* 2*,* 1*,* 1*,* 0.5*, and* 0 *respectively.*

Proof: Define $q(x)$ the true density function of x, and a kernel function $k_\sigma(x, y)$ between x and y with σ as neighborhood scaling parameter. Let

$$q_\sigma(x) = \int k_\sigma(x, y)q(y)dy, \tag{4.24}$$

which is an approximation of the true density $q(x)$, we can form the new kernel [28]:

$$k^\alpha_\sigma(x, y) = \frac{k_\sigma(x, y)}{q^\alpha_\sigma(x)q^\alpha_\sigma(y)}, \tag{4.25}$$

where $\alpha \in R$. Apply the Laplacian operator to this kernel as follows:

$$d^\alpha_\sigma(x) = \int k^\alpha_\sigma(x, y)q(y)dy, \tag{4.26}$$

the new anisotropic kernel can be defined as:

$$p_{\sigma}^{\alpha}(x,y) = \frac{k_{\sigma}^{\alpha}(x,y)}{d_{\sigma}^{\alpha}(x)}. \tag{4.27}$$

Therefore, based on the Laplacian operator, the infinitesimal generator of the Markov chain with $\sigma \to 0$ [28] can be defined as:

$$L_{\sigma,\alpha} = \frac{I - P_{\sigma,\alpha}}{\sigma}, \tag{4.28}$$

where $P_{\sigma,\alpha}f(x) = \int p_{\sigma}^{\alpha}(x,y)f(y)q(y)dy$ with any function f. If $\sigma \to 0$ we have:

$$lim_{\sigma\to 0}L_{\sigma,\alpha}f = \frac{\Delta(fq^{1-\alpha})}{q^{1-\alpha}} - \frac{\Delta(q^{1-\alpha})}{q^{1-\alpha}}f. \tag{4.29}$$

Hence the infinitesimal operator can be given by

$$\Delta\varphi - \frac{\Delta(q^{1-\alpha})}{q^{1-\alpha}}\varphi, \tag{4.30}$$

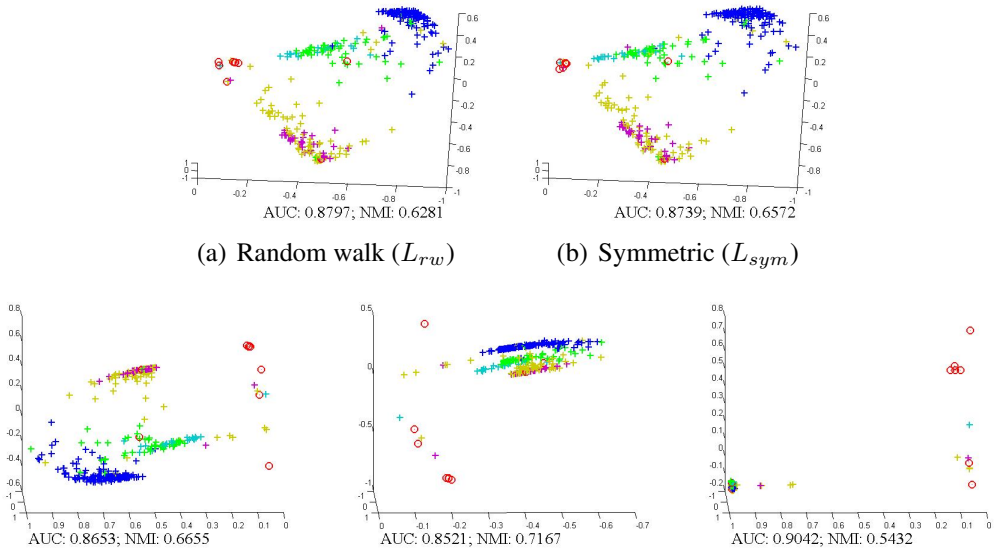
where $\varphi = fq^{1-\alpha}$.

For L_{rw} normalization, $\alpha^{L_{rw}} = 0$ [28] so the density impact power is $1 - \alpha^{L_{rw}} = 1$. For L_{fp} normalization, $\alpha^{L_{fp}} = 0.5$ [28] hence the density impact power $1 - \alpha^{L_{fp}} = 0.5$. For L_{lbn} normalization, $\alpha^{L_{lbn}} = 1$ [28] thus its density impact power is $1 - \alpha^{L_{lbn}} = 0$.

L_{sym} normalization can be transformed from L_{rw} normalization by $L_{sym} = D^{1/2}L_{rw}D^{-1/2}$. From Equation 4.26 we know that D (d) is proportional to the density impact power q, therefore $lim_{\sigma\to 0}L_{sym,\sigma}f$ depends on density function $q^{-1/2}q^{1-\alpha^{L_{rw}}}q^{1/2} = q^1$ where $\alpha^{L_{rw}} = 0$. On this account, its density impact power is also 1.

For L_{nn}, since $L_{nn} = DL_{rw}$, and $lim_{\sigma\to 0}L_{nn,\sigma}f$ depends on density function $q \times q^{1-\alpha^{L_{rw}}} = q^2$ where $\alpha^{L_{rw}} = 0$. Accordingly L_{nn} has the greatest density impact power 2. □

Proof of Theorem 1 demonstrates that L_{nn} is the best option for FDD. As an illustration, Figure 4.9 shows the effects of different normalizations on ecoli dataset (Section 4.8.1). We only plot the first three non-trivial eigenvectors derived from the graph Laplacian. The red circles indicate anomalous instances while crosses



(a) Random walk (L_{rw}) (b) Symmetric (L_{sym})

(c) Fokker-Planck (L_{fp}) (d) Laplace-Beltrami (L_{lbn}) (e) No-normalization (L_{nn})

Figure 4.9: The comparison from different graph Laplacians effect on ecoli dataset for the purpose of anomaly detection (measured by AUC) and clustering (measured by NMI). Red circles indicate anomalous instances, while crosses in other color represent different clusters of normal instances. We can see that L_{nn} is the best choice for anomaly detection since it magnifies the distance and density differences between anomalies and normal instances. On the contrary L_{lbn} is the worst choice for anomaly detection purpose but the best option for clustering.

with other colors represent different clusters of normal instances respectively. We also show AUC score (Section 4.8.1) of anomaly detection result, and NMI score (the detailed definition of NMI can be referred to [60]) of clustering result from different graph Laplacians.

This experiment shows that the L_{lbn} normalization (Figure 4.9(d)) reorganizes points with larger intra-cluster similarity and smaller inter-cluster similarity. Therefore L_{lbn} normalization has the highest NMI (0.7167). Nevertheless, the over-diffusion and the consequent unresponsiveness of density distribution generate a tail of normal instances connected to anomalous instances, which leads to the lowest AUC (0.8521). Compared with L_{lbn} normalization, L_{fp} (Figure 4.9(c)) normalization spreads the instances with a slightly more dispersive distribution (e.g., cluster in dark yellow), which makes a lower NMI 0.6655 but a slightly higher AUC

0.8653.

L_{rw} (Figure 4.9(a)) and L_{sym} (Figure 4.9(b)) normalizations reconstruct circle-like shape in their manifold space. But they also show more mixture of different clusters since they preserve the same density as in the input space with impact power equal to 1. Consequently it gives higher AUC (0.8739 for L_{sym} and 0.8797 for L_{rw}) but lower NMI (0.6572 for L_{sym} and 0.6281 for L_{rw}) compared with L_{lbn} and L_{fp}.

L_{nn} has the most polarized manifold reconstruction. The reason is that the density difference is amplified by L_{nn} compared with the four normalizations. It results in that the normal instances with higher density shrink to a condensed area while anomalous instances are far away from the collapsed center of normal instances. Consequently, L_{nn} has the strongest ability (with AUC 0.9042) to separate anomaly from normal instances even though the clustering based on it will miserably fail (with NMI 0.5432). Section 4.8.4 will show more convincing experiment results which confirm that L_{nn} is the best Laplacians for FDD.

4.6.3 Algorithmic Framework of FDD

Let X be a matrix of size $n \times m$, where n is the number of instances and m is the number of dimensions, our algorithm is detailed in Algorithm 6.

ALGORITHM 6: FermiDensityDescriptorGlobal(X, σ, T)

Input: Input data $X \in R^{n \times m}$, σ the Gaussian scaling parameter, T is the environmental temperature.

Output: FDD score for each instance

1 Construct Anisotropic Gaussian Kernel (AGK) $W^{(AGK)}$ using Equation 4.7 and σ;

2 Construct L_{nn} (Equation 2.1) ;

3 Compute generalized eigenvectors $\psi(i)$ and corresponding eigenvalues λ_i of L, $i = 1, 2, ..., n$;

4 Construct Fermi Density Descriptor (FDD) with temperature T using Equation 4.23 ;

Step 1 (details in Section 4.3) constructs AGK similarity matrix and L_{nn} is operated on top of it in Step 2 (details in Section 2.1) to generate density "polarized" manifold projection. Then we perform the eigen-decomposition (Step 3) and

compute FDD for each instance (Step 4, details in Section 4.6). FDD value is used as the final measurement of anomalousness.

Similar to algorithm of LAD in Section 4.4.2, the computation of FDD is dominated by affinity construction ($O(n^2m^2)$ if AGK and $O(n^2m)$ if GAU), and eigendecomposition in Step 3 ($O(n^3)$ [112]). Also, if we only use a small portion of eigen-system (say the first d eigenvalues and eigenvectors) to compute FDD, the computational complexity of eigendecomposition would be drop to $O(n^2d)$. We will analyze the performance of this fast version in Section 4.8.6.

4.6.4 Discussion of FDD

FDD satisfies the two properties of our anomaly definition (Section 4.1.2) in a more concise and effective way. In that FDD relies on the "polarized" manifold reconstruction which magnifies the distances between anomalies and normal instances. Consequently, anomalous instances will be more singular and distinctive. The dense neighborhood will become even denser, with analogous instances aggregated together. The sparse neighborhood, on the contrary, will be more sparse. In this fashion, FDD considers **the locally low neighborhood density** and **amount of similar instances** simultaneously and effectively.

Besides, FDD has more robust performance against different physics parameter settings (especially the extreme cases). Part of the reasons lie in the stable energy function FD which was already scrutinized in Section 4.6.1. The other reason is because the "polarized" manifold reconstruction that "breaks" the connections between anomalies and normal instances. Figure 4.6 and 4.7 illustrates the stability comparison of FDD, LAD and HKS, which once again confirm that FDD maintains desired result more stably with different parameter tuning.

4.7 Discussion of Theoretical Perspectives

We now justify the utility of our proposed two algorithms LAD and FDD by briefly documenting their theoretical connections with a few existing methods, which also lays a solid foundation for their attractive properties for practical use.

4.7.1 Comparison between LAD and FDD

LAD and FDD are all based on Laplace operator on the affinity matrix, and the subsequent manifold reconstruction. They all try to describe the density information in a retained but informative neighborhood. However, their different theoretical background leads to quite different interpretation and performance.

Theoretical Backgrounds. LAD is inspired by heat diffusion, which is highly related to Markov chain. It describes the amount of heat being transfer in a certain time scale, therefore its conception is continuous in both time and space. On the contrary, FDD measures the probability that a particle (fermion) shows up at a certain position. It is built upon quantum mechanics, whose key idea is that the motion of a particle is discontinuous and random.

Manifold Reconstruction. Due to the close theoretical connections, LAD uses random walk normalization L_{rw} by natural and projects origin instances onto a diffusion space. However the diffusion process is hard to control and usually gets over-diffused, leading to a blending of local anomalies into normal instances. But FDD applies L_{nn} to construct a "polarized" manifold projection, which concentrates on magnifying the difference between anomalies and normal instances. Roughly speaking, in the "polarized" manifold the similar points with higher density tend to collapse to the center of mass, therefore clusters of normal instance are topologically isomorphic to extremely condensed convex sets. Conversely, anomalies will be more singular and distinctive from the normal instances. Although this type of mapping is non-isometric and the original distribution is changed, it is of central interest in anomaly detection, as it becomes more sensitive to locally low neighborhood density and the preservation of intra-cluster distance or distribution is not a concern at all. In Section 4.8.4 we will further confirm our choice of Laplacians for LAD and FDD with support from more experiment results.

Strategies against parameter sensitivity. To overcome the narrow scope of small t, LAD integrates a scale-dependent umbrella operator on the projected diffusion space, which bridges the gap between global and local properties. Its advantage

compared with HKS is that, although with the same small t, LAD covers a sufficiently large neighborhood for each instance $x(i)$ since LAD also considers the t scale neighborhood of $x(i)$'s neighbors. On the other aspect, it takes the quantity of similar instances into consideration. But FDD approaches stability against parameter tuning in a different way: Equation 4.23 has two special terms which stabilize FDD performance: the constant smoothing term "plus one" and the balancing term μ in the denominator part. Both of these two terms can damp the contribution of the denominator from being too small, which results from the extreme setting of scaling parameter.

Stability. Although LAD provides more robust performance under very small t compared with HKS, it still suffers when t becomes too large due to over-diffusion of heat dissipation. But FDD has stronger stability than LAD in that it can conquer the negative side-effect from extreme scaling parameter, regardless whether it is too small or too large.

4.7.2 Connections between LAD/FDD and Other Anomaly Detection Algorithms

kNN-based Approaches. kNN-based methods [14] [34] [158] approach local density for each instance using its neighborhood information. Like LAD and FDD, they require (scaling) parameters to capture a reasonably large neighborhood, and the density information is based on this prescribed local region. However, kNN-based methods have strictly local context in that they simply fix the neighborhood size with k. In contrast, LAD employs locally adaptive neighborhood size which directly benefited from the physics-inspired properties of heat diffusion, while FDD makes uses of stabilization terms to smooth out the performance fluctuation from off-the-sweet-spot parameters. Moreover, kNN-based methods rely on Euclidean distance in the input space which is a pair-wise local quantity, while our methods consider the relationship between instances in manifold space, which is more comprehensive. For example, heat kernel used in LAD considers all the possible paths between two instances within time t. Therefore our proposed methods are more intrinsic and informative than kNN-based methods.

Attribute-based Approaches. Attribute-based methods [96] [97] [138] try to compute local density by adding up a sequence of values from an attribute-based function [138], which to some extent is equivalent to a kernel density function such as heat kernel. Their measurement of global instance distribution is based on each attribute and how deviated each instance is from the other instances in that specific attribute, which indeed is more informative than kNN-based approaches. However, the strong emphasis on input attribute distribution is also a "double-sided sword": on one hand it is much faster without any distance calculation, on the other hand, such distribution simply hinged on attributes still fails to consider local anomalies. Although our methods undergo a step of dimension reduction or manifold projection at first, they map all the correlated attributes onto a few lower dimensions. Therefore both LAD and FDD are more capable of stably finding local anomalies.

4.7.3 Connections between LAD and Other Related Techniques

Biharmonic Operator. HKS is directly derived from the Laplace operator and its eigen-decomposition, therefore HKS is intrinsically a second-order property relevant to the Laplace's equation. The derivation of LAD, or the scale-dependent umbrella operator, can be intuitively related to the biharmonic process, because the Laplace operator is essentially applied twice (to compute both HKS and the subsequent scale-dependent umbrella operator). It provides a good balance in the sense that it decays slowly in small cluster around the source instance and fast enough to be structurally inherent in dense areas. This specific "balancing" is intimately derived from the biharmonic equation with properties such as local support and global informativeness [94].

Signal Processing. LAD also has strong connection to signal processing. In low-pass filtering, the divergence of a sample from its average neighborhood is the easiest way to pinpoint those inconsistent instances if the desired signal has significant high frequency content. As in traditional signal processing [135], it is possible for LAD to quantify the frequency response by computing an adjoining sum of the Laplace operator in its immediate vicinity. As a result, this enables LAD to distinguish between normal instances and inconsistent instances (anomalies) with greater

precision.

Diffusion-based Clustering. Some recent researches [122] [120] [71] proposed the probabilistic clustering approaches based on diffusion space. By integrating all time scales of kernel function into one single term, this kind of techniques completely removes the diffusion time scaling parameter, therefore has the built-in robustness to data perturbation and scaling parameter tuning [71]. However, as a side-effect, this process of "integration" easily assimilates local anomalous instances into normal instance clusters since the excessive-diffusion tends to connect everything together. LAD, in sharp contrast, is built upon kernel function with small time scale and scale-dependent umbrella operator instead of integrating all time scales together. Therefore it avoids the excessive-connection problem.

4.7.4 Connections between FDD and Quantum-based Clustering/Classification

Many data mining researches [66] [65] [108] [147] used the Schrödinger equation from quantum mechanics to allow the clusters, or over dense regions in the data, to reveal themselves.

As an example, the intuition behind Quantum clustering [66] is based upon the fact that in the quantum system local maxima in the ground state wave-function correspond to the local minima of potential [147]. And such minima are likely to be good candidates of the cluster centroid locations [147]. Instances lying in the basin of attraction of particular minima were identified as a single cluster. Advanced methods have been proposed [65] [147] which differ in how to handle the problem of identifying data points with local minima of the function in high-dimensions. Specifically, Schrödinger equation in [108] is used to calculate the probability of locating a particle given its potential energy.

Although our proposed FDD also applies Schrödinger equation, it ignores the potential energy in Equation 4.12. The reason is that we are not trying to cluster instances to certain centroids but rather focusing on the local density measurement to distinguish anomalies. In our study, the Schrödinger equation acts as a cost function separating instances with different density instead of clustering/classifying

instances according to the local potential.

Table 4.1: Statistics of our evaluation datasets.

	Dataset	# Instance	# Attribute	% Anomalies(classes)	References
1	breastcancer	683	9	35.0% (malignant)	[110]
2	wdbc	569	29	37.3% (malignant)	[110]
3	pima	768	8	34.9% (positives)	[96]
4	arrhythmia	452	279	45.0% (abnormal)	[34]
5	arcene	200	10000	44.0% (positives)	[59]
6	prostatetumor	102	10509	50.9% (abnormal)	[130]
7	gse24417	417	6864	31.2% (abnormal)	[117]
8	hayesroth	132	5	22.7% (class 3)	[110]
9	ecoli	336	7	2.7% (omL,imL and imS)	[110]
10	yeast	1484	8	3.7% (vac, pox and erl)	[110]
11	abalone	4177	7	8.0% ($age < 5$ or > 15)	[110]
12	glass	214	9	4.2% (tableware)	[110]
13	ionosphere	351	34	35.9% (bad)	[96]
14	pageblocks	5473	10	4.2% (graphic, vertline and picture)	[110]
15	magic04	19020	10	35.2% (hadron)	[110]

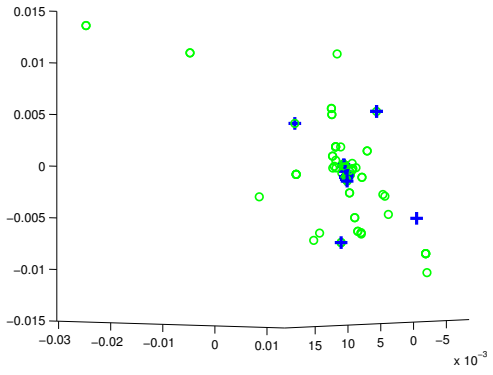


Figure 4.10: Dataset "wdbc" shown on the first three nontrivial eigenvectors. Anomalous instances in green (37.3% of #instance) are more scattered and sparse than normal instances in blue (62.7% of #instance). Therefore these anomalies, although have a large amount of instances, should be treated as many small abnormal clusters instead of a single cluster.

4.8 Experimental Analysis

4.8.1 Experimental Setup

Datasets. To demonstrate the performance of our proposed FDD and LAD, we evaluate our algorithms on fifteen benchmark datasets including seven medical datasets (breastcancer, wdbc, pima, arrhythmia, arcene, prostateTumor and gse24417), four biological datasets (hayeRoth, ecoli, yeast, and abalone), and four physics datasets (glass, ionosphere, pageblocks and magic04), whose statistics are summarized in Table 4.1. All these datasets have been popularly used in anomaly detection research (related references for each dataset are listed in Table 4.1). Such diverse combination of data is intended for our comprehensive studies. In the data preprocessing step, all nominal (including binary) attributes or attributes with missing value are removed.

Anomalies in some of the datasets (wdbc, arrhythmia, prostatetumor etc.), although carrying a large number of instances, have scattered and sparse distribution as shown in Figure 4.10. Therefore the anomalies in these datasets should be treated as a combination of many small anomalous clusters instead of one or a few normal clusters with high density [34] [110], which is consistent with our anomaly definition in Section 4.8.1.

Baselines. We choose seven state-of-the-art competitors in three categories to show the outstanding performance of our proposed FDD and LAD. For kNN-based algorithms, we choose Local Outlier Detection (**LOF**) [14] and Local Correlation Integral (**LOCI**) [113]. Specially, LOCI provides an automatic, data-dictated cut-off to determine whether an instance is an anomaly based on probabilistic reasoning. For attribute-based methods, we include **IForest** [96] and **Mass** [138]. For manifold-based methods, we choose two different manifold-based techniques used in [2] including locally linear embeddings (**LLE**), and isometric feature mapping (**ISM**), followed by LOF to obtain anomalousness measurement. We also include S-trangeness based Outlier Detection algorithm (**StrOUD**) presented in [9]. StrOUD is based on Transductive Confidence Machines, which have been previously proposed as a mechanism to provide individual confidence measures on classification decisions [9].

Evaluation Metrics. Since we have the ground truth of labels for each dataset, we compare our anomaly detection results with labels. For the purpose of theoretical analysis and practical use, we use three evaluation metrics: AUC, F1-Score and macro paired t-tests.

AUC. AUC measures the area under the Receiver Operating Characteristics Curve, which can be interpreted as the probability that the classifier will assign a higher score to a randomly chosen positive example than to a randomly chosen negative example. AUC is commonly used to evaluate anomaly detectors and it is cut-off independent. Detailed definition of AUC can be referred to [102].

F1-Score. In practical use, the anomalousness of all the instances are usually sorted and those instances with higher value are assigned as anomalies. We assume the number of anomalies h is already known (calculated with the ground truth), then the first h instances with the highest anomalousness are selected. We evaluate the estimated results with F1-score of the anomaly class. For more details of F1-score we refer readers to [118].

Macro Paired T-Tests. During the experiment, we also show that our FDD provides more stable anomaly detection accuracy for all the datasets by using macro paired t-tests [164] against each competitor respectively. Note that a score of macro paired t-tests (p-value) should be no more than 0.05 to be considered statistically

significant.

Parameters. First of all, we introduce the parameter settings in our FDD and LAD. Our algorithm FDD has two scaling parameters, the Gaussian scales σ and the environmental temperature T. These two parameters are also used in our LAD (heat diffusion time t has been replaced by $\frac{1}{T}$, check Section 4.6.1 for details). Besides these two parameters, LAD also has another tuning parameter k, the size of neighborhood scope, which is used in the scale-dependent umbrella operator. In our experiments they are set as follows:

- $\boldsymbol{\sigma}$: For local sensitivity, σ in both FDD and LAD are always fixed to be the average distance of each point to its 2-NN (second nearest neighbor).
- $\boldsymbol{T(\frac{1}{t})}$: Specifically we fix $t = 1$ in LAD in all the experiments (except Figure 4.11(h) and 4.12(h)) to avoid the heat dissipation from over-diffusion. For all the FDD experiments and LAD in Figure 4.11(h) and 4.12(h), the range of $T(\frac{1}{t})$ is in $10\text{^}\{-4, -3.8, -3.6, ..., 3.8, 4\}$.
- $\boldsymbol{k}$: k is fixed to be $k = \lceil 1\% \times n \rceil$ (n is the number of instances) in Figure 4.11(i) and 4.12(i). But in the other LAD experiments, the stability of LAD with different k is tested with $k \in \lceil \{1\%, 2\%, 3\%..., 100\%\} \times n \rceil$.

The size of neighborhood scope, k, is a commonly used parameter which also appears in LLE, ISM, LOF, LOCI, and StrOUD. For these algorithms k is also tested in $k \in \lceil \{1\%, 2\%, 3\%..., 100\%\} \times n \rceil$.

The parameter settings of the other algorithms in our experiments are briefly introduced as follows. For LLE and ISM, we fixed $d = 5$ to measure across different k in Table 4.2 and 4.3, and in Figure 4.11(a) 4.11(b) 4.12(a) 4.12(b) as well. But in Figure 4.11(c) 4.11(d) 4.12(c) 4.12(d) we show the stability of LLE and ISM across different $d \in [1, 30]$ by choosing the best k from the previous test for each dataset. In LOCI, radius coefficient is set as $\alpha = 0.5$ which is the same to their paper [113]. As for IForest, to conduct safe and fair comparison, we set ρ and the number of trees nt as the following six combinations: $\rho = 8$ and $nt = 100$ (the number of trees); $\rho = 8$ and $nt = 1000$; $\rho = 256$ and $nt = 10$; $\rho = 256$ and $nt = 100$; $\rho = 256$ and $nt = 500$; $\rho = 256$ and $nt = 1000$. For the same reason, in Mass we set

the sub-sampling size ρ and the number of mass estimation ne as the following six combinations: $\rho = 8$ and $ne = 100$; $\rho = 8$ and $ne = 1000$; $\rho = 256$ and $ne = 10$; $\rho = 256$ and $ne = 100$; $\rho = 256$ and $ne = 500$; $\rho = 256$ and $ne = 1000$. On the other hand, IForest and Mass are based on random sub-sampling which makes their performance unstable. In an attempt to get more stable statistics, for each dataset and parameter setting we run 30 times and compute the average AUC and F1-Score. In Table 4.2 and 4.3, we document the average AUC/F1-score of the best four (out of all six) parameter settings for each dataset.

4.8.2 Comparison of Average Performance

In this subsection we evaluate our proposed FDD and LAD, and the other seven anomaly detection algorithms. Table 4.2 documents the average AUC of each method across their corresponding tuning parameters, and the relative p-value w.r.t to FDD; while Table 4.3 records the average F1-score of each method across their corresponding tuning parameters, and the relative p-value w.r.t to FDD.

In Table 4.2 our proposed FDD and LAD show the first and the second best average AUC score (0.7818 and 0.7758). They boost up the AUC close to or more than 8% compared with the best performance (0.7214 from IForest) among the other methods. For most of the datasets, FDD and LAD has the best or very close to the best performance. Specifically, FDD is the top-three-ranked for all the datasets, meanwhile our LAD, although not all the time, outperforms the other algorithms in most cases. In fact the only two cases, arrhythmia and yeast, when LAD is not among the best four ranks , LAD still reach more than 95% of the best AUC result.

Although some algorithms, such as LOF, IForest and StrOUD, are more efficient in measuring the anomalousness, their methodologies are based on Euclidean space and therefore under the curse of dimensionality. As the number of feature increases, their performances drop significantly on the datasets such as arcene and prostatetumor. The manifold-based algorithms like LLE and ISM are to reduce the vulnerability of simple kernel under the high dimensions. In spite of the fact that LOF measurement on the projection of LLE and ISM show better quality compared with LOF on the input space, it suffers from the inferior manifold reconstruction. Comparably, our FDD and LAD, built upon optimal embedding structure derived

from solid physics theory, provide stronger capability of detecting anomalies in terms of AUC.

As for the macro paired t-tests across all the datasets in Table 4.2, compared with all the other algorithms, our quantum-theory-based FDD has extremely small p-value (less than 1%). Even compared with the other proposed method LAD, FDD still has very small p-value less than 5% with statistical significance. This proves that our FDD has the most stable average performance in terms of AUC.

To have a comprehensive test with a more practical view, we also measure the F1-score and document in Table 4.3. Here we only record the F1-score of the anomalous subset (class) because it is the focus of anomaly detection. Although the heat-diffusion-based LAD shows slightly fluctuating performance compared with its AUC production in Table 4.2, it still surpasses (0.5338 vs 0.5026) IForest, the best among the other algorithms (except FDD) for more than 6%. But the quantum-theory-based FDD, acquires the best F1-score (0.5542), which improves more than 10% based on IForest. Furthermore, the same as demonstrated in AUC, F1-score by FDD is almost persistently (except for glass) ranking top-three among all the algorithms. In the actual application of anomaly detection, the users tend to focus on the detected anomalous subset, instead of the whole label distribution, therefore F1-score tells more story than AUC. In this case, FDD shows more convincing quality in terms of F1-score. This is to say, our proposed FDD has the capability of providing the most desirable label results of anomalies in pratice.

Compared with the basic LOF algorithm, IForest shows passable AUC and F1-score on average, which supports the argument that it is able to take both global and local contexts into consideration. This is different from kNN-based methods (LOF and LOCI) which only concern with instance-wise local context. Compared with LOF, LOCI has more than 8% better AUC and more than 7% better F1-score. This moderately stable and stronger performance comes from the built-in concept of a multi-granularity deviation factor [113]. Although Mass cannot always maintain competitive quality of anomalousness measurement, it has the fastest computation speed compared with all the other competitors.

Table 4.2: Comparison of average AUC of our FDD and LAD, and other seven popular methods across their corresponding parameters (indicated in the parentheses after each method in the first row). For each dataset, the bold-faced number indicates the best method, and the numbers in the parentheses indicate the ranks of our FDD and LAD. Average is the average AUC of each method across all the datasets respectively. A $*$ indicates a p-value of 5% or lower and $**$ indicates a p-value of 1% or lower in the statistical significance test w.r.t. FDD.

Dataset	LLE(k)	ISM(k)	LOF(k)	LOCI(k)	Mass	IForest	StrOUD(k)	LAD(k)	FDD(T)
breastcancer	0.8448	0.6411	0.6423	0.8637	0.3572	0.9914	**0.9926**	0.9820 (4)	0.9870 (3)
wdbc	0.6502	0.7280	0.7289	0.8548	0.7823	0.8218	**0.9285**	0.9005 (3)	0.9049 (2)
pima	0.5964	0.6224	0.6188	0.6165	0.6094	0.6848	0.6434	0.7101 (2)	**0.7119** (1)
arrhythmia	0.6720	0.6942	0.7356	0.7475	0.6924	0.7425	**0.7530**	0.7192 (6)	0.7472 (3)
arcene	0.5118	0.5300	0.4417	0.3964	**0.5710**	0.3112	0.3242	0.5551 (2)	0.5551 (2)
prostatetumor	0.4637	0.4732	0.4757	0.4667	0.4203	0.4090	0.4279	0.5314 (2)	**0.5359** (1)
gse24417	0.5055	0.5176	0.5743	0.5625	0.5693	0.5688	0.5860	0.5860 (2)	**0.5895** (1)
hayesroth	0.8413	0.5713	0.5828	0.6438	0.6843	0.9846	0.7020	0.9903 (2)	**0.9905** (1)
ecoli	0.8502	0.8279	0.8178	0.8754	0.7690	0.8748	0.8972	0.8960 (3)	**0.9052** (1)
yeast	0.5146	0.6160	**0.6285**	0.6063	0.5665	0.6127	0.6149	0.6115 (6)	0.6212 (2)
abalone	0.6612	0.6793	0.6720	0.6767	0.6345	0.6933	0.6989	0.7299 (2)	**0.7332** (1)
glass	0.6346	0.6206	0.6205	0.7647	**0.8813**	0.6940	0.7962	0.8732 (3)	0.8737 (2)
ionosphere	0.5814	0.6206	0.7719	0.8482	0.8167	0.8517	0.8605	**0.9335** (1)	0.9253 (2)
pageblocks	0.7619	0.7740	0.7043	0.7759	0.8813	0.8561	0.5511	0.8893 (2)	**0.8939** (1)
magic04	0.5883	0.6466	0.6599	0.6747	0.6798	0.7214	0.6775	0.7286 (2)	**0.7520** (1)
Average	0.6452**	0.6396**	0.6390**	0.6916**	0.6611**	0.7214**	0.6861**	0.7758* (2)	**0.7818** (1)

Table 4.3: Comparison of average F1-score of our FDD and LAD, and other seven methods across their corresponding parameters (indicated in the parentheses after each method in the first row). For each dataset, the bold-faced number indicates the best method, and the numbers in the parentheses indicate the ranks of our FDD and LAD. Average is the average F1-score of each method across all the datasets respectively. A $*$ indicates a p-value of 5% or lower and $**$ indicates a p-value of 1% or lower in the statistical significance test w.r.t. FDD.

Dataset	LLE(k)	ISM(k)	LOF(k)	LOCI(k)	Mass	IForest	StrOUD(k)	LAD(k)	FDD(T)
breastcancer	0.7282	0.5277	0.5235	0.6971	0.2468	0.9411	**0.9439**	0.9144 (4)	0.9351 (3)
wdbc	0.5064	0.6149	0.6159	0.7406	0.6544	0.6745	**0.8161**	0.7445 (3)	0.7610 (2)
pima	0.4524	0.4772	0.4734	0.4892	0.4361	0.5289	0.5154	**0.5458** (1)	0.5426 (2)
arrhythmia	0.5968	0.6086	0.6469	**0.6711**	0.6067	0.6535	0.6654	0.6630 (4)	0.6667 (2)
arcene	0.4659	0.4639	0.3802	0.3786	0.4838	0.2730	0.3025	0.4882 (2)	**0.4941** (1)
prostatetumor	0.4808	0.4894	0.5046	0.5082	0.4849	0.4794	0.4569	0.5412 (2)	**0.5503** (1)
gse24417	0.3000	0.3017	0.3874	0.4005	0.3772	0.3766	0.4038	0.3774 (5)	**0.4200** (1)
hayesroth	0.5980	0.3233	0.3387	0.3760	0.4113	0.8888	0.3811	0.8997 (2)	**0.9042** (1)
ecoli	**0.5778**	0.4544	0.3444	0.3333	0.2570	0.5681	0.5578	0.4422 (6)	0.5691 (2)
yeast	0.0278	**0.2962**	0.2778	0.2909	0.0627	0.2505	0.2909	0.2909 (2)	0.2909 (2)
abalone	0.2061	0.2072	0.2845	0.2909	0.2891	0.3057	0.2402	0.3005 (3)	**0.3089** (1)
glass	0.1067	0.1367	0.1278	0.1111	**0.1875**	0.1083	0.1111	0.1133 (4)	0.1111 (5)
ionosphere	0.4047	0.4633	0.6312	0.6975	0.5955	0.6631	0.6587	**0.8331** (1)	0.8216 (2)
pageblocks	0.1991	0.2251	0.2009	0.2165	**0.3675**	0.2449	0.1905	0.3448 (3)	0.3588 (2)
magic04	0.4002	0.4079	0.4951	0.5166	0.5070	**0.5836**	0.4981	0.5081 (4)	0.5775 (2)
Average	0.4035**	0.3980**	0.4177**	0.4479**	0.3978**	0.5026**	0.4500**	0.5338* (2)	**0.5542** (1)

4.8.3 Comparison of Stability

To systematically manifest the stability against parameter tuning of each method, we run experiments for LLE, ISM, LOF, LOCI, StrOUD, and our proposed FDD and LAD across their corresponding parameter tuning respectively, and record the AUC in Figure 4.11 and F1-score in Figure 4.12. We select seven small datasets: wdbc, pima, arrhythmia, ecoli, yeast, glass and ionosphere for the stability test. In theory, smaller datasets should be more sensitive to the change of scaling parameters. Therefore these seven datasets are the more effective choices to show whether the algorithms perform robustly during adjusting their parameters.

For the size of neighborhood k and the number of embedding dimension d, LLE undergoes fluctuation especially on wdbc, ecoli and glass. It is mainly because LLE has strong assumption that the data is densely sampled and the embedding structure is locally approximately linear, yet it is not true for many real world datasets. Similarly, ISM's results vary dramatically as k changes especially for ecoli, glass, and ionosphere, although later ISM is comparably stable while tuning d. The reason is that ISM is highly vulnerable to the local data perturbation, as the embedding given by the ISM tends to recover the geodesic distances between points on the manifold which is very locally sensitive compared with random walk [84] [142].

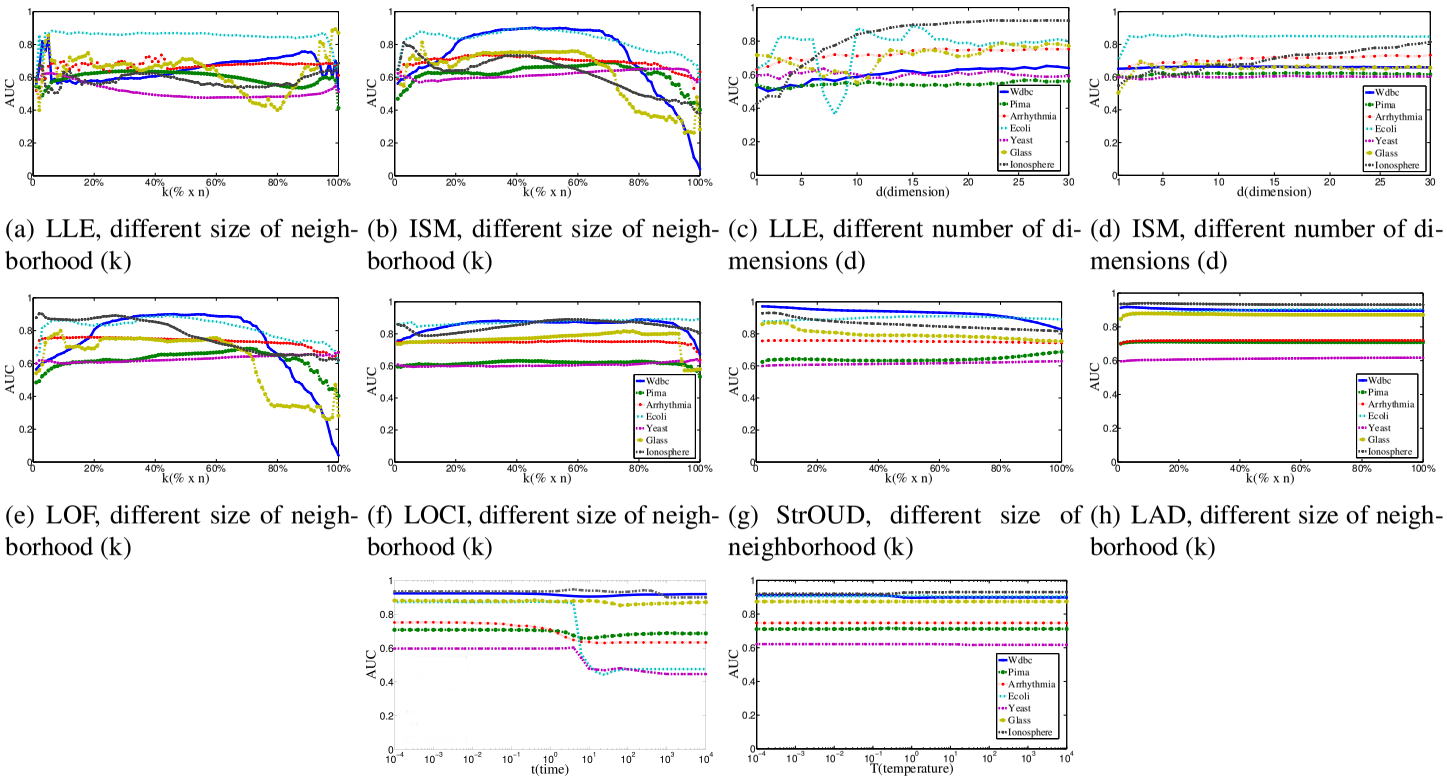
Compared with LOF, LOCI performs robustly with different k, which results from that its proposed multi-granularity deviation factor can more intuitively cope with local density variations in the feature space and detect both isolated anomalies as well as outlying clusters [113]. LOF, although occasionally beats LOCI with certain k, shows seriously unstable performance as k changes, which can be simply explained as follows: LOF is based on a direct normalization of anomaly scores for an inadaptive neighborhood.

StrOUD demonstrates not only its effectiveness and efficiency (since it is totally based on the input space without any projection), but also its AUC stability during the change of k. However, in terms of AUC result shown in Figure 4.11(g), the curves have different patterns: StrOUD reaches higher AUC with smaller k on wdbc, glass and ionosphere datasets, but it has better AUC result with larger k on pima and ecoli. In the test of F1-score in Figure 4.12(g), StrOUD shows serious instability on ecoli as k changes. Part of the reason comes from that StrOUD is

principally built upon Euclidean distance on the input space, which cannot faithfully reveal the intrinsic dissimilarity and density on the non-linear distributed data. Furthermore, it becomes even worse on the more complex datasets with large number of features, as already confirmed in Table 4.2 and 4.3.

Compared with the above algorithms, our proposed LAD shows the best stability against the change of k, as demonstrated in both AUC (Figure 4.11(h)) and F1-score (Figure 4.12(h)). This is because LAD has an inherent relationship with heat diffusion and random walk. More specifically, LAD has a strong probabilistic interpretation, with provides a power against noise appearance or neighborhood size perturbation, as long as they are not severe enough to perturb the general neighborhood statistics.

Importantly, we test the performance of our FDD and LAD with different physical parameters: heat diffusion time t in LAD and environmental temperature T. The same as what we already described in Figure 4.7 and Section 4.7.1, LAD may lost the power of local density description especially for local anomalies when t goes large, which means over-diffusion. Therefore the AUC curves by LAD of ecoli and yeast significantly drop in Figure 4.11(i). Likewise, the F1-score by LAD in Figure 4.12(i) shows comparably unstable trends as t increases. On the contrary, Figure 4.11(j) establishes the robustness of FDD. Compared with LAD, FDD has more potential in combating off-the-sweet-spot physical parameter since it is constructed on "polarized" manifold space and it has additional stabilizing factors which help to balance the riskiness from extreme cases. Another thing worth paying attention here is that in Figure 4.12(j), FDD does not always maintain strong stability across all the datasets. But comparatively, our proposed FDD still retains certain level of anomaly detection quality as parameter changes. And most importantly, FDD outperforms the existing baselines in terms of average performance and steadiness with the purpose of detecting anomalies. The robustness property is equally significant for domain experts who do not have strong machine learning background. Since there is no too much clue to tune the traditional yet unstable algorithms such as LLE, ISM and LOF, it is much more comfortable for the domain experts to utilize robust anomaly detection algorithms for the domain data analysis. Therefore our proposed FDD is very hands-on and effective on many real world applications.



(a) LLE, different size of neighborhood (k)

(b) ISM, different size of neighborhood (k)

(c) LLE, different number of dimensions (d)

(d) ISM, different number of dimensions (d)

(e) LOF, different size of neighborhood (k)

(f) LOCI, different size of neighborhood (k)

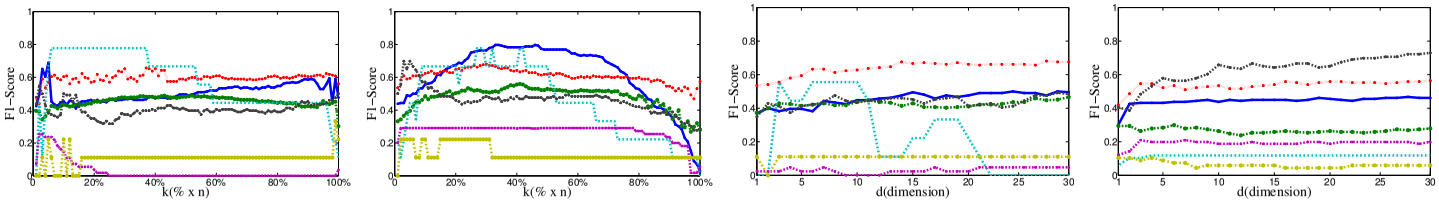
(g) StrOUD, different size of neighborhood (k)

(h) LAD, different size of neighborhood (k)

(i) LAD, different diffusion time (t)

(j) FDD, different temperature (T)

Figure 4.11: AUC stability with different parameters.

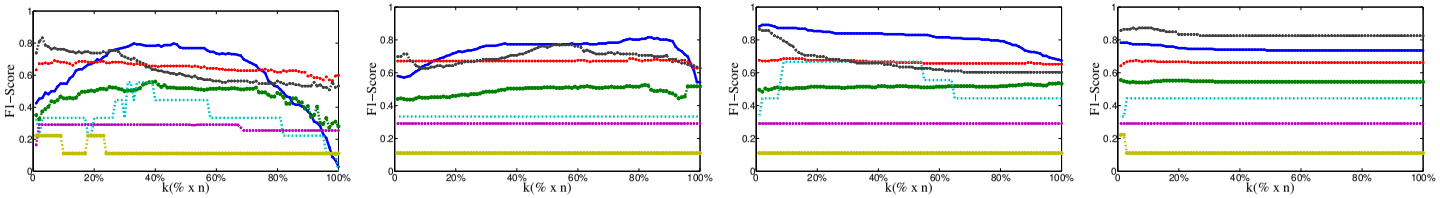


(a) LLE, different size of neighborhood (k)

(b) ISM, different size of neighborhood (k)

(c) LLE, different number of dimensions (d)

(d) ISM, different number of dimensions (d)

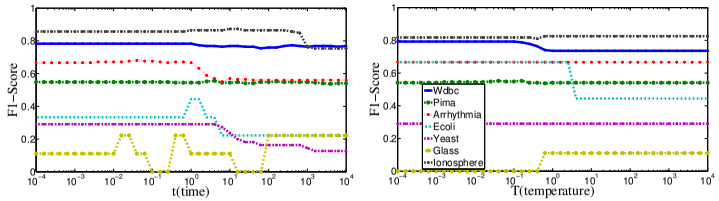


(e) LOF, different size of neighborhood (k)

(f) LOCI, different size of neighborhood (k)

(g) StrOUD, different size of neighborhood (k)

(h) LAD, different size of neighborhood (k)



(i) LAD, different diffusion time (t)

(j) FDD, different temperature (T)

Figure 4.12: F1-Score stability with different parameters.

4.8.4 Comparison of Different Laplacians

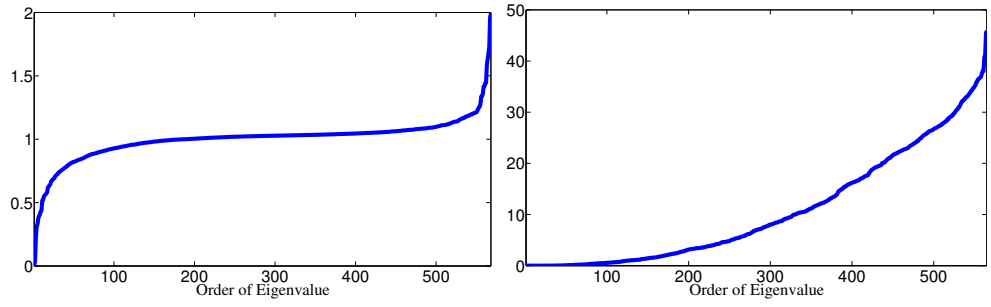
In Section 4.2.1 and 4.6.2 we introduce our selection of Laplacian in LAD and FDD. Here we analyze the reasons through experiments of LAD and FDD with the five Laplacians and 15 datasets respectively. To save space, we only list the average AUC in Table 4.4 and 4.5.

Table 4.4 shows the effect of different Laplacians on LAD, and L_{rw} has the best average performance. We also note that except L_{nn}, there is no too much difference among the four (normalized) Laplacians. Detailed analysis are listed below:

1. The similar performance of L_{rw}, L_{sym}, L_{fp} and L_{lbn} can be explained by the use of umbrella operator in LAD, which gives attention on the weighted distance between each point and its neighborhood. Therefore as long as the eigenvalues are normalized, and there is deviation between the normalized eigen-components, especially, the corresponding value of any anomaly and its surrounding normal instances in the eigenvectors, LAD can capture such deviation regardless the choice of normalized Laplacians. The reason we emphasize L_{rw} on LAD is that LAD is based on heat diffusion, and heat diffusion in classical physics has better interpretation with particles' random walk.

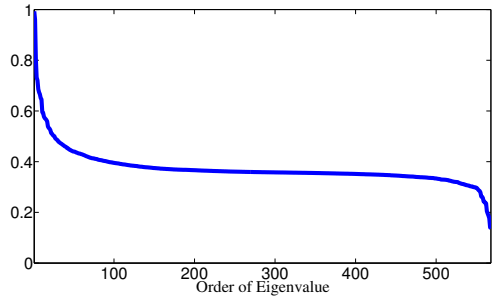
2. The reason why LAD fails on top of L_{nn} relates to the unnormalized eigenvalue distribution. Figure 4.13(a) and 4.13(b) show the eigenvalues (sorted in ascending order) derived from L_{rw} and L_{nn} on dataset "wdbc" correspondingly. Without normalization, the eigenvalues in Figure 4.13(b) increase "exponentially", and only a small portion of eigencomponents in Heat Diffusion (Figure 4.13(d)) are given stable and large enough weights, while the other eigencomponents, even those informative, are gone away quickly. Comparatively, the eigenvalues derived from L_{rw} show an inverse-hyperbolic-tangent-like distribution. So the consequent Heat Diffusion (Figure 4.13(c) with $t = 1$) gives very high weights on the first a few eigencomponents and less but non-negligible weights on most of the following ones. Therefore the normalized Laplacians such as L_{rw} weights the eigencomponents more safely, even conservatively, compared with L_{nn}.

3. Note that if diffusion time goes too large, Heat Diffusion will only emphasize the first few eigencomponents and ignore all the following, as shown in Figure 4.13(g). Therefore HKS and even LAD fail with too large t.

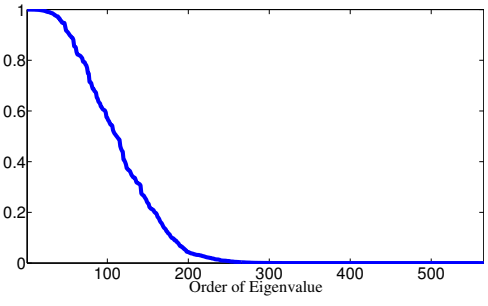


(a) Eigenvalue by L_{rw}

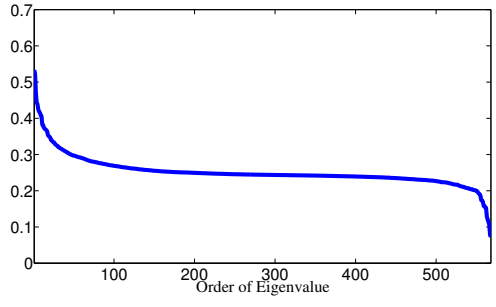
(b) Eigenvalue by L_{nn}



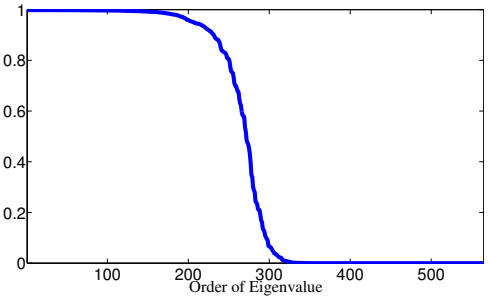
(c) HD by L_{rw} with $t = 1$



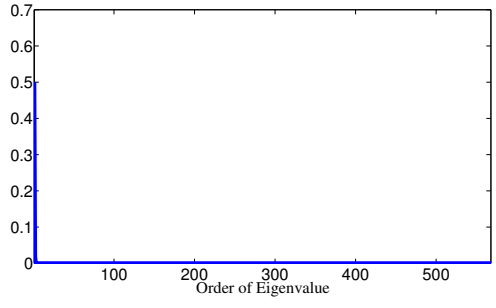
(d) HD by L_{nn} with $t = 1$



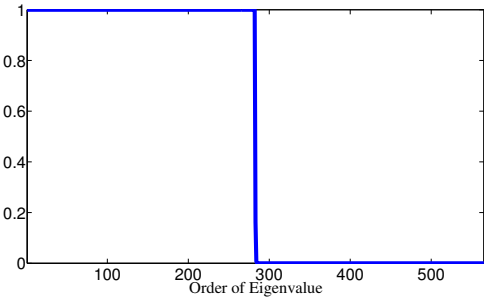
(e) FD by L_{rw} with $T = 1$



(f) FD by L_{nn} with $T = 1$



(g) HD by L_{rw} with $t = 100$



(h) FD by L_{nn} with $T = 0.01$ ($t = 100$)

Figure 4.13: Eigenvalue by L_{rw} and L_{nn}, and the corresponding weighted function of Heat Diffusion (HD, Equation 4.21) and Fermi-Dirac Distribution (FD, Equation 4.17). The dataset is "wdbc".

Table 4.4: Comparison of average AUC by LAD with different Laplacians. For each dataset, the numbers in the parentheses indicate the ranks of each Laplacian. Average is the average AUC of each Laplacian across all the datasets respectively.

Dataset	LAD with L_{nn}	LAD with L_{sym}	LAD with L_{fp}	LAD with L_{lbn}	LAD with L_{rw}
breastcancer	0.9821(4)	0.9861(2)	0.9865(1)	0.9822(3)	0.9820(5)
wdbc	0.7241(5)	0.9044(2)	0.9047(1)	0.9015(3)	0.9005(4)
pima	0.5806(5)	0.7130(1)	0.7121(2)	0.7103(3)	0.7101(4)
arrhythmia	0.7473(1)	0.7403(2)	0.7261(3)	0.7183(5)	0.7192(4)
arcene	0.4316(5)	0.5369(3)	0.5101(4)	0.5523(2)	0.5551(1)
prostatetumor	0.5136(5)	0.5396(3)	0.5608(2)	0.5808(1)	0.5314(4)
gse24417	0.5587(4)	0.5888(1)	0.5791(3)	0.5426(5)	0.5860(2)
hayesroth	0.9916(3)	0.9926(2)	0.9929(1)	0.9903(4)	0.9903(4)
ecoli	0.4084(5)	0.8949(4)	0.8951(3)	0.8955(2)	0.8960(1)
yeast	0.4135(5)	0.6114(4)	0.6115(1)	0.6115(1)	0.6115(1)
abalone	0.6335(5)	0.7668(1)	0.7667(2)	0.7666(3)	0.7299(4)
glass	0.7132(5)	0.8765(1)	0.8744(2)	0.8732(3)	0.8732(3)
ionosphere	0.9466(1)	0.9336(2)	0.9324(5)	0.9335(3)	0.9335(3)
pageblocks	0.7091(5)	0.7958(2)	0.7932(4)	0.7947(3)	0.8893(1)
magic04	0.6513(5)	0.7300(1)	0.7289(3)	0.7296(2)	0.7286(4)
Average	0.6681(5)	0.7740(2)	0.7716(4)	0.7722(3)	0.7758(1)

In Section 4.6.2 we prove that L_{nn} is the best choice for FDD. Table 4.5 confirms that L_{nn} has the best average performance on FDD. Here we give brief analysis:

1. FDD, different from LAD, doesn't use umbrella operator but instead relies on the energy distribution functions and the eigendecomposition on Laplacians. Hence the Laplace operator becomes more essential on the construction of FDD.

2. In Figure 4.13(f) and 4.13(h), Fermi-Dirac distribution function (FD) with L_{nn} robustly assigns similar and very stable weight to the first $200+$ eigencomponents of "wdbc" dataset, regardless of the value of T. Comparatively, FD with normalized Laplacians such as L_{rw} (Figure 4.13(e)) embraces too much eigencomponents, even including noisy ones. Without the help of umbrella operator, these noisy components will bring unstable anomaly detection results, as shown in Table 4.5.

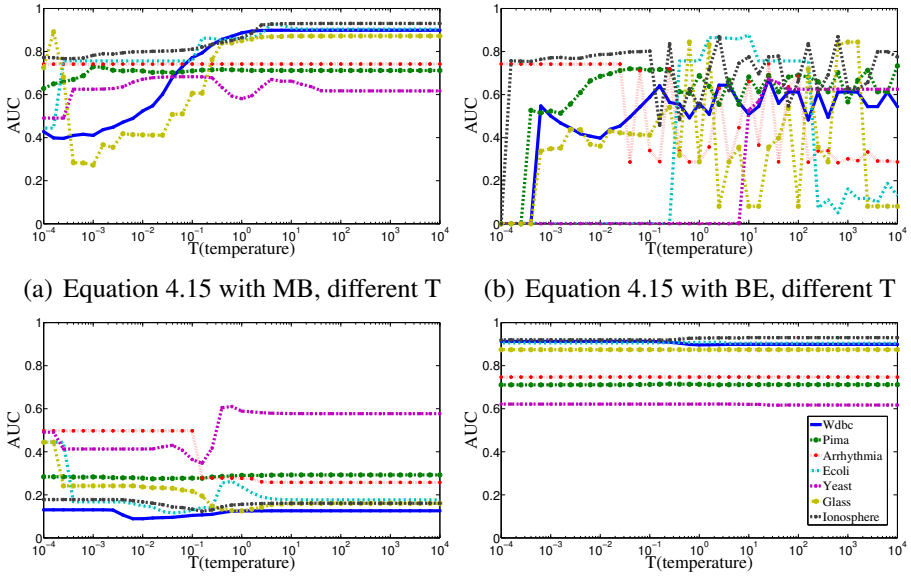
Table 4.5: Comparison of average AUC by FDD with different Laplacians. For each dataset, the numbers in the parentheses indicate the ranks of each Laplacian. Average is the average AUC of each Laplacian across all the datasets respectively.

Dataset	FDD with L_{nn}	FDD with L_{sym}	FDD with L_{fp}	FDD with L_{lbn}	FDD with L_{rw}
breastcancer	0.9870(1)	0.9187(2)	0.9088(3)	0.3171(5)	0.5612(4)
wdbc	0.9049(1)	0.8236(2)	0.5337(3)	0.2464(4)	0.2139(5)
pima	0.7119(1)	0.6886(3)	0.7027(2)	0.4344(5)	0.6032(4)
arrhythmia	0.7472(1)	0.7384(2)	0.7050(3)	0.6131(5)	0.6639(4)
arcene	0.5551(1)	0.5538(2)	0.5384(4)	0.5433(3)	0.4648(5)
prostatetumor	0.5359(5)	0.5415(4)	0.5792(2)	0.5427(3)	0.5904(1)
gse24417	0.5895(1)	0.5815(2)	0.5529(3)	0.4964(5)	0.5504(4)
hayesroth	0.9905(1)	0.6990(4)	0.9418(2)	0.1000(5)	0.9183(3)
ecoli	0.9052(1)	0.8913(3)	0.8977(2)	0.0832(5)	0.8906(4)
yeast	0.6212(2)	0.6205(3)	0.6614(1)	0.3654(5)	0.5927(4)
abalone	0.7332(2)	0.7526(1)	0.5190(3)	0.4291(4)	0.2504(5)
glass	0.8737(2)	0.7702(3)	0.8883(1)	0.3149(5)	0.6141(4)
ionosphere	0.9253(1)	0.6320(2)	0.3558(4)	0.1736(5)	0.5781(3)
pageblocks	0.8939(1)	0.6888(2)	0.2993(4)	0.4201(3)	0.2247(5)
magic04	0.7520(1)	0.7502(2)	0.3411(4)	0.6722(3)	0.3357(5)
Average	0.7818(1)	0.7100(2)	0.6283(3)	0.3835(5)	0.5268(4)

4.8.5 Comparison of Energy Distribution Functions

To have a better understanding of different distribution function effects (introduced in Section 4.6.1) on anomaly detection, we test their stability. Here we integrate all the four functions, namely Maxwell-Boltzmann distribution (MB), Bose-Einstein distribution (BE), Gaussian distribution (GD) and our chosen Fermi-Dirac distribution (FD), into Equation 4.15 with L_{nn} operator, and calculate the anomaly detection scores in AUC and F1-Score.

The results are illustrated in Figure 4.14 and 4.15. The stability of GD is reasonable, but the scores are apparently lower than the other three. BE shows the most fluctuant results in both AUC and F1-Score because it doesn't have the smoothing term "plus one". MB suffers from extremely small temperature T, which is similar to the fact that HKS suffers from large diffusion time t, therefore generally it has a dropping trend when T becomes smaller. Our FDD, although not always maintains the best performance, has the best average result and the most stability in both AUC and F1-score.



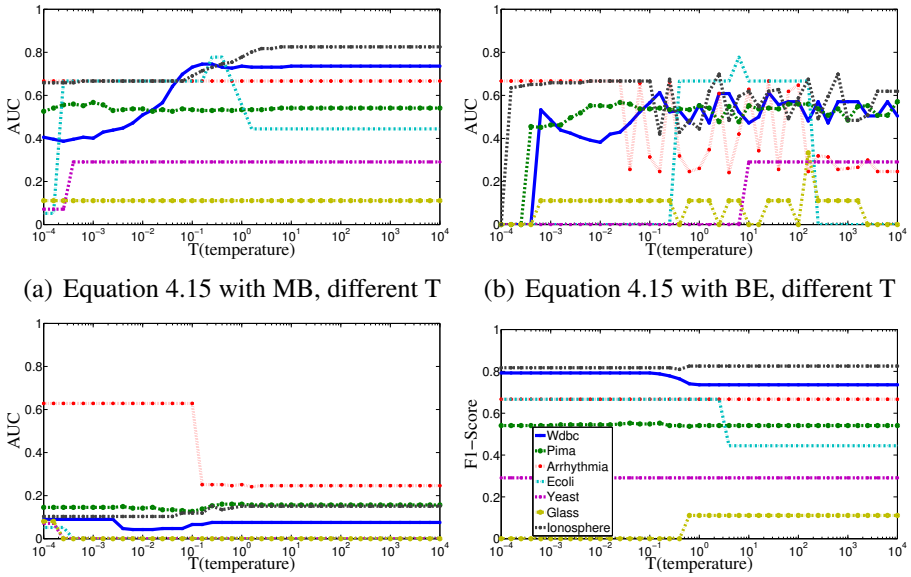
(a) Equation 4.15 with MB, different T

(b) Equation 4.15 with BE, different T

(c) Equation 4.15 with GD, different T

(d) FDD (Equation 4.15 with FD), different T

Figure 4.14: AUC stability with different energy distribution functions.



(a) Equation 4.15 with MB, different T

(b) Equation 4.15 with BE, different T

(c) Equation 4.15 with GD, different T

(d) FDD (Equation 4.15 with FD), different T

Figure 4.15: F1-Score stability with different energy distribution functions.

4.8.6 Comparison of Efficiency and Effectiveness

In this subsection we analyze the efficiency and effectiveness of LAD and FDD with a small portion of eigencomponents. Also to obtain a short amount of running time we only use GAU ($O(n^2m)$) instead of AGK ($O(n^2m^2)$) here. Suppose the size of dataset (number of instances) is n, the first $max(\lfloor n/50 \rfloor, 10)$ eigencomponents are used to compute LAD and FDD, which note as LAD$_f(GAU)$ and FDD$_f(GAU)$. The AUC results are shown in Table 4.6. LAD$_f(GAU)$ obtains about 94% of AUC by full version LAD(AGK), while FDD$_f(GAU)$ only gets 67% of full FDD(AGK). Apparently LAD doesn't suffer a lot from small amount of eigenvectors compared with FDD, which can be explained by the effect of umbrella operator and the illustration results in Figure 4.13. Table 4.7 shows the running time comparison of a few algorithms. The parameter settings are documented in previous Section 4.8.1. Specifically IForest's running time is measured by the average of the best four parameter settings listed in Section 4.8.1. LAD$_f(GAU)$ is two times faster than LAD on average, and also more efficient than StrOUD and LOCI. The most efficient one among these five algorithms is IForest, which is extremely fast

on large dataset such as “magic04” and “pageblocks”. But it is worth to notice that $\mathrm{LAD}_f(GAU)$ is more efficient on the small datasets.

Table 4.6: Comparison of AUC between full and fast version of LAD and FDD.

Dataset	LAD(AGK)	LAD$_f(GAU)$	FDD(AGK)	FDD$_f(GAU)$
breastcancer	0.9820	0.9874	0.9870	0.4856
wdbc	0.9005	0.9681	0.9049	0.4883
pima	0.7101	0.4804	0.7119	0.5105
arrhythmia	0.7192	0.5666	0.7472	0.4973
arcene	0.5551	0.5510	0.5551	0.5055
prostatetumor	0.5314	0.6185	0.5359	0.5041
gse24417	0.5860	0.4449	0.5895	0.5054
hayesroth	0.9903	0.9311	0.9905	0.4908
ecoli	0.8960	0.9377	0.9052	0.8763
yeast	0.6115	0.6013	0.6212	0.5548
abalone	0.7299	0.7387	0.7332	0.4250
glass	0.8732	0.8232	0.8737	0.7545
ionosphere	0.9335	0.8241	0.9253	0.3000
pageblocks	0.8893	0.7188	0.8939	0.4992
magic04	0.7286	0.7307	0.7520	0.5064
Average	0.7758	0.7282	0.7818	0.5269

Table 4.7: Comparison of running time (in seconds).

Dataset	LAD(AGK)	LAD$_f(GAU)$	LOCI	StrOUD	IForest
breastcancer	0.9125	0.2732	45.5966	6.2400	4.8025
wdbc	0.8137	0.2822	129.7696	3.9162	3.8211
pima	1.3304	0.4589	231.3238	6.4149	5.3805
arrhythmia	0.5204	0.2265	78.8057	3.1139	3.8083
arcene	0.1772	0.0842	12.6367	8.3564	1.4613
prostatetumor	0.1442	0.0871	3.1904	7.0358	0.5022
gse24417	0.4080	0.1791	65.8869	8.6048	3.2912
hayesroth	0.1058	0.0735	4.2058	0.2284	0.6599
ecoli	0.3294	0.1223	32.6413	1.4403	2.7602
yeast	5.9344	0.5448	851.1367	25.5289	4.8630
abalone	72.5052	8.8877	17112.6534	192.0656	5.9136
glass	0.1414	0.0867	14.7054	0.6044	1.7353
ionosphere	0.3251	0.1003	41.3065	1.4513	2.9742
pageblocks	92.3223	35.4092	33725.4086	320.3760	6.3389
magic04	1297.0366	471.1242	252425.9877	864.9672	6.6197
Average	98.2004	34.5293	20318.3503	96.6896	3.6621

4.9 Chapter Summary

This chapter documents physics-based methodology of unsupervised anomaly detection. The first algorithm we propose is Local Anomaly Descriptor (LAD), which is based on heat diffusion and scale-dependent umbrella operator. Its capability of representing local density relies on a short time heat dissipation and an informative neighborhood that is guaranteed by the scale-dependent umbrella operator. Another anomaly detection method we proposed is Fermi Density Descriptor (FDD). It is built upon a "polarized" manifold projection and quantum motion probability measured by Fermi-dirac energy distribution. We also analyze the utilization of Anisotropic Gaussian Kernel (AGK) and the best choice of graph Laplacian with the purpose of anomaly detection. Compared with the existing algorithms, our proposed LAD and FDD exhibit better average performance and stability in our extensive experiments. Moreover, FDD demonstrates its robustness across different physics scaling parameters compared with LAD.

Chapter 5

Noise-Resistant Unsupervised Feature Selection via Multi-Perspective Correlations

5.1 Chapter Introduction

Many real world applications have high dimensionality in their feature space. A larger number of features can be associated with expensive data collection cost, more difficulty in model interpretation, expensive computational cost, and sometimes decreased generalization ability. These challenges are commonly referred to "the curse of dimensionality", and motivate a plethora of research to find a well representative feature subset and thereby reduce the number of features before actual machine learning and analysis. Many feature selection approaches have been developed [114] [98] [123] [35] [43] [106] [44]. In many applications, usually data has no label information, since it is too expensive or difficult to assign labels by experts. Therefore, it is important to develop an unsupervised approach which can perform feature selection task without labels. Compared with the supervised case, the unsupervised feature selection is much more challenging because of the lack of prior knowledge. In this chapter, we focus on an unsupervised feature selection due to its broad applicability.

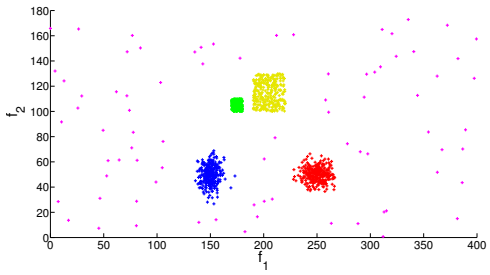
The goal of feature selection is to minimize information loss when removing

the noise and redundancy in the feature space [111], therefore can achieve better 1) model interpretation, 2) computational efficiency, and 3) generalization ability. However, there are significant challenges associated with many existing unsupervised feature selection algorithms:

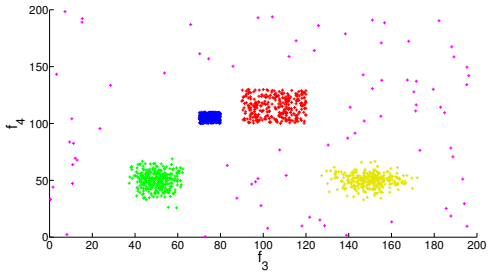
1. Feature importance is usually more about a "local" conception than a "global" one. To obtain a better representative feature subsets, the feature impact associates with different low-embeddings or spectrums need to be considered [35]. Besides, the perspective of instances is also indispensable since some features may only have strong correlation with certain instances with respect to certain spectrums. Therefore it is necessary to design a feature selection algorithm based on such **multi-perspective correlation**.

2. Real world datasets contain many **noisy features** (such as f_5 and f_6 shown in Figure 5.1(c)). These noisy features have negative impacts and make it difficult to identity the informative features, especially for the existing unsupervised feature selection algorithms [35] [76] [62] [89].

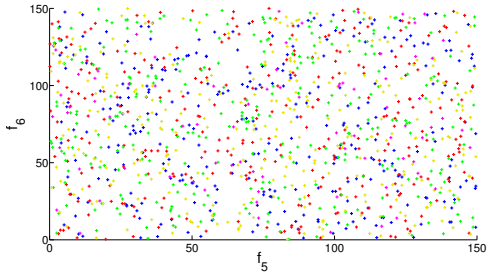
3. **Noisy observations/instances** (colored as purple in Figure 5.1(a) and 5.1(b)) are also very common in real world applications. When a dataset has a significant number of noisy instances, feature importance are hard to discover by most of unsupervised feature selection algorithms [35] [89] [62] [88] due to that the weights of feature become influenced by noisy instances.



(a) Distribution on f_1 and f_2



(b) Distribution on f_3 and f_4



(c) Distribution on f_5 and f_6

Figure 5.1: Synthetic dataset with four clusters (colored with red, blue, yellow and green respectively), each has 300 instances and 34 features. In addition, there are 80 noisy instances (colored as purple). Figure 5.1(a) shows the feature subspace of f_1 and f_2, where the blue and red clusters have a Gaussian distribution, while green and yellow clusters show a uniform distribution in a rectangle area. Figure 5.1(b) shows the feature subspace of f_3 and f_4, where blue and red clusters show uniform distribution in a rectangle area, while green and yellow clusters have a Gaussian distribution. The other 30 features are all noisy, for example f_5 and f_6 shown in Figure 5.1(c). Through the experimental results listed in Table 5.1 we can see that noisy instances can become a hurdle for feature selection, and noisy features, with their quantity even more than that of the informative (useful) ones, could be another issue.

Table 5.1: Clustering results of synthetic dataset in Figure 5.1. The size of selected feature subset is 4 for all the five feature selection algorithms. We run each algorithm 30 times on the dataset with all instances (including normal and noisy instances), and also on the subset of the normal instances (without any noisy instance). We report the average NMI score only on the normal instances.

Algorithms	K-means	NJW	SPEC[161]	LS[62]	MCFS[35]	NDFS[89]	NRFS (our algorithm)
NMI(all instances)	0.1571	0.1002	0.0017	0.0069	0.0032	0.3147	1.0000
NMI(normal instances)	0.2665	0.1097	0.0032	0.0071	0.0094	0.5004	1.0000

To solve these problems, our proposed method, called **Noise-Resistant Feature Selection (NRFS)**, designs a feature selection strategy based on multi-perspective correlation measurement which is effective and robust to both noisy observations and noisy features. By selecting representative instances via density distribution statistics, we reduce the occurrence of the noise observations.For each feature, we compute its local correlation with regard to instance representative. Such local correlations are evaluated with respect to each global spectrum (or trend) of data to find the informative features. Noisy features tend to have lower local correlations across all of the global spectrums compared to the informative ones, while the locally informative features tend to show strong association to at least one global spectrum. We comprehensively considerate all the correlation scores and obtain the informative feature subset. Our work in this chapter has the following contributions:

1. Our proposed NRFS selects features **under local context** instead of global context. We build a set of similarity matrices, where each similarity matrix is constructed using a local feature subspace (each feature and its nearest neighbor features) (Section 5.3.1). By doing this, we have a local perspective w.r.t each instance and feature, and measure their local correlation with the global spectrums (Section 5.3.2).

2. In order to mitigate the influence of noisy instances, we propose the **Noise-Resistant Density-Preserving Sampling** (Section 5.4). It combines both anomaly detection [70] and Density-Preserving Sampling [15], and selects only instance representatives from the original dataset. By only analyzing the feature impact on these representatives, we have a noise-instance-resistant algorithm.

3. Our proposed NRFS has a **more stable performance** in that it selects features by comprehensively considering multi-perspective correlation for each feature, each instance representative, and each global spectrum (Section 5.3.3).

4. Our proposed NRFS combines all the above contributions in a **well-organized framework** (Section 5.5), to deliver a more robust feature selection algorithm, as shown in our systematic benchmark evaluation (Section

5.6).

5.1.1 Related Works

He et al. [62] proposed Laplacian Score (LS) which is one of the earliest work to seek features with respect to the manifold structure. It uses a nearest neighbor graph to model the local geometric structure of the data and selects those features which are smoothest on the manifold graph [35]. Similarly, Spectral Feature Selection (SPEC) [161] obtains the feature importances by estimating the feature consistency with the spectrum of a matrix derived from a similarity matrix on the whole feature space. Jiang et al. pointed out the untrustworthiness of the similarity matrix due to noise, and designed Eigenvalue Sensitive Criteria (EVSC) [76] which evaluates the feature importances by measuring the change of graph Laplacian's eigenvalues. Although these methods could find features that are related to the manifold structure to some degree, they cannot necessarily discriminate the feature importances because they are only based on the global context without local perspective and noise resistance.

Recently many algorithms perform feature selection simultaneously during the model building process [162]. In their work, the embedded modeling usually treats feature selection as a part of training process. The feature importances are obtained by optimizing the objective function of the learning model. The method in [149] puts a l_0-norm constraint into the proposed objective function to achieve sparse and efficient solution. l_1-norm has been used in [150] and Multi-Cluster Feature Selection (MCFS) [35] to recover the global distribution pattern on either similarity or dimensionality on the manifold space. Algorithms in [154] [67] and Nonnegative Discriminative Feature Selection (NDFS) [89] use $l_{2,1}$-norm regularization to achieve similar objectives. Although these methods are effective and robust to some degree, they only focus on the global feature importances by measuring how much each feature can preserve the global distribution pattern on the low embedding dimensions (eigenvectors). Therefore they cannot reveal the local correspondence between each feature-instance pair.

In general, the aforementioned unsupervised feature selection algorithms conduct feature selection globally by producing a common feature subset across all

instances at the same time. This, however, might fail to deal with real world noisy datasets in practice, where feature selection becomes challenging in the presence of noisy observations, and where the local intrinsic property of data plays more important role [87]. Li et al. proposed the Localized Feature Selection algorithms [86] [87] which tend to find the optimal feature subsets for each cluster. But these algorithms are either based on K-means or Bayesian variational learning, and not practically robust to real world datasets due to the lack of manifold awareness and noise effect mitigation.

Although projected clustering [1], subspace clustering [54] [83] and co-clustering algorithms [26] [37] can detect local structure through simultaneously clustering on instances and features of a dataset, they cannot provide the relative importance value of each feature. Secondly, finding the correct subspace to define a suitable group of objects is a difficult problem, since cluster objects may reside in arbitrarily oriented, affine subspaces [83]. In addition, most of subspace clustering methods are formulated only for a mixture of linear manifolds and do not work well in the presence of nonlinear manifolds [54].

5.1.2 Motivations

We illustrate our motivation using a synthetic noisy dataset with 1280 instances and 34 features in Figure 5.1. The dataset contains noise in both instance space and feature space. It has four clusters, each cluster contains 300 instances and colored with red, blue, yellow and green respectively. We also added 80 noisy instances which are colored with purple. On the other hand, only the first four features are significantly important: the subspace of f_1 and f_2 in Figure 5.1(a) shows that the blue and red clusters have a Gaussian distribution, while green and yellow clusters have a uniform distribution in the rectangle area; the subspace of f_3 and f_4 (Figure 5.1(b)) shows that the blue and red clusters have a uniform distribution in the rectangle area, while green and yellow clusters have a Gaussian distribution. Except these four features, all the other 30 features show noisy distribution, such as f_5 and f_6 shown in Figure 5.1(c).

There are two characteristics about this synthetic dataset: 1) it has a certain amount of noisy instances that cannot be neglected (corresponds to challenge 1 in

Section 5.1). 2) The dataset contains more noisy features than useful features (30 $v.s.$ 4, which corresponds to challenge 2 in Section 5.1). These two characteristics exist in many real world datasets, such as microarray or text datasets.

These two characteristics make the popular unsupervised feature selection algorithms to be difficult to handle. In Table 5.1, we reveal the challenges of the other popular feature selection algorithms. We evaluate K-means clustering results on the selected four-feature subspace from a few popular feature selection algorithms (SPEC [161], Laplacian Scores (LS) [62], MCFS [35] and NDFS [89]). From Table 5.1, we can see that if the noisy observations are filtered out, all the baseline algorithms have better performance (although only slightly better for some algorithms), which indicates that the noisy instances lower the performance. Among the four popular feature selection algorithms, NDFS has the most noticeable improvement after filtering out the noisy observations, since it performs a joint and iterative learning between cluster labels and feature selection matrix that optimizes the objective functions [89]. However, NDFS, as well as the other existing algorithms, still suffers a lot from noisy features and observations.

We here design an advanced unsupervised feature selection algorithm which not only reduces noisy instance effects (challenge 1), but also effectively filter out the noisy features (challenge 2).

5.2 Notations and Background

We use $X_{**} \in R^{n \times m}$ to denote a high-dimensional dataset with n instances and m features. The corresponding global similarity matrix $W_{**} \in R^{n \times n}$ can be constructed to represent the relationship among instances considering the whole feature space. Gaussian similarity is one of the most generally used options for constructing W_{**}:

$$W_{ij}^{(GAU)} = exp(- \parallel X_{i*} - X_{j*} \parallel^2 /(2\sigma^2)), \tag{5.1}$$

where σ controls the width of neighborhood [100]. For some datasets with nonuniform sizes such as text datasets we tend to use cosine similarity:

$$W_{ij}^{(COS)} = \frac{X_{i*} \cdot X_{j*}}{\parallel X_{i*} \parallel_2 \cdot \parallel X_{j*} \parallel_2}. \tag{5.2}$$

The degree matrix D_{**} on W_{**} is defined by $D_{ij} = \sum_{k=1}^{n} W_{ik}$ if $i = j$, and 0 otherwise. Given W_{**} and the corresponding D_{**}, the Laplacian matrix $\mathcal{L}_{**}$ and symmetric normalized Laplacian matrix L_{**}^{sym} are defined as:

$$\mathcal{L} = D - W, \tag{5.3}$$

$$L^{sym} = D^{-1/2}\mathcal{L}D^{-1/2}. \tag{5.4}$$

From L_{**}^{sym} we can compute the eigenvectors $Y_{**} \in R^{n\times c}$ ($c \ll m$) which in theory provide the manifold structure of the high-dimensional dataset X_{**} [100]. By carefully setting the value of c, the first c eigenvectors reveal the global distribution pattern of X_{**}. In practice c is usually set as the number of clusters [109].

In 2010, Cai et al. proposed a method called Multi-Cluster Feature Selection (MCFS) [35]. They measured the importance of each feature w.r.t. each column of Y_{**} which corresponds to the contribution of each feature for differentiating clusters [35] by minimizing the following equation:

$$min_{a_{k*}} \parallel Y_{*k} - Xa_{k*} \parallel^2 + \beta \mid a_{k*} \mid, \tag{5.5}$$

where Y_{*k} is the k-th column/eigenvector in Y_{**}, a_{k*} is a $m \times 1$ vector and β is a parameter controls the a_{k*}'s approximation speed to zero. For each feature f_j, they defined the feature importance as:

$$MCFS(f_j) = max_k \mid a_{kj} \mid, \tag{5.6}$$

where a_{kj} is the j-th element of vector a_{k*}.

5.3 Multi-perspective Unsupervised Feature Selection

The notion of correlation is essential since it allows us to discover signals with similar patterns and, consequently for feature selection applications, discover each feature contribution to the global spectrums. In this section we consider the correlation among features and global spectrums, and exhibit two important properties:

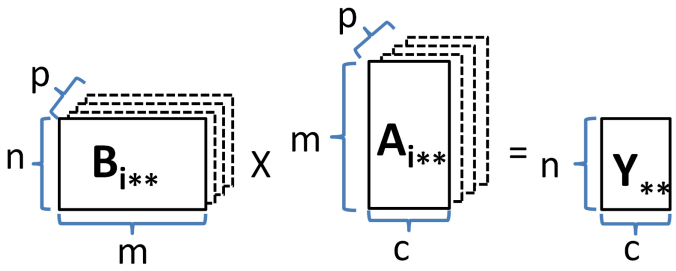


Figure 5.2: Multi-layers of matrix (cube) used in our algorithm. Each layer shows a case of Equation 5.7 with a similarity matrix B_{i**}, coefficient matrix A_{i**} and global spectrums Y_{**}. Equation 5.7 shows how to construct A_{i**} which represents the multi-perspective correlations [1].

1. The effect of each feature may change over different instances or global spectrums. In this case, a single and static score for each feature regardless of different instances and spectrums would be misleading. It is desirable to have a notion of multi-perspective correlation that evolves with each instance, each feature and each global spectrum.

2. The second property is that some informative features w.r.t. certain instance subset exhibit strong but fairly complex, non-linear correlations with global spectrums. Traditional linear measures, such as [35] are less effective in capturing these non-linear relationships. Here we seek a powerful model that can capture such correlations on certain dataset applications.

We introduce a powerful model that can capture multi-perspective correlations inside the high-dimensional dataset. It starts with global spectrum derivation and make the spectrums as regression target. Then the association score is measured by comparing the similarity between each global spectrum and each feature on certain instance (representatives). Higher value of association score means higher possibility that the corresponding feature is an informative feature with respect to the related global spectrum.

[1] In practice we added one column-vector $\mathbf{1} \in R^n$ in B_{i**} which plays a role of intercept.

5.3.1 Constructing Similarity Cube

To learn a model of comprehensive feature weighting, we learn from multiple instance representatives simultaneously, since each instance representative usually only provides "strong feedback" to a subset of features. We will explain how to choose instance representative in Section 5.4. To obtain each instance representative perspective, we acquire the similarity information between the representative and all the other instances within each local feature subspace. In this way, the influence of each feature to the neighborhood of each representative can be revealed.

Specifically, for each instance representative x_i ($i \leq p$, where p denotes the number of instance representatives) and each feature f_j ($j \leq m$), we construct x_i's similarity vector B_{i*j} (to all instances), which is a $1 \times n$ vector based on the q neighboring features of f_j (including f_j). Using f_j's q neighborhood instead of only f_j itself can generate more stable and informative similarity distribution for each x_i. For those applications with a large feature size, we use fast approximate k-nearest neighborhood search [51] to obtain the neighbors of each feature. After we extract q neighbors for each feature, we construct the corresponding similarity matrix (on the instance representatives) within this feature subspace. Therefore for each feature f_j and each instance representative x_i, we obtain a $1 \times n$ similarity vector B_{i*j}. So we have a $p \times n \times m$ three dimensional cube B_{***} shown in Figure 5.2, where p is the number of instance representatives, m is the number of features and n is the number of total instances.

In practice, for those Gaussian distributed dataset we use Gaussian kernel (Equation 5.1) to reveal the non-linear correlation between global spectrums and original features. For text datasets, we use cosine similarity (Equation 5.2) to construct similarity matrix.

Each B_{i**} shows x_i's similarity with all the instances within each local feature subspace. Next subsection explains, by learning the similarity of these local information to the global spectrums Y_{**}, we can measure how much each feature contributes to the global spectrums for each instance representative. The more it contributes, the more important the corresponding feature is.

5.3.2 Learning Coefficient Cube

On the other hand, different instances (representatives) may have very different feature preferences. To qualify these preferences, we here resort to a regression procedure, which typically requires learning from the low-rank model, or global spectrums on instance space, in order to measure the feature contribution across instance representatives to different spectrums.

Intuitively we want to extract the "key information" locally contained in the similarity cube B_{***} and measure how close they are to the global spectrums. This is where the spectral decomposition Y_{**} helps. Here Y_{**} is set as the regression target that consists of the first c global spectrums. These spectrums capture the key aperiodic and oscillatory trends that explain the largest fraction of the data variance. Thus, we only consider the low-rank subspace spanned by the first c global spectrums/eigenvectors. Specifically, we compare the feature impact for each instance representative on this low-rank subspace, and extract the correlation/similarity score.

For each instance representative x_i in Cube B_{***}, there is one $n \times m$ similarity layer B_{i**}, which contains x_i's information related to all n instances and all m features. Given B_{i**}, we propose the following equation to characterize the correlation between each feature and each global spectrum in the perspective of x_i, i.e. A_{i**}:

$$B_{i**} \times A_{i**} = Y_{**}, \qquad i = 1, 2, ..., p. \tag{5.7}$$

Equation 5.7, shown in Figure 5.2, is a simple regression problem. In practice we solve it with the following ridge regression equation:

$$argmin_{A_{i**}} \|B_{i**} \times A_{i**} - Y_{**}\|^2 + \lambda \|A_{i**}\|^2. \tag{5.8}$$

which can be solved by using Moore-Penrose pseudoinverse [31]. A_{i**} is a $m \times c$ matrix which represents the coefficients to reconstruct Y_{**} given B_{i**} [1]. This equation is to find the matrix factorization that has minimal reconstruction error on Y_{**}. Because the layer/perspective is independent to each other, more advanced techniques such as Lasso regression would not be necessary. The advantage of using pseudoinverse here is that it is a relatively simple and non-iterative method, and the weights/coefficients can be solved analytically.

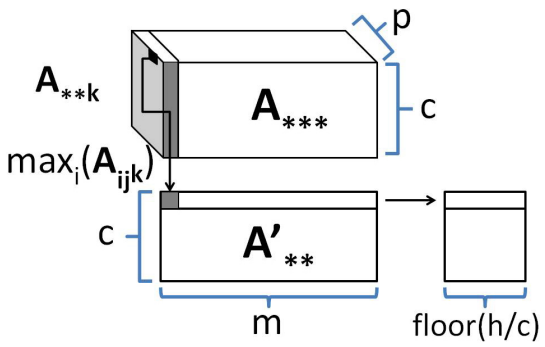


Figure 5.3: The selection of feature subset based on the coefficient cube A_{***} (Section 5.3.3).

The coefficient matrix A_{i**} is of interest because it reflects the correlation between the pattern of the corresponding feature in B_{i**} and the global spectrum Y_{**}. When the value of such coefficients, or interdependence scores are high, the contribution of the corresponding features to the global spectrums are high. These measures can also help us to filter out the noisy features since they tend to have very low correlation with the low-rank embeddings of the whole dataset.

In particular, A_{i*k} provides the correlations of all the features to the global spectrum Y_{*k} with respect to the instance representative x_i. Therefore, for each instance representative x_i, we obtain a $m \times c$ coefficient matrix. The final coefficient cube A_{***} is $p \times m \times c$ (Figure 5.2). The three dimensional cube A_{***} provides a multi-perspective model of different feature weighting across all the instance representatives and global spectrums. Therefore, it provides a comprehensive "platform" for an informative feature selection.

5.3.3 Feature Selection with Coefficient Cube

Based on the coefficient cube A_{***}, we now select feature subset in a more comprehensive way compared with the other existing methods.

1. First of all, we need to make all the coefficient measures have the same sign. The coefficients generated from Equation 5.8 usually have mixed positive and negative values, while the extremes of both sides show a strong correlation. In our algorithm we take the absolute value of coefficient (similar to Equation

5.6). Also since the "localized" feature selection may result in different value ranges of coefficient, each coefficient vector A_{i*k} should thereby be properly normalized. In our implementation, we use L_2-normalization for each A_{i*k}, therefore the above processing could be represented as:

$$A_{ijk} = |A_{ijk}| / \sqrt{(\sum_{g} |A_{igk}|^2)}, \tag{5.9}$$

Now the higher the coefficient value is, the more important the feature is to the corresponding pair of instance representative and global spectrum.

2. We then select the feature subset based on the normalized A_{***}. To preserve the global spectrums with a small amount of observed features, we select representative features from the perspective of each global spectrum. Suppose we need to select no more than h features (usually $h > c$), then $\lfloor h/c \rfloor$ features are chosen for each global spectrum, where c is the number of global spectrums.

   In the coefficient cube A_{***}, each global spectrum Y_{*k} corresponds to a $p \times m$ matrix A_{**k}. The first dimension p correlates with the number of instance representatives, while the second dimension m corresponds to the number of original features. To study how much a global spectrum values each feature, we need to compress this $p \times m$ matrix A_{**k} into a $1 \times m$ vector A'_{*k}, in which each value A'_{jk} is the weight of feature f_j w.r.t. the corresponding global spectrum Y_{*k}. As shown in Figure 5.3, we choose the maximum along all the instance representatives:

$$A'_{jk} = max_i\{A_{ijk}\}. \tag{5.10}$$

   Now we have a $m \times c$ correlation matrix A'_{**} which shows the relation of features and global spectrums.

3. For each global spectrum we select $\lfloor h/c \rfloor$ features. Every time when we select w.r.t. A'_{*k}, we choose the $\lfloor h/c \rfloor$ features with the highest coefficient value. And set the elements in the same positions but on the unprocessed columns as 0, in order to avoid duplicate features. Finally we successfully choose $\lfloor h/c \rfloor \times c$ features out of the original feature space.

5.4 Noise-Resistant and Density-Preserving Sampling

This section introduces how to select instance representatives by our proposed noise-resistant density-preserving sampling. It consists of two components: outlier removal and density-preserving sampling to fulfill the needs of our proposed feature selection algorithm.

5.4.1 Noisy Observation Removal

The first step is to remove noisy observations. Here we assume noisy observations are those instances with small neighborhood density, which also called outliers or anomalies. We resort to anomaly detection algorithms [9] [69] [70], which distinguish normal instances from a small portion of abnormal instances (noisy observations). Particularly we apply FDD (Fermi Density Descriptor) [70] due to its effectiveness and stability. It measures the average probability of a fermion appearing at a specific location (corresponds to each instance in high-dimensional coordinates) in the "polarized" manifold space. The computed probability provides the value of anomalousness for each instance. By choosing the stable energy distribution function, FDD steadily distinguishes anomalies from normal instances. In our algorithm, we sort all instances in the descending order of their anomalousness, and remove the first 10% instances. We assume that the majority of the noisy observations are removed after we apply this approach.

5.4.2 Density-Preserving Sampling

The second step is down-sampling. Many sampling methods have been proposed [80] [45]. But most of them are stochastic and their sampling results vary significantly from one repetition to another. There is no guarantee that the sample results are inclusively representing the original dataset [15]. In this work, we adopt a more intelligent sampling approach aiming to produce representative splits with minimum duplications. We use the newly appeared density-preserving sampling (DPS) [15] to eliminate the need for repeating an error estimation procedure by dividing available data into subsets that are guaranteed to represent the input data.

The idea of DPS is inspired by the concept of correntropy which is a nonparametric similarity measurement between two random variables. Since correntropy can be used to measure similarity, it can also be used to measure the quality of a sample to preserve the representative of the whole dataset [15]. DPS uses correntropy as an optimization criterion, guiding the sampling process to split a given dataset into two or more maximally representative subsets. In their paper, Budka et al. proposed correntropy-inspired similarity index (CiSI) between two random variables (datasets) X and Y:

$$\begin{aligned} CiSI(X,Y) \approx \frac{1}{n} \sum_{i \in (1 \ldots n)} G(x_i - y_j, 2\sigma^2 I), \\ i,j = argmin_{i,j} \|x_i - y_j\|, j \in J_{avail}, \end{aligned} \tag{5.11}$$

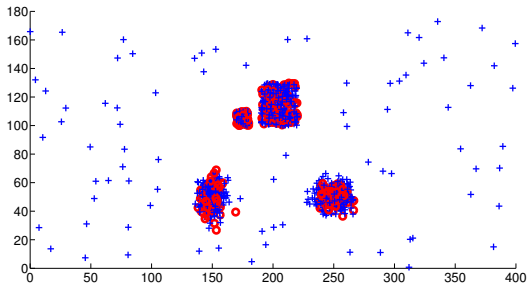
where $G(x_i - y_j, 2\sigma^2 I)$ denotes a Gaussian kernel centered at $(x_i - y_j)$ to avoid the ordering effect, $\sigma^2 I$ is a diagonal covariance matrix of the Gaussian kernel, $\|\cdot\|$ denotes the Euclidean norm, and the set J_{avail} contains the indices of y which have not yet been used, and it ensures that each y_k is used only once. Since a Gaussian kernel peaks at the 0 Euclidean distance regardless the value of σ, CiSI provides a σ-independent iterative binary procedure to split dataset into subsets X and Y. It selects instances z_i and z_j from dataset Z at each step such that the following equation holds:

$$i,j = argmin_{i,j} \|z_i - z_j\|. \tag{5.12}$$

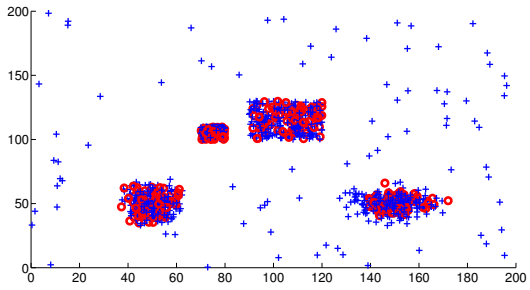
Subsequently, z_i and z_j are added into X and Y to maximize CiSI(X,Y). The procedure can be iteratively applied to split X or Y furthermore to get a small enough sample size.

Note that the density-preserving sampling is not guaranteed to remove noisy observations/instances. We have to combine both noisy observation detection and density-preserving sampling to obtain the final informative instance representatives.

The main property of the above sampling strategy is to produce only representatives while excluding noisy observations. The down-sampling also reduces the running time complexity as shown in Section 5.6.3. In Figure 5.4, we show the effect of our sampling strategy with a 25% sample size (means $p = 0.9n/4$ after removing $0.1n$ noisy observations), which demonstrates that the proposed sampling



(a) Sampling result on f_1 and f_2



(b) Sampling result on f_3 and f_4

Figure 5.4: Sampling result of synthetic dataset in Figure 5.1. Instances marked with red circles are one of the 25% sampling subsets after noisy instance removal.

strategy is not only noise-resistant, but also selects representatives with density-preserving.

It is worth noting that given proper normalization, the above sampling strategy can be also applied on text datasets.

5.5 Noise-Resistant Feature Selection and Theoretical Connections

5.5.1 Noise-Resistant Feature Selection

In this section, we propose the integrated framework that documents the whole process of NRFS. Let X_{**} be the dataset matrix of size $n \times m$ where n is the number of instances and m is the number of features. Algorithm 7 describes NRFS step by step.

Through Step $1 - 3$, we obtain the global spectrum Y_{**} (Section 5.2) as our

ALGORITHM 7: NRFS(X_{**}, h, σ (if use Gaussian kernel), p, q)

Input: Input data $X_{**} \in R^{n \times m}$; h is the #selected features; σ is the Gaussian scaling parameter; p is the #instance representatives; q is the size of local feature subspaces.

Output: Selected feature subset.

1 Construct similarity matrix W_{**} using Equation 5.2, or Equation 5.1 with σ (Section 5.2);
2 Construct symmetric normalized Laplacian matrix L_{**}^{sym} using Equation 5.3 and 5.4 (Section 5.2);
3 Compute generalized eigenvectors Y_{**} (Section 5.2);
4 Remove noisy observations using anomaly detection algorithm (Section 5.4.1);
5 Down sample the remaining dataset to p instance representatives (Section 5.4.2) ;
6 Construct cube B_{***} for each sample instance and each local feature subspace with q (Section 5.3.1);
7 Learn the coefficient cube A_{***} (Section 5.3.2);
8 Obtain the final feature subset (Section 5.3.3)

later regression target. We simply use all instances (including normal and noisy observations) to construct Y_{**}, in that we need to stably rebuild the low embeddings. However it is both sensitive and useless to detect the local correlation w.r.t. noisy instances between features and global spectrums. We thereby remove noisy observations and only focus on the informative representatives, by applying Step 4 and 5 which constitute the Noise-Resistant Density-Preserving Sampling (Section 5.4). On the other hand, noisy features can be filtered out based on their values of the coefficients in Step $6-8$ (Section 5.3). Here the noisy features are coincident with the low correlation values between the global spectrums and local perspective of the instance representatives.

Regarding computational complexity, NRFS is dominated by the eigendecomposition (that gives Y_{**}) which takes $O(n^3)$ and pseudoinverse in Equation 5.8 that takes $O(p(mn^2 + n^3))$. However, the pseudo inverse can be done parallelly for different representative instance layer.

We run NRFS 30 times on the synthetic dataset in Figure 5.1 with $p = 288$ and $q = 1$. Each time the four selected features are always f_1, f_3, f_2, f_4 which generate

the highest K-means clustering result $NMI = 1$.

5.5.2 Connections with Other Techniques

Our proposed NRFS has close connection with recommendation techniques, of which one popular approach for characterizing the multi-user personalization problem is collaborative modeling [81] [155]. In collaborative modeling, users provide feedback on an absolute scale and the model integrates these feedback and obtain final results. Most of these approaches are motivated by the intuition that even though users have different preferences, many users share preference with other users. Therefore the integrated result can be stable and informative. Similarly, our NRFS treats instance representatives and global spectrums as two different kinds of "users". Each of them has its own perspective (feedback) of feature importance. The coefficient cube A_{***} of our NRFS (Section 5.3.2) reveals the two different perspectives to each feature.

On the other hand, different from the target of collaborative modeling, our NRFS tries to locally weight features with multi-perspective correlations. This step is closely related to matrix factorization [82] and fuzzy feature weighting [145] [73]. Our proposed NRFS learns from a low-dimensional latent model Y_{**} which reliably characterize the space of the "user's" dominative yet diverse preferences. It computes a factorization that has a minimal reconstruction error on the latent-variable matrix Y_{**}. Finally, instead of assigning a global importance for each feature, NRFS weights feature according to different perspectives, namely, different global spectrums Y_{*k}. Therefore it is a more comprehensive strategy compared with the other feature selection algorithms.

5.6 Experimental Analysis

5.6.1 Experimental Setup

Datasets and Preprocessing. To demonstrate the performance of our proposed method, we evaluate our algorithm on four microarray datasets and four text datasets (statistics are summarized in Table 5.2).

Table 5.2: Statistics of experimental datasets.

	Dataset	#instances	#features	#clusters
1	11Tumors	174	12534	11
2	Leukemia2	72	11225	3
3	BrainTumor2	50	10368	4
4	Lung	181	12533	3
5	RCV1-4Classes	1200	11370	4
6	Reuter21578A	1000	18933	5
7	20NewsgroupA	800	11269	4
8	20NewsgroupB	800	11217	4

The microarray datasets were mainly produced by oligonucleotide based technology [130]. We took the advantage of all available information in order to increase the number of categories or diagnoses for outcome variable, as described in [130]. In summary, the ten microarray datasets have 3-11 distinct diagnostic categories, 50-181 patients (instances) and about $10,000$-$13,000$ genes (features). In the preprocessing phase, we relied on the following three commonly used steps: 1) $base$-10 logarithm [26], 2) standard quantile normalization [13] over multiple chips, and 3) double centering [26] for background correction.

All the four text datasets we used came from large and popularly used datasets: 20Newsgroups, Reuters21578 and RCV1. The original 20Newsgroups has $18,846$ documents (instances) and $26,214$ words (features). 20NewsgroupA has 800 documents, namely 200 documents from four categories: alt. atheism, comp. graphics, rec. autos, and sci. med. 20NewsgroupB has 800 documents and four categories: comp. windows, rec. motorcycles, sci. space, and talk. religion. misc, and each of them takes 200 documents. Note that there is no repetitive category in the above two datasets. The origin Reuters21578 has $8,293$ documents and $18,933$ words. We select 200 documents from each of the first five clusters. The origin RCV1 is a dataset contains $810,000$ documents. In order to obtain a smaller dataset, we choose samples from only four categories: "C15", "ECAT", "GCAT" and "MCAT", with 300 documents from each category. Our text data preprocessing steps include 1) removing stop words; 2) applying stemming to the remaining words; 3) applying tf-idf transformation; 4) applying the l_2-norm normalization on document; 5) applying bi-normalization to the data matrix as in [37].

Baselines and Evaluation Metric. We choose four state-of-the-art competitors to show the outperformance of our proposed NRFS: Laplacian Score (LS) [62]; Spectral Feature selection (SPEC) [161]; Multi-Cluster Feature Selection (MCFS) [35]; and Nonnegative Discriminative Feature Selection (NDFS) [89].

It would be the best to evaluate feature selection results based on ground truth of feature importance value. However, in real world application, we cannot easily find such ground truth because: 1) it is highly subjective to select candidate features because there are many similar features/terms, and 2) feature selection is an intermediate step for the rest of data analysis pipeline. However, even though we don't have the ground truth for feature importance, we do have the ground truth of cluster labels to indirectly evaluate the quality of feature selection, by comparing clustering performance of the feature-reduced dataset.

In our experiment, we evaluate the feature selection algorithms by performing K-means clustering on the selected feature space. To give a more general perspective, we also test K-means clustering (WCSS [60]) without any feature selection. Normalized Mutual Information (NMI) is used as our only evaluation metric among all being described because most of clustering algorithm papers make use of NMI as their primary evaluation metric. The detailed definition of NMI can be found in [132].

Parameters. The number of selected features are set as $\{$ 200, 300, 500, 800, 1000, 1200, 1500, 1800 $\}$. For the similarity function used in the microarray dataset experiments, we use Gaussian similarity (Equation 5.1). We need to construct similarity matrices with both local feature subspace and the whole feature space. Here we adopt an adaptive width of neighborhood σ for each local feature subspace, instead of a fixed value. In our implementation, we assign σ to be the average Euclidean distance of each instance to its K' nearest neighbor, where K' is the average size of clusters ($K' = round(n/c)$). For text datasets, cosine similarity (Equation 5.2) is a reasonable choice to compare texts with different sizes. For all the kNN based similarity methods $k = 5$, where k specifies the size of neighborhood. The number of eigenvectors c is set as the number of instance clusters, which assume to be already known [109] [35].

Especially, for MCFS, we keep $min\{M, n\}$ non-zero entries in each eigenvector when trying to select M features. For NDFS, we set $\alpha = 1e-006$, $\beta = 1e-006$ and $\gamma = 10^8$. We follow the suggestions in [35] [89] to set default values for these parameters.

Our proposed algorithm NRFS has two specific parameters: the sampling rate p and the number of neighbors q for each feature. We set p according to DPS [15] with $level = 2$ and pick one out of four sampling subsets), and $q = 50$ which is appropriate for maintaining stable performance and alleviating noise effects adaptively. We also test the performance stability of NRFS across different size of feature subspace q later.

To guarantee a fair comparison, for each size of feature subsets, we run every algorithm 30 times and record the average NMI in Figure 5.5. Whenever we get the reduced feature subspace, we apply the K-means clustering (the version with minimizing within-cluster sum of square (WCSS) [60]), with 100 inner loops and 100 outer loops.

5.6.2 Overall Algorithm Performance Analysis

Figure 5.5 documents the performance of a few feature selection algorithms, including our proposed NRFS and K-means clustering on the whole feature space. The experiments are measured by NMI derived from the K-means clustering on feature subspaces generated by by the feature selection algorithms. The experimental results offer the following observations:

1. Generally speaking, NRFS results on text datasets showed an "improving" trend as the feature size increases, i.e. NRFS started with a suboptimal performance for text datasets when the size of feature subset is small (eg. 200, 300), and surpassed the other algorithms when the size increases. The reason is that the number of informative features/words in text datasets is usually much higher (e.g. hundreds) than those in microarray datasets (e.g. dozens). For microarray dataset, a small number of informative features contain sufficient information to achieve a good clustering quality. However, for text datasets, if a feature subset is too small, it cannot provide enough descriptive capability to differentiate different document categories.

2. Second, feature selection algorithms help to obtain a refined description of the feature space. Compared with the K-means clustering on the whole feature space, most of the five feature selection algorithms have better performance in their reduced feature space. In particular, our proposed NRFS, has more than 25% for the microarray datasets and $180\% \sim 200\%$ improvement for the text datasets in average.

3. Our proposed NRFS outperforms not only the similarity-based methods such as LS and SPEC, but also regression-based methods such as MCFS and NDFS in terms of average NMI. Moreover, NRFS shows more stable performance as the number of feature change. Our NRFS outperforms MCFS, the second best algorithm, by a margin of more than 10% for microarray datasets and 25% for text datasets in average. It confirms that our proposed NRFS algorithm is capable to find better representative feature subsets by detecting and taking advantage of multi-perspective correlation.

4. MCFS [35] and NDFS [89], to some extent, are capable to exploit discriminative information among different features, which result in more accurate result than LS [62] and SPEC [161].

We conduct experiments with controlled size of feature neighborhood q to examine the NRFS's stability. The datasets used in the experiments are 11Tumors, BrainTumor2, Reuter21578A and 20NewsgroupB with $q = [30, 50, 80, 100]$. As shown in Figure 5.6, our proposed NRFS consistently shows a robust performance across different q.

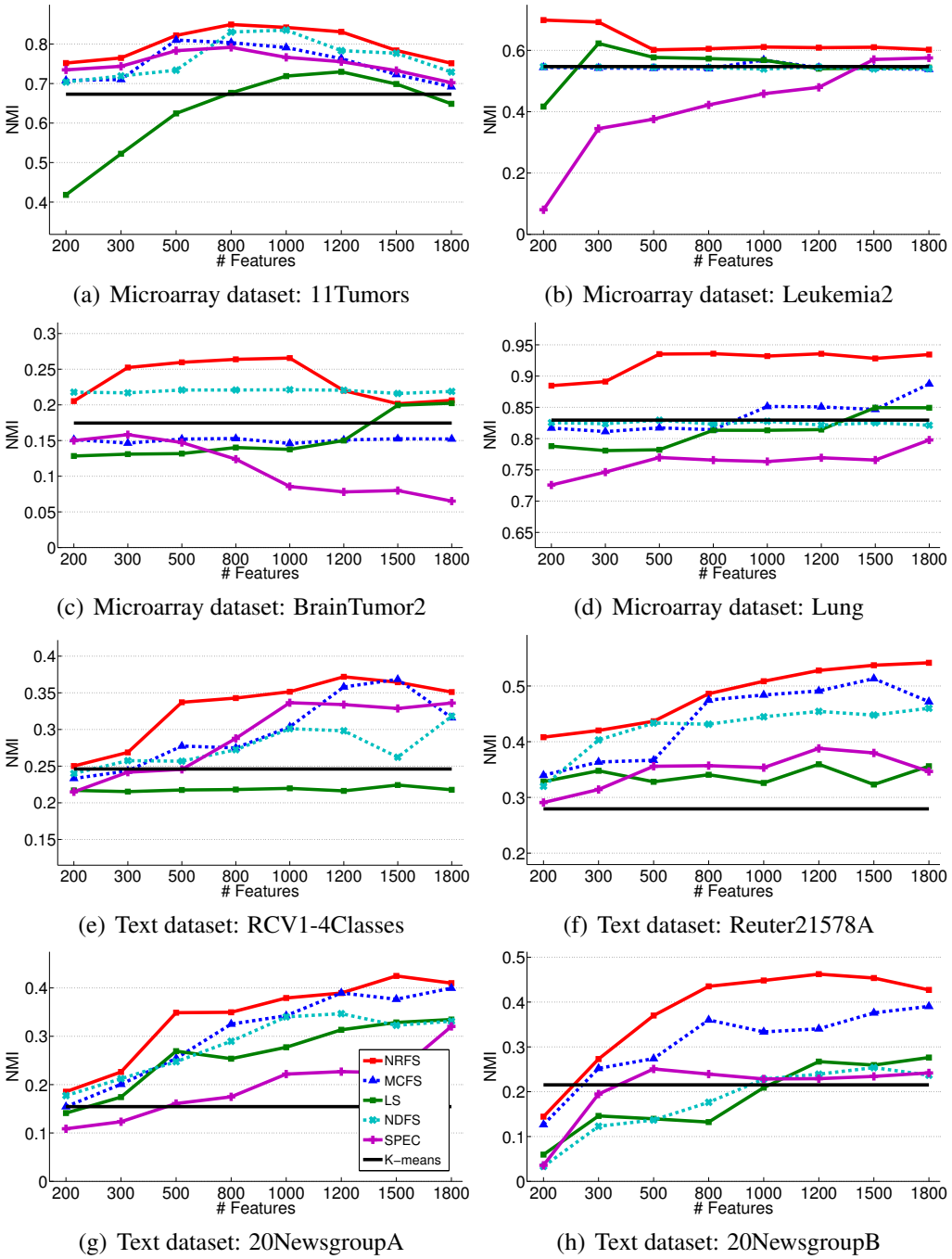
5.6.3 Comparison of Time Complexity

Figure 5.7 shows the comparison results of time complexity among the six algorithms including two versions of NRFS: NRFS-1 is with noise-resistant and density-preserving sampling, while NRFS-2 uses the full instance space without representative selection. With the help of our sampling strategy, NRFS-1 is 57% faster than NRFS-2. Moreover, NRFS-1 has comparable running time with MCFS, but it is more than 2.5 times faster than NDFS. Although SPEC and LS are more

efficient, their effectiveness shown in Figure 5.5 is actually much worse than our proposed NRFS.

5.7 Chapter Summary

In this chapter we propose an unsupervised feature selection algorithm called Noise-Resistant Feature Selection (NRFS). It has two main advantages: firstly, NRFS is a collaborative feature selection algorithm based on multi-perspective correlation, in that it probes the feature effect via local perspective from instance representatives and global spectrums, and thereby effectively distinguishes diverse and informative features from the remaining ones. Secondly, NRFS applies noise-resistant and density-preserving sampling to improve its efficiency while reducing the negative affect incurred by noisy instances. Compared with existing algorithms, our proposed NRFS demonstrates much more stable and better performance in the experiments on microarray and text datasets.



(a) Microarray dataset: 11Tumors

(b) Microarray dataset: Leukemia2

(c) Microarray dataset: BrainTumor2

(d) Microarray dataset: Lung

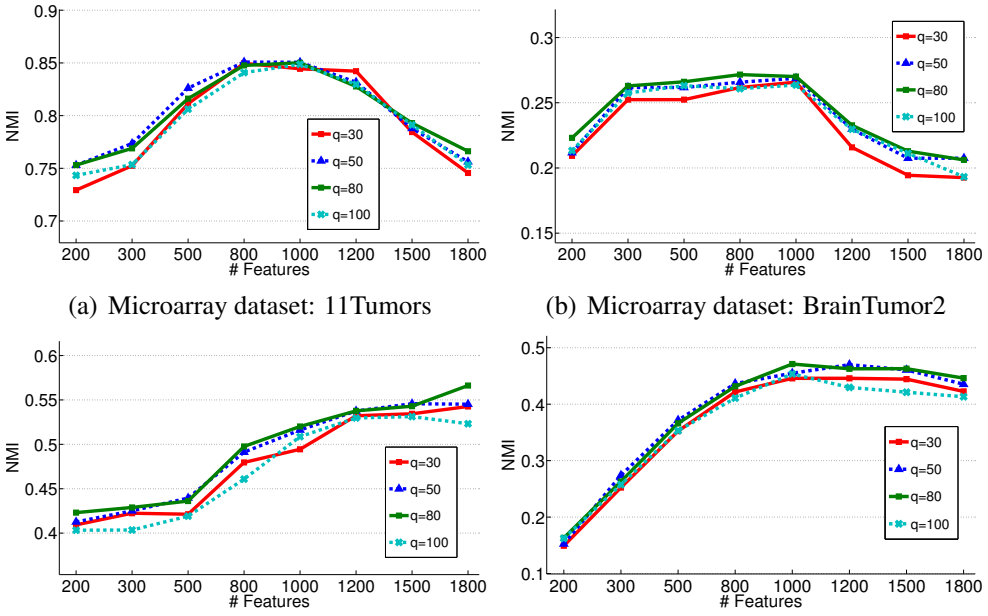
(e) Text dataset: RCV1-4Classes

(f) Text dataset: Reuter21578A

(g) Text dataset: 20NewsgroupA

(h) Text dataset: 20NewsgroupB

Figure 5.5: Comparison of feature selection performance. Results are evaluate by K-means clustering on the selected feature subset using NMI score. It shows that our proposed NRFS (in red) outperforms the other competitors.



(a) Microarray dataset: 11Tumors

(b) Microarray dataset: BrainTumor2

(c) Text dataset: Reuter21578A

(d) Text dataset: 20NewsgroupB

Figure 5.6: Performance stability of NRFS across different size of feature neighborhood q.

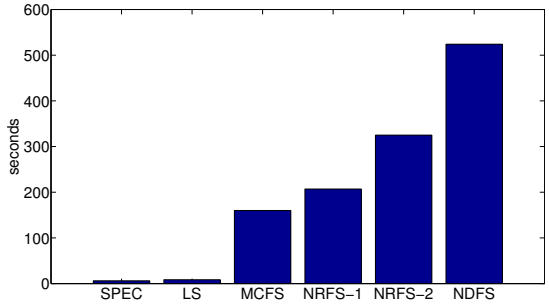


Figure 5.7: Comparison of time complexity. NRFS-1 is NRFS with our sampling strategy, while NRFS-2 is NRFS without any sampling.

Chapter 6

Diverse Power Iteration Embeddings and Its Applications

6.1 Chapter Introduction

Spectral Embedding is one of the methods to calculate low dimensional embeddings. It was used in clustering [109] [71] at first but later applied to many other data mining applications such as anomaly detection [69] [70] and feature selection [35]. Spectral Embedding uses a spectral decomposition of the graph Laplacian[100]. The generated graph can be considered as a discrete approximation of the low dimensional manifold embedded in the original high-dimensional data space. Minimizing a cost function based on the graph ensures neighboring data points that are close to each other on the manifold to be still mapped to neighboring ones in the low dimensional space, i.e. preserving local distances/neighborhood.

Although Spectral Embedding gained an increasing popularity in recent years, its associated high complexity in both time $O(n^3)$ and space $O(n^2)$ prevents it from practical utilization in many real-world applications. For instance, we cannot do spectral clustering directly on popular RCV1 benchmark dataset due to its large data size of nearly $200,000$ documents. Given a dataset with n data points, spectral methods create an $n \times n$ affinity matrix and apply eigen-decomposition on the subsequent Laplacian normalized matrix with the time complexity of $O(n^3)$ in general.

To overcome these limitations, several methods are proposed such as [92]

[137] [90]. Among them, Power Iteration Clustering (PIC) [92] is one of the most promising candidates due to its speed, small memory requirements and yet effectiveness in obtaining clustering results for datasets with small number of clusters. However, PIC cannot handle well those datasets with a large number of clusters, even with the new PIC-k (with k power iteration vectors) method [91]. In addition, it is also an impediment to apply this type of power iteration embedding in many other data mining applications, such as feature selection and anomaly detection.

This chapter proposes Diverse Power Iteration Embeddings (DPIE) which overcomes the limitations of PIC/PIC-k and applies it in a broad scope of spectral analysis. Moreover, it requires a far less amount of time and space, which is similar to PIC-k. Our contributions in DPIE are as follows:

1. We proposed a novel power-iteration-based method that aims to find diverse and yet informative low dimensional embeddings, which is different from the single or similar embedding vectors from previous PIC methods.

2. In theory, our proposed DPIE has the same or similar representational power of low dimensional projection with classic spectral embeddings, so that it can be applicable to various spectral analysis.

3. Our proposed DPIE, compared with the existing spectral embedding approximations, achieves a similar or even lower time and space computational complexity, but a more desired quality.

4. We systematically evaluated DPIE along with several closely-related algorithms on a number of important applications. The results confirmed that our new algorithm significantly outperformed other existing algorithms in terms of effectiveness and efficiency.

6.2 Spectral Embeddings Construction

Spectral embedding construction already gained its popularity in the last decade because of its ability to reveal embedded data structure. It has a strong connection with a graph cut, i.e., it uses eigenspace to solve a relaxed form of a normalized graph partitioning problem [109]. Its second desirable aspect is that it

can capture the nonlinear structure of data with the help of nonlinear kernel, which is difficult for k-means or other linear clustering algorithms.

ALGORITHM 8: SpectralEmbeddingConstruction(X, c)

Input: $X \in R^{n \times m}$ where n is #instances and m is #features, and c is #low-dimensions.

Output: Spectral embeddings $Y \in R^{n \times c}$.

1 Construct the affinity matrix $W \in R^{n \times n}$ of X;

2 Compute the diagonal matrix $D \in R^{n \times n}$ where $D(i,i) = \sum_{j=1}^{n} W(i,j)$ and $D(i,j) = 0$ if $i \neq j$;

3 Construct a graph Laplacian L using $L_{nn} = D - W$, $L_{rw} = I - D^{-1}W$ or $L_{sym} = I - D^{-1/2}WD^{-1/2}$;

4 Extract the first c nontrivial eigenvectors Ψ of L, $\Psi = \{\psi_1, \psi_2, \ldots, \psi_c\}$;

5 Re-normalize the rows of $\Psi \in R^{n \times c}$ into $Y_i(j) = \psi_i(j)/(\sum_l \psi_i(l)^2)^{1/2}$;

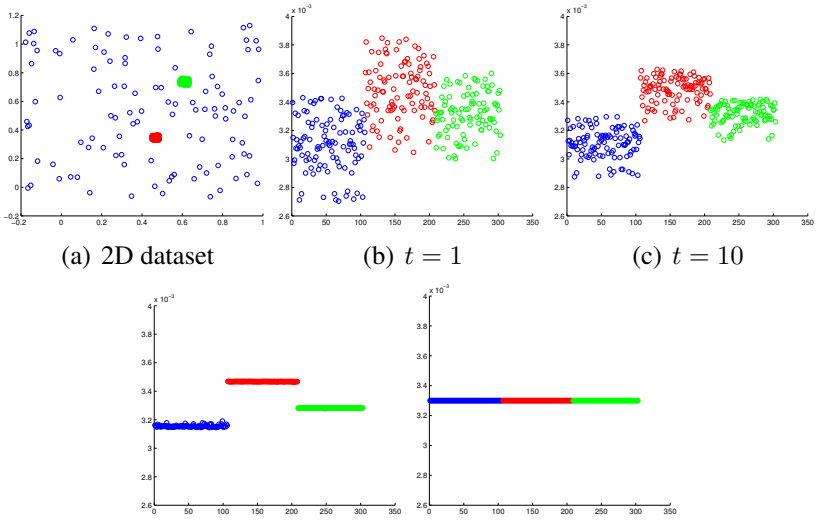
Spectral embedding construction as shown in Algorithm 8, starts with local information encoded in a weighted graph that is constructed from input data with a certain similarity kernel, and selects embedding vectors from the global eigenvectors of the corresponding (normalized) affinity matrix.

Although it demonstrated its effectiveness in clustering [109], feature selection [35], and anomaly detection [69], it is infeasible for large-scale data analysis due to its time and space complexities. The space requirement for constructing affinity matrix (Step 1) is $O(n^2)$, and the computing time for eigen-decomposition in Step 4 is $O(n^3)$. A mechanism is needed to approximate Algorithm 8 with less time and space requirements while retaining similar effectiveness.

6.3 Power Iteration Embeddings and Its Limitations

6.3.1 Power Iteration Embeddings

To address the complexity of classic spectral embedding construction, Lin et.al [92] proposed power iteration clustering (PIC), which finds a one dimensional data embedding using truncated power iteration on a Laplacian normalized affinity matrix. PIC is based on a simple iterative method called power iteration, which we will briefly introduce here.



(a) 2D dataset (b) $t = 1$ (c) $t = 10$

(d) $t = 20$ (e) $t = 200$

Figure 6.1: Single power iteration embedding (the embedding v_*^t provided by [92] or Equation 6.3) for 2D dataset in Figure 6.1(a) with three clusters, of which each cluster is represented with a different color. In Figure 6.1(b), 6.1(c), 6.1(d) and 6.1(e), the value of each component of v_*^t is plotted against its index. We can see that although v_*^t eventually converges to a uniform vector (Figure 6.1(e) when $t = 200$), the intermediate vectors (eg. v_*^t when $t = 20$) reveal the manifold embedding of the dataset. This example shows that PIE could be an efficient alternative to eigenvectors from traditional eigen-decomposition.

ALGORITHM 9: PowerIterationEmbedding(X)

Input: $X \in R^{n \times m}$ where n is #instances and m is #features.
Output: Power iteration embedding $v^t \in R^{n \times 1}$.

Construct the affinity matrix $W \in R^{n \times n}$ of X;
Perform positive random normalization $W \leftarrow D^{-1}W$;
Initialize $v^0 \in R^{n \times 1}$;
Repeat
    $v^{t+1} \leftarrow \frac{Wv^t}{\|Wv^t\|_1}$;
    $\delta^{t+1} \leftarrow |v^{t+1} - v^t|$;
    $t \leftarrow t + 1$;
until $\| \delta^t - \delta^{t+1} \|_{max} \simeq 0$;

According to [100], the c smallest eigenvectors of graph Laplacian L_{rw} happen to be the c largest eigenvectors of random walk normalized affinity matrix $W_{rw} =$

$D^{-1}W$. For our notational convenience, we will use W for W_{rw} in the rest of this chapter. Let $W \in R^{n \times n}$ and recall that if ψ is an eigenvector for W with eigenvalue λ, then $W\psi = \lambda\psi$. Therefore in general, there is $W^t\psi = \lambda^t\psi$ for any t. This observation is the very foundation of the power iteration method.

Suppose $\Psi = \{\psi_1, \psi_2, \ldots, \psi_n\}$, the set of unit eigenvectors of W, forms a basis in $R^{n \times n}$, and has corresponding real eigenvalues $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$. We assume that the first c eigenvectors carry informative signals and the rest eigenvectors are noise [100]. From the spectral theorem, for the properly normalized affinity matrix W such as random walk normalization, there are eigenvalues as follows:

$$1 = \lambda_1 > \lambda_2 > \ldots > \lambda_c \gg \lambda_{c+1} > \ldots > \lambda_n. \tag{6.1}$$

Note that power iteration embeddings assume 1) there is at least a large enough eigen-gap between c and $c+1$ and 2) $\lambda_2 \sim \lambda_3 \sim \ldots \sim \lambda_c$. Now let $v^{(0)} \in R^n$ be a randomly generated vector, since Ψ is a basis of $R^{n \times n}$, we have:

$$v^{(0)} = a_1\psi_1 + a_2\psi_2 + \ldots + a_n\psi_n, \tag{6.2}$$

where a_i is the weight of i-th eigenvector. Then, the power iteration will be:

$$\begin{aligned} v^t = W^t v^{(0)} &= a_1\lambda_1^t\psi_1 + a_2\lambda_2^t\psi_2 + \ldots + a_n\lambda_n^t\psi_n \\ &= a_1\psi_1 + \lambda_2^t\left(\sum_{i=2}^{n} a_i(\frac{\lambda_i}{\lambda_2})^t\psi_2\right). \end{aligned} \tag{6.3}$$

The power iteration will finally converge to $a_1\psi_1$ which is useless because it is a constant vector. However, if the number of iteration t is cleverly set from being too large as shown in [92], $W^t v^{(0)}$ is a linear combination of the first c informative eigenvectors, while all the other eigenvectors are gone away due to the eigen-gap. In other word, the whole process should be controlled very well in order to remove the terms of $\psi_{c+1} \ldots \psi_n$ with diminishing rate $(\frac{\lambda_{c+1}}{\lambda_2})^t$, but still keep the rate of $(\frac{\lambda_c}{\lambda_2})^t$ big enough. Fortunately, if the power iteration reaches the eigen-gap, then the convergence rate will be relatively slow because the similar values from λ_2 to λ_c. PIC defines the velocity at t as $\delta^t = |v^t - v^{t-1}|$ and acceleration at t as $\varepsilon = ||\delta^t - \delta^{t-1}||_{max}$ as a measure of the convergence rate and stop power iterations if ε is very small to do early stopping. Figure 6.1 shows the effect of different number of power iterations and $t = 20$ shows a pretty good clustering embedding. Lin and

Cohen [92] proposed the described procedure as Power Iteration Embedding (PIE) algorithm, also shown in Algorithm 9.

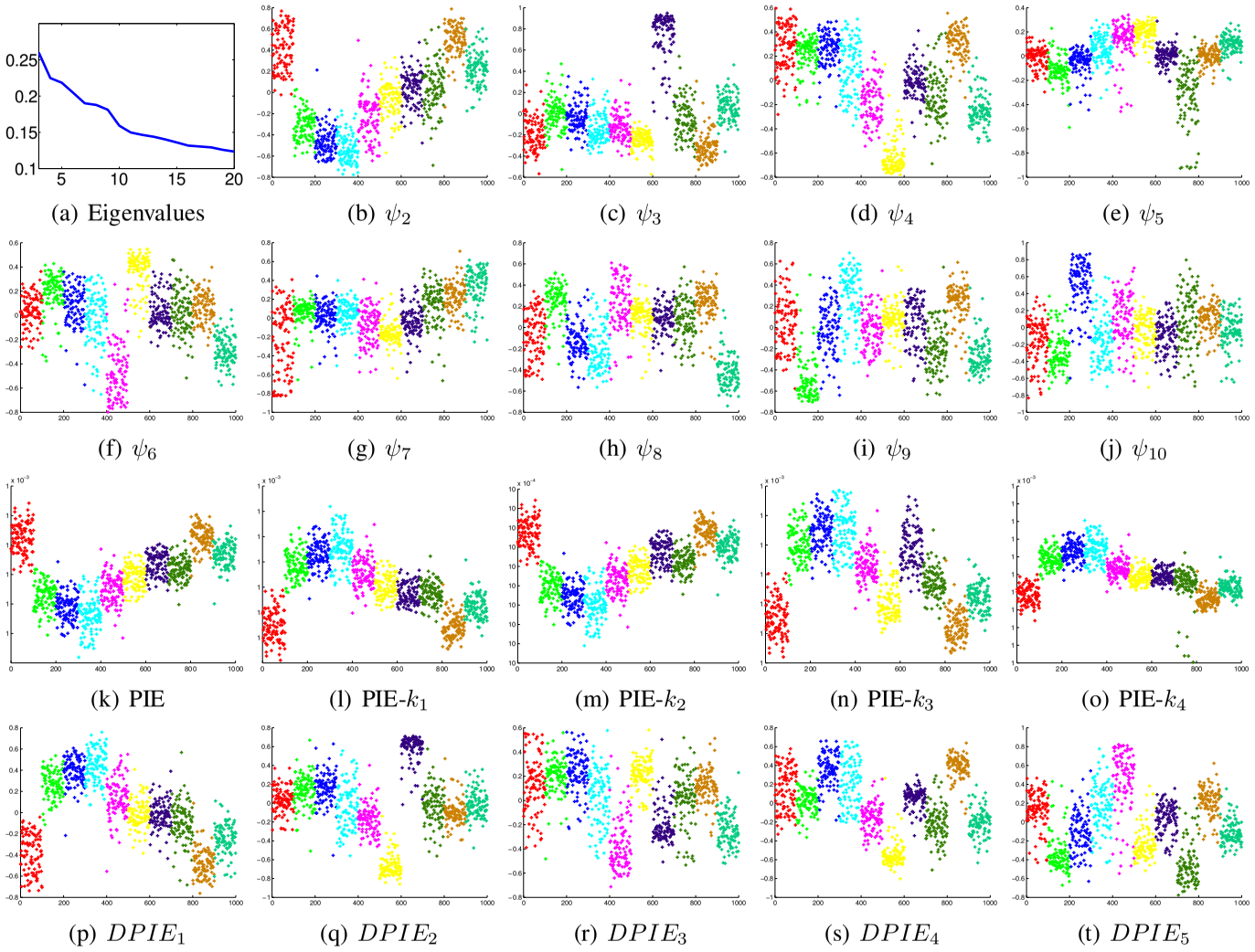


Figure 6.2: Different low dimensional embeddings of 20NG-10 dataset, which consists of 10 cluster subsets from 20Newsgroups dataset (Section 6.7.1). Eigenvectors ψ (Figure 6.2(b) to 6.2(j)) are sorted by eigenvalues in descending order (Figure 6.2(a)). PIE (Figure 6.2(k)) and PIE-k (Figure 6.2(l) to 6.2(o)) are quite similar to ψ_2 in Figure 6.2(b). Relatively DPIEs (Figure 6.2(p) to 6.2(t)) reveal more diverse yet informative signals than PIE and PIE-k.

6.3.2 The Limitations of PIE

Although it showed a pretty good embedding in Figure 6.1, it is not good enough to handle large c clusters or different spectral applications. If the dataset has a relatively large number of clusters, it is quite difficult to discriminate clusters with a single PIE. The obvious reason is that if c is sufficiently large, the number of required eigenvectors increases. But in PIE, the first few (or even one) nontrivial eigenvectors dominate the whole vector. For instance, Figure 6.2 showed ten selected clusters from 20Newsgroups (see Section 6.7.1) violates two PIE assumptions; the biggest eigen-gap is between λ_2 and λ_3 and the second biggest is between λ_3 and λ_4, which also violates similar eigenvalues before c eigenvectors. So, the PIE is quite similar to ψ_2, which is not good enough to distinguish the ten clusters. But the ten eigenvectors together reveal more information such as the blue cluster from ψ_3, the pink cluster from ψ_6, etc.

Different random starting vectors v^0 may reveal different degrees of impact on top c eigenvectors due to different a_i in Equation 6.2. Suppose ψ_k ($k > 2$ and $\lambda_2 > \lambda_k$) is a very informative eigenvector and there happens to be $a_k \gg a_2$. By attentively controlling the number of iteration we may have $a_2\lambda_2^t \simeq a_k\lambda_k^t \gg a_{k+1}\lambda_{k+1}^t$, which means that v^t holds essential information from ψ_k without concealing by the first few ψ_i. So by increasing the number of initial vectors to generate multiple PIE or PIE-k ($k = \lceil log(c) \rceil$ according to [91]), the quality of the generated embedding vectors has potential to improve to a certain degree. For instance, the PIE-k of Figure 6.2 share the similar general trends with the second eigenvector but it reveals slightly different distributions.

But there is still a crucial and unsolved problem: the first few eigenvectors still overshadow the other less important but indispensable eigenvectors. Under this circumstance, these first few eigenvectors are still dominant in the result vector v^t. We can easily see this from Equation 6.3 as well : each v_k^t is still dominated by the first few $\psi_1, \psi_2, \ldots$ because of $\lambda_1^t \gg \lambda_2^t \gg \ldots \gg \lambda_n^t$. Therefore, for large c clustering problems or the other spectral applications such as spectral feature selection or anomaly detection, PIE and PIE-k are not practical, which we can also verify in Section 6.7.

6.4 Diverse Power Iteration Embeddings

As analyzed in the last session, the fundamental problem in PIE/PIE-k is the essential influences by the first few eigenvectors in each converged embedding vector. To deal with this problem, we propose Diverse Power Iteration Embeddings (DPIE) $\Psi' = \psi'_1, \psi'_2, \ldots, \psi'_n$. We design DPIE to be a collection of informative and yet divergent embedding vectors where each ψ'_k reveals the corresponding eigenvector ψ_k more considerably than any other eigenvector. To achieve this goal, all the previous eigenvectors $\Psi_{1:k-1} = [\psi_1, \psi_2, \ldots, \psi_{k-1}]$ must be removed from ψ'_k, which is the major difference between our DPIE and PIE/PIE-k.

In our DPIE, the first nontrivial embedding vector ψ'_2 would be quite similar to PIE but the subsequent DPIEs will be different in the sense that we take out all the already-found DPIEs from the current one. Let v_i^0 denotes the i-th starting random seed vector and $v_i^t = W^t v_i^0$, and the power iteration was stopped at t-th iteration, we compute ψ'_k from the normalized linear fitting residue of the already-found $k-1$ DPIEs:

$$\psi'_k = \frac{v_i^t - \Psi'_{1:k-1} f^*}{\| v_i^t - \Psi'_{1:k-1} f^* \|_1}, \tag{6.4}$$

where $f^* \in R^{(k-1)\times 1}$ is the weight coefficient vector of those already-found DPIEs, and is derived from solving the linear equation $argmin_f = \|v_i^t - \Psi'_{1:k-1} f\|$. In other words, we treat the (unnormalized) ψ'_k as residue or regression error, which is obtained by subtracting the effects of the already-found DPIEs from v_i^t. After normalization ψ'_k becomes the next found DPIE.

However, if we apply the same stopping criteria as that used in PIE or PIE-k, we cannot discover good quality of DPIE. The primary reason is that PIE stopping criteria will suppress the rest of eigenvector signals except the first few because $(\lambda_k/\lambda_2)^t \ll 1$ if t is as large as the PIE stopping criteria. To avoid this problem, we need to increase the acceleration threshold ε of PIE as we find more DPIEs. So, our new stopping criteria for DPIE is as follows:

$$\varepsilon_i = i * \lceil log(c) \rceil * \varepsilon / n, \tag{6.5}$$

where ε is a tuning parameter and we used 10^{-6} by default as in [92] [93].

When ε is too small; or the random seed is similar to one of what we have used; or v_i^t can be well represented by the existing DPIEs, DPIE cannot find any

ALGORITHM 10: DPIE(X, e, E, T, ε_i, η)

Input: $X \in R^{n \times m}$ where n is #instances and m is #features, e is the maximum #DPIE, E is #random seed vectors ($E > e$), T is the maximum #iterations, ε_i defines the acceleration threshold for the i-th random seed, and η is the normalized residual threshold.

Output: Diverse Power iteration embeddings Ψ'.

Construct the affinity matrix of X;
Perform positive random walk normalization on the affinity matrix and denote as W ;
Initialize $v^0 = [v_2^0 \mid v_3^0 \mid \ldots \mid v_E^0] \in R^{n \times E}$, $\Psi' = \{\mathbf{1} \in R^{n \times 1}\}$;
For each v_i^0 ($i = 1, 2, \ldots, E$)
    Repeat
        $v^{t+1} \leftarrow \frac{Wv^t}{\|Wv^t\|_1}$;
        $\delta^{t+1} \leftarrow \left|v^{t+1} - v^t\right|$;
        $t \leftarrow t + 1$;
    until ($\| \delta^t - \delta^{t+1} \|_{max} \leq \varepsilon_i$) or ($t \geq T$) ;
    Solve equation $f^* = argmin_f = \|v_i^t - \Psi'_{1:k-1} f\|$;
    $r_i^t \leftarrow v_i^t - \Psi' f^*$;
    If $\frac{\|r_i^t\|_1}{\|v_i^t\|_1} > \eta$
        $\psi'_i \leftarrow \frac{r_i^t}{\|r_i^t\|_1}$;
        Insert ψ'_i into Ψ' ;
        If size of Ψ' equals to e
            Break ;
        End;
    End ;
End ;
Remove $\mathbf{1}$ from Ψ';

new PIE. In that case, we check the normalized residual (line 12 in Algorithm 10):

$$\vartheta = \frac{\| v_k^t - \Psi'_{1:k-1} f^* \|_1}{\| v_k^t \|_1}. \tag{6.6}$$

If ϑ is smaller than a certain threshold, we do not add such PIEs. In practice, we used $\lceil log(c) \rceil * \eta/n$ as our threshold and $\eta = 10^{-6}$ by default. For notational convenience, we denote the normalized residual threshold as η from now on.

In terms of stabilities, if ε is too large which means we do very early stopping, then we might not be able to find good eigenvector approximations because PIE is a mixture of interesting and noisy eigenvectors. Relatively, the small ε is not a big problem because the normalized residual threshold η can detect the duplicated information and it is just a little bit slower. However, if ε becomes too small then it will lead to over-convergent. In case of η, it is easy to tune because η has the direct meaning of how much new information is added through the new candidate PIE and it is not relevant to eigen-gaps of specific dataset. We present the DPIE stability results in regards to ε and η in Experiment Section 6.7.5.

On the other hand, the power of DPIE can be also interpreted by diffusion theorem. Note that $\Psi_{1:k-1}$ has been removed from ψ'_k, so the explicit formula of ψ'_k is:

$$\psi'_k = b_k \lambda_k^t \psi_k + b_{k+1} \lambda_{k+1}^t \psi_{k+1} + \ldots + b_n \lambda_n^t \psi_n, \tag{6.7}$$

where b_i is the weight coefficient. Considering the 1-norm distance between x and y on ψ'_k there is:

$$D_k^t(x, y) = |\psi'_k(x) - \psi'_k(y)| = \sum_{i=k}^{n} b_i \lambda_i^t |\psi_i(x) - \psi_i(y)|. \tag{6.8}$$

It is actually the same as the diffusion process [28], where $\psi'_k(x)$ is the diffusion coordinate of x after t steps/time diffusion process, with all the directions of ψ_i $(i \geq k)$ taken into account. So $D_k^t(x, y)$ is a family of 1-norm diffusion distances between x and y with Markov diffusion process in time t. It reflects the connectivity in the graph of the data: $D_k^t(x, y)$ will be small if there are a large number of short paths connecting x and y, and large enough walking time t. In other words, there is a large transition probability from x to y [28]. In this sense, t plays the role of a scaling parameter. Therefore DPIE has a potential to be more stable to the noise perturbation.

The whole procedure for DPIE is defined in Algorithm 10. Note that 1) we added one vector **1** from line 3 and take it out from the final results to simulate the first eigenvector ψ_1 which is a constant vector and it plays a role of intercept in line 10 in Algorithm 10, and 2) we start v^0 with v_2^0 instead of v_1^0 due to the same reason. We can see the final DPIEs are quite instructive yet different from each other in Figure 6.2. But like PIE/PIE-k, DPIE is mainly relying on matrix vector multiplication and enjoys the same speed-up and scalability, and it can be easily implemented as distributed matrix vector computation (Section 6.5). Since the most time consuming part (from line 5 to line 9) does not depend on the other DPIE computations, we can further parallelize Algorithm 10.

In the rest of this Section, we provide a simple proof of why DPIE can obtain Ψ' (Equation 6.7), of which each ψ'_k has dominant eigenvector ψ_k while removing the previous eigenvectors $\Psi_{1:k-1}$.

Proposition 1 *Assume that t is sufficient large and clear eigengap exists between every two successive eigenvalues, the linear equation solver (Step 10 to 11 in Algorithm 3) can remove the eigencomponents $\Psi_{1:k-1}$ in order to construct DPIE.*

Proof: Let us assume the first nontrivial DPIE ψ'_2 is found, and the constant eigencomponent (ψ_1) has been removed from ψ'_2 and v_3^t. We now prove we can get ψ'_3 from v_3^t:

$$\begin{aligned} v_3^t &= a_2\lambda_2^t\psi_2 + a_3\lambda_3^t\psi_3 + \ldots + a_n\lambda_n^t\psi_n, \\ \psi'_2 &= b_2\lambda_2^T\psi_2 + b_3\lambda_3^T\psi_3 + \ldots + b_n\lambda_n^T\psi_n, \end{aligned} \tag{6.9}$$

where $T = t + \Delta t$ with $\Delta t \geq 1$ (since we use earlier stopping by controlling ϵ_i when i increases). We assume $argmin_f \|v_3^t - \psi'_2 \times f\| = f_2$ and all $\lambda_j \leq t/(t+\Delta t)$ with $j \geq 1$, there is:

$$(\frac{1}{\lambda})^{\Delta t} > \frac{1}{\lambda} \geq \frac{t+\Delta t}{t}, \tag{6.10}$$

therefore:

$$\frac{\lambda^{t-1}}{\lambda^{T-1}} > \frac{T}{t} \Rightarrow \frac{d(\lambda^t - \lambda^T)}{d\lambda} = t\lambda^{t-1} - T\lambda^{T-1} > 0. \tag{6.11}$$

Since t is sufficiently large, the ratio between a_j and b_j can be ignored. Equation 6.11 means that $\lambda^t - \lambda^T$ becomes larger when λ is larger. Therefore to minimize

the least square $\|v_3^t - \psi_2' \times f\|_2$, there should be $f^* = f_2 \sim \lambda_2^t/\lambda_2^T$, which means the first nontrivial eigenvector ψ_2 is removed from the residue:

$$v_3^t - \psi_2' \times f_2 = \sum_{j=2}^{n}(\lambda_j^t - \lambda_j^T \frac{\lambda_2^t}{\lambda_2^T})\psi_j = \sum_{j=3}^{n}(\lambda_j^t - \lambda_j^T \frac{\lambda_2^t}{\lambda_2^T})\psi_j, \tag{6.12}$$

in which ψ_3 is the dominant vector. For all $j \geq 3$, we assume $\lambda_2/\lambda_j \geq (t + \Delta t)/t$, there is:

$$(\frac{\lambda_2}{\lambda_j})^{\Delta t} > \frac{t + \Delta t}{t} \Rightarrow \frac{d(\lambda_j^t - \lambda_j^T \frac{\lambda_2^t}{\lambda_2^T})}{d\lambda_j} > 0. \tag{6.13}$$

which also leads to the removal of ψ_3 on the following ψ'. Similarly the other eigencomponents can be removed from the coming DPIEs. The above Proposition did not guarantee the eigenvectors if the eigengap is not big between every two successive eigenvalues. However, DPIE procedure guarantees to find diverse PIEs, which are good enough as an approximated eigenvector solution for our proposed applications.

6.5 Efficient Kernel Computation and Complexity Analysis

DPIE provides a scalable and effective alternative to spectral embedding construction, but it still requires the construction of normalized affinity matrix W (line 1 and 2 in Algorithm 10), which is a huge space cost. This section first describes how to avoid the overhead for storing the affinity matrix by using exact cosine similarity or an approximated Gaussian kernel, and then analyzes the time and space complexity of the whole algorithm.

6.5.1 Cosine Similarity

A popular similarity kernel for text dataset is the cosine angle between two vectors, which is defined as:

$$W_{(COS)}(i,j) = \frac{X(i) \times X(j)}{\| X(i) \|_2 \times \| X(j) \|_2}. \tag{6.14}$$

X is usually $tf - idf$ weighted sparse matrix and the two norm normalizations in the denominator term enable us to fairly compare documents with different length.

We apply implicit manifold [93] which is represented with a series of sparse matrix multiplications. As described in [93], for the denominator term an additional diagonal matrix $N_{ii} = 1/\sqrt{X(i)X(i)^T}$ is computed and the affinity matrix A and degree matrix D can be calculated with:

$$\begin{aligned} A &= N \times X \times X^T \times N, \\ D &= N \times X \times X^T \times N \times \mathbf{1}, \end{aligned} \tag{6.15}$$

where $\mathbf{1}$ is a constant vector of all 1's, and X^T denotes the transpose of X. To remove the diagonal on A, we use a modified equation $D = N \times X \times X^T \times N \times \mathbf{1} - 1$. Therefore we can represent random walk power iteration as:

$$Wv^t = D^{-1} \times (N \times (X \times (X^T \times (N \times v^t))) - v^t). \tag{6.16}$$

Since v^t is a $n \times 1$ vector, and D and N are diagonal matrix which can be stored in a sparse format, Equation 6.16 is a lot more efficient to implement and at the same time keeps the same output as the conventional implementation. It is also worth to mention that in anomaly detection application we use bi-normalization instead of one-side random walk normalization to make the anomalies more salient:

$$Wv^t = D^{-1} \times (N \times (X \times (X^T \times (N \times (D^{-1} \times v^t)))) - D^{-1} \times v^t). \tag{6.17}$$

6.5.2 Gaussian Kernel Approximation

One of the most commonly used similarity measurements is the Gaussian kernel:

$$W_{(GAU)}(i,j) = exp(\frac{- \parallel X(i) - X(j) \parallel^2}{2\sigma^2}), \tag{6.18}$$

where σ controls the width of neighborhood [100].

Gaussian kernel is a little bit more complicated than Cosine similarity since it is not a linear construction. In our implementation we approximate it in a space-efficient way by using random Fourier bases [121] [91] shown as follows:

1. Draw d i.i.d. samples $\varpi(1), \ldots, \varpi(d)$ from $p(\varpi \sim \frac{1}{\sigma^2}\mathcal{N}(0,1))$ where $p(*)$ is fast Fourier transform;

Table 6.1: Notations used in the complexity analysis.

	Notations	Meanings
1	n	the number of instances
2	m	the number of features
3	d	the number of samples
4	T	maximum power iterations in DPIE
5	e	maximum number of DPIEs
6	κ	condition number of data eigensystem

2. Draw d i,i,d. samples (offsets) $b(1), \ldots, b(d)$ from uniform distribution on $[0, 2\pi]$;

3. Compute R where $R(i,j) = \sqrt{2/d}[cos(\varpi(j)^T x(i) + b)]$;

4. Use Equation 6.16 or 6.17 by replacing X with R.

This approximation can be interpreted as a random projection with Gaussian basis. It projects each point onto a random direction and passes it through a sinusoidal function with σ as bandwidth, and then slides the function by a random amount (offset) [91]. According to the analysis in [121], as the number of samples d increases, the error of this random Fourier bases approximation goes to zero.

6.5.3 Analysis of Complexity

Space Complexity. Cosine similarity compresses every intermediate result in a vector form $O(n)$, while the Gaussian kernel approximation is based on sampling matrix of which size is $O(nd)$. Therefore, the space complexity is at most $O(nm)$, which is only as the size of original dataset X, which is much smaller than $O(n^2)$ in general.

Time Complexity. Since a matrix vector multiplication requires $O(nm)$, the process from line 5 to line 9 in Algorithm 10 takes $O(nmT)$, while the operation of solving linear systems takes $O(ne\sqrt{\kappa})$ when using conjugated gradient method ($\kappa = \lambda_1^*/\lambda_2^*$ is the condition number of Ψ' where λ_1^* and λ_2^* are the first and second eigenvalue of Ψ') [125]. Note that these time complexities are much smaller than $O(n^3)$.

6.6 Discussion of Theoretical Perspectives

This section justifies the utility of our proposed DPIE by briefly discussing the theoretical distinctions and connections with a few existing methods, which also lays a solid foundation for DPIE's attractive properties for practical use.

Instance-sampling based Methods. Researches like [144] [21] hold a subset of original instances and extend the clustering result to the whole dataset. Other researches like [40] generate a sparser version of matrix by sampling which can be stored more efficiently and multiplied faster. Alternatively the similarity matrix can also be sampled, which is known as the Nyström method [49]. These methods, although reduce the computation cost, are quite sensitive to the sampling quality [144]. Therefore the embedding quality deteriorates with poor sampling. On the contrary, our proposed DPIE does not rely on any sampling strategy.

Random-projection based Methods. Yan et.al. proposed a general framework [152] for fast approximate spectral clustering. It leverages random projection tree to produce a set of reduced representatives and uses them as centroids to cluster all the instances. Gittens et.al. [53] used randomized sketching to approximate the eigenvectors. Their qualities rely on the subspace embedding techniques which result from random projections. However the generated embeddings, because of the indeterministic process, could contain a lot of noisy signals and fail to provide desirable result. In spite of the fact that our DPIE also has random seed vectors as initial status, the seed vectors eventually converge to certain patterns of eigenvector combination during power iteration.

Frequent-direction based Methods. Recent researches drew on the similarity between matrix sketching and the item frequency estimation problems, and proposed frequent-direction based methods [90] with two major contributions: 1) because it is one-pass streaming algorithm, it can be implemented in space and time efficiently, and 2) it approximates the truncated Singular Value Decompositions (SVD). These methods are claimed to be deterministic since they have no sampling or any randomized components. However, their quality is highly related to the input

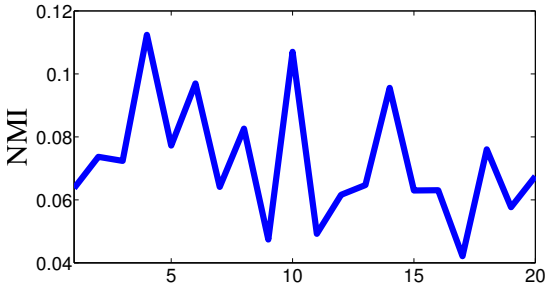


Figure 6.3: MatrixSketching [90] clustering results (recorded in NMI) on 20NG-10 dataset, which is a subset of 20Newsgroups with 10 clusters. We ran the algorithm 20 times and every time we shuffled the input order randomly. Obviously the results are NOT stable against different input order, and a lot worse than our DPIE result (NMI = 0.4373).

order. For instance, we evaluated the matrix sketching quality of [90] on 20NG-10 dataset 20 times and each time we randomly shuffled the order of input, and performed K-means clustering on the final sketched matrix (evaluated by NMI [132]). Figure 6.3 shows its poor results and the instability recorded in NMI across the 20 randomly shuffled experiments. On the other hand, our proposed DPIE is constructed with close connections with random walk process. Thereby, DPIE is more stable against perturbation or noisy features.

Power Iteration based Methods. Power iteration clustering [92] computes a linear combination of the important eigenvectors. It is extremely simple and elegant, and efficient in practice and this is why our work shares the same foundation. Different from the sampling methods and random projection methods, PIC in theory does not modify the original data distribution thus there is no lost information. However the major drawback it suffers is that it tends to return only the first few (or even only one) eigenvectors, which are not enough to represent the datasets with multiple classes or patterns. Although an advanced version, PIE-k, has been proposed later in [91] with multiple output vectors, it does not solve the signal-overlapping problem. Recently deflation-based power iteration method was proposed [137]. It applies Schur complement deflation to remove the previously found

pseudo-eigenvectors from the current matrix, so that it computes multiple orthogonal vectors without redundancy. However, strict orthogonality is also a "double-edged sword" since it requires more iterations to extract certain eigenvectors with smaller eigengaps, therefore deflation-based methods take more time to converge compared with PIE-based methods. On the other hand, our DPIE also intends to eliminate the previously found embedding vectors from the next one. But it does not require the embeddings to be orthogonal to each other: each embedding is a different linear combination of eigenvectors. DPIE has similar representation power as real eigenvectors but takes much less iterations than the deflation PIC, resulted in faster computational speed.

6.7 Experimental Analysis

The low rank embeddings can be used on many data mining applications. We evaluate the quality of the generated embedding vectors through three different application areas: clustering, anomaly detection, and feature selection. For a fair comparison, we constrain each test within a single thread to measure the actual running time. But we want to emphasize that all the algorithms, especially our DPIE, can be implemented and run in a parallel environment.

- **Clustering.** We perform K-means on the generated low-rank embeddings and evaluate the clustering result with NMI (Normalized Mutual Information [132]).
- **Anomaly Detection.** We approximately compute Heat Kernel Signature (HKS) [133] [69] score using the generated low-rank embeddings and evaluate the score with AUC (Area under Receiver Operating Characteristics Curve [102]) which is commonly used to evaluate anomaly detectors and is cut-off independent [96].
- **Feature Selection.** We apply Multi-Cluster Feature Selection (MCFS) [35] with the low-rank embeddings as input to extract feature subset. Although it would be the best to evaluate results based on ground truth of feature importance, it is difficult to find such ground truth. Therefore we evaluate with NMI by applying K-means clustering on the selected feature space.

Table 6.2: Statistics of datasets (including number of instances, features, clusters or anomalies).

	Dataset	# ins.	# fea.	# clu.
1	20Newsgroups	18846	26214	20
2	Reuters21578	8293	18933	65
3	RCV1	193844	47236	103
4	USPS	9298	256	10
5	MNIST	70000	784	10
	Dataset	# ins.	# fea.	# ano.
6	20NG-10-11	4991	26214	100
7	Reuters21578AD	6261	18933	493
8	RCV1AD	7803	29992	200
9	magic04	19020	10	6688
10	satellite	6435	36	2036

6.7.1 Datasets, Baselines and Parameters.

Datasets. All datasets used in the experiments are summarized in Table 6.2. To demonstrate the quality of the generated embedding on **clustering**, we evaluate our algorithm on three text datasets: 20Newsgroups, Reuters21578 and RCV1, and two image datasets USPS and MNIST. Both of the USPS and MNIST datasets are 10 classes of handwritten digits. Reuters21578 and USPS are unbalanced datasets with quite different size of clusters. For **feature selection** evaluation, we focus on two datasets: 20Newsgroups and Reuters21578. In case of **anomaly detection**, we choose three text datasets and two scientific datasets. 20NG-10-11 is a subset of 20Newsgroups, which consists of all the samples from 6 computer-related clusters (from "comp.graphics" to "comp.windows.x" and treated as regular samples) and 100 randomly-selected samples from "talk.religion.misc" (anomalous samples). Reuters21578AD is a subset of Reuters21578 which is composed of the first two largest categories as regular documents and the smallest 45 categories as anomalous documents. RCV1AD is a subset of RCV1 which is made up of four categories "C15", "ECAT", "GCAT", and "MCAT" and we selected 200 "C15" category documents as anomalies and the rest of three categories as regular documents. Satellite consists of the multi-spectral values of pixels in 3×3 neighborhoods in a satellite image which has unbalanced classification associated with each

neighborhood central pixel. Magic04 is a binary classification dataset from the UCI repository which was generated to simulate registration of high energy gamma particles.

For text datasets, cosine similarity (Section 6.5.1) is a reasonable choice. For USPS, MNIST, magic04 and satellite, Gaussian kernel (Section 6.5.2) is used. To adopt an adaptive width of neighborhood σ instead of a fixed value, we assign σ to be the average Euclidean distance of each instance to its second nearest neighbor.

Baselines. For clustering we choose five baselines: NJW (one of the conventional spectral clustering, or Spectral Embedding (SE) when we mention in feature selection) [109], Power Iteration Embedding (PIE) [92], PIE-k [91], Matrix Sketching (MatSket) [90] and DeflationPIC [137]. Once we get the embeddings, we performed a 2-norm normalization along instance side and a WCSS (minimizing within-cluster sum of squares, with 100 inner loops and 100 outer loops) K-means to obtain the cluster assignments.

The anomaly detection experiment is inspired by HKS [69] which is a measure of $X(i)$'s anomalousness using $H_t(i) = \sum_p [e^{\lambda_p t}(\psi_p(i))^2]$ (λ and ψ are derived from positive random walk Laplacian). We name HKS with true eigenvectors as HKS-SE. However, since eigenvalues are not explicitly extracted by PIE, PIE-k, MatSket, DeflationPIC and our proposed DPIE, we use the approximated equation $H'(i) = \sum_p [(v_p(i))^2]$ where v_p is the p-th embedding vector, and call them HKS-PIE, HKS-PIEK, HKS-MatSket, HKS-DFL and HKS-DPIE respectively. To have a more comprehensive comparison, we also include IForest [96] which is a very efficient and effective anomaly detection method. IForest detects data-anomalies with binary trees, using the property that anomalies are more susceptible to isolation.

The feature selection experiment is integrated with MCFS [35] which measures the importance of each feature along each generated embedding that corresponds to the contribution of each cluster by minimizing $\{min_{s_p}(\parallel v_p - X s_p \parallel^2 + \beta \mid s_p \mid)\}$ where s_p is a m-dimensional vector and β controls the s_p's approximation speed to zero. For the j-th feature, MCFS defines the feature importance as $max_p |s_{p,j}|$ where $s_{p,j}$ is the j-th element of vector s_p. We evaluate the output feature subsets by WCSS K-means clustering.

Parameters. Firstly, the number of generated embeddings plays an essential role on the embedding quality. It should be large enough to cover all the signals but small enough to stay away from noise. For clustering and feature selection, we use the first c embeddings from NJW, MatSket and DeflationPIC. PIE generates only one vector while PIE-k set $k = \lceil log(c) \rceil$ [91]. We set the maximum number of DPIEs to be $e = \lceil log(c) \rceil \times 6$ out of $E = max(\lceil log(c) \rceil \times 30, 2c)$ random seeds. In anomaly detection experiment, for HKS-SE we use all the eigenvectors with eigenvalue-weighted, as the original definition in [133] and [69]. HKS-PIE use only one embedding. For a fair comparison, we compute H' with (the first) 5 output embeddings for HKS-PIEK, HKS-MatSket HKS-DFL and our HKS-DPIE. It is also worth to mention the followings: 1) As the other methods, we use the same normalized affinity matrix as the input in Matrix Sketching to provide manifold insight; 2) For text dataset on IForest, we use l_2-norm normalized X as input to make sure that the result is not sensitive to the document length; and 3) For MCFS in feature selection, we perform 2-norm normalization along sample side of X to evaluate uniform feature scales.

The heat diffusion time variable t in HKS-SE is set to be 1 in order to avoid over-diffusion [69]. In IForest, to conduct a safe and fair comparison, we set the sub-sampling size $\rho = 4000$ and the number of trees $nt = 100$ because these parameters are the authors' recommendation [95].

When we use Gaussian kernel approximation (Section 6.5.2) we set the number of samples $d = 2000$ and $\sigma = 2000$. The maximum number of power iteration T is fixed to be 1000. Acceleration convergence rate in PIE and PIE-k is set to be $\varepsilon = 10^{-5}/n$ where n is the number of samples, as described in [92] and [91]. In our proposed DPIE, we set $\varepsilon_i = i \times \lceil log(c) \rceil \times \varepsilon/n$ with $\varepsilon = 10^{-6}$, and normalized residual threshold as $\lceil log(c) \rceil \times \eta/n$ with $\eta = 10^{-6}$ by default. In Section 6.7.5 we test DPIE stability with different ε and η.

Finally, for each method with sampling steps or random seeds, we run 50 times and report the average performance.

Table 6.3: Clustering results in NMI and time consuming. For each dataset, the bold-faced number indicates the best approximation method (**except NJW**), and the numbers in the parentheses indicate the ranks of our DPIE. Average is the average NMI and Time of each method across all the datasets respectively.

NMI	NJW	PIE	PIE-k	MatSket	DeflationPIC	DPIE
20Newsgroups	0.5326	0.2519	0.3266	0.4877	0.4847	**0.5061** (1)
Reuters21578	0.5048	0.2557	0.2718	**0.5322**	0.5014	0.5143 (2)
RCV1	0.2875[1]	0.1022	0.1237	0.1521	0.1941	**0.2644** (1)
USPS	0.6207	0.2026	0.2401	0.4667	**0.5871**	0.5786 (2)
MNIST	0.4433	0.0022	0.0028	0.3523	0.3788	**0.4032** (1)
Average	0.4778	0.1629	0.1930	0.3982	0.4292	**0.4533** (1)
Time(s)	NJW	PIE	PIE-k	MatSket	DeflationPIC	DPIE
20Newsgroups	5653.0193	0.1461	5.0816	4131.7741	35.4688	5.0834
Reuters21578	1958.5777	0.0671	2.3548	830.7118	13.7681	1.6388
RCV1	——	5.1961	110.5477	108998.2234	923.6324	127.6903
USPS	1665.3840	0.0675	1.9807	395.9329	7.2451	0.6584
MNIST	201581.2017	4.0707	38.8645	46072.8311	196.3723	43.6582
Average	——	1.9095	31.7659	32085.8947	235.2973	35.7458

6.7.2 Clustering Result Analysis

The clustering results are summarized in Table 6.3. We reported the time used for the affinity matrix and embeddings constructions but we excluded the final K-means steps. For NJW, we also excluded the affinity matrix construction time.

Generally speaking, NJW has the best average performance in NMI since it has full knowledge of the real eigenvectors, but at the same time requires the most expensive cost in time. Compared with PIE, PIE-k is 15 times slower on average since it requires more input and output, but PIE-k improves 20% on average NMI since it has the potential to contain different aspects of signal resulting from different starting vectors. However, it only gets 40% of NJW in NMI. By truncated SVD on normalized affinity matrix, MatSket can deterministically extract the low rank approximation. So it covers additional signals in a more effective way than PIE-k (more than two times better in NMI). But at the same time MatSket is also $1000+$ times slower than PIE-k since it requires lots of SVD calculations. DeflationPIC, on the other hand, computes multiple orthogonal pseudo-eigenvectors using deflation technique, so that it could approximate the original eigenvectors to certain degree. It shows improved performance in USPS and MNIST compared with MatSket. But since it requires more matrix computations in the deflation equation, it is noticeably much slower than PIE-k. Our DPIE, although not always the best among all the (approximate) methods, achieves more than 95% performance of NJW in NMI, and at the same time only requires quite a short running time which is close to PIE-k. Especially, DPIE only takes about 2 minutes to process RCV1 dataset but more than 35% better than the second best approximation method with 7 times faster speed.

Due to out-of-memory problem, the NJW experiment on RCV1 could not be finished since it requires full affinity matrix construction. However, using the space-efficient ways introduced in Section 6.5 it is not a problem for the other listed methods, especially our proposed DPIE.

[1]We couldn't run NJW on RCV1 dataset due to out-of-memory error, but instead cite its NJW score from [129] for reference.

Table 6.4: Anomaly detection results in AUC and time consuming. For each dataset, the bold-faced number indicates the best approximation method (**except HKS-SE**), and the numbers in the parentheses indicate the ranks of our HKS-DPIE. Average is the average AUC and time of each method across all the datasets respectively.

AUC	HKS-SE	HKS-PIE	HKS-PIEK	HKS-MatSket	HKS-DFL	IForest	HKS-DPIE
20NG-10-11	0.9042	0.3294	0.4858	0.6331	0.2318	0.6176	**0.8844** (1)
Reuters21578AD	0.7845	0.3034	0.5131	0.4824	0.7863	0.6048	**0.9271** (1)
RCV1AD	0.5428	0.4403	0.5049	0.4619	**0.5925**	0.4879	0.5547 (2)
magic04	0.7286	0.5757	0.5757	0.5799	0.4205	**0.7506**	0.7179 (3)
satellite	0.7078	0.3378	0.3378	0.5062	0.5416	0.7173	**0.7193** (1)
Average	0.7336	0.3973	0.4835	0.5327	0.5145	0.6356	**0.7607** (1)
Time(s)	HKS-SE	HKS-PIE	HKS-PIEK	HKS-MatSket	HKS-DFL	IForest	HKS-DPIE
20NG-10-11	876.9247	0.0297	0.8683	181.7283	5.7138	7.6199	0.8193
Reuters21578AD	4141.9718	0.0528	1.1995	170.0181	7.3392	8.2016	1.0608
RCV1AD	4199.1405	0.0476	1.3253	475.9983	10.6519	5.5944	1.1128
magic04	14732.0387	0.1252	0.3402	3241.6766	20.3112	53.8751	2.2759
satellite	779.7334	0.0145	0.1121	152.7320	8.9713	49.3959	0.5889
Average	4945.9618	0.0540	0.7691	844.4307	10.5975	24.9374	1.1715

6.7.3 Anomaly Detection

Table 6.4 shows the anomaly detection results. Similar to the clustering comparisons, HKS-PIEK performed better than HKS-PIE (21% improvement), with the reason that PIE-k is possible to provide more informative signals. HKS-DFL and HKS-MatSket can capture supplementary yet important eigenvectors, which leads to a 6% and 10% boost up respectively compared with HKS-PIEK, but still much worse than HKS-SE (less than 73%). IForest is efficient in that it detects the anomalies by recording the short expected path lengths, so that it has 200% faster running time than HKS-SE and still acquires $86 + \%$ performance of HKS-SE. However, our proposed HKS-DPIE is 4220 times faster than HKS-SE and yet reach the best average performance.

6.7.4 Feature Selection

We tested all the embedding construction methods using MCFS [35] with $\{50$, 200, 800, 1200, $1800\}$ selected features, and reported the result in Table 6.5. Similar to clustering experiments, DeflationPIC and MatSket perform better than PIE-k and PIE. But DPIE extracts more representative features, which are even with better quality than those derived from original spectral embeddings (SE). This can be explained by the fact that DPIE formulates all the informative signals within diffusion space, which is a more compact and profound way than discrete eigenvectors.

Table 6.5: Feature selection results in NMI. For each dataset, the bold-faced number indicates the best approximated method, and the numbers in the parentheses indicate the ranks of our DPIE. Average is the average NMI of each method. Due to space limitation and the close connections between clustering and feature selection technique we used in this experiment we do not list the time consuming here.

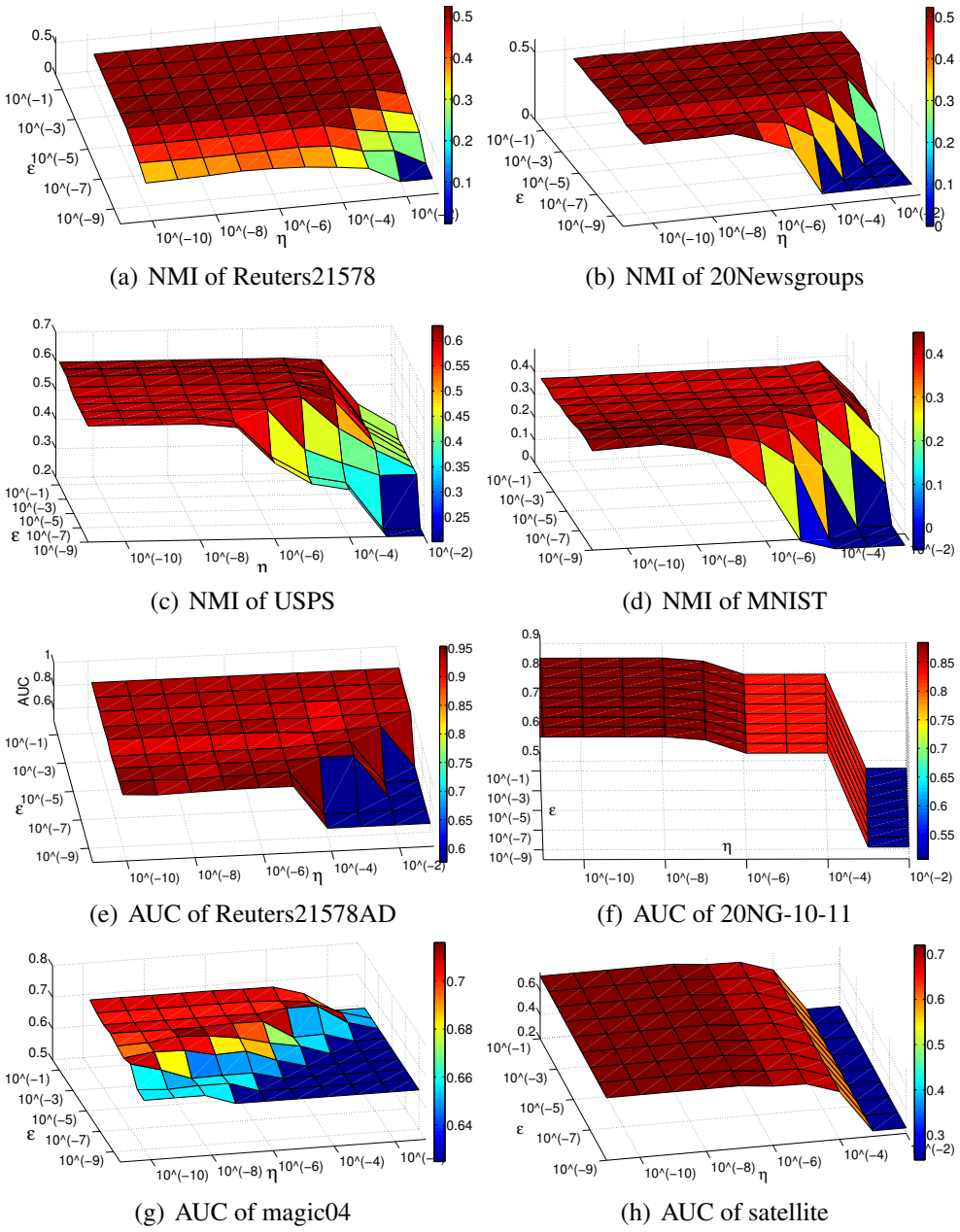
20Newsgroups	MCFS-SE	MCFS-PIE	MCFS-PIEK	MCFS-MatSket	MCFS-DFL	MCFS-DPIE
50	0.2971	0.1691	0.1590	0.2691	0.2552	**0.3446** (1)
200	0.3361	0.3089	0.3181	0.3603	0.3274	**0.3834** (1)
800	0.4118	0.3899	0.4115	0.4061	0.4256	**0.4372** (1)
1200	0.4256	0.4696	0.4498	0.4692	0.4335	**0.4819** (1)
1800	0.4865	0.4671	0.4587	0.4340	0.4748	**0.4993** (1)
Reuters21578	MCFS-SE	MCFS-PIE	MCFS-PIEK	MCFS-MatSket	MCFS-DFL	MCFS-DPIE
50	0.3957	0.3959	0.3889	**0.4399**	0.3973	0.4366 (2)
200	0.4607	0.4539	0.4598	0.4745	0.4677	**0.4814** (1)
800	0.5125	0.5021	**0.5183**	0.5113	0.4993	0.5176 (2)
1200	0.5125	0.4783	0.4882	0.4971	0.5122	**0.5297** (1)
1800	0.5081	0.5104	0.5078	0.4980	0.5200	**0.5308** (1)
Average	0.4347	0.4145	0.4160	0.4360	0.4313	**0.4646** (1)

6.7.5 Stability Experiments

We conduct experiments with different acceleration threshold ε and normalized residual threshold η to study the parameter tuning sensitivities of DPIE. The results are illustrated in Figure 6.4. It indicates that DPIE has a stable range of performance on clustering with large enough ε and small enough η. The reason is that for clustering we need more number of embeddings which cover enough informative eigenvectors. Consequently the iteration should have early stopping controlled by increasing ε to prevent the iteration procedure to remove the less strong eigencomponents, and lowering η to include more diverse DPIEs. Similarly, for anomaly detection DPIE performs stably with large ε and small η. If the anomalies only take a small percentage of total instances, more PIEs are required to separate anomalies from the normal ones. By assigning large enough ε and small enough η, we ensure to obtain enough PIEs while removing the negative influence from the later (noisy) ones.

6.8 Chapter Summary

In this chapter we propose a power-iteration-based low dimensional embeddings to cope with the time and space complexities of traditional spectral analysis. Our proposed Diverse Power Iteration Embedding (DPIE), inspired by the power iteration embedding (PIE [92]), can eliminate duplicated information due to a few dominant eigenvectors, which makes it achieve outstanding performance compared with PIE and other related methods [91]. DPIE can be used for not only clustering but also various spectral analysis including feature selection and anomaly detection. Extensive experiments and evaluations on the three spectral analysis applications have demonstrated that our proposed DPIE is the most effective in improving the clustering, anomaly detection, and feature selection methods in the comparison with state-of-the-art baseline approximation algorithms. Meanwhile, DPIE remains efficient in terms of time and space complexity, i.e. being as efficient as PIE-k and much faster than MatrixSketching [90] and DeflationPIC [137].



(a) NMI of Reuters21578

(b) NMI of 20Newsgroups

(c) NMI of USPS

(d) NMI of MNIST

(e) AUC of Reuters21578AD

(f) AUC of 20NG-10-11

(g) AUC of magic04

(h) AUC of satellite

Figure 6.4: Stability experiment with different acceleration threshold ε and normalized residual threshold η.

Chapter 7

Conclusion and Future Work

This chapter summarizes our finished works, ongoing works and future research directions. Our works focus on seeking novel modeling techniques based on physics principles that not only have solid theoretical foundation, but also have intrinsic and informative insight on those high-dimensional and heterogeneous datasets from various domains. We also demonstrate that our scalable physics-based modeling framework has great potential in many valuable applications.

7.1 Contribution Summary

In our research, we have investigated and presented a physics-based methodology framework, with which we can combine clustering, local anomaly detection, and feature selection together. Moreover, we have proposed a practical way to apply this framework on real world large-scale datasets. Our work is generally applicable to most of the application domains without specific assumptions. Our salient contributions include:

1. We have developed a new spectral clustering algorithm, called **Aggregated Heat Kernel (AHK)**, with robustness to both scaling parameter tuning and data perturbation. This research was originated from heat kernel and diffusion maps. In technical essence, our AHK permits reorganizing the spectral-embedded structure regardless sub-optimal scaling parameter selection and noise perturbation.

2. We have developed a novel clustering methodology that seamlessly integrates two powerful concepts: **Local Density Affinity Transformation (LDAT)** and AHK to achieve remarkable performance under the heterogeneous density distributions. Three primary advantages of this work include: (1) It suppresses local density bias of different density in affinity matrix; (2) It functions well with any affinity measurement in a universally-applicable way; and (3) It reconstructs different density manifold with high fidelity and utilizes them to offer guidance during clustering.

3. We have designed an original unsupervised anomaly detection algorithm, called **Local Anomaly Descriptor (LAD)**, which is based on the physics-inspired diffusion space and weighted umbrella operator. Compared with the existing algorithms, our proposed LAD has demonstrated many important properties such as intrinsicality to local density, and stability to small parameter perturbation.

4. We have devised a new unsupervised anomaly detection algorithm, called **Fermi Density Descriptor (FDD)**, with strong performance and robustness to parameter tuning. This algorithm was originated from quantum mechanics, and has a physically-rigorous probability interpretation. To further enhance the functionality of our algorithm, we first explored the best choice among different Laplacian normalizations with the goal of mining anomalous instances.

5. We have proposed an unsupervised **Noise-Resistant Feature Selection (NRFS)**. NRFS is a collaborative feature selection algorithm based on multi-perspective correlation, in that it probes the feature effect via local view from instance representatives and global spectrums, and thereby effectively distinguishes informative, non-redundant and diverse features from the remaining ones. Moreover, NRFS applies noise-resistant and density-preserving sampling to improve its efficiency while reducing the negative affect incurred by noisy instances.

6. To make the above spectral-embedding-based methods applicable on large-scale datasets, we proposed a power-iteration-based low-rank embedding construction, called **Diverse Power Iteration Embedding (DPIE)**, to approximate the classic yet inefficient spectral embedding construction. Compared with the other approximation methods, DPIE has outstanding performance with fast running

time and low storage requirement. Extensive experiments and evaluations have demonstrated that DPIE is very effective and efficient in improving the clustering, anomaly detection, and feature selection methods.

Besides the aforementioned contributions, we have also accomplished a few works on real-world application, including short-term solar energy prediction and nanoscale structure mining using our proposed machine learning and data mining techniques. Since they are not under the theme of this dissertation, we did not list them here.

7.2 On-going Works

There are still many immediate and valuable research topics that can be included into our current framework. Following are several of them that can be directly extend from work we have done so far.

7.2.1 Coclustering for Microarray Datasets

On many application datasets such as document datasets and biological microarray datasets, there are usually corresponding correlations between instance subsets and attribution subsets. In other words, a cocluster of instances in these datasets can be composed of instances with similarity over only a subset of attributions. Unsupervisedly discovering such interrelations is not achievable by traditional clustering methods. The major reason is that the traditional methods can only tackle each data type independently, which will lose the interaction that are essential to gain a full understanding of the data.

Coclustering (Biclustering) algorithms simultaneously cluster row and column of a dataset. Row and column usually represent data instance and feature respectively. Therefore, the results of the coclustering algorithm reveal which group of features maximally response to which group of data instances, or vice versa. Through coclustering, we are able to discover a hidden global structure in the heterogeneous data, and it will seamlessly integrate multiple data types to provide us a better picture of the underlying data distribution with high value in many real world applications.

There are so many coclustering algorithms categorized by the type of their cost functions [79] [26] [37] [22] [139] [32] [99]. The existing algorithms are either too sensitive to parameter tuning, or to initialization [22] [139] [99] and nonuniform density distribution [79] [26] [37] [32] in instance space or attribute space. Moreover, they are lack of simultaneous consideration between instance density and attribution density distribution.

We attempt to simultaneously provide manifold and density awareness between instance space and attribute space, and to integrate the interaction measurement into coclustering methods. This can improve the discovery power of interrelations inside those complex datasets. By building the global affinity of not only intra-instances or intra-attribute, but also between instances and attributes, and applying our density-sensitive affinity transformation, we seek to recognize the subset structure with strong correlation between instance and feature/attribute.

7.2.2 Heat Spreading Clustering with Boundary Constrains

We have proposed heat diffusion based clustering algorithms in our finished work. One of the advantages of such techniques is that, both diffusion and its kernel function afford robust description of manifold reconstruction, with solid probabilistic interpretation. While these techniques have shown their promising potentials in robust clustering, there still remain certain limitations in the current state-of-the-art, including initial heat source locating, and boundary area over-connection.

First, existing work oftentimes emphasizes the equal probability of heat diffusion (or more generally speaking, random walk) starting point among all the instances, while paying far less attention to locating/selecting the initial heat source. The heat spreading tends to lose control if we assign all the instances as heat source, since the diffusion process focus on the global trend instead of only trace heat spreading of representative heat source. To avoid such problems, the heat sources should be those representative center points inside each clusters. The heat from such heat sources gradually diffuse inside the intra-cluster area without bridging inter-cluster structure together. On the other hand, if the boundary instances are assigned as initial heat sources, after a number of random walk or diffusion steps,

the heat dissipation among these instances tends to connect different clusters together. In short, these characteristics of over-diffusion fail to refine information to describe and reconstruct the manifold. Moreover, they are sensitive to density changes between clusters with overlapping boundary area, which goes against the manifold-aware nature of the diffusion.

To better depict the characteristics of a manifold, an informative and stable initial heat source locating algorithm is strongly desirable. Furthermore, the affinity between boundary points should be set as zero or very small to avoid over connection among clusters. In this way the heat diffusions inside each cluster are isotropic in nature, which are based on isotropic heat kernels inside manifolds, while diffusion among boundary area will be anisotropic to control the diffusion direction via small weighted boundary affinity.

7.3 Future Directions

In long term, we expect to extend our scalable physics-based data modeling framework to more complex structure discovery such as structural pattern recognition [124] and time-series learning [47]. Recently graph kernels [124] have received considerable interest within the machine learning and data mining community. We plan to introduce a novel approach enabling kernel methods to utilize additional information hidden in the structural neighborhood of the graphs. Our assumption is that graph similarity can not only be described by the similarity between instances, but also by the similarity between structural neighborhood. Furthermore, we will also explore the potential to apply our proposed methodology on time-series data [47] by constructing multi-scale coordinates, or dynamic parameterization. We look forward that this series of works will provide us more insight of physics-based data modeling under different applications.

Bibliography

[1] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu. A framework for projected clustering of high dimensional data streams. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 852–863. VLDB Endowment, 2004.

[2] A. Agovic, A. Banerjee, A. R. Ganguly, and V. Protopopescu. Anomaly detection in transportation corridors using manifold embedding. In *Proceedings of the First International Workshop on Knowledge Discovery from Sensor Data, ACM KDD Conference*, 2007.

[3] N. O. Andrews and E. A. Fox. Recent developments in document clustering. *Technical Report TR-07-35*, 2007.

[4] M. Ankerst, M. M. Breunig, H. P. Kriegel, and J. Sander. Optics: Ordering points to identify the clustering structure. *ACM SIGMOD*, pages 49–60, 1999.

[5] W. Arendt. Vector-valued laplace transforms and cauchy problems. *Springer*, 96, 2011.

[6] E. Arias-Castro, D. Donoho, and X. Huo. Near-optimal detection of geometric objects by fast multiscale methods. *Stanford Statistics Department Technical Report*, 2003.

[7] M. Aubry, U. Schlickewei, and D. Cremers. The wave kernel signature - a quantum mechanical approach to shape analyis. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1626–1633. IEEE, 2011.

[8] R. Badeau, B. David, and G. Richard. Fast approximated power iteration subspace tracking. *IEEE Signal Processing*, pages 2931–2941, 2005.

[9] D. Barbará, C. Domeniconi, and J. P. Rogers. Detecting outliers using transduction and statistical testing. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 55–64. ACM, 2006.

[10] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 6(15):1373–1396, 2003.

[11] M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger. Robust anisotropic diffusion. *Image Processing, IEEE Transactions on 1998*, 7(3):421–432, 1998.

[12] G. Blanchard, G. Lee, and C. Scott. Semi-supervised novelty detection. *The Journal of Machine Learning Research*, 11:2973–3009, 2010.

[13] B. Bolstad. Probe level quantile normalization of high density oligonucleotide array data. *Unpublished manuscript*, 2001.

[14] M. M. Breunig, H. P. Kriegel, R. T. Ng, and J. Sander. Lof: Identifying density-based local outliers. *ACM SIGMOD*, pages 93–104, 2000.

[15] M. Budka and G. Bogdan. Density-preserving sampling: Robust and efficient alternative to cross-validation for error estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):22–34, 2013.

[16] T. Buhler and M. Hein. Spectral clustering based on the graph p-laplacian. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 81–88. ACM, 2009.

[17] G. Camps-Valls, L. Bruzzone, J. L. Rojo-Álvarez, and F. Melgani. Robust support vector regression for biophysical variable estimation from remotely sensed images. *Geoscience and Remote Sensing Letters, IEEE*, 3(3):339–343, 2006.

[18] G. Carlsson, A. Collins, L. Guibas, and A. Zomorodian. Persistent homology and shape description. *preprint, Stanford Math Department*, 2003.

[19] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: a survey. *ACM Computing Surveys*, 41(3):1–72, 2009.

[20] H. Chang and D. Y. Yeung. Robust path-based spectral clustering. *Pattern Recognition*, 41(1):191–203, 2008.

[21] X. Chen and D. Cai. Large scale spectral clustering with landmark-based representation. In *AAAI*, pages 313–318, 2011.

[22] Y. Chen, L. Wang, and M. Dong. Non-negative matrix factorization for semisupervised heterogeneous data coclustering. *IEEE Knowledge and Data Engineering*, 22(10):1459–1474, 2010.

[23] S. Y. Cheng, P. Li, and S. T. Yau. Heat equations on minimal submanifolds and their applications. *American Journal of Mathematics*, 106(5):1033–1065, 1984.

[24] V. Cherkassky and Y. Ma. Multiple model regression estimation. *Neural Networks, IEEE Transactions on*, 16(4):785–798, 2005.

[25] Y. Chi, X. Song, D. Zhou, K. Hino, and B. L. Tseng. On evolutionary spectral clustering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 3(4):17, 2009.

[26] H. Cho and I. S. Dhillon. Coclustering of human cancer microarrays using minimum sum-squared residue coclustering. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 5(3):385–400, 2008.

[27] R. R. Coifman, I. G. Kevrekidis, S. Lafon, M. Maggioni, and B. Nadler. Diffusion maps, reduction coordinates and low dimensional representation of stochastic systems. *Multiscale Modeling & Simulation*, 7(2):842–864, 2008.

[28] R. R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006.

[29] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, 2002.

[30] C. D. Correa and P. Lindstrom. Locally-scaled spectral clustering using empty region graphs. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1330–1338. ACM, 2012.

[31] P. Courrieu. Fast computation of moore-penrose inverse matrices. *Nueural Information Processing-Letters and Review*, pages 303–308, 2005.

[32] W. Dai. Co-clustering based classification for out-of-domain documents. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, volume 12, pages 210–219. ACM, 2007.

[33] R. N. Davé and R. Krishnapuram. Robust clustering methods: a unified view. *Fuzzy Systems, IEEE Transactions on*, 5(2):270–293, 1997.

[34] T. de Vires, S. Chawla, and M. Houle. Finding local anomalies in very high-dimensional space. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 128–137. IEEE, 2010.

[35] C. Deng, C. Zhang, and X. He. Unsupervised feature selection for multi-cluster data. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 333–342. ACM, 2010.

[36] M. Desbrun, M. Meyer, P. Schrder, and A. Barr. Implicit fairing of arbitrary meshes using diffusion and curvature flow. *ACM SIGGRAPH*, pages 317–324, 1999.

[37] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 269–274. ACM, 2001.

[38] D. L. Donoho and C. Grimes. When does isomap recover the natural parametrization of families of articulated images? *Technical Report 2002-27: Department of Statistics, Stanford University*, 2002.

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

[40] P. Drineas and A. Zouzias. A note on element-wise matrix sparsification via a matrix-valued bernstein inequality. *Information Processing Letters*, 111(8):385–389, 2011.

[41] K. B. Driver. Compact and fredholm operators and the spectral theorem. *Analysis tools with applications, Lecture Notes*, 35:579–600, 2003.

[42] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. John Wiley & Sons,, 1999.

[43] J. G. Dy. Unsupervised feature selection. *Computational Methods of Feature Selection*, pages 19–39, 2008.

[44] J. G. Dy and C. E. Brodley. Feature selection for unsupervised learning. *The Journal of Machine Learning Research*, 5:845–889, 2004.

[45] B. Efron and R. Tibshirani. *An Introduction to the Bootstrap*, volume 57. CRC press, 1994.

[46] L. Ertoz, M. Steinbach, and V. Kumar. A new shared nearest neighbor clustering algorithm and its applications. *In Workshop on Clustering High Dimensional Data and its Applications at 2nd SIAM International Conference on Data Mining*, pages 105–115, 2002.

[47] P. Esling and C. Agon. Time-series data mining. *ACM Computing Surveys (CSUR)*, 45(1):12, 2012.

[48] M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. *KDD*, 96:226–231, 1996.

[49] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the nyström method. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(2):214–225, 2004.

[50] J. Gao, H. Cheng, and P. N. Tan. Semi-supervised outlier detection. In *Proceedings of the 2006 ACM symposium on Applied computing*, pages 635–636. ACM, 2006.

[51] V. Garcia, E. Debreuve, and M. Barlaud. Fast k nearest neighbor search using gpu. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–6. IEEE, 2008.

[52] J. Giesen and W. U. Shape dimension and intrinsic metric from samples of manifolds with high co-dimension. In *Proceedings of the nineteenth annual symposium on Computational geometry*, pages 329–337. ACM, 2003.

[53] A. Gittens, P. Kambadur, and C. Boutsidis. Approximate spectral clustering via randomized sketching. *Ebay/IBM Research Technical Report*, 2013.

[54] A. Goh and R. Vidal. Segmenting motions of different types by unsupervised manifold clustering. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–6. IEEE, 2007.

[55] G. Greenstein and A. Zajonc. *The Quantum Challenge: Modern Research on the Foundations of Quantum Mechanics*. Jones & Bartlett Learning, 2006.

[56] A. Grigoryan. Estimates of heat kernels on riemannian manifolds. *London Math. Soc. Lecture Note Ser*, 273:140–225, 1999.

[57] S. Guha, R. Rastogi, and K. Shim. Rock: A robust clustering algorithm for categorical attributes. pages 512–521, 1999.

[58] I. Gutman and W. Xiao. The generalized inverse of the laplacian matrix and some applications. *Bulletin, Sciences mathematiques*, 29:1–9, 2004.

[59] I. Guyon, J. Li, T. Mader, P. A. Pletscher, G. Schneider, and M. Uhr. Competitive baseline methods set new standards for the nips 2003 feature selection benchmark. *Pattern recognition letters*, 28(12):1438–1444, 2007.

[60] J. A. Hartigan and M. A. Wong. Algorithm as 136: a k-means clustering algorithm. *Applied Statistics*, 28(1):100–108, 1978.

[61] D. Hawkins. *Identification of outliers*, volume 11. Springer, 1980.

[62] X. He, D. Cai, and P. Niyogi. Laplacian score for feature selection. In *Advances in neural information processing systems*, pages 507–514, 2005.

[63] K. Hempstalk, E. Frank, and I. H. Witten. One-class classification by combining density and class probability estimation. In *Machine Learning and Knowledge Discovery in Databases*, pages 505–519. Springer, 2008.

[64] A. Hinneburg and D. A. Keim. Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering. In *VLDB*, volume 99, pages 506–517. Citeseer, 1999.

[65] D. Horn and A. Gottlieb. Algorithm for data clustering in pattern recognition problems based on quantum mechanics. *Physical Review Letters*, 88(1):018702, 2001.

[66] D. Horn and A. Gottlieb. The method of quantum clustering. *NIPS*, pages 769–776, 2001.

[67] C. Hou, F. Nie, D. Yi, and Y. Wu. Feature selection via joint embedding learning and sparse regression. *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence*, 2:1324–1329, 2011.

[68] E. Hsu. *Stochastic Analysis on Manifolds*, volume 38. American Mathematical Soc., 2002.

[69] H. Huang, H. Qin, S. Yoo, and D. Yu. Local anomaly descriptor: a robust unsupervised algorithm for anomaly detection based on diffusion space. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 405–414. ACM, 2012.

[70] H. Huang, H. Qin, S. Yoo, and D. Yu. A new anomaly detection algorithm based on quantum mechanics. In *Proceedings of the 2012 IEEE 12th International Conference on Data Mining*, pages 900–905. IEEE Computer Society, 2012.

[71] H. Huang, S. Yoo, H. Qin, and D. Yu. A robust clustering algorithm based on aggregated heat kernel mapping. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 270–279. IEEE, 2011.

[72] P. J. Huber. *Robust statistics*. Springer, 2011.

[73] W. Hung, M. Yang, and D. Chen. Bootstrapping approach to feature-weight selection in fuzzy c-means algorithms with an application in color image segmentation. *Pattern Recognition Letters*, 29(9):1317–1325, 2008.

[74] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, Inc., 1988.

[75] R. A. Jarvis and E. A. Patrick. Clustering using a similarity measure based on shared nearest neighbors. *In IEEE Transactions on Computers*, pages 1025–1034, 1973.

[76] Y. Jiang and J. Ren. Eigenvalue sensitive feature selection. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 89–96, 2011.

[77] T. Joachims. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. *Computer Science Technical Report CMU-CS-96-118*, 1996.

[78] G. Karypis, E. H. Han, and V. Kumar. Chameleon: Hierarchical clustering using dynamic modeling. *In IEEE Transactions on Computers*, pages 68–74, 1999.

[79] Y. Kluger, R. Basri, J. T. Chang, and M. Gerstein. Spectral biclustering of microarray data: Coclustering genes and conditions. *Genome Research*, 13(4):703–716, 2003.

[80] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation andmodel selection. In *IJCAI*, volume 14, pages 1137–1145, 1995.

[81] Y. Koren and R. Bell. Advances in collaborative filtering. *Recommender Systems Handbook*, pages 145–186, 2011.

[82] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[83] H. P. Kriegel, P. Kroger, and A. Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 3(1):1, 2009.

[84] S. Lafon, Y. Keller, and R. R. Coifman. Data fusion and multi-cue data matching by diffusion maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1784–1797, 2006.

[85] D. D. Lewis, Y. Yang, T. G. Rose, F. Li, and G. Dietterich. Rcv1: A new benchmark collection for text categorization research. *The Journal of Machine Learning Research*, 5:361–397, 2004.

[86] Y. Li, M. Dong, and J. Hua. Localized feature selection for clustering. *Pattern Recognition Letters*, 29(1):10–18, 2008.

[87] Y. Li, M. Dong, and J. Hua. Simultaneous localized feature selection and model detection for gaussian mixtures. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(5):953–960, 2009.

[88] Z. Li, J. Liu, S. Chen, and X. Tang. Noise robust spectral clustering. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[89] Z. Li, Y. Yang, J. Liu, X. Zhou, and H. Lu. Unsupervised feature selection using nonnegative spectral analysis. In *AAAI*, pages 1026–1032, 2012.

[90] E. Liberty. Simple and deterministic matrix sketching. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 581–588. ACM, 2013.

[91] F. Lin. Scalable methods for graph-based unsupervised and semi-supervised learning. *Doctoral dissertation, Carnegie Mellon University*, 2012.

[92] F. Lin and W. W. Cohen. Power iteration clustering. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 655–662, 2010.

[93] F. Lin and W. W. Cohen. A very fast method for clustering big text datasets. In *ECAI*, pages 303–308, 2010.

[94] Y. Lipman, R. Rustamov, and T. Funkhouser. Biharmonic distance. *ACM Transactions on Graphics (TOG)*, 29(3):27, 2010.

[95] F. T. Liu and K. M. Ting. Can isolation-based anomaly detectors handle arbitrary multi-modal patterns in data? *Technical Report*, 2010.

[96] F. T. Liu, K. M. Ting, and Z. H. Zhou. Isolation forest. *IEEE ICDM*, pages 413–422, 2008.

[97] F. T. Liu, K. M. Ting, and Z. H. Zhou. Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(1):3, 2012.

[98] H. Liu and L. Yu. Toward integrating feature selection algorithms for classification and clustering. *IEEE Knowledge and Data Engineering*, 17(4):491–502, 2005.

[99] B. Long, Z. M. Zhang, and P. S. Yu. Co-clustering by block value decomposition. *SIGKDD*, pages 635–640, 2005.

[100] U. V. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[101] U. V. Luxburg, A. Radl, and M. Hein. Getting lost in space: Large sample analysis of the resistance distance. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 2622–2630. Curran Associates, Inc., 2010.

[102] C. Marzban. A comment on the roc curve and the area under it as performance measures. *Technical Report*, 2004.

[103] M. Meila and J. Shi. A random walks view of spectral segmentation. *the 8th International Workshop on Artificial Intelligence and Statistics*, 2001.

[104] F. Memoli and G. Sapiro. Distance functions and geodesics on point clouds. *Technical Report 1902, IMA, University of Minnesota, Minneapolis*, 2003.

[105] N. J. Mitra and A. Nguyen. Estimating surface normals in noisy point cloud data. In *Proceedings of the Nineteenth Annual Symposium on Computational Geometry*, pages 322–328. ACM, 2003.

[106] P. Mitra, C. A. Murthy, and S. K. Pal. Unsupervised feature selection using feature similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):301–312, 2002.

[107] B. Nadler, S. Lafon, R. Coifman, and I. Kevrekidis. Diffusion maps, spectral clustering and eigenfunctions of fokker-planck operators. *NIPS*, pages 955–962, 2005.

[108] N. Nasios and A. G. Bors. Kernel-based classification using quantum mechanics. *Pattern recognition*, 40(3):875–889, 2007.

[109] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems*, 14:846–856, 2002.

[110] K. Noto, C. E. Brodley, and D. Slonim. Anomaly detection using an ensemble of feature models. *IEEE ICDM*, pages 953–958, 2010.

[111] S. K. Pal and P. Mitra. *Pattern recognition algorithms for data mining*. CRC press, 2004.

[112] V. Y. Pan and Z. Q. Chen. The complexity of the matrix eigenproblem. *ACM STOC*, pages 507–516, 1999.

[113] S. Papadimitriou, H. Kitagawa, P. B. Gibbons, and C. Faloutsos. Loci: Fast outlier detection using the local correlation integral. *IEEE ICDE*, pages 315–326, 2003.

[114] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2010.

[115] P. Perona and W. T. Freeman. A factorization approach to grouping. *ECCV*, pages 655–670, 1998.

[116] B. Pogorelc and M. Gams. Discovery of gait anomalies from motion sensor data. *IEEE ICTAI*, pages 331–336, 2010.

[117] V. Popovici, W. Chen, B. G. Gallas, and C. Hatzis. Effect of training-sample size and classification difficulty on the accuracy of genomic predictors. *Breast Cancer Res*, 12(1):R5, 2010.

[118] D. M. W. Powers. Evaluation: From precision, recall and f-measure to roc., informedness, markedness and correlation. *Journal of Machine Learning Technologies*, 2(1):37–63, 2011.

[119] M. L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*, volume 414. John Wiley & Sons, 2009.

[120] H. Qiu and E. R. Hancock. Clustering and embedding using commute times. *IEEE TPAMI*, 29(11):1873–1890, 2007.

[121] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2007.

[122] J. W. Richards, P. E. Freeman, A. B. Lee, and C. M. Schafer. Accurate parameter estimation for star formation history in galaxies using sdss spectra. *MNRAS*, pages 1044–1057, 2009.

[123] Y. Saeys, I. Inza, and P. Larranaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, 2007.

[124] M. Seeland, A. Karwath, and S. Kramer. A structural cluster kernel for learning on graphs. *SIGKDD*, pages 516–524, 2012.

[125] J. R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. *Technical Report CMU-CS-TR-94-125*, 1994.

[126] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE TPAMI*, 22(8):888–905, 2000.

[127] A. Singer and R. R. Coifman. Non-linear independent component analysis with diffusion maps. *Applied and Computational Harmonic Analysis*, 25(2):226–239, 2008.

[128] A. Singer and R. R. Coifman. Anisotropic diffusion on sub-manifolds with application to earth structure classification. *Technical Report*, 2011.

[129] Y. Song, W. Chen, H. Bai, C. Jin, and E. Y. Chang. Parallel spectral clustering. In *Machine Learning and Knowledge Discovery in Databases*, pages 374–389. Springer, 2008.

[130] A. Statnikov, C. F. Aliferis, I. Tsamardinos, D. Hardin, and S. Levy. Gene expression model selector. *http://www.gems-system.org/*, 2005.

[131] M. Steinbach, P. Tan, V. Kumar, S. Klooster, and C. Potter. Discovery of climate indices using clustering. *KDD*, pages 446–455, 2003.

[132] A. Strehl and J. Ghosh. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, pages 583–617, 2003.

[133] J. Sun, M. Ovsjanikov, and L. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *Computer Graphics Forum*, volume 28, pages 1383–1392. Wiley Online Library, 2009.

[134] Z. Syed and I. Rubinfeld. Unsupervised risk stratification in clinical datasets: Identifying patients at risk of rare outcomes. *ICML*, pages 1023–1030, 2010.

[135] G. Taubin. A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 351–358. ACM, 1995.

[136] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[137] N. D. Thang, Y. K. Lee, and S. Lee. Deflation-based power iteration clustering. *Applied intelligence*, 39(2):367–385, 2013.

[138] K. M. Ting, G. T. Zhou, F. T. Liu, and J. S. Tan. Mass estimation and its applications. *ACM KDD*, pages 989–998, 2010.

[139] W. C. Tjhi and L. Chen. Dual fuzzy-possibilistic coclustering for categorization of documents. *IEEE Fuzzy Systems*, 17(3):532–543, 2009.

[140] T. N. Tran, K. Drab, and M. Daszykowski. Revised dbscan algorithm to cluster data with dense adjacent clusters. *Chemom. Intell. Lab. Syst. 120*, pages 92–96, 2013.

[141] H. Valizadegan and R. Jin. Generalized maximum margin clustering and unsupervised kernel learning. 19:1417–1424, 2006.

[142] L. van der Maaten, E. Postma, and J. van der Herik. Dimensionality reduction: A comparative review. *Technical report*, 2009.

[143] D. Verma and M. Meila. Comparison of spectral clustering methods. *Advances in Neural Information Processing Systems*, 15:38, 2003.

[144] L. Wang, C. Leckie, K. Ramamohanarao, and J. Bezdek. Approximate spectral clustering. In *Advances in Knowledge Discovery and Data Mining*, pages 134–146. Springer, 2009.

[145] X. Wang, Y. Wang, and L. Wang. Improving fuzzy c-means clustering based on feature-weight learning. *Pattern Recognition Letters*, 25(10):1123–1132, 2004.

[146] F. L. Wauthier, N. Jolic, and M. I. Jordan. Active spectral clustering via iterative uncertainty reduction. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1339–1347. ACM, 2012.

[147] M. Weinstein. Strange bedfellows: Quantum mechanics and data mining. *Nuclear Physics B-Proceedings Supplements*, 199(1):74–84, 2010.

[148] I. Weiss. Noise-resistant invariants of curves. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(9):943–948, 1993.

[149] J. Weston, A. Elisseeff, B. Scholkopf, and M. Tipping. Use of the zero norm with linear models and kernel methods. *The Journal of Machine Learning Research*, 3:1439–1461, 2003.

[150] D. M. Witten and R. Tibshirani. A framework for feature selection in clustering. *Journal of the American Statistical Association*, 105(490), 2010.

[151] M. Wu and J. Ye. A small sphere and large margin approach for novelty detection using training data with outliers. *IEEE TPAMI*, 31(11):2088–2092, 2009.

[152] D. Yan, L. Huang, and M. I. Jordan. Fast approximate spectral clustering. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 907–916. ACM, 2009.

[153] P. Yang, Q. Zhu, and B. Huang. Spectral clustering with density sensitive similarity function. *Knowledge-Based Systems*, 24:621–628, 2011.

[154] Y. Yang, H. T. Shen, Z. Ma, Z. Huang, and X. Zhou. l2,1-norm regularized discriminative feature selection for unsupervised learning. *IJCAI*, 2:1589–1594, 2011.

[155] Y. Yue, C. Wang, K. El-Arini, and C. Guestrin. Personalized collaborative clustering. In *Proceedings of the 23rd international conference on World wide web*, pages 75–84. International World Wide Web Conferences Steering Committee, 2014.

[156] L. Zelnik-manor and P. Perona. Self-tuning spectral clustering. *Advances in Neural Information Processing Systems*, 17:1601–1608, 2004.

[157] H. Zha, X. He, C. Ding, H. D. Simon, and M. Gu. Spectral relaxation for k-means clustering. *In Advances in neural information processing systems*, pages 1057–1064, 2001.

[158] K. Zhang, M. Hutter, and H. Jin. A new local distance-based outlier detection approach for scattered real-world data. *Advances in Knowledge Discovery and Data Mining*, pages 813–822, 2009.

[159] W. Zhang, X. Wang, D. Zhao, and X. Tang. Graph degree linkage: Agglomerative clustering on a directed graph. *ECCV Lecture Notes in Computer Science*, pages 428–441, 2012.

[160] X. Zhang, J. Li, and H. Yu. Local density adaptive similarity measurement for spectral clustering. *Pattern Recognition Letters*, 32:352–358, 2011.

[161] Z. Zhao and H. Liu. Spectral feature selection for supervised and unsupervised learning. *Proceedings of the 24th international conference on Machine learning*, pages 1151–1157, 2007.

[162] X. Zhu, Z. Huang, Y. Yang, H. T. Shen, C. Xu, and J. Luo. Self-taught dimensionality reduction on the high-dimensional small-sized data. *Pattern Recognition*, 46(1):215–229, 2013.

[163] X. Zhu, X. Wu, and C. Zhang. Vague one-class learning for data streams. *IEEE ICDM*, pages 657–666, 2009.

[164] D. W. Zimmerman. A note on interpretation of the paired-samples t test. *Journal of Educational and Behavioral Statistics*, 22(3):349–360, 1997.