MACHINE LEARNING METHODS BASED ON DIFFUSION PROCESSES

BY

CHENCHAO ZHAO

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Physics
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2018

Urbana, Illinois

Doctoral Committee:

Professor Sergei Maslov, Chair
Professor Jun S. Song, Director of Research
Professor Yann Robert Chemla
Assistant Professor Sihai Dave Zhao

Abstract

This thesis presents three distinct machine learning algorithms based on the mathematical formalism and physical idea of diffusion processes.

First, the idea of using heat diffusion on a hypersphere to measure similarity has been previously proposed and tested by computer scientists [37], demonstrating promising results based on a heuristic heat kernel obtained from the zeroth order parametrix expansion; however, how well this heuristic kernel agrees with the exact hyperspherical heat kernel remains unknown. This thesis presents a higher order parametrix expansion of the heat kernel on a unit hypersphere and discusses several problems associated with this expansion method. We then compare the heuristic kernel with an exact form of the heat kernel expressed in terms of a uniformly and absolutely convergent series in high-dimensional angular momentum eigenmodes. Being a natural measure of similarity between sample points dwelling on a hypersphere, the exact kernel often shows superior performance in kernel SVM classifications applied to text mining, tumor somatic mutation imputation, and stock market analysis. The improvement in classification accuracy compared with kernels based on Euclidean geometry may arise from ameliorating the curse of dimensionality on compact manifolds.

Second, the effective dissimilarity transformation (EDT) on empirical dissimilarity

hyperspheres is proposed and studied using synthetic and gene expression data sets. Iterating the EDT turns a static data distribution into a dynamical process purely driven by the empirical data set geometry and adaptively ameliorates the curse of dimensionality, partly through changing the topology of a Euclidean feature space $\mathbb{R}^n$ into a compact hypersphere S^n. The EDT often improves the performance of hierarchical clustering via the automatic grouping of information emerging from global interactions of data points. The EDT is not restricted to hierarchical clustering, and other learning methods based on pairwise dissimilarity should also benefit from the many desirable properties of EDT.

Finally, quantum time evolution exhibits rich physics, attributable to the interplay between the density and phase of a wave function. However, unlike classical heat diffusion, the wave nature of quantum mechanics has not yet been extensively explored in modern data analysis. We propose that the Laplace transform of quantum transport (QT) can be used to construct an ensemble of maps from a given complex network to a circle S^1, such that closely-related nodes on the network are grouped into sharply concentrated clusters on S^1. The resulting QT clustering (QTC) algorithm is as powerful as the state-of-the-art spectral clustering in discerning complex geometric patterns and more robust when clusters show strong density variations or heterogeneity in size. The observed phenomenon of QTC can be interpreted as a collective behavior of the microscopic nodes that evolve as macroscopic cluster "orbitals" in an effective tight-binding model recapitulating the network.

In summary, the three machine learning methods are based on three distinct diffusion processes. The dynamic diffusion processes serve as a promising foundation for future development in machine learning methods.

To my family.

Acknowledgements

I would like to thank my advisor Professor Jun S. Song who has been continuously encouraging me to push myself to unlock my potentials. This project would not have been possible without the help and comments from my friends and colleagues. The research was supported by a Distinguished Scientist Award from Sontag Foundation and the Grainger Engineering Breakthroughs Initiative.

Contents

1 Introduction **1**

2 Hyperspherical heat kernel and applications **6**

2.1 Introduction 6

2.2 The hyperspherical map 9

2.3 Laplacian on a Riemannian manifold 10

2.3.1 The induced metric on S^{n-1} 12

2.3.2 Geodesic polar coordinates 13

2.4 Parametrix expansion 14

2.5 Exact hyperspherical heat kernel 21

2.5.1 Euclidean heat kernel 21

2.5.2 Spherical Laplacian and its eigenfunctions 22

2.5.3 Generalization to S^{n-1} 23

2.5.4 Lemmas for the proof of convergence 29

2.5.5 The sweet spot of t 30

2.6 SVM classifications 31

2.6.1 VC dimension and effective sample size 38

2.7 Data preparation 40
2.8 Discussion 42

3 Effective dissimilarity transformation **44**
3.1 Introduction 44
3.2 Formulation of effective dissimilarity transformation (EDT) 45
3.3 Gedankenexperimente of EDT 49
3.4 Effective dissimilarity transformation 55
3.4.1 Perspective contraction 55
3.4.2 Cluster condensation 58
3.4.3 Local deformation 59
3.4.4 Global deformation 59
3.4.5 EDT and the curse of dimensionality 60
3.5 Application of EDT in two gene expression data sets 61
3.6 Data preparation 62
3.7 Discussion 63

4 Clustering via quantum time evolution **77**
4.1 Introduction 77
4.2 Schrödinger equation implies fluid dynamics 78
4.3 Graph Laplacians 80
4.4 Laplace transform of time evolution 81
4.5 Phase information of Laplace-transformed wave function 83
4.6 Spectrum of graph Laplacian reflects the number of clusters 88
4.7 Effective tight-binding model 88

4.8 Two-level toy model . . . 94
4.9 The algorithm . . . 100
4.9.1 Direct Extraction . . . 102
4.9.2 Consensus matrix . . . 104
4.10 Data preparation . . . 106
4.10.1 Synthetic data sets . . . 106
4.10.2 Time series stock price data . . . 106
4.10.3 Genomic data . . . 108
4.11 Comparison with other methods . . . 109
4.11.1 Spectral embedding . . . 109
4.11.2 Time-averaged transition amplitude . . . 115
4.11.3 Density information of Laplace-transformed wave functions . . . 119
4.11.4 Jensen-Shannon divergence of density operators . . . 120
4.12 Discussion . . . 121

5 Conclusions **123**

Bibliography **125**

A Multidimensional scaling **136**
A.1 Multidimensional scaling . . . 136
A.2 Visualization of EDT iterations . . . 137
A.3 Visualization of similarity matrix . . . 139

Chapter 1

Introduction

As the techniques for analyzing large data sets continue to grow, diverse quantitative sciences – including computational biology, observation astronomy, and high energy physics – are becoming increasingly data driven. Moreover, modern business decision making critically depends on quantitative analyses such as community detection and consumer behavior prediction. Consequently, statistical learning has become an indispensable tool for modern data analysis. Data acquired from various experiments are usually organized into an $m \times n$ matrix, where m samples are represented as feature vectors in $\mathbb{R}^n$. The simplest distribution in Euclidean space given mean and covariance is the multivariate Gaussian distribution; many learning algorithms, such as Gaussian mixture model, linear regression, and linear discriminant analysis [31], assume that data points are approximately normal. If the data samples are generated from a non-Euclidean space with intrinsic curvature, then Gaussian distribution is not a good approximation for data clouds scattered in a curved space. However, the Gaussian probability density function shares the same functional form as Euclidean heat kernel,

where the variance is linear in diffusion time t. As the heat kernel can be interpreted as a heat distribution at time t initialized from a point source, we can thus generalize the Gaussian distribution to non-Euclidean spaces using heat diffusion initialized at any point in the space. In Chapter 2, I will show that the heuristic hyperspherical heat kernel [38] is an unsatisfactory generalization of Gaussian distribution to a high-dimensional sphere, and then systematically develop the exact form of hyperspherical heat kernel using high-dimensional angular momentum eigenfunctions. Finally, the heat kernels will be tested in support vector machine (SVM) using three real-world data sets.

In the context of Riemannian geometry and Einstein's theory of gravity, the "geometry" is completely encoded in the metric tensor. Based on a similar idea, most statistical learning algorithms utilize a pairwise dissimilarity measure $d_{ij}^{(0)}$ that depends only on the (i, j)-pair of samples, The Euclidean ℓ_p-metric directly defined on the feature space $\mathbb{R}^n$ is among the most common pairwise dissimilarities. In high dimensions, however, the relative contrast between the farthest and nearest points measured by the ℓ_p-metric diminishes; consequently, the concept of nearest neighbors, which serves as the foundation for clustering, becomes increasingly ill-defined as the feature dimension increases [71, 9, 33]. This phenomenon is termed "the curse of dimensionality," analogous to the idea of "more is different" for many-body systems [3]. Modifications of Euclidean distances are found to improve the relative contrast for an artificial data cloud drawn from a single distribution [71, 9], but fail in data drawn from several distributions [33]. One way to address the loss of contrast in high dimensions for multi-distribution data is to introduce an effective dissimilarity measure calculated from the number of shared nearest neighbors of two data points, where each point is allowed to have a fixed num-

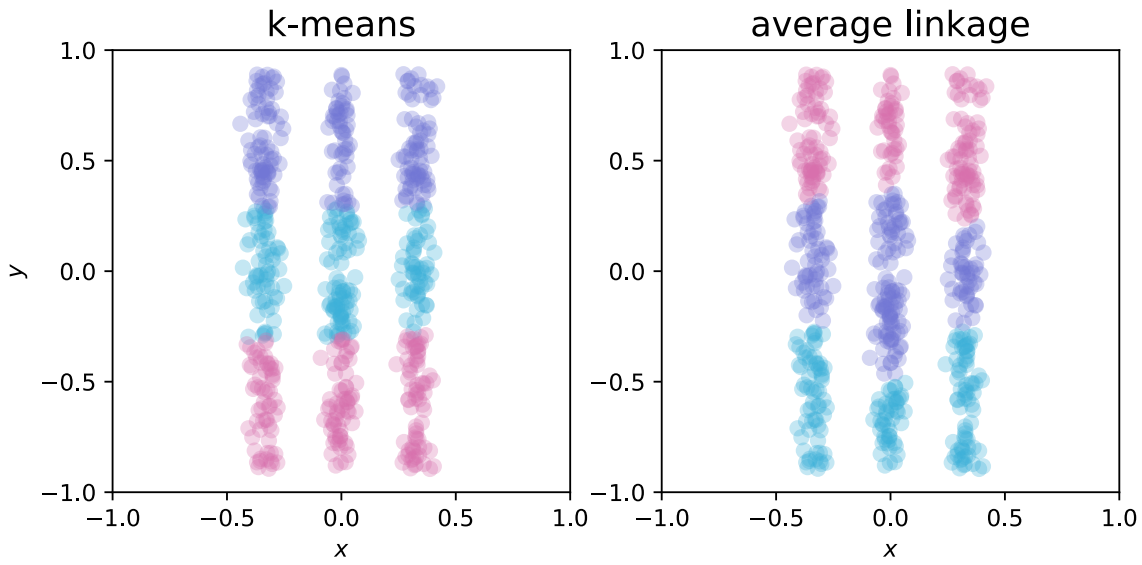

Figure 1.0.1: "Cheese-sticks" confused k-means and hierarchical clustering with average linkage.

ber of nearest neighbors [33]. The use of effective dissimilarity reduces the effect of high feature dimensions in subsequent computations; however, the choice of effective dissimilarity function actually dictates the improvement. In Chapter 3, I will introduce a transformation of dissimilarity matrix $d_{ij}^{(0)}$ which changes the geometry of data space and aggregates similar points. In this effective dissimilarity transformation (EDT), all data points in the primary feature space participate in redefining the effective dissimilarity between any two given data points, and thus, the effective similarity globally captures relations to all available sample points. Iteratively applying the transformation yields a sequence of EDT, where microscopic structures condense locally, while inter-cluster macroscopic distinctions become more evident. Thus the transformation turns a static distribution of points into a diffusion process and often amplifies the power of cluster separation in high dimensions.

Grouping similar objects into sets is a fundamental task in modern data science. So far, we have assumed that data points are embedded in some data space. However, many data sets, such as social networks, and gene regulation networks, cannot be automatically embedded in Euclidean space. Many clustering algorithms have thus been devised to automate the partitioning of samples into clusters, or communities, based on some similarity or dissimilarity measures between the samples that form nodes on a graph [35, 31]. In particular, physics-inspired approaches based on classical spin-spin interaction models [39, 50] and Schrödinger equation [32] have been previously proposed; however, the former usually requires computationally intensive Monte Carlo simulations which may get trapped in local optima, while the latter essentially amounts to Gaussian kernel density estimation. These intriguing physical ideas thus have been under the shadow of popular contemporary approaches that are simple and computationally efficient, such as the dissimilarity-based k-means [42, 23, 6] and hierarchical clustering [54, 17], density-based DBSCAN [19], distribution-based Gaussian mixture [67], and kernel-based spectral clustering [63]. In addition to high dimensionality, geometric complexity remains an outstanding challenge, e.g., in Fig. 1.0.1, the simple "cheese-sticks" confused k-means and hierarchical clustering with average linkage. In Chapter 4, I will demonstrate an efficient clustering algorithm which is especially robust against geometric complexity in data distribution. It is based on the physics of quantum walks – the quantum extension of classical random walks modeling discrete diffusion processes; the clustering information is contained in the phase information of wave functions at the data points. The performance of quantum transport clustering (QTC) is comparable to the state-of-the-art spectral clustering when the clusters exhibit non-spherical, geometrically complex shapes; at the same time, QTC is less sensitive to the

choice of parameters in the measure of adjacency or similarity. Moreover, unlike spectral clustering, the QTC representation of data on a circle does not jump in dimension when the specified number of clusters changes. Python source code implementing the algorithm and examples are available at https://github.com/jssong-lab/QTC.

In summary, the three machine learning methods are based on three distinct diffusion processes: heat diffusion on a hypersphere, evolution of pairwise distances, and quantum transport respectively. The dynamic diffusion processes often trace out hidden structures in the data sets and improve the performance machine learning algorithms.

Chapter 2

Hyperspherical heat kernel and applications

2.1 Introduction

Lafferty and Lebanon proposed a multinomial interpretation of non-negative feature vectors and an accompanying transformation of the multinomial simplex to a hypersphere, demonstrating that using the heat kernel on this hypersphere may improve the performance of kernel support vector machine (SVM) [37, 31, 20, 11, 15]. Despite the interest that this idea has attracted, only approximate heat kernel is known to date. We here present an exact form of the heat kernel on a hypersphere of arbitrary dimension and study its performance in kernel SVM classifications of text mining, genomic, and stock price data sets.

To date, sparse data clouds have been extensively analyzed in the flat Euclidean space endowed with the ℓ_2-norm using traditional statistical learning algorithms, in-

cluding k-means, hierarchical clustering, SVM, and neural network [31, 35, 20, 11, 24, 28]; however, the flat geometry of the Euclidean space often poses severe challenges in clustering and classification problems when the data clouds take non-trivial geometric shapes or class labels are spatially mixed. Manifold learning and kernel-based embedding methods attempt to address these challenges by estimating the intrinsic geometry of a putative submanifold from which the data points were sampled and by embedding the data into an abstract Hilbert space using a nonlinear map implicitly induced by the chosen kernel, respectively [8, 5, 49]. The geometry of these curved spaces may then provide novel information about the structure and organization of original data points.

Heat equation on the data submanifold or transformed feature space offers an especially attractive idea of measuring similarity between data points by using the physical model of diffusion of relatedness ("heat") on curved space, where the diffusion process is driven by the intrinsic geometry of the underlying space. Even though such diffusion process has been successfully approximated as a discrete-time, discrete-space random walk on complex networks, its continuous formulation is rarely analytically solvable and usually requires complicated asymptotic expansion techniques from differential geometry [10]. An analytic solution, if available, would thus provide a valuable opportunity for comparing its performance with approximate asymptotic solutions and rigorously testing the power of heat diffusion for geometric data analysis.

Given that a Riemannian manifold of dimension d is locally homeomorphic to $\mathbb{R}^d$, and that the heat kernel is a solution to the heat equation with a point source initial condition, one may assume in the short diffusion time limit ($t \downarrow 0$) that most of the heat is localized within the vicinity of the initial point and that the heat kernel on a

Riemannian manifold locally resembles the Euclidean heat kernel. This idea forms the motivation behind the parametrix expansion, where the heat kernel in curved space is approximated as a product of the Euclidean heat kernel in normal coordinates and an asymptotic series involving the diffusion time and normal coordinates. In particular, for a unit hypersphere, the parametrix expansion in the limit $t \downarrow 0$ involves a modified Euclidean heat kernel with the Euclidean distance $\|\mathbf{x}\|$ replaced by the geodesic arc length θ. Computing this parametrix expansion is, however, technically challenging; even when the computation is tractable, applying the approximation directly to high-dimensional clustering and classification problems may have limitations. For example, in order to be able to group samples robustly, one needs the diffusion time t to be not too small; otherwise, the sample relatedness may be highly localized and decay too fast away from each sample. Moreover, the leading order term in the asymptotic series is an increasing function of θ and diverges as θ approaches π, yielding an incorrect conclusion that two antipodal points are highly similar. For these reasons, the machine learning community has been using only the Euclidean diffusion term without the asymptotic series correction; how this resulting kernel, called the parametrix kernel [37], compares with the exact heat kernel on a hypersphere remains an outstanding question, which is addressed in this paper.

Analytically solving the diffusion equation on a Riemannian manifold is challenging [34, 61, 10]. Unlike the discrete analogues – such as spectral clustering [45] and diffusion map [14], where eigenvectors of a finite dimensional matrix can be easily obtained – the eigenfunctions of the Laplace operator on a Riemannian manifold are usually intractable. Fortunately, the high degree of symmetry of a hypersphere allows the explicit construction of eigenfunctions, called hyperspherical harmonics, via the

projection of homogeneous polynomials [7, 64]. The exact heat kernel is then obtained as a convergent power series in these eigenfunctions. Then we compare the analytic behavior of this exact heat kernel with that of the parametrix kernel and analyze their performance in classification. This chapter is based on [70].

2.2 The hyperspherical map

The heat kernel is the fundamental solution to the heat equation $(\partial_t - \Delta_x)u(x,t) = 0$ with an initial point source [57], where Δ_x is the Laplace operator; the amount of heat emanating from the source that has diffused to a neighborhood during time $t > 0$ is used to measure the similarity between the source and proximal points. The heat conduction depends on the geometry of feature space, and the main idea behind the application of hyperspherical geometry to data analysis relies on the following projective map from a non-negative feature space to a unit hypersphere: A hyperspherical projective map $\varphi : \mathbb{R}^n_{\geq 0} \setminus \{0\} \to S^{n-1}$ maps a vector $\mathbf{x}$, with $x_i \geq 0$ and $\sum_{i=1}^n x_i > 0$, to a unit vector $\hat{x} \in S^{n-1}$ where $(\hat{x})_i \equiv \sqrt{x_i / \sum_{j=1}^n x_j}$. We will henceforth denote the image of a feature vector $\mathbf{x}$ under the hyperspherical projective map as $\hat{x}$. The notion of neighborhood requires a well-defined measurement of distance on the hypersphere, which is naturally the great arc length – the geodesic on a hypersphere. Both parametrix approximation and exact solution employ the great arc length, which is related to the following definition of cosine similarity:

Definition 1. The generic cosine similarity between two feature vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n \setminus \{0\}$ is

$$\cos\theta \equiv \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \, \|\mathbf{y}\|},$$

where $\|\cdot\|$ is the Euclidean ℓ_2-norm, and $\theta \in [0, \pi]$ is the great arc length on S^{n-1}. For unit vectors $\hat{x} = \varphi(\mathbf{x})$ and $\hat{y} = \varphi(\mathbf{y})$ obtained from non-negative feature vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n_{\geq 0} \setminus \{0\}$ via the hyperspherical map, the cosine similarity reduces to the dot product $\cos\theta = \hat{x} \cdot \hat{y}$; the non-negativity of $\mathbf{x}$ and $\mathbf{y}$ guarantees that $\theta \in [0, \pi/2]$ in this case.

2.3 Laplacian on a Riemannian manifold

The Laplacian on a Riemannian manifold $\mathcal{M}$ with metric $g_{\mu\nu}$ is the operator

$$\Delta : C^\infty(\mathcal{M}) \to C^\infty(\mathcal{M})$$

defined as

$$\Delta \equiv \frac{1}{\sqrt{g}} \partial_\mu \left(\sqrt{g} g^{\mu\nu} \partial_\nu \right), \tag{2.3.1}$$

where $g = |\det g|$, and the Einstein summation convention is used. It can be also written in terms of the covariant derivative ∇_μ as

$$\Delta = g^{\mu\nu} \nabla_\mu \nabla_\nu. \tag{2.3.2}$$

The covariant derivative satisfies the following properties

$$\nabla_\mu f = \partial_\mu f, \quad f \in C^\infty(\mathcal{M})$$

$$\nabla_\mu V^\nu = \partial_\mu V^\nu + \Gamma^\nu_{\lambda\mu} V^\lambda, \quad V \in T_p\mathcal{M}$$

$$\nabla_\mu \omega_\nu = \partial_\mu \omega_\nu - \Gamma^\lambda_{\nu\mu}\omega_\lambda, \quad \omega \in T^*_p\mathcal{M},$$

where $\Gamma^\lambda_{\alpha\beta}$ is the Levi-Civita connection satisfying $\Gamma^\lambda_{\alpha\beta} = \Gamma^\lambda_{\beta\alpha}$ and $\nabla_\lambda g_{\mu\nu} = 0$.

To show Eq. 2.3.2, recall that the Levi-Civita connection is uniquely determined by the geometry, or the metric tensor, as

$$\Gamma^\lambda_{\alpha\beta} = \frac{1}{2} g^{\lambda\rho} \left(\partial_\alpha g_{\beta\rho} + \partial_\beta g_{\alpha\rho} - \partial_\rho g_{\alpha\beta} \right).$$

Using the formula for determinant differentiation

$$\left[\log \left(\det \mathbf{A} \right) \right]' = \mathrm{tr} \left(\mathbf{A}' \mathbf{A}^{-1} \right),$$

we can thus write

$$\Gamma^\lambda_{\lambda\mu} = \partial_\mu \log \sqrt{g}.$$

Hence, for any $f \in C^\infty(\mathcal{M})$,

$$\begin{aligned} g^{\mu\nu}\nabla_\mu\nabla_\nu f &= \nabla_\mu (g^{\mu\nu}\partial_\nu f) && (2.3.3) \\ &= \partial_\mu (g^{\mu\nu}\partial_\nu f) + \Gamma^\lambda_{\lambda\mu}(g^{\mu\nu}\partial_\nu f) && (2.3.4) \\ &= \partial_\mu (g^{\mu\nu}\partial_\nu f) + (\partial_\mu \log\sqrt{g})(g^{\mu\nu}\partial_\nu f) && (2.3.5) \\ &= \frac{1}{\sqrt{g}} \partial_\mu \left(\sqrt{g} g^{\mu\nu} \partial_\nu \right), && (2.3.6) \end{aligned}$$

proving the equivalence of Eq. 2.3.1 and Eq. 2.3.2.

2.3.1 The induced metric on S^{n-1}

The $(n-1)$-sphere embedded in $\mathbb{R}^n$ can be parameterized as

$$\begin{aligned} x_1 &= \cos\theta_1 \\ x_2 &= \sin\theta_1 \cos\theta_2 \\ x_3 &= \sin\theta_1 \sin\theta_2 \cos\theta_3 \\ &\vdots \\ x_{n-1} &= \sin\theta_1 \cdots \sin\theta_{n-2} \cos\theta_{n-1} \\ x_n &= \sin\theta_1 \cdots \sin\theta_{n-2} \sin\theta_{n-1}, \end{aligned} \tag{2.3.7}$$

where $0 \leq \theta_i \leq \pi$, for $i = 1, \ldots, n-2$, and $0 \leq \theta_{n-1} \leq 2\pi$. Let $\lambda := (\partial x_i / \partial \theta_j)$ denote the $n \times (n-1)$ Jacobian matrix for the above coordinate transformation.

The square of the line element in $\mathbb{R}^n$ is given by

$$ds_n^2 = \sum_{i=1}^{n} dx_i dx_i. \tag{2.3.8}$$

Restricted to S^{n-1},

$$dx_i = \sum_{j=1}^{n-1} \frac{\partial x_i}{\partial \theta_j} d\theta_j = \sum_{j=1}^{n-1} \lambda_{ij} d\theta_j. \tag{2.3.9}$$

Therefore, on S^{n-1}, we have

$$ds_{n-1}^2 = \sum_{i=1}^{n} \sum_{j,j'=1}^{n-1} \lambda_{ij} \lambda_{ij'} d\theta_j d\theta_{j'} \tag{2.3.10}$$

$$= \sum_{j,j'=1}^{n-1} \left(\sum_{i=1}^{n} \lambda_{ij} \lambda_{ij'} \right) d\theta_j d\theta_{j'} \,. \tag{2.3.11}$$

Hence, the induced metric on S^{n-1} embedded in $\mathbb{R}^n$ is

$$g_{\mu\nu} = \left(\lambda^T \lambda\right)_{\mu\nu}. \tag{2.3.12}$$

After some algebraic manipulations, it can be shown that the metric is in fact diagonal and its determinant takes the form

$$g = \sin^{2(n-2)}\theta_1 \sin^{2(n-3)}\theta_2 \cdots \sin^4\theta_{n-3} \sin^2\theta_{n-2}. \tag{2.3.13}$$

The geodesic arc length θ between $\hat{x}$ and $\hat{x}'$ on S^{n-1} is the angle given by

$$\theta \equiv \arccos \hat{x} \cdot \hat{x}' = \arccos \sum_{i=1}^{n} \hat{x}_i \hat{x}'_i. \tag{2.3.14}$$

2.3.2 Geodesic polar coordinates

In geodesic polar coordinates (r, ξ) around a point, one can show using Eq. 2.3.2 that the Laplacian on a d-dimensional Riemannian manifold $\mathcal{M}$ takes the form

$$\Delta = \partial_r^2 + (\partial_r \log\sqrt{g})\partial_r + \Delta_{S_r^{d-1}}, \tag{2.3.15}$$

where $\Delta_{S_r^{d-1}}$ is the Laplacian induced on the geodesic sphere S_r^{d-1} of radius r. If function f depends only on the geodesic distance r from the fixed point, then

$$\Delta f(r) = f''(r) + (\log\sqrt{g})' f'(r), \tag{2.3.16}$$

where $'$ denotes the radial derivative.

For the special case when $\mathcal{M}$ is S^{n-1}, the coordinates $\theta_1, \ldots, \theta_{n-1}$ described above correspond to the geodesic polar coordinates around the north pole, with $r = \theta_1$. From Eq. 2.3.13, we get

$$\log \sqrt{g(x)} = (n-2) \log \sin r + (n-3) \log \sin \theta_2 + \cdots \tag{2.3.17}$$

$$+ \log \sin \theta_{n-2}. \tag{2.3.18}$$

Note that only the first terms contributes to the radial derivative.

2.4 Parametrix expansion

The parametrix kernel K^{prx} previously used in the literature is just a Gaussian RBF function with $\theta = \arccos \hat{x} \cdot \hat{y}$ as the radial distance [37]: The parametrix kernel is a non-negative function

$$K^{\text{prx}}(\hat{x}, \hat{y}; t) = \mathrm{e}^{-\frac{\arccos^2 \hat{x} \cdot \hat{y}}{4t}} = \mathrm{e}^{-\frac{\theta^2}{4t}},$$

defined for $t > 0$ and attaining global maximum 1 at $\theta = 0$. The normalization factor $(4\pi t)^{-\frac{n-1}{2}}$ is numerically unstable as $t \downarrow 0$ and complicates hyperparameter tuning; as a global scaling factor of the kernel can be absorbed into the misclassification C-parameter in SVM, this overall normalization term is ignored in this paper. Importantly, the parametrix kernel K^{prx} is merely the Gaussian multiplicative factor without any asymptotic expansion terms in the full parametrix expansion G^{prx} of the heat kernel [37, 10], as described below.

The Laplace operator on manifold $\mathcal{M}$ equipped with a Riemannian metric $g_{\mu\nu}$ acts on a function f that depends only on the geodesic distance r from a fixed point as

described in Eq. 2.3.16. Due to the non-vanishing metric derivative in Eq. 2.3.16, the canonical diffusion function

$$G(r,t) = \left(\frac{1}{4\pi t}\right)^{\frac{d}{2}} \exp\left(-\frac{r^2}{4t}\right) \tag{2.4.1}$$

does not satisfy the heat equation; that is, $(\Delta - \partial_t)G(r,t) \neq 0$. For sufficiently small time t and geodesic distance r, the parametrix expansion of the heat kernel proposes an approximate solution

$$K_p(r,t) = G(r,t)\left(u_0(r) + u_1(r)t + u_2(r)t^2 + \cdots + u_p(r)t^p\right),$$

where the functions u_i should be found such that K_p satisfies the heat equation to order $t^{p-d/2}$, which is small for $t \ll 1$ and $p > d/2$; more precisely, we seek u_i such that

$$(\Delta - \partial_t)K_p = G\, t^p\, \Delta u_p. \tag{2.4.2}$$

Taking the time derivative of K_p yields

$$\partial_t K_p = G\cdot\left[\left(-\frac{d}{2t} + \frac{r^2}{4t^2}\right)\left(u_0 + u_1 t + u_2 t^2 + \cdots + u_p t^p\right) + \left(u_1 + 2u_2 t + \cdots + p u_p t^{p-1}\right)\right], \tag{2.4.3}$$

while the Laplacian of K_p is

$$\Delta K_p = \left(u_0 + u_1 t + \cdots + u_p t^p\right)\Delta G + G\Delta\left(u_0 + u_1 t + \cdots + u_p t^p\right) + 2G'\left(u_0 + u_1 t + \cdots + u_p t^p\right)'. \tag{2.4.4}$$

One can easily compute

$$\Delta G = \left[\left(-\frac{1}{2t} + \frac{r^2}{4t^2}\right) - \frac{r}{2t}(\log\sqrt{g})'\right] G \tag{2.4.5}$$

and

$$G'\left(u_0 + u_1 t + \cdots\right)' = -\frac{r}{2t}\left(u_0' + u_1' t + \cdots\right) G. \tag{2.4.6}$$

The left-hand side of Equation 2.4.2 is thus equal to G multiplied by

$$\begin{aligned}\left(u_0 + \cdots + u_p t^p\right)\left[-\frac{r}{2t}(\log\sqrt{g})' + \frac{d-1}{2t}\right] + \Delta\left(u_0 + \cdots + u_p t^p\right) + \\ -\frac{r}{t}\left(u_0' + \cdots + u_p' t^p\right) - \left(u_1 + 2u_2 t + \cdots + p u_p t^{p-1}\right),\end{aligned}$$

and we need to solve for u_i such that all the coefficients of t^q in this expression, for $q < p$, vanish.

For $q = -1$, we need to solve

$$u_0 \frac{r}{2}\left[-(\log\sqrt{g})' + \frac{d-1}{r}\right] = r u_0' \ , \tag{2.4.7}$$

or equivalently,

$$(\log u_0)' = -\frac{1}{2}(\log\sqrt{g})' + \frac{d-1}{2r}. \tag{2.4.8}$$

Integrating with respect to r yields

$$\log u_0 = -\frac{1}{2}\left[\log\sqrt{g} - (d-1)\log r\right] + \text{const.}, \tag{2.4.9}$$

where we implicitly take only the radial part of $\log\sqrt{g}$. Thus, we get

$$u_0 = \text{const.} \times \left(\frac{\sqrt{g}}{r^{d-1}}\right)^{-\frac{1}{2}} \propto \left(\frac{\sin r}{r}\right)^{-\frac{d-1}{2}} \tag{2.4.10}$$

as the zeroth-order term in the parametrix expansion. Using this expression of u_0, the remaining terms become

$$r\left[\left(u_1 + u_2 t + \cdots\right)(\log u_0)' - \left(u_1' + u_2' t + \cdots\right)\right] +$$

$$+ \left(\Delta u_0 + t\Delta u_1 + \cdots\right) - \left(u_1 + 2u_2 t + \cdots\right),$$

and we obtain the recursion relation

$$u_{k+1}(\log u_0)' - u_{k+1}' = -\frac{\Delta u_k - (k+1)u_{k+1}}{r}. \tag{2.4.11}$$

Algebraic manipulations show that

$$(\log r^{k+1} - \log u_0 + \log u_{k+1})' u_{k+1} = r^{-1}\Delta u_k\,, \tag{2.4.12}$$

from which we get

$$\left(\frac{u_{k+1} r^{k+1}}{u_0}\right)' = r^{(k+1)-1} u_0^{-1} \Delta u_k. \tag{2.4.13}$$

Integrating this equation and rearranging terms, we finally get

$$u_{k+1} = r^{-(k+1)} u_0 \int_0^r d\tilde{r}\; \tilde{r}^k u_0^{-1} \Delta u_k. \tag{2.4.14}$$

Setting $k = 0$ in this recursion equation, we find the second correction term to be

$$u_1 = \frac{u_0}{r} \int_0^r d\tilde{r}\, u_0^{-1} \Delta u_0 \tag{2.4.15}$$

$$= \frac{u_0}{r} \int_0^r d\tilde{r}\, u_0^{-1} \left(u_0'' + u_0' (\log \sqrt{g})' \right). \tag{2.4.16}$$

From our previously obtained solution for u_0, we find

$$u_0' = \frac{1}{2} \left(\frac{d-1}{r} - \frac{g'}{2g} \right) u_0. \tag{2.4.17}$$

and

$$u_0'' = \frac{1}{4} \left[\frac{(d-1)(d-3)}{r^2} - \frac{g'(d-1)}{gr} - \frac{g''}{g} + \frac{5}{4} \left(\frac{g'}{g} \right)^2 \right] u_0. \tag{2.4.18}$$

Substituting these expressions into the recursion relation for u_1 yields

$$u_1 = \frac{u_0}{4r} \int_0^r dr \left[\frac{(d-1)(d-3)}{r^2} - \frac{g''}{g} + \frac{3}{4} \left(\frac{g'}{g} \right)^2 \right]. \tag{2.4.19}$$

For the hypersphere S^d, where $d \equiv n-1$ and $g = \text{const.} \times \sin^{2(d-1)} r$, we have

$$\frac{g'}{g} = \frac{2(d-1)}{\tan r} \tag{2.4.20}$$

and

$$\frac{g''}{g} = 2(d-1) \left(\frac{2d-3}{\tan^2 r} - 1 \right). \tag{2.4.21}$$

Thus,

$$\begin{aligned} u_1 &= \frac{u_0}{4r}\int_0^r d\tilde{r}\left[\frac{(d-1)(d-3)}{\tilde{r}^2} - (d-1)\left(\frac{d-3}{\tan^2\tilde{r}} - 2\right)\right] \\ &= \frac{u_0(d-1)}{4r^2}\left[3 - d + (d-1)r^2 + (d-3)r\cot r\right]. \end{aligned} \tag{2.4.22}$$

Notice that $u_1(r) = 0$ when $d = 1$ and $u_1(r) = u_0(r)$ when $d = 3$. For $d = 2$, u_1/u_0 is an increasing function in r and diverges to ∞ at $r = \pi$. By contrast, for $d > 3$, u_1/u_0 is a decreasing function in r and diverges to $-\infty$ at $r = \pi$; u_1/u_0 is relatively constant for $r < \pi$ and starts to decrease rapidly only near π. Therefore, the first order correction is not able to remove the unphysical behavior near $r = 0$ in high dimensions where, according to the first order parametrix kernel, the surrounding area is hotter than the heat source.

Next, we apply Equation 2.4.14 again to obtain u_2 as

$$u_2 = \frac{u_0}{r^2}\int_0^r d\tilde{r}\,\tilde{r}u_0^{-1}\Delta u_1 \tag{2.4.23}$$

$$= \frac{u_0}{r^2}\int_0^r d\tilde{r}\,\tilde{r}u_0^{-1}\left(u_1'' + u_1'(\log\sqrt{g})'\right). \tag{2.4.24}$$

After some cumbersome algebraic manipulations, we find

$$\begin{aligned} \frac{u_2}{u_0} &= \frac{d-1}{32}\left[(d-3)^3 + \frac{(d-3)(d-5)(d-7)}{r^4} - \frac{(d-3)^2(d-5)}{r^3\tan r}\right. \\ &\quad \left. + \frac{2(d-1)^2(d-3)}{r\tan r} + \frac{(d+1)(d-3)(d-5)}{r^2\sin r}\right]. \end{aligned} \tag{2.4.25}$$

Again, $d = 1$ and $d = 3$ are special dimensions, where $u_2(r) = 0$ for $d = 1$, and

$u_2(r) = u_0/2$ for $d = 3$; for other dimensions, $u_2(r)$ is singular at both $r = 0$ and π. Note that on S^1, the metric in geodesic polar coordinate is $g_{11} = 1$, so all parametrix expansion coefficients $u_k(r)$ must vanish identically, as we have explicitly shown above.

Thus, the full G^{prx} defined on a hypersphere, where the geodesic distance r is just the arc length θ, suffers from numerous problems. The zeroth order correction term $u_0 = (\sin\theta/\theta)^{-\frac{n-2}{2}}$ diverges at $\theta = \pi$; this behavior is not a major problem if θ is restricted to the range $[0, \frac{\pi}{2}]$. Moreover, G^{prx} is also unphysical as $\theta \downarrow 0$ when $(n-2)t > 3$; this condition on dimension and time is obtained by expanding $\mathrm{e}^{-\theta^2/4t} = 1 - \frac{\theta^2}{4t} + \mathcal{O}(\theta^4)$ and $(\sin\theta/\theta)^{-\frac{n-2}{2}} = 1 + \frac{\theta^2}{12}(n-2) + \mathcal{O}(\theta^3)$, and noting that the leading order θ^2 term in the product of the two factors is a non-decreasing function of distance θ when $\frac{n-2}{12} \geq \frac{1}{4t}$, corresponding to the unphysical situation of nearby points being hotter than the heat source itself. As the feature dimension n is typically very large, the restriction $(n-2)t < 3$ implies that we need to take the diffusion time to be very small, thus making the similarity measure captured by G^{prx} decay too fast away from each data point for use in clustering applications. In this work, we further computed the first and second order correction terms, denoted u_1 and u_2 in Equation 2.4.22 and Equation 2.4.25, respectively.

In high dimensions, the divergence of u_1/u_0 and u_2/u_0 at $\theta = \pi$ is not a major problem, as we expect the expansion to be valid only in the vicinity $\theta \downarrow 0$; however, the divergence of u_2/u_0 at $\theta = 0$ (to $-\infty$ in high dimensions) is pathological, and thus, we truncate our approximation to $\mathcal{O}(t^2)$. Since $u_1(\theta)$ is not able to correct the unphysical behavior of the parametrix kernel near $\theta = 0$ in high dimensions, we conclude that the parametrix approximation fails in high dimensions. Hence, the only remaining part of G^{prx} still applicable to SVM classification is the Gaussian factor, which is clearly

not a heat kernel on the hypersphere. The failure of this perturbative expansion using the Euclidean heat kernel as a starting point suggests that diffusion in $\mathbb{R}^d$ and S^d are fundamentally different and that the exact hyperspherical heat kernel derived from a non-perturbative approach will likely yield better insights into the diffusion process.

2.5 Exact hyperspherical heat kernel

2.5.1 Euclidean heat kernel

Heat kernels in general are solutions to the heat equation

$$(\partial_t - \Delta)\,\phi = 0$$

with a point-source (Dirac delta) initial condition. The heat kernel in $\mathbb{R}^d$ is easily found to be

$$G(\mathbf{x}, \mathbf{y}; t) = \left(\frac{1}{4\pi t}\right)^{\frac{d}{2}} K(\mathbf{x}, \mathbf{y}; t) \tag{2.5.1}$$

where

$$K(\mathbf{x}, \mathbf{y}; t) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{4t}\right). \tag{2.5.2}$$

K is known as the Gaussian RBF kernel with parameter $\gamma = 1/4t$. $G(\mathbf{x}, \mathbf{y}; t)$ is the solution to the heat equation satisfying the initial condition $G(\mathbf{x}, \mathbf{y}; 0) = \delta(\mathbf{x} - \mathbf{y})$. Note that formally,

$$G(\mathbf{x}, \mathbf{y}; t) = \mathrm{e}^{t\Delta}\delta(\mathbf{x} - \mathbf{y});$$

using the Fourier transform representation of the right-hand side then yields the expression in Eq. 2.5.1.

We treat the hypersphere S^{n-1} as being embedded in $\mathbb{R}^n$ and use the induced metric on S^{n-1} to define the Laplacian. The Laplacian in $\mathbb{R}^n$ takes the usual form

$$\Delta = \frac{1}{r^{n-1}}\partial_r\left(r^{n-1}\partial_r\right) - \frac{\hat{L}^2}{r^2} \tag{2.5.3}$$

where the differential operator $\hat{L}^2$ depends only on the angular coordinates. $-\hat{L}^2$ is the spherical Laplacian operator. [64]

2.5.2 Spherical Laplacian and its eigenfunctions

For $n = 3$, the Laplacian on $\mathbb{R}^3$ is

$$\Delta = \frac{1}{r^2}\partial_r\left(r^2\partial_r\right) - \frac{\hat{L}^2}{r^2} \tag{2.5.4}$$

where $\hat{L}^2$ is the squared orbital angular momentum operator in quantum mechanics. Restricted to $r = 1$, the Laplacian reduces to the spherical Laplacian on S^2, which is exactly the operator $-\hat{L}^2$ whose eigenfunctions are the spherical harmonics $Y_{lm}(\theta, \phi)$ with eigenvalue $-\ell(\ell + 1)$. In this setting, $Y_{lm}(\theta, \phi)$ can be viewed as the angular component of homogeneous harmonic polynomials in $\mathbb{R}^3$, and this perspective will be used in the subsequent discussion of hyperspherical Laplacian. By convention, our spherical harmonics satisfy the normalization condition

$$\sum_{m=-\ell}^{\ell} |Y_{\ell m}(\theta, \phi)|^2 = \frac{2\ell + 1}{4\pi} \tag{2.5.5}$$

and the completeness condition

$$\sum_{\ell=0}^{\infty}\sum_{m=-\ell}^{\ell} Y_{\ell m}(\theta,\phi)Y^*_{\ell m}(\theta',\phi') = \delta(\cos\theta-\cos\theta')\delta(\phi-\phi'). \tag{2.5.6}$$

Analogous to the Euclidean case, applying the evolution operator $\exp(-\hat{L}^2 t)$ on the initial delta distribution yields the following eigenfunction expansion of the heat kernel on S^2:

$$G(\hat{x},\hat{y};t) = \sum_{l=0}^{\infty} \mathrm{e}^{-\ell(\ell+1)t} \sum_{m=-\ell}^{\ell} Y_{\ell m}(\hat{x})Y_{\ell m}(\hat{y})^*. \tag{2.5.7}$$

Applying the addition theorem of spherical harmonics,

$$\frac{4\pi}{2\ell+1}\sum_{m=-\ell}^{\ell} Y_{\ell m}(\hat{x})Y_{\ell m}(\hat{y})^* = P_\ell(\hat{x}\cdot\hat{y}), \tag{2.5.8}$$

we finally get

$$G(\hat{x}\cdot\hat{y};t) = \sum_{\ell=0}^{\infty}\left(\frac{2\ell+1}{4\pi}\right)\mathrm{e}^{-\ell(\ell+1)t}P_\ell(\hat{x}\cdot\hat{y}). \tag{2.5.9}$$

2.5.3 Generalization to S^{n-1}

Similar to the spherical harmonics, the hyperspherical harmonics arise as the angular part of degree-ℓ homogeneous harmonic polynomials h_ℓ that satisfy $\Delta h_\ell = 0$. In spherical coordinates (r,ξ), we can decompose $h_\ell(\mathbf{x}) = r^\ell \tilde{Y}_\ell(\xi)$ [7, 64], where $\tilde{Y}_\ell(\xi)$ is the desired hyperspherical harmonic. Using the spherical coordinate Laplacian in $\mathbb{R}^n$ shown in Eq. 2.5.3, we get

$$0 = \Delta h_\ell(\mathbf{x}) = \tilde{Y}_\ell(\hat{x})\frac{1}{r^{n-1}}\partial_r\left(r^{n-1}\partial_r r^\ell\right) - r^{\ell-2}\hat{L}^2\tilde{Y}_\ell(\xi), \tag{2.5.10}$$

which can be simplified to yield the following eigenvalue equation for the hyperspherical Laplacian:

$$\hat{L}^2 Y_{\ell\{m\}} = \ell(\ell + n - 2) Y_{\ell\{m\}}, \tag{2.5.11}$$

where the set $\{m\}$ indexes the degenerate eigenstates.

By definition, the exact heat kernel $G^{\text{ext}}(\hat{x}, \hat{y}; t)$ is the fundamental solution to heat equation $\partial_t u + \hat{L}^2 u = 0$ where $-\hat{L}^2$ is the hyperspherical Laplacian [57, 27, 34, 61]. In the language of operator theory, $G^{\text{ext}}(\hat{x}, \hat{y}; t)$ is an integral kernel, or Green's function, for the operator $\exp\{-\hat{L}^2 t\}$ and has an associated eigenfunction expansion. Because $\hat{L}^2$ and $\exp\{-\hat{L}^2 t\}$ share the same eigenfunctions, obtaining the eigenfunction expansion of $G^{\text{ext}}(\hat{x}, \hat{y}; t)$ amounts to solving for the complete basis of eigenfunctions of $\hat{L}^2$. The spectral decomposition of the Laplacian is in turn facilitated by embedding S^{n-1} in $\mathbb{R}^n$ and utilizing the global rotational symmetry of S^{n-1} in $\mathbb{R}^n$. The Euclidean space harmonic functions, which are the solutions to the Laplace equation $\nabla^2 u = 0$ in $\mathbb{R}^n$, can be projected to the unit hypersphere S^{n-1} through the usual separation of radial and angular variables [7, 64]. In this formalism, the hyperspherical Laplacian $-\hat{L}^2$ on S^{n-1} naturally arises as the angular part of the Euclidean Laplacian on $\mathbb{R}^n$, and $\hat{L}^2$ can be interpreted as the squared angular momentum operator in $\mathbb{R}^n$ [64].

The resulting eigenfunctions of $\hat{L}^2$ are known as the hyperspherical harmonics and generalize the usual spherical harmonics in $\mathbb{R}^3$ to higher dimensions. Each hyperspherical harmonic is equipped with a triplet of parameters or "quantum numbers" $(\ell, \{m_i\}, \alpha)$: the degree ℓ, magnetic quantum numbers $\{m_i\}$ and $\alpha = \frac{n}{2} - 1$. In the eigenfunction expansion of $\exp\{-\hat{L}^2 t\}$, we use the addition theorem of hyperspherical harmonics to sum over the magnetic quantum number $\{m_i\}$ and obtain the following main result:

Theorem 2. *The exact hyperspherical heat kernel* $G^{\text{ext}}(\hat{x}, \hat{y}; t)$ *can be expanded as a uniformly and absolutely convergent power series*

$$G^{\text{ext}}(\hat{x}, \hat{y}; t) = \sum_{\ell=0}^{\infty} \mathrm{e}^{-\ell(\ell+n-2)t} \frac{2\ell+n-2}{n-2} \frac{1}{A_{S^{n-1}}} C_\ell^{\frac{n}{2}-1}(\hat{x}\cdot\hat{y})$$

in the interval $\hat{x}\cdot\hat{y} \in [-1,1]$ *and for* $t > 0$*, where* $C_\ell^\alpha(w)$ *are the Gegenbauer polynomials and* $A_{S^{n-1}} = \frac{2\pi^{\frac{n}{2}}}{\Gamma\left(\frac{n}{2}\right)}$ *is the surface area of* S^{n-1}*. Since the kernel depends on* $\hat{x}$ *and* $\hat{y}$ *only through* $\hat{x}\cdot\hat{y}$*, we will write* $G^{\text{ext}}(\hat{x}, \hat{y}; t) = G^{\text{ext}}(\hat{x}\cdot\hat{y}; t)$*.*

Proof. We will obtain an eigenfunction expansion of the exact heat kernel by using the lemmas. The completeness of hyperspherical harmonics (Lemma 1) states that

$$\delta(\hat{x}, \hat{y}) = \sum_{\ell=0}^{\infty} \sum_{\{m\}} Y_{\ell\{m\}}(\hat{x}) Y^*_{\ell\{m\}}(\hat{y}). \tag{2.5.12}$$

Applying the addition theorem (Lemma 2) to Equation 2.5.12, we get

$$\delta(\hat{x}, \hat{y}) = \frac{1}{A_{S^{n-1}}} \sum_{\ell=0}^{\infty} \frac{2\ell+n-2}{n-2} C_\ell^{\frac{n}{2}-1}(\hat{x}\cdot\hat{y}).$$

Next, we apply time evolution operator $e^{-t\hat{L}^2}$ on this initial state to generate the heat kernel

$$G(\hat{x}\cdot\hat{y}; t) = \mathrm{e}^{-\hat{L}^2 t}\delta(\hat{x}, \hat{y}) \tag{2.5.13}$$

$$= \sum_{\ell=0}^{\infty} \mathrm{e}^{-\ell(\ell+n-2)t} \frac{2\ell+n-2}{n-2} \frac{1}{A_{S^{n-1}}} C_\ell^{\frac{n}{2}-1}(\hat{x}\cdot\hat{y}). \tag{2.5.14}$$

□

To show that it is a uniformly and absolutely convergent series for $t > 0$, note that

$$|G(w;t)| \leq \frac{1}{(n-2)A_{S^{n-1}}} \sum_{\ell=0}^{\infty} \mathrm{e}^{-\ell(\ell+n-2)t}(2\ell+n-2) \left| C_\ell^{\frac{n-2}{2}}(w) \right|,$$

where $w = \hat{x} \cdot \hat{y}$.

The terms involving Gegenbauer polynomials can be bounded by using Lemma 3 as

$$\begin{aligned}
\left| C_\ell^{\frac{n-2}{2}}(w) \right| &\leq \left[w^2 \frac{\Gamma(\ell+n-2)}{\Gamma(n-2)\Gamma(\ell+1)} + (1-w^2) \frac{\Gamma(\frac{\ell+n-2}{2})}{\Gamma(\frac{n-2}{2})\Gamma(\frac{\ell}{2}+1)} \right] \\
&= \left[\frac{\Gamma(\frac{\ell+n-2}{2})}{\Gamma(\frac{n-2}{2})\Gamma(\frac{\ell}{2}+1)} + \left(\frac{\Gamma(\ell+n-2)}{\Gamma(n-2)\Gamma(\ell+1)} - \frac{\Gamma(\frac{\ell+n-2}{2})}{\Gamma(\frac{n-2}{2})\Gamma(\frac{\ell}{2}+1)} \right) w^2 \right] \\
&\leq \frac{\Gamma(\frac{\ell+n-2}{2})}{\Gamma(\frac{n-2}{2})\Gamma(\frac{\ell}{2}+1)} + \left| \frac{\Gamma(\ell+n-2)}{\Gamma(n-2)\Gamma(\ell+1)} - \frac{\Gamma(\frac{\ell+n-2}{2})}{\Gamma(\frac{n-2}{2})\Gamma(\frac{\ell}{2}+1)} \right| w^2 \\
&\leq \frac{\Gamma(\frac{\ell+n-2}{2})}{\Gamma(\frac{n-2}{2})\Gamma(\frac{\ell}{2}+1)} + \left| \frac{\Gamma(\ell+n-2)}{\Gamma(n-2)\Gamma(\ell+1)} - \frac{\Gamma(\frac{\ell+n-2}{2})}{\Gamma(\frac{n-2}{2})\Gamma(\frac{\ell}{2}+1)} \right| \\
&\equiv M_\ell.
\end{aligned}$$

We thus have

$$\begin{aligned}
|G(w;t)| &\leq \frac{1}{(n-2)A_{S^{n-1}}} \sum_{\ell=0}^{\infty} \mathrm{e}^{-\ell(\ell+n-2)t}(2\ell+n-2) \left| C_\ell^{\frac{n-2}{2}}(w) \right| \\
&\leq \frac{1}{(n-2)A_{S^{n-1}}} \sum_{\ell=0}^{\infty} \mathrm{e}^{-\ell(\ell+n-2)t}(2\ell+n-2) M_\ell \\
&\equiv \frac{1}{(n-2)A_{S^{n-1}}} \sum_{\ell=0}^{\infty} Q_\ell.
\end{aligned}$$

But, in the large ℓ limit, we have

$$M_\ell \sim \frac{\Gamma(\ell+n-2)}{\Gamma(n-2)\Gamma(\ell+1)} \sim \frac{\ell^{n-3}}{(n-3)!}\,;$$

thus,

$$\lim_{\ell\to\infty} \frac{Q_{\ell+1}}{Q_\ell} = \lim_{\ell\to\infty} \frac{\mathrm{e}^{-(2\ell+n-1)t}(2\ell+n)M_{\ell+1}}{(2\ell+n-2)M_\ell} = 0 < 1,$$

for any $t > 0$. The sequence $\{Q_\ell\}$ is thus convergent, and hence, the Weiestrass M-test implies that the eigenfunction expansion of the heat kernel is uniformly and absolutely convergent in the indicated intervals. Q.E.D.

As before, we will rescale the kernel by self-similarity and define: The exact kernel $K^{\mathrm{ext}}(\hat{x},\hat{y};t)$ is the exact heat kernel normalized by self-similarity:

$$K^{\mathrm{ext}}(\hat{x},\hat{y};t) = \frac{G^{\mathrm{ext}}(\hat{x}\cdot\hat{y};t)}{G^{\mathrm{ext}}(1;t)},$$

which is defined for $t > 0$, is non-negative, and attains global maximum 1 at $\hat{x}\cdot\hat{y} = 1$.

Note that unlike $K^{\mathrm{prx}}(\hat{x},\hat{y};t)$, $K^{\mathrm{ext}}(\hat{x},\hat{y};t)$ explicitly depends on the feature dimension n. In general, SVM kernel hyperparameter tuning can be computationally costly for a data set with both high feature dimension and large sample size. In particular, choosing an appropriate diffusion time scale is an important challenge. On the one hand, choosing a very large value of t will make the series converge rapidly; but, then, all points will become uniformly similar, and the kernel will not be very useful. On the other hand, a too small value of t will make most data pairs too dissimilar, again limiting the applicability of the kernel. In practice, we thus need a guideline for a finite time scale at which the degree of "self-relatedness" is not singular, but still larger than

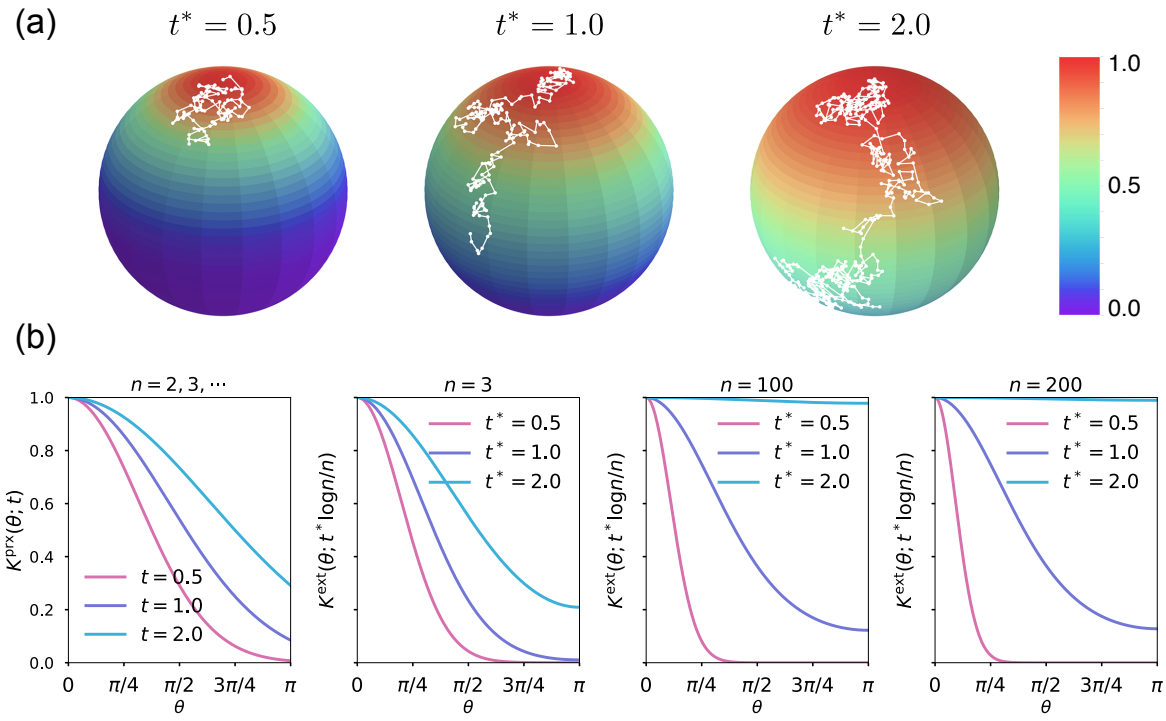

Figure 2.5.1: (A) Color maps of the exact kernel K^{ext} on S^2 at rescaled time $t^* = 0.5, 1.0, 2.0$; the white paths are simulated random walks on S^2 with the Monte Carlo time approximately equal to $t = t^* \log 3/3$. (B) Plots of the parametrix kernel K^{prx} and exact kernel K^{ext} on S^{n-1}, for $n = 3, 100, 200$, as functions of the geodesic distance.

the "relatedness" averaged over the whole hypersphere. Examining the asymptotic behavior of the exact heat kernel in high feature dimension n shows that an appropriate time scale is $t \sim \mathcal{O}(\log n/n)$; in this regime the numerical sum in Theorem 2 satisfies a stopping condition at low orders in ℓ and the sample points are in moderate diffusion proximity to each other so that they can be accurately classified.

Figure 2.5.1A illustrates the diffusion process captured by our exact kernel $K^{\mathrm{ext}}(\hat{x}, \hat{y}; t)$ in three feature dimensions at time $t = t^* \log 3/3$, for $t^* = 0.5, 1.0, 2.0$. In Figure 2.5.1B, we systematically compared the behavior of (1) dimension-independent parametrix kernel K^{prx} at time $t = 0.5, 1.0, 2.0$ and (2) exact kernel K^{ext} on S^{n-1} at $t = t^* \log n/n$ for

$t^* = 0.5, 1.0, 2.0$ and $n = 3, 100, 200$. By symmetry, the slope of K^{ext} vanished at the south pole $\theta = \pi$ for any time t and dimension n. In sharp contrast, K^{prx} had a negative slope at $\theta = \pi$, again highlighting a singular behavior of the parametrix kernel. The "relatedness" measured by K^{ext} at the sweet spot $t = \log n/n$ was finite over the whole hypersphere with sufficient contrast between nearby and far away points. Moreover, the characteristic behavior of K^{ext} at $t = \log n/n$ did not change significantly for different values of the feature dimension n, confirming that the optimal t for many classification applications will likely reside near the "sweet spot" $t = \log n/n$.

2.5.4 Lemmas for the proof of convergence

To construct the eigenfunction expansion of the exact heat kernel and prove its convergence, we need the following lemmas [7, 64, 43]:

Lemma 3. *The hyperspherical harmonics are complete on S^{n-1} and resolve the δ-function*

$$\delta(\hat{x}, \hat{y}) = \sum_{\ell=0}^{\infty} \sum_{\{m\}} Y_{\ell\{m\}}(\hat{x}) Y^*_{\ell\{m\}}(\hat{y}).$$

Lemma 4. *The hyperspherical harmonics satisfy the generalized addition theorem*

$$\sum_{\{m\}} Y_{\ell\{m\}}(\hat{x}) Y_{\ell\{m\}}(\hat{y})^* = \frac{1}{A_{S^{n-1}}} \frac{2\ell + n - 2}{n-2} C_\ell^{\frac{n}{2}-1}(\hat{x} \cdot \hat{y}),$$

where $C_\ell^\nu(w)$ are the Gegenbauer polynomials and $A_{S^{n-1}} = 2\pi^{n/2}/\Gamma\left(\frac{n}{2}\right)$ is the surface area of S^{n-1}.

Lemma 5. *The Gegenbauer polynomials $C_\ell^\alpha(w)$ with $\alpha > 0$ and $\ell \geq 0$ are bounded in the interval $w \in [-1, 1]$: in particular, $C_0^\alpha(w) = 1$, $C_1^\alpha(w) = \alpha w$, and thus, $|C_1^\alpha(w)| \leq \alpha$*

for $w \in [-1, 1]$. *Finally, for* $\ell \geq 2$,

$$|C_\ell^\alpha(w)| \leq \left[w^2 c_{2\ell,2\alpha} + (1-w^2)c_{\ell,\alpha}\right], \tag{2.5.15}$$

where

$$c_{\ell,\alpha} = \frac{\Gamma(\frac{\ell}{2}+\alpha)}{\Gamma(\alpha)\Gamma(\frac{\ell}{2}+1)}. \tag{2.5.16}$$

2.5.5 The sweet spot of t

Choosing an appropriate diffusion time t for the heat kernel is important for machine learning applications. Here, we use the degree of self-similarity measured by the heat kernel as a function of t, and propose a choice for which the self-similarity is neither too large nor too small. If t is too large, then the self-similarity is roughly the uniform similarity $1/A_{S^{n-1}}$, thereby losing contrast between neighbors and outliers. By contrast, as t approaches 0, the self-similarity becomes infinite, and the sense of neighborhood becomes too localized. We thus need an intermediate value of t, for which the self-similarity interpolates between the two limits.

The self-similarity is a special value of the heat kernel

$$\begin{aligned} G(1;t) &= \sum_{\ell=0}^{\infty} \mathrm{e}^{-\ell(\ell+n-2)t} \frac{2\ell+n-2}{n-2} \frac{1}{A_{S^{n-1}}} C_\ell^{\frac{n}{2}-1}(1) && (2.5.17) \\ &= \frac{1}{A_{S^{n-1}}} \sum_{\ell=0}^{\infty} \mathrm{e}^{-\ell(\ell+n-2)t} \frac{2\ell+n-2}{n-2} \frac{\Gamma(\ell+n-2)}{\Gamma(\ell+1)\Gamma(n-2)}. && (2.5.18) \end{aligned}$$

Because the series converges rapidly for sufficiently large t, we can truncate the series

at $\ell = \ell_{\max}$; i.e.

$$G(1;t) \approx \frac{1}{A_{S^{n-1}}} \sum_{\ell=0}^{\ell_{\max}} \mathrm{e}^{-\ell(\ell+n-2)t} \frac{2\ell+n-2}{n-2} \frac{\Gamma(\ell+n-2)}{\Gamma(\ell+1)\Gamma(n-2)}. \tag{2.5.19}$$

In the large n limit, we can bound the sum as

$$G(1;t) \leq \frac{1}{A_{S^{n-1}}} \sum_{\ell=0}^{\ell_{\max}} \left(\mathrm{e}^{-nt}\right)^{\ell} \frac{n^{\ell}}{\ell!} \leq \frac{\exp\left(n\mathrm{e}^{-nt}\right)}{A_{S^{n-1}}}. \tag{2.5.20}$$

To keep the self-similarity finite, but larger than the uniform similarity, suggests the choice for t of order $\log n/n$, at which the self-similarity is roughly $\mathrm{e}/A_{S^{n-1}}$. We thus search for an optimal value of t around $\log n/n$.

2.6 SVM classifications

We evaluated the performance of kernel SVM using the

1. linear kernel $K^{\mathrm{lin}}(\mathbf{x},\mathbf{y}) = \mathbf{x}\cdot\mathbf{y}$,
2. Gaussian RBF $K^{\mathrm{rbf}}(\mathbf{x},\mathbf{y};\gamma) = \exp\{-\gamma|\mathbf{x}-\mathbf{y}|^2\}$,
3. cosine kernel $K^{\cos}(\hat{x},\hat{y}) = \hat{x}\cdot\hat{y}$,
4. parametrix kernel $K^{\mathrm{prx}}(\hat{x},\hat{y};t)$, and
5. exact kernel $K^{\mathrm{ext}}(\hat{x},\hat{y};t)$,

on two independent data sets: (1) WebKB data of websites from four universities (WebKB-4-University) [16], and (2) glioblastoma multiforme (GBM) mutation data from The Cancer Genome Atlas (TCGA) with 5-fold cross-validations (CV).

m_r	lin	rbf	cos	prx	ext
100	74.2%	75.1%	84.4%	85.4%	**85.6%**
200	80.9%	82.0%	89.2%	89.6%	**89.9%**
300	83.2%	84.1%	89.9%	90.5%	**91.1%**
400	86.7%	86.1%	91.3%	91.7%	**92.3%**

Table 2.1: WebKB-4-University Document Classification. Performance test on four-class (*student*, *faculty*, *course*, and *project*) classification of WebKB-4-University word count data with different number m_r of representatives for each class, for $m_r = 100, 200, 300, 400$. The entries show the average of optimal 5-fold cross-validation mean accuracy scores of five runs. The exact kernel (ext) reduced the error of parametrix kernel (prx) by $1\% \sim 7\%$ and the Gaussian RBF (rbf) by $41\% \sim 45\%$; the cosine kernel (cos) also reduced the error of linear kernel (lin) by $34\% \sim 43\%$.

The WebKB-4-University data contained 4199 documents in total comprising four classes: student (1641), faculty (1124), course (930), and project (504); in our analysis, however, we selected an equal number of representative samples from each class, so that the training and testing sets had balanced classes. Table 2.1 shows the average optimal prediction accuracy scores of the five kernels for a varying number of representative samples, using 393 most frequent word features. The exact kernel outperformed the Gaussian RBF and parametrix kernel, reducing the error by $41\% \sim 45\%$ and by $1\% \sim 7\%$, respectively. Changing the feature dimension did not affect the performance much (Table 2.2).

In the TCGA-GBM data, there were 497 samples, and we aimed to impute the mutation status of one gene – i.e., mutant or wild-type – from the mutation counts of other genes. For each imputation target, we first counted the number m_r of mutant samples and then selected an equal number of wild-type samples for 5-fold CV. Imputation tests were performed for top 102 imputable genes.

Table 2.3 shows the average prediction accuracy scores for 5 biologically interesting genes known to be important for cancer [30]:

n	m_{r}	lin	rbf	cos	prx	ext
393	400	86.73%	86.27%	91.57%	91.99%	**92.44%**
726	400	86.78%	86.95%	92.62%	92.91%	**93.00%**
1023	400	85.56%	86.11%	92.62%	92.74%	**92.91%**
1312	400	85.78%	86.75%	92.56%	92.81%	**93.03%**

Table 2.2: WebKB-4-University Document Classification. Comparison of kernel SVMs on the WebKB-4-University data with a fixed sample size m_{r}, but varying feature dimension n. To account for the randomness in selecting the representative samples using k-means, we performed fives runs of representative selection, and then performed CV using the training and test sets obtained from each run. Finally, we averaged the five mean CV scores to assess the performance of each classifier on the imbalanced WebKB-4-University data set. The exact (ext) and cosine (cos) kernels outperformed the Gaussian RBF (rbf) and linear (lin) kernels in various feature dimensions $n = 393, 726, 1023$, and 1312, with fixed and balanced class size $m_{\mathrm{r}} = 400$. A word was selected as a feature if its total count was greater than 1/10, 1/20, 1/30 or 1/40 times the total number of web pages in the WebKB-4-University data set, with the different thresholds corresponding to the different rows in the table. The exact kernel reduced the errors of Gaussian RBF and parametrix kernels by $45 \sim 48\%$ and $1 \sim 6\%$, respectively; the cosine kernel reduced the errors of linear kernel by $36 \sim 49\%$.

1. *ZMYM4* ($m_{\mathrm{r}} = 33$) is implicated in an antiapoptotic activity [55, 52];

2. *ADGRB3* ($m_{\mathrm{r}} = 37$) is a brain-specific angiogenesis inhibitor [72, 36, 29];

3. *NFX1* ($m_{\mathrm{r}} = 42$) is a repressor of *hTERT* transcription [65] and is thought to regulate inflammatory response [56];

4. *P2RX7* ($m_{\mathrm{r}} = 48$) encodes an ATP receptor which plays a key role in restricting tumor growth and metastases [1, 26, 41];

5. *COL1A2* ($m_{\mathrm{r}} = 61$) is overexpressed in the medulloblastoma microenvironment and is a potential therapeutic target [4, 40, 51].

For the remaining genes, the exact kernel generally outperformed the linear, cosine and parametrix kernels (Fig. 2.6.1). However, even though the exact kernel dramatically

	lin	rbf	cos	prx	ext
ZMYM4	82.9%	84.0%	83.6%	84.1%	**85.1%**
ADGRB3	75.7%	**81.0%**	78.0%	79.5%	79.3%
NFX1	73.0%	81.2%	80.9%	**82.7%**	82.5%
P2RX7	79.2%	84.1%	**85.0%**	84.0%	**85.0%**
COL1A2	68.4%	70.5%	72.9%	73.9%	**74.2%**

Table 2.3: TCGA-GBM Genotype Imputation. Performance test on binary classification of *mutant* vs. *wild-type* in TCGA-GBM mutation count data. The rows are different genes, the mutation statuses of which were imputed using m_{r} samples in each mutant and wild-type class. The entries show the average of optimal 5-fold cross-validation mean accuracy scores of five runs.

outperformed the Gaussian RBF in the WebKB-4-University classification problem, the advantage of the exact kernel in this mutation analysis was not evident (Fig. 2.6.1). It is possible that the radial degree of freedom $\sum_{i=1}^{n} x_i$ in this case, corresponding to the genome-wide mutation load in each sample, contained important covariate information not captured by the hyperspherical heat kernel. The difference in accuracy between the hyperspherical kernels (cos, prx, and ext) and the Euclidean kernels (lin and rbf) also hinted some weak dependence on class size m_{r} (Fig. 2.6.1), or equivalently the sample size $m = 2m_{\mathrm{r}}$. In fact, the level of accuracy showed much stronger correlation with the "effective sample size" $\tilde{m}$ related to the empirical Vapnik-Chervonenkis (VC) dimension [60, 11, 28, 59, 48] of a kernel SVM classifier (Fig. 2.6.2a-e); moreover, the advantage of the exact kernel over the Gaussian RBF kernel grew with the effective sample size ratio $\tilde{m}_{\cos}/\tilde{m}_{\mathrm{lin}}$ (Fig. 2.6.2f).

By construction, our definition of the hyperspherical map exploits only the positive portion of the whole hypersphere, where the parametrix and exact heat kernels seem to have similar performances. However, if we allow the data set to assume negative values, i.e. the feature space is the usual $\mathbb{R}^n \backslash \{0\}$ instead of $\mathbb{R}^n_{\geq 0} \backslash \{0\}$, then we may apply

the usual projective map, where each vector in the Euclidean space is normalized by its L^2-norm. As shown in Figure 2.5.1B, the parametrix kernel is singular at $\theta = \pi$ and qualitatively deviates from the exact kernel for large values of θ. Thus, when data points populate the whole hypersphere, we expect to find more significant differences in performance between the exact and parametrix kernels. For example, Table 2.4 shows the kernel SVM classifications of 91 S&P500 *Financials* stocks against 64 *Information Technology* stocks ($m = 155$) using their log-return instances between January 5, 2015 and November 18, 2016 as features. As long as the number of features was greater than sample size, $n > m$, the exact kernel outperformed all other kernels and reduced the error of Gaussian RBF by $29 \sim 51\%$ and that of parametrix kernel by $17 \sim 51\%$.

n	m	lin	rbf	cos	prx	ext
475	155	98.06%	98.69%	98.69%	98.69%	**99.35%**
238	155	95.50%	96.77%	94.82%	96.13%	**98.06%**
159	155	94.86%	95.48%	95.48%	96.13%	**96.79%**
119	155	92.86%	93.53%	91.57%	**94.15%**	**94.15%**
95	155	91.55%	**95.50%**	94.19%	94.15%	94.79%

Table 2.4: S&P500 Stock Classification. Classifications were performed on $m = 155$ stocks from S&P500 companies: 91 *Financial* vs. 64 *Information Technology*. The 475 log-return instances between January 5, 2015 and November 18, 2016 were used as features. We uniformly subsampled the instances to generate variations in the feature dimension n. Here, we report the mean 5-fold CV accuracy score for each kernel. Although the two classes were slightly imbalanced, all scores were much larger than the "random score" $91/155 \approx 58.7\%$, calculated from the majority class size and sample size. For $n > m$, the exact (ext) kernel outperformed all other kernels and reduced the errors of Gaussian RBF (rbf) and parametrix (prx) kernels by $29 \sim 51\%$ and $17 \sim 51\%$, respectively. When $n < m$, the exact kernel started to lose its advantage over the Gaussian RBF kernel.

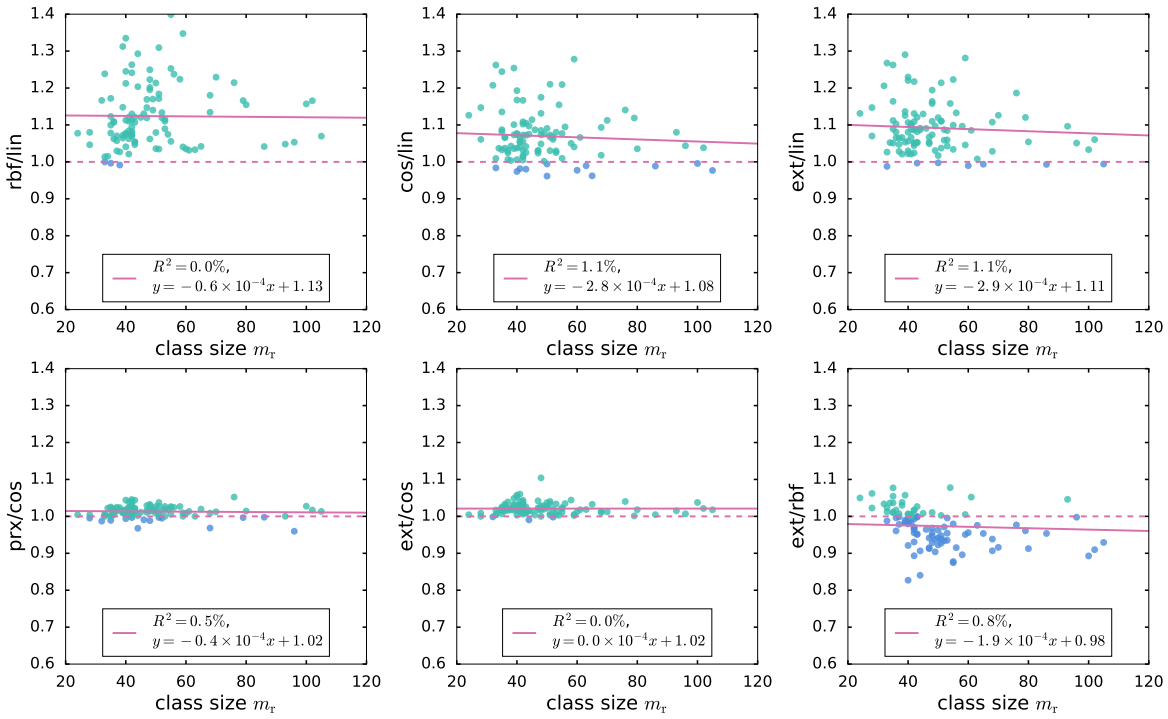

Figure 2.6.1: Comparison of the classification accuracy of SVM using linear (lin), cosine (cos), Gaussian RBF (rbf), parametrix (prx), and exact (ext) kernels on TCGA mutation count data. The plots show the ratio of accuracy scores for two different kernels. For visualization purpose, we excluded one gene with $m_r = 250$. The ratios rbf/lin, prx/cos, and ext/cos were essentially constant in class size m_r and greater than 1; in other words, the Gaussian RBF (rbf) kernel outperformed the linear (lin) kernel, while the exact (ext) and parametrix (prx) kernels outperformed the cosine (cos) kernel uniformly over all values of class size m_r. However, the more negative slope in the linear fit of cos/lin hints that the accuracy scores of cosine and linear kernels may depend on the class size m_r; the exact kernel also tended to outperform Gaussian RBF kernel when m_r was small.

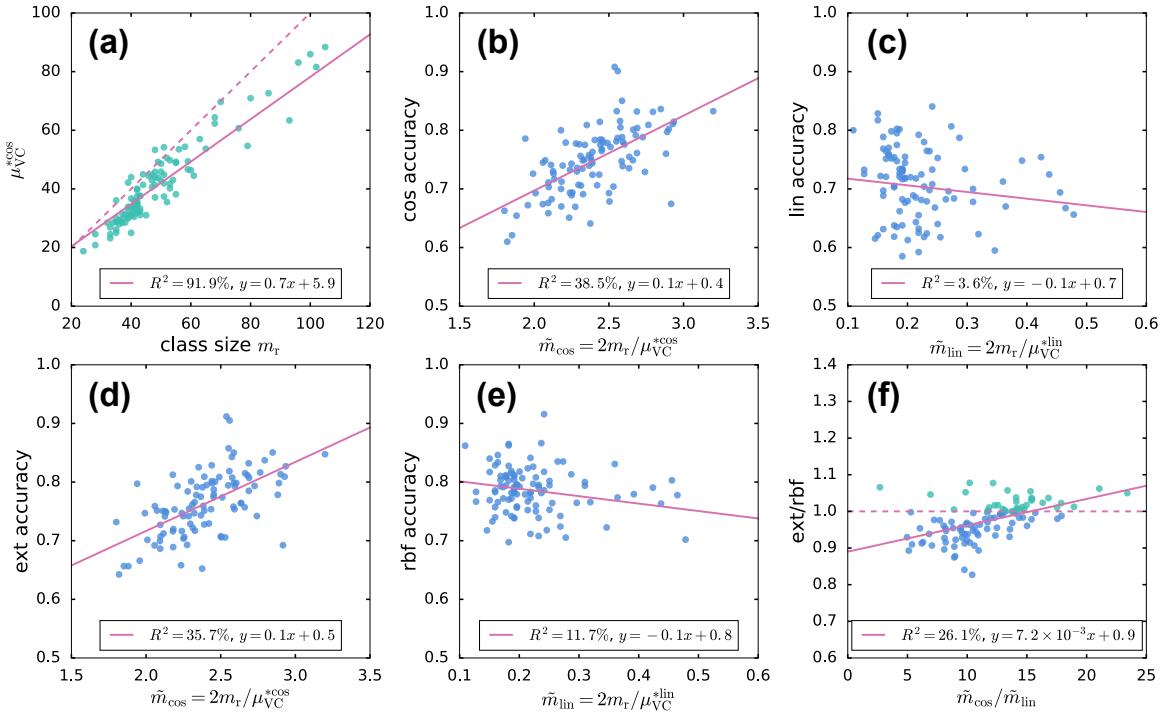

Figure 2.6.2: (A) A strong linear relation is seen between the VC-bound for cosine kernel $\mu^{*\cos}_{\text{VC}}$ and class size m_r. The dashed line marks $y = x$; the VC-bound for linear kernel, however, was a constant $\mu^{*\text{lin}}_{\text{VC}} = 439$. (B-E) The scatter plots of accuracy scores for cosine (cos), linear (lin), exact (ext), and Gaussian RBF (rbf) kernels vs. the effective sample size $\tilde{m} = 2m_r/\mu^*_{\text{VC}}$; the accuracy scores of exact and cosine kernels increased with the effective sample size, whereas those of Gaussian RBF and linear kernels tended to decrease with the effective sample size. (F) The ratio of ext vs. rbf accuracy scores is positively correlated with the ratio $\tilde{m}_{\cos}/\tilde{m}_{\text{lin}}$ of effective sample sizes.

2.6.1 VC dimension and effective sample size

We have applied the linear (lin), Gaussian RBF (rbf), cosine (cos), parametrix (prx), and exact (ext) kernels in SVM to (1) classify WebKB-4-University web pages into four classes: *student*, *faculty*, *course*, and *project*; and (2) impute the binary mutation status of genes in TCGA-GBM data. The kernel SVM classification results indicated that the cosine kernel usually outperformed the linear kernel, most likely as a pure consequence of the hyperspherical geometry, as we argue below. The exact kernel outperformed the Gaussian RBF kernel for the WebKB document data, but the advantage of exact kernel diminished in the TCGA mutation count data. Fig. 2.6.1 compares the accuracy of SVM using different kernels on the TCGA-GBM data, where the accuracy ratios rbf/lin, cos/lin, ext/lin, prx/cos, and ext/cos were greater than 1 for most class sizes m_{r}. Interestingly, the ratio cos/lin showed some dependence on the sample size m_{r}, and the exact kernel also tended to outperform the Gaussian RBF kernel when m_{r} was small; in general, we noted that the hyperspherical kernels tended to outperform the Euclidean kernels in small-sample-size classification problems. This pattern may be understood by examining the generalization error of kernel SVM as follows.

Intuitively, if a generic classifier were closely acquainted with the population distribution of data through a large sample size, then its predictions would be more generalizable to unseen samples. The "largeness" of sample size m, however, is not explicitly quantifiable unless we have a natural unit for it. Statistical learning theory [60, 59, 28] provides such a unit associated with a probabilistic upper bound on generalization errors. That is, with probability at least $1 - \eta$, the generalization error of a

binary SVM classification is bounded from above by

$$F(\tilde{m};\mu_{\rm VC},\eta)=\sqrt{\frac{1}{\tilde{m}}\left[(\log 2\tilde{m}+1)-\frac{\log\frac{\eta}{4}}{\mu_{\rm VC}}\right]}$$

where $\mu_{\rm VC}$ is the VC-dimension of the classifier, and $\tilde{m}=m/\mu_{\rm VC}$ is the effective sample size. The derivative of $F(\tilde{m};\mu_{\rm VC},\eta)$ with respect to $\tilde{m}$ is proportional to a positive factor times $-\log\left[(2\tilde{m})^{\mu_{\rm VC}}4/\eta\right]$. Thus, the upper bound decreases with $\tilde{m}$ when $(2\tilde{m})^{\mu_{\rm VC}}>\eta/4$, and increases otherwise; the critical effective sample size $\tilde{m}_{\rm crt}=\frac{1}{2}\cdot(\eta/4)^{1/\mu_{\rm VC}}\approx\frac{1}{2}$ for typical values of $\mu_{\rm VC}>100$ and $\eta\in[10^{-3},0.1]$. The VC dimension of a linear kernel SVM can be estimated using an empirical upper bound [59, 48]

$$\mu_{VC}\leq\mu^{*}_{VC}=\min\left\{n,\frac{R^2}{M^2}\right\}+1, \tag{2.6.1}$$

where n is the feature space dimension, R is the radius of the smallest ball in feature space that encloses all data points, and M is the SVM margin. We evaluated the bound $\mu^{*}_{\rm VC}$ for the TCGA-GBM mutation count data with $C=1$, and found that the linear kernel had $R^2/M^2\approx 6\times10^3$ and thus that $\mu^{*\rm lin}_{\rm VC}=n+1\approx 4\times10^2$. By contrast, the cosine kernel, which is a linear kernel in the hyperspherically transformed space with $R\leq 1$, had $\mu^{*\rm cos}_{\rm VC}$ approximately in the range $20\sim100\ll\mu^{*\rm lin}_{\rm VC}$, as shown in Fig. 2.6.2 (a). This reduction in the VC-dimension is likely responsible for the classification improvement of the cosine kernel over the linear kernel. We thus found that $\tilde{m}_{\rm cos}=2m_{\rm r}/\mu^{*\rm cos}_{\rm VC}>\tilde{m}_{\rm crt}$, while $\tilde{m}_{\rm lin}=2m_{\rm r}/\mu^{*\rm lin}_{\rm VC}<\tilde{m}_{\rm crt}$ for the TCGA-GBM data, and that the cosine kernel accuracy increased with effective sample size, whereas the linear kernel accuracy tended to decrease Fig. 2.6.2(b,c)), consistent with the analysis of the upper bound on generalization error $F(\tilde{m};\mu_{\rm VC},\eta)$. In addition, the Gaussian

RBF and exact kernels followed similar trends as the linear and cosine kernels, respectively (Fig. 2.6.2(d,e)). Similar to the cosine kernel, the exact kernel likely inherited the reduction in VC-dimension from the hyperspherical map; as a result, the accuracy of the exact kernel also increased with $\tilde{m}_{\cos}$, but with slightly higher accuracy due to the additional tunable parameter t that can adjust the curvature of nonlinear decision boundaries. Moreover, the cases of small sample size where the exact kernel outperformed the Gaussian RBF kernel corresponded to the cases of larger effective sample size ratio $\tilde{m}_{\cos}/\tilde{m}_{\text{lin}}$ (Fig. 2.6.2(f)).

2.7 Data preparation

The WebKB-4-University raw webpage data were downloaded from www.cs.cmu.edu and processed with the python packages Beautiful Soup and Natural Language Toolkit (NLTK). Our feature extraction excluded punctuation marks and included only letters and numerals where capital letters were all converted to lower case and each individual digit 0-9 was represented by a "#." Very infrequent words, such as misspelled words, non-English words, and words mixed with special characters, were filtered out. We selected top 393 most frequent words as features in our classification tests; the cutoff was chosen to select frequent words whose counts across all webpage documents are greater than 10% of the total number of documents. There were 4199 documents in total: student (1641), faculty (1124), course (930), and project (504).

The TCGA-GBM data were downloaded from the GDC Data Portal under the name TCGA-GBM Aggregated Somatic Mutation. The mutation count data set was extracted from the MAF file, while ignoring the detailed types of mutations and count-

ing only the total number of mutations in each gene. Very infrequently, mutated genes were filtered out if the total number of mutations in one gene across all samples is less than 10% of the total number of samples ($m = 497$ samples and $n = 439$ genes). We imputed the mutation status of one gene, mutant or wild-type, from the mutation counts of the remaining genes. The most imputable genes were selected using 5-fold cross-validation linear kernel SVM. Most of the mutant and wild-type samples were highly unbalanced, the ratio being typically around $1 : 9$; therefore, unthresholded area-under-the-curve (AUC) of the receiver operating characteristic (ROC) curve was used to quantify the classification performance of the linear kernel SVM. Mutated genes with AUC greater than 60% were selected for the subsequent imputation tests.

To balance the sample size between classes, we performed k-means clustering of samples within each class, with a specified number m_{r} of centroids and took the samples closest to each centroid as representatives. For the WebKB document classifications, we used $m_{\mathrm{r}} \leq \min\{m_{\mathrm{student}}, m_{\mathrm{faculty}}, m_{\mathrm{course}}, m_{\mathrm{project}}\}$, and k-means clustering was performed in each of the four classes separately; for the TCGA-GBM data, m_{r} was chosen to be the number of samples in each mutant (minority) class, and k-means clustering was performed in the wild-type (majority) class. Since k-means might depend on the random initialization, we performed the clustering 50 times and selected the top m_{r} most frequent representatives. Five-fold stratified cross-validations (CV) were performed on the resulting balanced data sets, where training and test samples were drawn without replacement from each class. The mean CV accuracy scores across the five folds were recorded.

2.8 Discussion

We have constructed the exact hyperspherical heat kernel using the complete basis of high-dimensional angular momentum eigenfunctions and tested its performance in kernel SVM. We have shown that the exact kernel and cosine kernel, both of which employ the hyperspherical projections, often outperform the Gaussian RBF and linear kernels. The advantage of using hyperspherical kernels likely arises from the hyperspherical projections of feature space, and the exact kernel may further improve the decision boundary flexibility of the raw cosine kernel. To be specific, the hyperspherical maps project out the less informative radial degree of freedom in a nonlinear fashion and compactify the Euclidean feature space into a unit hypersphere where all data points may then be enclosed within a finite radius. By contrast, our numerical estimations using TCGA-GBM data show that for linear kernel SVM, the margin M tends to be much smaller than the data range R in order to accommodate the separation of strongly mixed data points of different class labels; as a result, the ratio R/M was much larger than that for cosine kernel SVM. This insight may be summarized by the fact that the upper bound on the empirical VC-dimension of linear kernel SVM tends to be much larger than that for cosine kernel SVM, especially in high dimensions, suggesting that the cosine kernel SVM is less sensitive to noise and more generalizable to unseen data. The exact kernel is equipped with an additional tunable hyperparameter, namely the diffusion time t, which adjusts the curvature of nonlinear decision boundary and thus adds to the advantage of hyperspherical projections. Moreover, the hyperspherical kernels often have larger effective sample sizes than their Euclidean counterparts and, thus, may be especially useful for analyzing data with a small sample size in high feature dimensions.

The failure of the parametrix expansion of heat kernel, especially in dimensions $n \gg 3$, signals a dramatic difference between diffusion in a non-compact space and that on a compact manifold. It remains to be examined how these differences in diffusion process, random walk and topology between non-compact Euclidean spaces and compact manifolds like a hypersphere contribute to ameliorating the "curse of dimensionality," as hinted by the results of this paper.

Acknowledgments

I thank Alex Finnegan and Hu Jin for critical reading of the manuscript and helpful comments, and Mohith Manjunath for his help with the TCGA data.

Chapter 3

Effective dissimilarity transformation

3.1 Introduction

Community detection, better known as clustering in the literature of statistical learning [35, 31, 48, 62, 45, 14, 44], is a process of merging similar nodes of a complex network into communities (clusters) and often shows a hierarchical organization of communities at different levels of similarity. Akin to the idea of renormalization group in physics, decreasing the threshold for similarity leads to increasingly coarse-grained pictures of the "microscopic" network. The reduction in complexity can sometimes yield more interpretable statistical models that could serve as a basis for further classification analysis. Along this line, we present an idea of transforming dissimilarity measures to allow dynamic agglomeration of data points into communities. This chapter is based on [68].

3.2 Formulation of effective dissimilarity transformation (EDT)

As observed in previous support vector machine (SVM) classification studies [38, 70], hyperspherical geometry often improves classification accuracy. Motivated by these results, we now introduce an effective dissimilarity transformation based on a hyperspherical representation of data clouds. To map sample points onto a hypersphere, we will utilize the following hyperspherical transformation from non-negative space $\mathbb{R}^m \setminus \{0\}$ to a unit hypersphere:

Definition 6. A hyperspherical projective map $\varphi : \mathbb{R}^m_{\geq 0} \setminus \{0\} \to S^{m-1}$ maps a vector $\mathbf{x}$, with $x_i \geq 0$ and $\sum_{i=1}^m x_i > 0$, to a unit vector $\hat{x} \in S^{m-1}$ where $(\hat{x})_i \equiv \sqrt{x_i / \sum_{j=1}^m x_j}$.

A useful measure of similarity on a hypersphere is the cosine similarity:

Definition 7. For unit vectors $\hat{x} = \varphi(\mathbf{x})$ and $\hat{y} = \varphi(\mathbf{y})$ obtained from non-negative vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m_{\geq 0} \setminus \{0\}$ via the hyperspherical projective map, the cosine similarity is the dot product $\hat{x} \cdot \hat{y}$.

The EDT relies on this notion of cosine similarity, as explained below. Many algorithms – such as hierarchical clustering, k-medoids, and k-means – directly rely on some notion of difference between samples. For example, the Euclidean distance function is a popular measure of the difference between two sample points in $\mathbb{R}^n$. In statistical learning approaches based on pairwise differences, however, we often relax the definiteness condition and triangular inequality satisfied by a distance function and utilize instead a more general and flexible measure of difference, called the dissimilarity function:

Definition 8. A dissimilarity function defined on a manifold $\mathcal{M}$ is a map $d : \mathcal{M} \times \mathcal{M} \to \mathbb{R}_{\geq 0}$ satisfying

non-negativity: $d(x, y) \geq 0$ for all $x, y \in \mathcal{M}$,

identity: $d(x, x) = 0$ for all $x \in \mathcal{M}$,

symmetry: $d(x, y) = d(y, x)$ for all $x, y \in \mathcal{M}$.

Usually $\mathcal{M} = \mathbb{R}^n$, representing the sample space of original data directly collected from experiments, and its nonlinear embedding into an abstract manifold is often only implicitly defined through the dissimilarity function.

Dissimilarity functions are relatively easy to construct; in particular, we can turn the cosine similarity on $\mathbb{R}^n_{\geq 0} \backslash \{0\}$ into a dissimilarity function by defining $d(\mathbf{x}, \mathbf{y}) = 1 - \hat{x} \cdot \hat{y} = \frac{1}{2} \|\hat{x} - \hat{y}\|^2$. We here show that this cosine dissimilarity function can be iteratively applied to an initial dissimilarity measure and that this simple iteration leads to several robust properties desirable for clustering applications.

More precisely, given an initial dissimilarity function $d(\cdot, \cdot)$ and m sample points, organize the pairwise dissimilarity of the samples into an $m \times m$ non-negative, symmetric dissimilarity matrix $d^{(0)}$. To apply our method, we only need to assume the mild condition that each column of $d^{(0)}$ is not a zero vector. We then define the effective dissimilarity transformation on the space of such matrices as follows:

Definition 9. The effective dissimilarity transformation (EDT) $\psi : \mathbb{R}^{m \times m}_{\geq 0} \to \mathbb{R}^{m \times m}_{\geq 0}$ is defined as

$$\left[\psi(d^{(0)})\right]_{ij} = \frac{1}{2} \left\|\varphi(\mathbf{p}_i) - \varphi(\mathbf{p}_j)\right\|^2,$$

where $\mathbf{p}_i$ is the i-th column of the dissimilarity matrix $d^{(0)}$ and φ is the hyperspherical projective map into S^{m-1}. We denote $d^{(1)} \equiv \psi(d^{(0)})$.

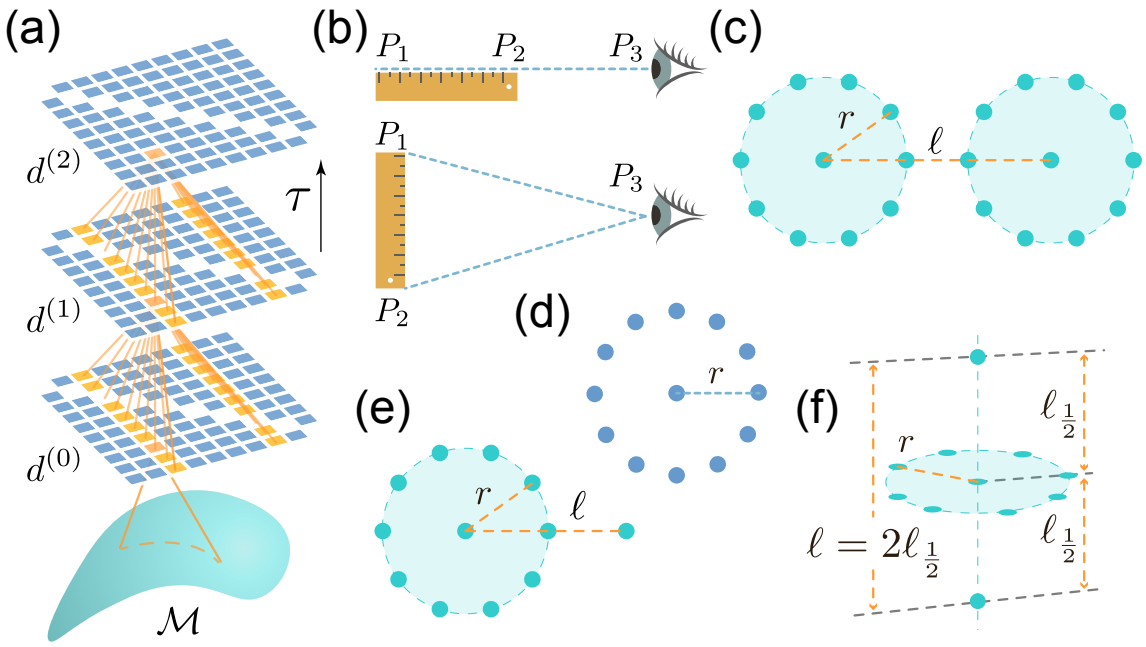

Figure 3.2.1: (a) A schematic illustration of the network structure of effective dissimilarity transformations (EDT) parameterized by τ. The (i, j)-th entry of $d^{(\tau)}$ arises from transforming the i- and j-th columns of $d^{(\tau-1)}$. (b) Illustrations of *perspective contraction* effect of EDT. (c) Two ideal clusters with radius r and centroid-centroid distance ℓ in $\mathbb{R}^2$. (d) The detector used in the measurement of *local deformation* of data distributions in $\mathbb{R}^2$. (e) An ideal cluster of radius r in $\mathbb{R}^2$ and an outlier at distance ℓ from the cluster centroid. (f) An ideal cluster of radius r in the xy-plane of $\mathbb{R}^3$ with symmetrically located outliers on the z-axis at distance $\ell_{\frac{1}{2}} = \frac{\ell}{2}$ from cluster centroid.

The resulting $d^{(1)}$ is thus a cosine dissimilarity matrix of the m samples newly represented by the columns of the dissimilarity matrix $d^{(0)}$. Importantly, the pairwise dissimilarity captured by $d^{(1)}$ between any two samples measures how dissimilar are their respective $d^{(0)}$ dissimilarities to all samples; in other words, each entry of $d^{(1)}$ depends on the global network structure encoded in $d^{(0)}$ as illustrated in Fig. 3.2.1(a). Iterating the map composition $\psi^{(\tau+1)} = \psi \circ \psi^{(\tau)}$ yields a sequence $\{\psi^{(\tau)}\}_{\tau=0}^{\infty}$ of EDTs and corresponding dissimilarity matrices $\{d^{(\tau)}\}_{\tau=0}^{\infty}$, where $\psi^{(0)}$ is the identity map and

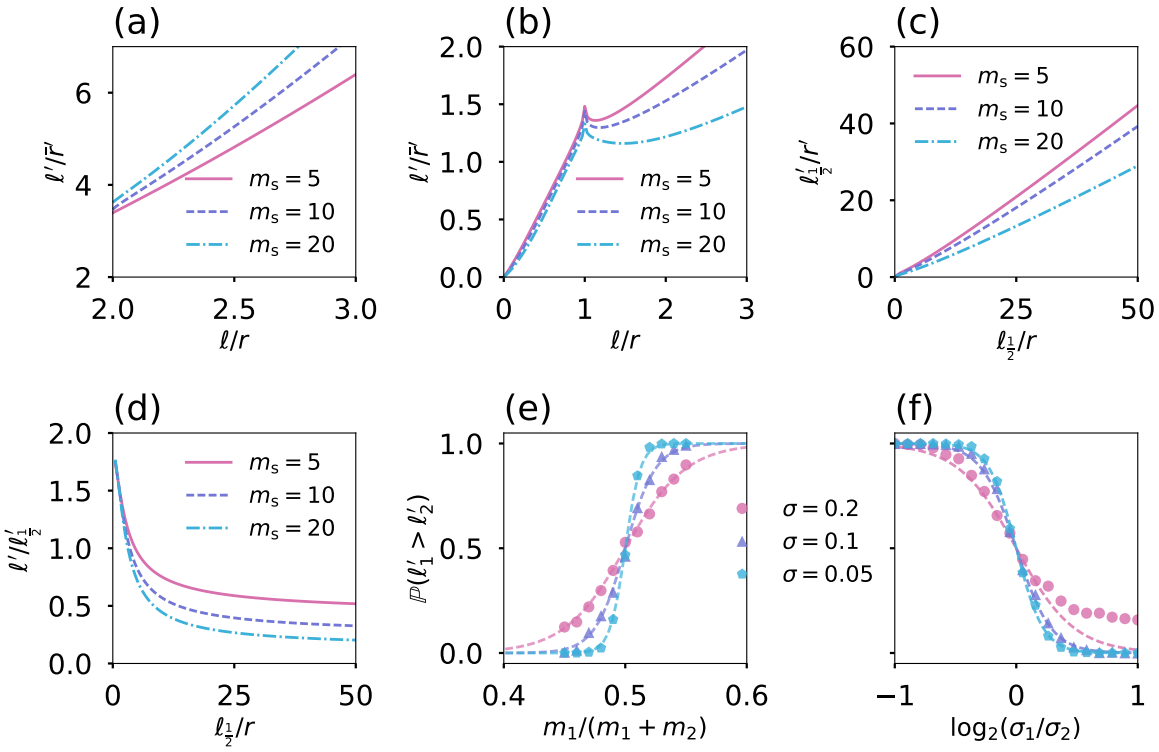

Figure 3.2.2: Results of Gedankenexperimente: (a) cluster condensation (Fig. 3.2.1(c)), (b) single outlier absorption (Fig. 3.2.1(e)), (c, d) two outliers perpendicular to an ideal cluster (Fig. 3.2.1(f)), (e, f) probabilistic sampling.

$d^{(\tau)} = \psi^{(\tau)}(d^{(0)})$. The sequence of dissimilarity matrices $\{d^{(\tau)}\}_{\tau=0}^{\infty}$ may be interpreted as inducing a data-driven evolution or flow of sample points parametrized by τ. We will show that the data-driven redefinition of dissimilarity resulting from an iterated application of EDT often leads to improved clustering results.

Even though EDT is simple in its definition and deterministic in nature, its nonlinearity makes the flow of data points difficult to study. Consequently, we first study the properties of EDT by performing Gedankenexperimente on carefully designed synthetic data sets shown in Fig. 3.2.1(b-f) (accompanying simulation results in Fig. 3.2.2(a-f)), and then test the power of these observed properties in the setting of real data sets.

3.3 Gedankenexperimente of EDT

First consider the simple data set consisting of 3 distinct points, P_1, P_2, and P_3, in $\mathbb{R}^n$, for any $n \geq 2$. Let P_1 and P_2 represent two ends of a ruler of length $d_{12}^{(0)} = a$, and let P_3 represent an observer at distance b to the center of the ruler; Fig. 3.2.1(b) shows two particular cases: (1) the ruler and observer are collinear, and $b > a/2$; (2) the observer and ruler form an isosceles triangle, and $d_{23}^{(0)} = d_{13}^{(0)} = c = \sqrt{\left(\frac{a}{2}\right)^2 + b^2}$. In scenario (1), the original distance $d_{12}^{(0)}$ between P_1 and P_2 is equal to the ruler length and is also the observed distance $d_{13}^{(0)} - d_{23}^{(0)}$ measured by the observer at P_3, irrespective of the location of P_3; after EDT, however, both $d_{12}^{(1)}$ and the ratio $(d_{13}^{(1)} - d_{23}^{(1)})/d_{12}^{(1)} = \sqrt{\frac{a}{2b}}$ shrink as the observer moves away (Section 3.4.1). That is, in the limit $b \gg a$, the effective dissimilarity between P_1 and P_2 approaches zero, and the observer at P_3 cannot distinguish between P_1 and P_2 on the scale set by $d_{12}^{(1)}$. In the language of hierarchical clustering, the single, average, and complete linkages become equivalent after EDT as P_3 becomes a clear outgroup. Similarly, in scenario (2), the effective ruler length also shrinks as the observer moves away from the other two points, i.e. $d_{12}^{(1)} = 1 - \frac{c}{a+c} \downarrow 0$ as $b/a \uparrow \infty$. We can thus summarize these properties as a *perspective contraction* effect:

Proposition 10. *The EDT dissimilarity between each pair of points shrinks as an observer moves away from the distribution of points. Consequently, compared to the original dissimilarity, hierarchical clustering using the EDT dissimilarity is insensitive to the choice of linkage.*

We verified this observation by comparing the performance of Euclidean distance with its EDT dissimilarity in the hierarchical clustering of three Gaussian clouds in $\mathbb{R}^2$ using single, average and complete linkages (Fig. 3.3.1). As often is the case with

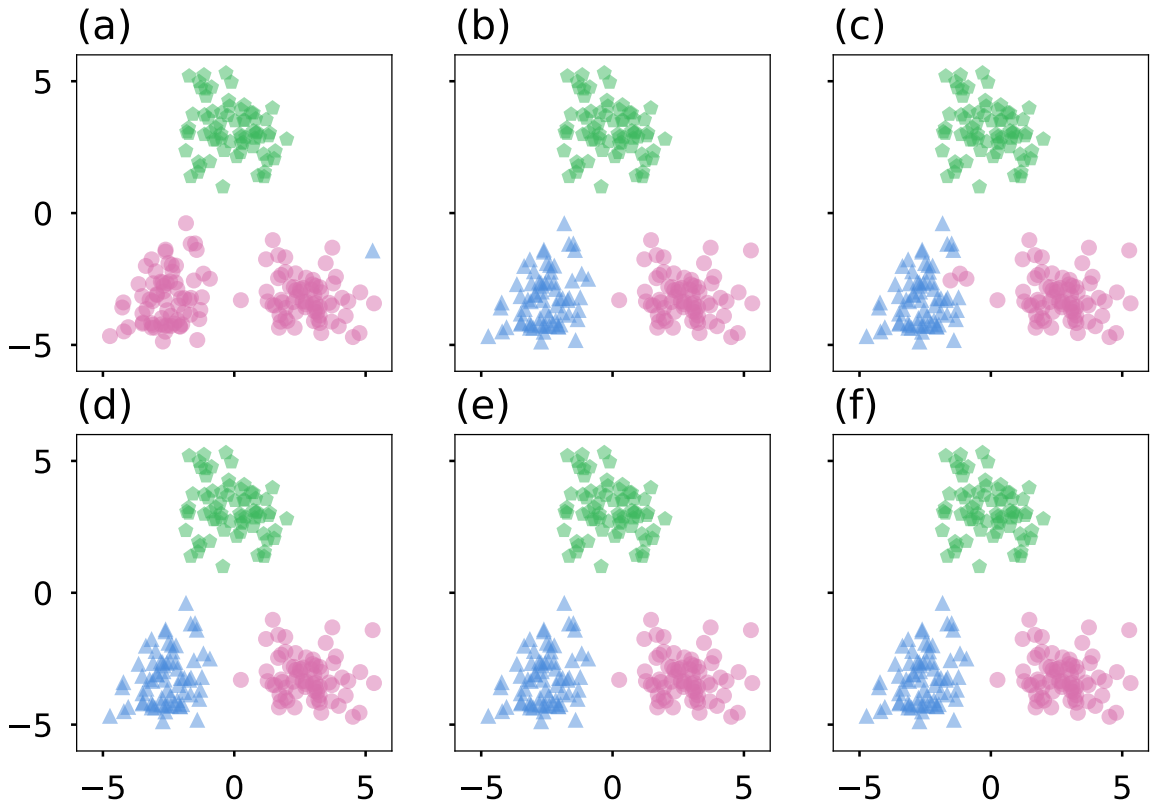

Figure 3.3.1: Comparison of hierarchical clustering results using (a-c) Euclidean distance vs. (d-f) EDT-enhanced Euclidean distance with (a,d) single, (b,e) average, and (c,f) complete linkages. The number of clusters was chosen to be three in the analysis.

real data, the three linkages based on the Euclidean distance led to different clustering results (Fig. 3.3.1 top row), whereas the EDT dissimilarity was insensitive to the choice of linkage (Fig. 3.3.1 bottom row).

We next replaced the ruler and observer in our first model with two identical ideal clusters, each of which consisted of a centroid point and m_s uniformly distributed satellites at radius $d_{cs}^{(0)} = r$ in $\mathbb{R}^2$ (Fig. 3.2.1(c)). The distance between the two centroids was set to $d_{cc}^{(0)} = \ell > 2r$, and data distribution had two global mirror reflection sym-

metries about (1) the line connecting two centroids, and (2) the perpendicular bisector thereof. We compared the changes in intra- and inter-cluster dissimilarities after EDT and found that the two circles were deformed, but the global mirror reflection symmetries were preserved. We further measured the mean $\bar{r}' \equiv \left\langle d_{\mathrm{cs}}^{(1)} \right\rangle$ and $\ell' \equiv d_{\mathrm{cc}}^{(1)}$ and found the ratio $\ell'/\bar{r}'$ to be an increasing function in both ℓ/r and m_{s}; moreover, $\ell'/\bar{r}' > \ell/r$ for any $\ell > 2r$ (Fig. 3.2.2(a)). Thus, the EDT had the effect of forcing the data points in each cluster to condense towards their respective centroid location, a potentially desirable effect that can help automatically merge data points into correct communities. We summarize our observation as a *cluster condensation* effect:

Proposition 11. *For separable clusters, the EDT condenses the points within a cluster, while inflating the space between clusters; this cluster condensation effect becomes stronger with the number of points in each cluster and also with the initial inter-cluster dissimilarity.*

The previous two Gedankenexperimente were performed on highly symmetric data sets. To probe the local deformation induced by EDT on a generic data distribution, we devised a detector, or a composite "test charge." The idea is generalizable to higher feature dimensions, but to simplify the interpretation, we performed the simulation in $\mathbb{R}^2$, with the detector being an ideal cluster of 12 sensor points at radius r from a centroid point (Fig. 3.2.1(d)). Deviations of the detector from a perfect circle in local ambient distributions were used to assess the EDT impact landscape. We captured the deviations through the transformed arm lengths $\{r_i'\}_{i=1}^{12}$ of the 12 sensors after EDT; we then derived two scalar quantities of interest: (1) the mean arm length $\nu = \langle r_i' \rangle$ that measures a volume change, and (2) the standard deviation of $\{r_i'/\nu\}_{i=1}^{m_{\mathrm{s}}}$, denoted κ, that measures anisotropy or the effect of "tidal force" from probed data points. The

observed volume changes were consistent with the effect of "perspective contraction," and the mean arm length ν of the detector shrank as it moved away from high density regions of the probed data distribution (Fig. 3.3.2). The κ-distributions were highly non-trivial, as illustrated in Fig. 3.3.3: κ attained high values whenever the rim of the detector was near a data point, indicating an intense tug-of-war between the data points and the detector that were both trying to capture the sensors; by contrast, the normalized κ almost vanished at the centers of two Gaussian distributions, within the inner circle of the two layers of circularly distributed points, and at the center of "O" in the "COS" data. The low values of κ in the interior of clustered data suggest a screening effect that shields the interior from anisotropic distortions, akin to the shielding effect of conductors in electrostatics; this effect may potentially protect sub-cluster structures within a dense cluster.

Inspired by the high values of κ near the boundary of a cluster, we performed additional experiments to test the effect of EDT on outliers, using (1) an ideal cluster in $\mathbb{R}^2$ with m_{s} satellites at radius r from the center point and an additional single point at varying distance ℓ from the center (Fig. 3.2.1(e)), and (2) the same ideal cluster in the xy-plane of $\mathbb{R}^3$ and two outliers located on the z-axis at $z = \pm\ell/2$ (Fig. 3.2.1(f)). For the first case, Fig. 3.3.4 shows how a cluster of points traps an outlier and prevents it from escaping the cluster. Furthermore, in both cases, we observed that the trapping power increased with cluster mass m_{s}: in case (1), increasing m_{s} reduced the relative effective outlier-centroid dissimilarity $\ell'/\bar{r}'$ and broadened the outlier region that got pulled back towards the cluster (Fig. 3.2.2(b)); in case (2), increasing m_{s} also decreased the relative effective outlier-centroid dissimilarity $\ell'_{\frac{1}{2}}/r'$ (Fig. 3.2.2(c)). We summarize the *local deformation* effect, or the "tidal force" exerted by local data distribution, as

follows:

Proposition 12. *Under the EDT, data points deform the local distribution of neighboring points such that potential outliers tend to be trapped by a massive cluster. The deformation is strong near the exterior of a cluster and almost completely screened inside the cluster.*

In case (2), we also observed an intriguing paradox: the transformed outlier-outlier dissimilarity ℓ' satisfied the condition $\ell' < 2\ell'_{\frac{1}{2}}$ for all $\ell_{\frac{1}{2}}/r > 0$, and it even satisfied the counter-intuitive inequality $\ell' < \ell'_{\frac{1}{2}}$ for sufficiently large $\ell_{\frac{1}{2}}/r$ and large m_{s} (Fig. 3.2.2(d)). A resolution of this paradox is achieved by noting that the points at infinity become identified under EDT. For example, for the particular case of circularly distributed data points in $\mathbb{R}^2$, as illustrated in Fig. 3.3.5, the outer rings of points become increasingly similar as τ, indexing the EDT iteration, increases; moreover, the effect becomes more pronounced as the density of points at the center of the distribution increases (bottom row in Fig. 3.3.5, Section 3.4.4). In mathematical terms, adding the point at infinity to $\mathbb{R}^2$ yields a compact sphere S^2, and the above process can be visualized as the outer rings diffusing towards the south pole (Fig. 3.3.5).

We tested whether this property of EDT can help improve clustering performance on synthetic data sets that are known to confound simple algorithms. For this purpose, we chose two clusters of data concentrically distributed with a gap in radial direction (Fig. 3.3.6). The EDT dramatically improved the performance of hierarchical clustering with Euclidean metric (Fig. 3.3.6); furthermore, the EDT-enhanced hierarchical clustering outperformed spectral clustering using Gaussian RBF as a measure of similarity (Fig. **??**). These observations can be summarized as EDT's *global deformation* effect:

Proposition 13. *EDT is able to globally warp the data space on the length scale comparable to inter-cluster distances, such that points far from the majority distribution become approximately identified. EDT thus topologically changes* $\mathbb{R}^n$ *to* S^n.

In application, the EDT will asymptotically group outliers that are very dissimilar to all clusters and may be dissimilar among themselves into one "unclassifiable" cluster in an automatic fashion.

Lastly, we considered the effect of EDT in a probabilistic sense. The initial dissimilarity $d^{(0)}$ can be thought of as a random matrix calculated from data sampled from a probability distribution. We replaced the ideal clusters in $\mathbb{R}^2$ in Fig. 3.2.1(c) by two independent bivariate Gaussian distributions $\mathcal{N}_1((-\ell_1, 0)^\top, \sigma_1^2)$ and $\mathcal{N}_2((\ell_2, 0)^\top, \sigma_2^2)$ located symmetrically about the origin, i.e. initially $\ell_1 = \ell_2$. We then placed a test point at the origin and two anchor centroids at $x = -\ell_1$ and $x = \ell_2$. Denoting the transformed value of ℓ_i after one application of EDT by ℓ_i', we used Monte Carlo simulations to compute the probability $\mathbb{P}(\ell_1' > \ell_2')$, which may be viewed as the probability that the test point is clustered with $\mathcal{N}_2$. We performed the calculation in two different settings: (1) $\mathcal{N}_1$ and $\mathcal{N}_2$ have same number of samples ($m_1 = m_2$), but different variances; and (2) $\mathcal{N}_1$ and $\mathcal{N}_2$ share the same variance ($\sigma_1^2 = \sigma_2^2$), but different number of samples. We found that the test point was more likely to join (1) a cluster drawn from the distribution with larger variance, consistent with the local deformation effect that absorbs an outlier near the boundary of a cluster into the cluster, or (2) a cluster with fewer samples, consistent with the global deformation effect of EDT that makes points from the majority distribution similar to each other. More precisely, we empirically found the $\mathbb{P}(\ell_1' > \ell_2')$ to be a hyperbolic tangent sigmoid function in $m_1/(m_1 + m_2)$ and

$-\log_2(\sigma_1/\sigma_2)$, as shown in Fig. 3.2.2(e-f).

3.4 Effective dissimilarity transformation

The following properties of EDT mentioned in the previous section were obtained from the Gedankenexperimente illustrated in Fig. 3.2.2(b-f).

3.4.1 Perspective contraction

The 3 points $\{P_1, P_2, P_3\}$ shown in Fig. 3.2.1(b) form two distinct configurations: (1) aligned in a line, and (2) forming a triangle in a plane. For case (1), let P_1 and P_2 be at $x = +a/2$ and $-a/2$, respectively, and P_3 at $x = b > a/2$. Then, the original dissimilarity matrix is

$$d^{(0)} = \begin{pmatrix} 0 & a & b+\frac{1}{2}a \\ a & 0 & b-\frac{1}{2}a \\ b+\frac{1}{2}a & b-\frac{1}{2}a & 0 \end{pmatrix},$$

and the transformed feature vectors are:

$$\hat{p}_1 = \frac{1}{\sqrt{b+\frac{3}{2}a}} \begin{pmatrix} 0 \\ \sqrt{a} \\ \sqrt{b+\frac{1}{2}a} \end{pmatrix},$$

$$\hat{p}_2 = \frac{1}{\sqrt{b + \frac{1}{2}a}} \begin{pmatrix} \sqrt{a} \\ 0 \\ \sqrt{b - \frac{1}{2}a} \end{pmatrix},$$

and

$$\hat{p}_3 = \frac{1}{\sqrt{2b}} \begin{pmatrix} \sqrt{b + \frac{1}{2}a} \\ \sqrt{b - \frac{1}{2}a} \\ 0 \end{pmatrix}.$$

From these feature vectors, we compute the first EDT dissimilarity matrix components to be

$$d_{12}^{(1)} = 1 - \hat{p}_1 \cdot \hat{p}_2 = 1 - \sqrt{\frac{b - \frac{1}{2}a}{b + \frac{3}{2}a}},$$

$$d_{13}^{(1)} = 1 - \hat{p}_1 \cdot \hat{p}_3 = 1 - \sqrt{\frac{a\left(b - \frac{1}{2}a\right)}{2b\left(b + \frac{3}{2}a\right)}},$$

and

$$d_{23}^{(1)} = 1 - \hat{p}_2 \cdot \hat{p}_3 = 1 - \sqrt{\frac{a}{2b}}.$$

As $b/a \uparrow \infty$, we have $d_{12}^{(1)} \downarrow 0$, $d_{13}^{(1)} \uparrow 1$, and $d_{23}^{(1)} \uparrow 1$; in other words, the EDT ruler length will shrink to zero if the observer moves away from the ruler. Next, we can calculate the relative dissimilarity, i.e. the observed difference between P_1 and P_2 from the perspective of P_3 measured in units of the transformed dissimilarity $d_{12}^{(1)}$, to be

$$\frac{d_{13}^{(1)} - d_{23}^{(1)}}{d_{12}^{(1)}} = \sqrt{\frac{a}{2b}}.$$

Therefore, as the observer moves away from the ruler, the EDT ruler length shrinks to zero, but the observed difference shrinks even faster. In the application of hierarchical

clustering, the diminishing difference between the nearest (P_2) and the farthest (P_1) point with respect to the outlier P_3 implies that clustering derived from the EDT tends to be robust against the choice of linkage, which may be single (nearest point), average, or complete (farthest point).

For case (2), we set up a Cartesian coordinate system in $\mathbb{R}^2$ such that P_1, P_2, and P_3 are located at $(0, a/2), (0, -a/2)$, and $(b, 0)$, respectively, where we assume $a, b > 0$. The original Euclidean distance matrix is thus

$$d^{(0)} = \begin{pmatrix} 0 & a & c \\ a & 0 & c \\ c & c & 0 \end{pmatrix},$$

where $c = \sqrt{\left(\frac{a}{2}\right)^2 + b^2}$. The transformed feature vectors are

$$\hat{p}_1 = \frac{1}{\sqrt{a+c}} \begin{pmatrix} 0 \\ \sqrt{a} \\ \sqrt{c} \end{pmatrix},$$

$$\hat{p}_2 = \frac{1}{\sqrt{a+c}} \begin{pmatrix} \sqrt{a} \\ 0 \\ \sqrt{c} \end{pmatrix},$$

and

$$\hat{p}_3 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}.$$

The corresponding transformed dissimilarity matrix elements are

$$d_{12}^{(1)} = 1 - \hat{p}_1 \cdot \hat{p}_2 = 1 - \frac{c}{a+c},$$

and

$$d_{23}^{(1)} = d_{13}^{(1)} = 1 - \hat{p}_1 \cdot \hat{p}_3 = 1 - \sqrt{\frac{1}{2}\frac{a}{a+c}}.$$

As b/a increases to infinity, $d_{12}^{(1)}$ monotonically decreases to zero. Thus, the effective ruler length d_{12} approaches 0 from the perspective of point P_3 as it moves far away.

3.4.2 Cluster condensation

When clustering real datasets, the contrast between the inter-cluster distance and the intra-cluster variance is often not very dramatic, making it very difficult to separate the clusters. Therefore, if the data points could condense to the respective centroid locations, then it would improve clustering accuracy considerably; this effect is precisely what EDT accomplishes. For the synthetic data shown in Fig. 3.2.1(c), the EDT centroid-centroid dissimilarity $d_{\mathrm{cc}}^{(1)}$ increased relative to the average centroid-satellite distance $\left\langle d_{\mathrm{cs}}^{(1)} \right\rangle$, or the contrast captured by the ratio $d_{\mathrm{cc}}^{(1)} / \left\langle d_{\mathrm{cs}}^{(1)} \right\rangle$ grew more rapidly than $d_{\mathrm{cc}}^{(0)}/d_{\mathrm{cs}}^{(0)}$. Moreover, for fixed $d_{\mathrm{cc}}^{(0)}/d_{\mathrm{cs}}^{(0)}$, increasing the number of satellites m_{s} around each centroid amplified the contrast ratio (Fig. 3.2.2(a)).

Throughout the simulations, we did not use any information about the cluster labels, and the improvement of contrast is purely driven by the data. The dense clusters condense while pushing themselves away from other clusters. In other words, within a cluster, the EDT acts similar to gravity, whereas the transformation inflates the space between clusters.

3.4.3 Local deformation

In Fig. 3.2.1(e), r denotes the radius of the cluster and ℓ the distance between a single test point and the cluster centroid. We simulated the effect of increasing ℓ on the EDT. As $\ell/r \downarrow 1$, we observed a window where the transformed ratio $\ell'/\bar{r}'$ was less than or equal to the local peak at $\ell/r = 1$ (Fig. 3.2.2(b)). This phenomenon can be interpreted as the cluster's trying to reabsorb the test point that is escaping to become an outlier. We also observed that the range of absorption window increased as the cluster size m_s increased, thus making it easier for an outlier to tunnel back to a denser cluster (Fig. 3.2.2(b)). Moreover, the test point also deformed the shape of the cluster, and the satellite points on the circle acted like an elastic membrane that trapped the test point and hindered it from escaping the cluster through elongation.

3.4.4 Global deformation

Consistent with the single test point example, the cluster in Fig. 3.2.1(f) tended to attract the two escaping outliers, as manifested by the fact that as m_s increased, the ratio ℓ'/r' decreased (Fig. 3.2.2(c)). Counterintuitively, $\ell'/\ell'_{\frac{1}{2}}$ also dropped below 1 as ℓ/r increased (Fig. 3.2.2(d)); that is, the two test points became more similar as they departed from the cluster centroid in opposite directions. This paradox can be resolved by merging the points at infinity to a single point, or by topologically transforming the Euclidean space into a hypersphere. We explicitly demonstrated our idea using two circularly distributed data sets shown in Fig. 3.3.5. We first observed that the effective dissimilarity between two neighboring points in the outer rings shrank faster than that between neighboring points in the inner rings. To better visualize this phenomenon, we then displayed the dissimilarities on a sphere using the following methods.

For a ring of k points distributed on a unit 2-sphere at constant colatitude $\theta \in [0, \pi]$ and uniformly partitioned longitude $\phi_i \in [0, 2\pi], i = 1, \ldots, k$, the latitude distance ℓ between any two neighboring points is equal to $\sin\theta\,\delta\phi$, where $\delta\phi = 2\pi/k$. Thus, ℓ attains its maximum value $\ell_{\max} = \delta\phi$ at the equator $\theta = \frac{\pi}{2}$. Note that regardless of the size of $\delta\phi$, we always have $\ell/\ell_{\max} = \sin\theta$; we will utilize this fact to display the EDT-deformed concentric rings shown in Fig. 3.3.5. For this purpose, it might appear natural to identify the centroid as the north pole of the sphere, and then identify the colatitude θ' of a ring as the EDT dissimilarity between the centroid and a point on the ring. However, while the distance between two neighboring data points on the sphere at such θ' would then be fixed to be $\sin\theta'\,\delta\phi$, the actual EDT dissimilarity ℓ' might be different. We thus empirically calculated the function $f(\theta')$ that satisfies $\ell' = f(\theta')\ell'_{\max}$. We then used the location $\theta_{\frac{\pi}{2}}$ of the global maximum of f to calibrate the equator location, and then calculated the effective colatitude $\tilde{\theta}$ defined as

$$\tilde{\theta} = \begin{cases} \arcsin \frac{\ell'}{\ell'_{\max}} & \theta' \leq \theta_{\frac{\pi}{2}} \\ \pi - \arcsin \frac{\ell'}{\ell'_{\max}} & \theta' > \theta_{\frac{\pi}{2}} \end{cases}$$

to display the concentric rings on the sphere, as shown in Fig. 3.3.5. Fig. 3.4.1 shows the $f(\theta')$ for the two circular data sets shown in Fig. 3.3.5 after τ iterations of EDT.

3.4.5 EDT and the curse of dimensionality

The loss of contrast in Euclidean distance is one of the symptoms of the curse of dimensionality; to be exact, the longest distance $d^{(0)}_{\max}$ and shortest distance $d^{(0)}_{\min}$ between any pair of points in a data set will both asymptotically approach the mean distance $\bar{d}^{(0)}$

in the large feature dimension limit $n \uparrow \infty$. To see whether EDT can help improve the contrast between clusters in high dimensions, we simulated two n-dimensional Gaussian distributions $\mathcal{N}((\pm\ell/2, 0, \ldots, 0, 0)^\top, \sigma^2 I_n)$, 100 points from each, for $\ell/\sigma = 0, 4$, and 10. We then computed the Euclidean distance matrix $d^{(0)}$ and subsequent effective dissimilarity matrices $\{d^{(\tau)}\}_{\tau=0}^{5}$. Fig. 3.4.2 shows the normalized pairwise maximum $(d_{\max}^{(\tau)}/\bar{d}^{(\tau)})$ and minimum $(d_{\min}^{(\tau)}/\bar{d}^{(\tau)})$ distance between data points in each dimension. The difference $(d_{\max}^{(\tau)} - d_{\min}^{(\tau)})/\bar{d}^{(\tau)}$ generally became larger as the EDT index τ increased, and the improvement in contrast over the original Euclidean distance in high dimensions was very dramatic when $\ell \gg \sigma$, as seen for $n = 1000$ in Fig. 3.4.2(c).

3.5 Application of EDT in two gene expression data sets

We tested the power of EDT on two publicly available gene expression data sets: (1) 59 cancer cell lines from NCI60 in 9 cancer types, (2) 116 blood cell samples in 4 cell types from human hematopoietic stem cell differentiation data set [46], with 4,000 most variable genes in each data set as features. We performed hierarchical clustering using the first few iterations of EDT dissimilarity. We used the variation of information (VI) as a well-defined distance between two clustering results [44]; using the given cell types as the reference clustering, we optimized the threshold for cutting the dendrogram into clusters and quantified the performance of clustering with the minimum distance to reference clustering (Fig. 3.5.1).

For the NCI60 data, the original Euclidean distance ($\tau = 0$) gave minimum VI of 1.042; but, after two rounds of EDT ($\tau = 2$), the VI reduced by 31.7% to 0.712 (top

two rows in Fig. 3.5.1). The original Euclidean distance failed to combine all leukemia (LE) cell lines, but EDT ($\tau = 2, 3$) brought LE cell lines together into a single cluster. From the very beginning ($\tau = 0$), the melanoma cell lines were in a distinct single cluster except for one outlier LOXIM-VI. Among the misclassified cell lines after two iterations of EDT, the LOXIM-VI found itself more similar to the mixture cluster of central nervous system (CNS) and breast cancer (BR) cell lines; the result is consistent with the fact that LOXIM-VI is a desmoplastic melanoma cell line and is biologically similar to neurofibroma [66].

For the blood cell data, the original Euclidean distance split the erythrocyte (Ek, where larger values of k indicate latter stages of of maturity) samples into several small sub-clusters, and the VI was 0.706 (bottom two rows in Fig. 3.5.1). After one iteration of EDT, the VI reduced by 54.0% to 0.325, and all Ek samples were grouped into a single cluster with two branches – immature red blood cells (E1, E2) and more mature blood cells (E3, E4, E5) – well separated from the immune cells (T-cells, B-cells, and, natural killer cells). These results support that the EDT can help improve clustering performance in real data analysis.

3.6 Data preparation

Two public data sets were used in the hierarchical clustering analysis: (1) NCI60 gene expression data in 59 cancer cell lines comprising 9 cancer types, and (2) 116 differentiated blood cell samples in 4 cell types from human hematopoietic cell (HHC) gene expression data [46]. The 9 cancer types in NCI60 data were 5 breast (BR), 6 central nervous system (CNS), 7 colon (CO), 6 leukemia (LE), 10 melanoma (ME), 8

non-small cell lung (LC), 7 ovarian (OV), 2 prostate (PR), and 8 renal (RE) cancer. The 4 cell types in the HHC data were red blood cells or erythrocytes (E), T-cells (T), B-cells (B), and natural killer cells (NK). For both data sets, approximately four thousand most variable genes were selected as features.

Samples from each data set were first clustered with the usual Euclidean distance $d_{ij}^{(0)} = \|\mathbf{x}_i - \mathbf{y}_j\|^2$, and then with the EDT dissimilarity $d_{ij}^{(\tau)}$ computed from $d_{ij}^{(0)}$. Average linkage was used in all hierarchical clustering analysis, unless indicated otherwise. To quantify the clustering performance unambiguously, the minimum distance from a given clustering to the standard reference clustering was found by measuring the variation of information (VI), which is a well-defined metric function that computes the distance between different partitions (clusterings) of a given set [44]. The reference clustering for NCI60 was the known 9 cancer types (BR, CNS, CO, LE, ME, LC, OV, PR, and RE); the reference clustering for HHC was the known 4 cell types (E, T, B, and NK).

3.7 Discussion

In this paper, we have developed the notion of effective dissimilarity transformation to enhance the performance of hierarchical clustering, utilizing only the geometric information of all pairwise dissimilarities. The nonlinear transformation adjusts the dissimilarities according to the global distribution of data points. The EDT can be interpreted either as deformation of the feature space or as the result of emergent interactions among all sample points. Specifically, we devised a probe to detect local "tension," or the force field due to ambient sample points, in a deformed feature space.

On a global scale, the EDT is able to change the topology of original Euclidean feature space into a compact sphere. Furthermore, iterating the EDT produces a discrete-time dynamical process purely driven by data set geometry. Using carefully designed Gedankenexperimente, we have shown that EDT has the following properties: (1) perspective contraction, (2) cluster condensation, (3) local deformation, and (4) global deformation effects. These properties arise as different facets of the same mathematical transformation and, thus, should be interpreted in a unified manner. The cosine similarity of EDT is akin to distance correlation [58] and measures the similarity of two random vectors obtained from pairwise similarities to all sample points. Properties (1), (2) and (4) can be understood as mutually enhancing the similarity among a subset of points that share common dissimilar points, while property (3) suggests that common similar points can enhance the similarity between "local" or slightly less similar points.

An adjustable regularizer is able to qualitatively improve an unsupervised learning algorithm. The index of the sequence of iterated EDT, or discrete "time" τ plays the role of tuning parameter in hierarchical clustering: increasing τ brings similar sample points into tighter proximity, while enhancing the contrast between clusters (communities). The EDT thus helps hierarchical clustering by utilizing information about the global data distribution. Furthermore, the improvement in clustering accuracy arises from the transformation of data set geometry; thus, any learning algorithm based on pairwise dissimilarity should also benefit from the desirable properties of EDT.

Although the key properties of EDT were first extracted in low feature dimensions in this paper, these advantages, arising from capturing the intrinsic geometry of data distribution, are independent of the feature space dimension, as demonstrated by our finding that EDT also improved the hierarchical clustering of two biological data sets

containing $4,000$ features. As an additional verification of the robustness of EDT in high feature dimensions, our simulation shows that the EDT helps increase the contrast in dissimilarity of bimodal Gaussian clouds even in feature dimensions as high as 10^3, where EDT adapts to the increase in feature dimension by increasing the "time" index τ (Section 3.4.5).

Table 3.1: Clustering variation of information as a function of α

		α						
	$\tau = 0$	$\frac{1}{20}$	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{2}$	1	2	5
NCI60	**1.042**	0.960	0.899	0.918	0.917	0.864	0.961	1.075
HHC	**0.706**	0.325	0.485	0.325	0.325	0.503	0.537	1.005

The EDT was motivated by the multinomial interpretation of a non-negative data vector and its mapping to a hypersphere [38]. In this view, the EDT first takes element-wise square root of the dissimilarity matrix, normalizes each column of $d_{ij}^{\frac{1}{2}}$ by its L^2-norm, and finally evaluates the cosine dissimilarity between the normalized column vectors lying on a hypersphere. One can consider generalizing this approach by raising the distance matrix to a different power $\alpha > 0$, i.e. taking d_{ij}^{α}. Large values of α will have the effect of selectively amplifying large elements in the column vector; in the limit $\alpha \to \infty$, all but the largest element in each column will be set to zero on the hypersphere. By contrast, small values of α will reduce the contrast between elements in a column; in the limit $\alpha \to 0$, all normalized column vectors will point in the direction $(1, 1, \ldots, 1)$, forming a single group. Thus, clustering will be poor when α is either too small or too large. Nevertheless, tuning α amounts to performing feature selection, and we have evaluated the effects of changing α on the hierarchical clustering of cancer cell lines (NCI60) and human differentiated blood cell (HHC) data (Table 3.1). Comparing the performance of hierarchical clustering without EDT ($\tau = 0$) and

with one iteration of EDT at $\alpha = \frac{1}{20}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1,\ 2,$ or 5 showed that the generalized EDT performed similarly well when $\alpha \approx \frac{1}{2}$, but tended to underperform when $\alpha \gg \frac{1}{2}$ or $\alpha \ll \frac{1}{2}$; however, different data sets may favor different choices of α.

In addition, we also tested how modulating α affects the clustering of our synthetic annulus data sets shown in Fig. 3.3.6. We found that (1) very large or small values, e.g. $\alpha = 5$ or $\frac{1}{20}$, gave poor clustering results, (2) $\alpha = \frac{1}{4}$ gave identical clustering results as $\alpha = \frac{1}{2}$, and (3) $\alpha = 1$ failed to cluster the "hard data set" correctly. Similar observations were found in the reanalysis of "single outlier absorption" effect illustrated in Fig. 3.2.1(e) for different values of α (Fig. 3.7.1): α values greater than 0.5 tended to weaken the absorption effect and shorten the trapping range, but values smaller than 0.5 tended to amplify the absorption.

To investigate the effect of α further, we generated a bivariate Gaussian cloud $\mathcal{N}((0,0)^\top, \sigma^2 = 0.01)$ of 50 samples, and benchmarked the volume effect captured by ν and anisotropy captured by κ using the detector shown in Fig. 3.2.1(d) (Fig. 3.7.2). Denoting the distance between the center of bivariate Gaussian distribution and the detector by r, our original EDT ($\alpha = \frac{1}{2}$) showed $\sim 1/r$ power law decay for both ν and κ. Increasing (decreasing) α induced faster (slower) power law decay of ν, but the κ anisotropy robustly followed $\sim 1/r$ for a range of moderate values of α. Near $\alpha \approx \frac{1}{2}$, we observed that ν followed a power law described by $\sim 1/r^{2\alpha}$. Both ν and κ showed significant deviations from the original EDT when $\alpha \gg \frac{1}{2}$. Therefore, within moderate values of α, one can control the volume effect by tuning the power α, where smaller α implies slower decay.

Acknowledgements

I thank Alex Finnegan and Hu Jin for critical reading of the manuscript and helpful comments.

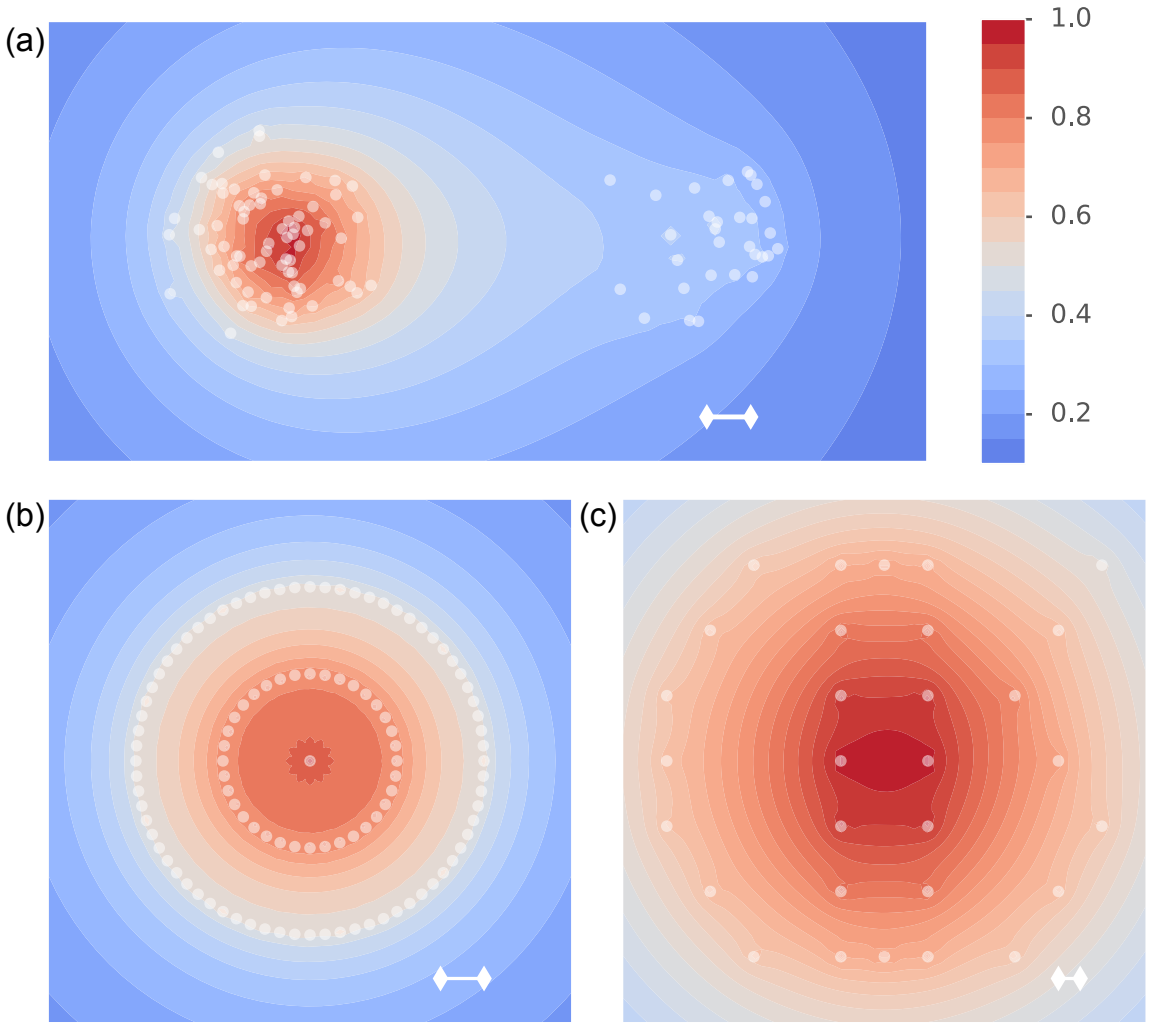

Figure 3.3.2: The ν-distribution for three data sets: (a) two Gaussian distributions with equal variance, but different sample sizes $m_{\text{left}} = 70$ and $m_{\text{right}} = 30$; (b) two layers of circularly distributed points with radius $r_{\text{outer}} = 2r_{\text{inner}}$ (bottom left); (c) points distributed in the shape of the word "COS." Each ν-distribution was normalized by dividing by its maximum; the white segment in each plot indicates the diameter of the detector used in the measurement of ν.

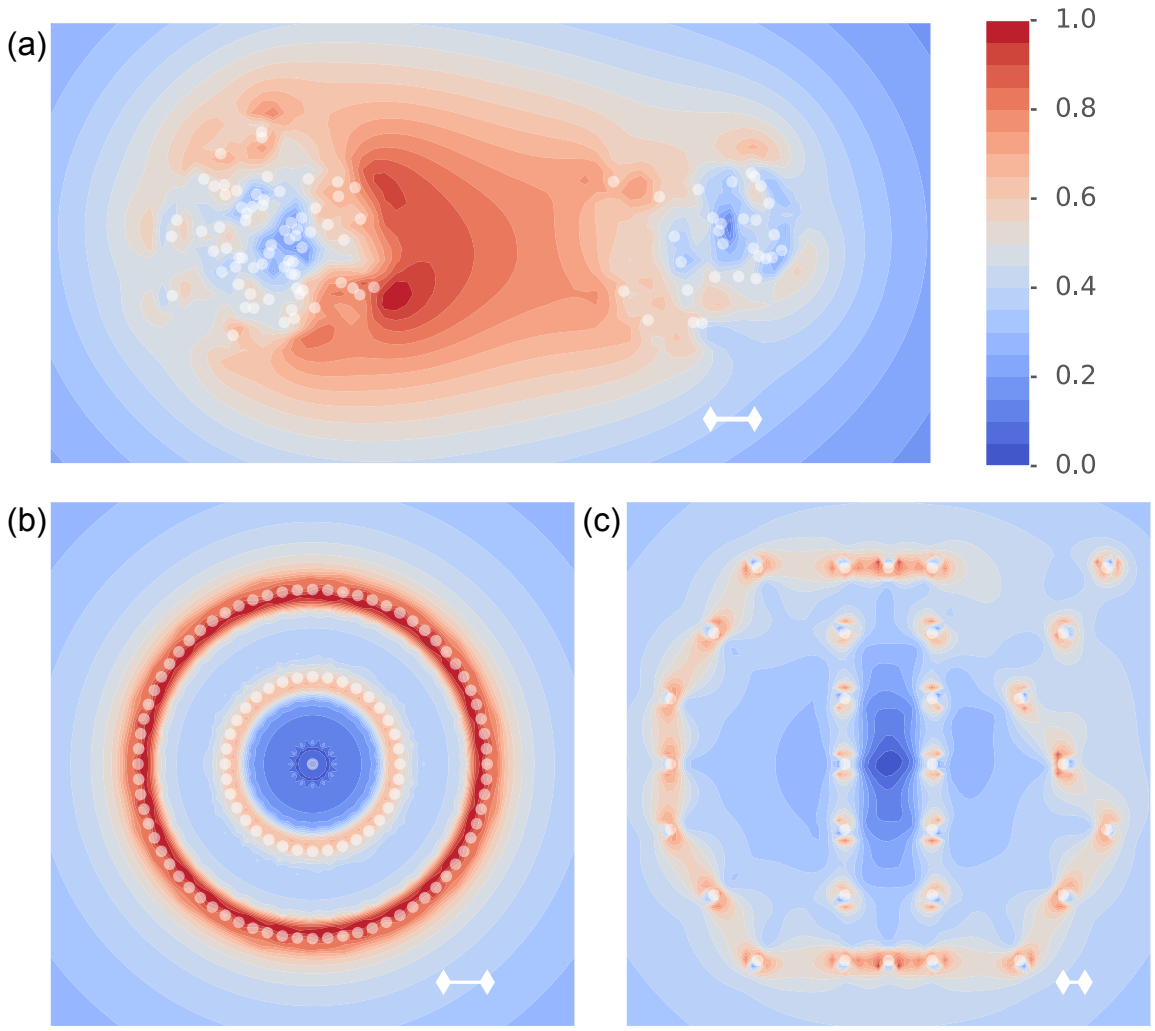

Figure 3.3.3: The κ-distribution for three data sets: (a) two Gaussian distributions with equal variance, but different sample sizes $m_{\text{left}} = 70$ and $m_{\text{right}} = 30$; (b) two layers of circularly distributed points with radius $r_{\text{outer}} = 2r_{\text{inner}}$; (c) points distributed in the shape of the word "COS." Each κ-distribution was normalized by dividing by its maximum; the white segment in each plot indicates the diameter of the detector used in the measurement of κ.

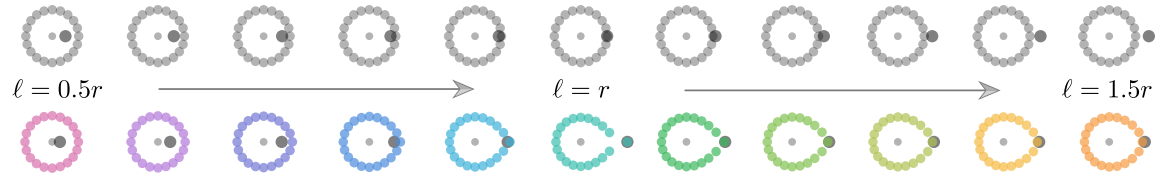

Figure 3.3.4: A cluster of points can pull back or "trap" an outlier. Figure shows the case illustrated in Fig. 3.2.1(e) for varying values of the ratio ℓ/r in the range $[0.5, 1.5]$ and for 20 satellite points. The top gray circles indicate the actual locations of points in $\mathbb{R}^2$; the bottom colored circles illustrate the corresponding effective locations after EDT, where we doubled the distortions to visualize the effect more clearly. As ℓ/r increased from left to right, the deformed circle behaved like an elastic membrane trying to trap the outlier from escaping and demonstrated singular behavior at $\ell = r$.

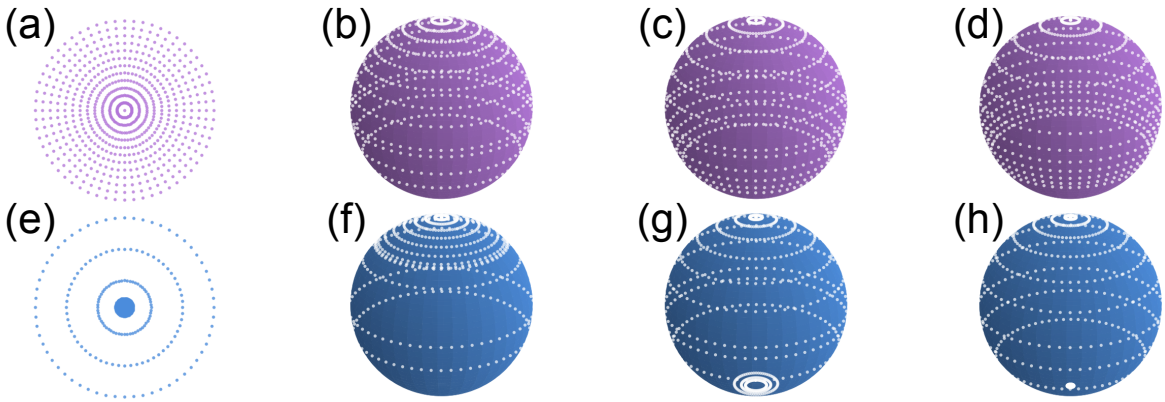

Figure 3.3.5: EDT approximately identifies the points at infinity. We designed two uniformly circularly distributed data sets with (a) a uniform increment in radius, or (e) a small increment in radius near the center and a large increment for the outermost three circles. For both data sets, the outer circles became relatively closer as τ increased. The values $\tau = 1, 2$, and 3 correspond to (b, f), (c, g), and (d, h), respectively. The effect was more pronounced in the case (e), and the outermost three circles were visibly mapped to the south pole. The mapping method can be found in Section 3.4.4.

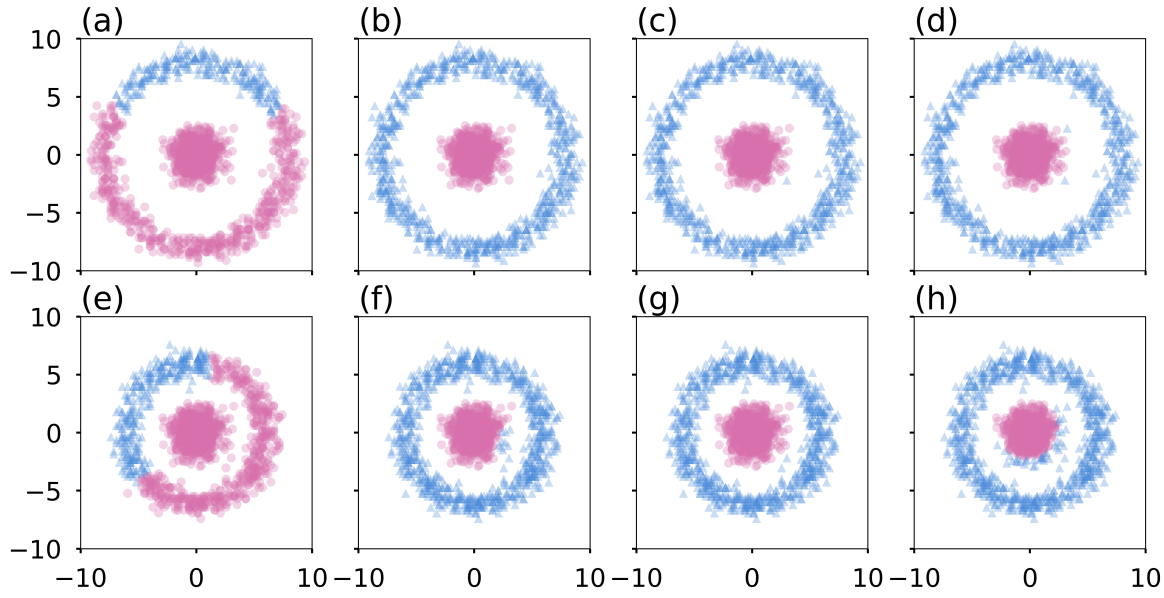

Figure 3.3.6: Hierarchical clustering results on (a-d) "easy" and (e-h) "hard" annulus data sets using Euclidean metric ($\tau = 0$) or EDT-enhanced dissimilarities up to three iterations and using average linkage. From left to right, (a, e), (b, f), (c, g), and (d, h) correspond to $\tau = 0, 1, 2$, and 3, respectively. Dramatic improvements were seen after just one iteration of EDT.

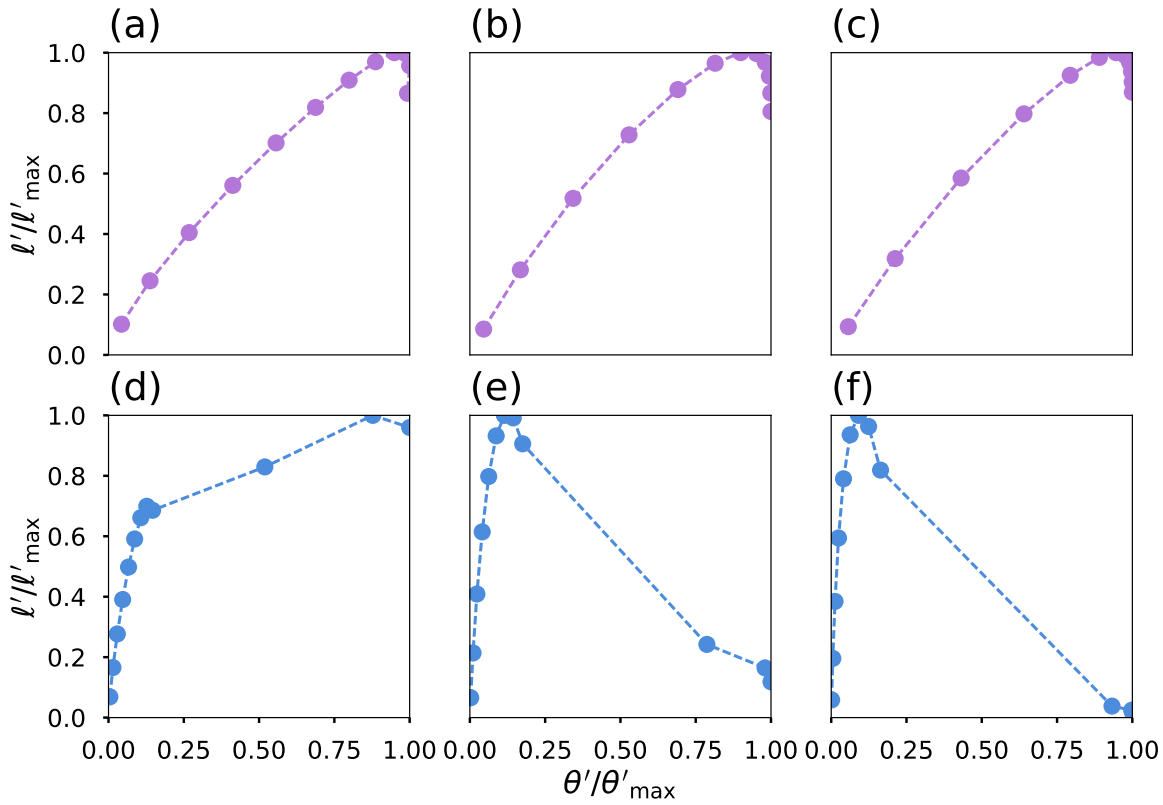

Figure 3.4.1: Plots of the empirical function f that satisfies $\ell' = f(\theta')\ell'_{\max}$, where ℓ' is the EDT dissimilarity between two neighboring points on a circle and θ' is the EDT dissimilarity between the centroid and the circle. The three plots on the top (a-c) correspond to the top three spheres in Fig. 3.3.5; and similarly for the three plots on the bottom (d-f).

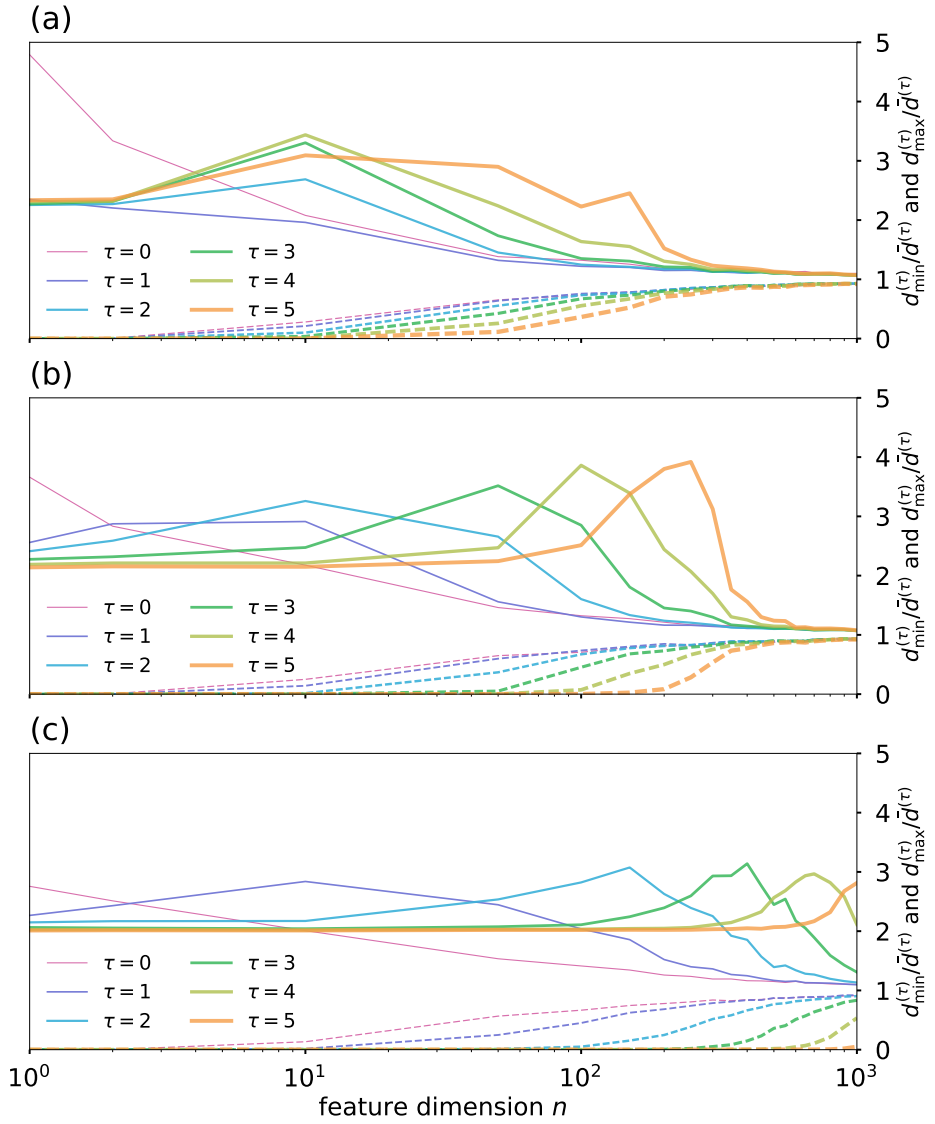

Figure 3.4.2: Plots of maximum and minimum dissimilarities normalized by mean dissimilarity: $d_{\max}^{(\tau)}/\bar{d}^{(\tau)}$ (solid) and $d_{\min}^{(\tau)}/\bar{d}^{(\tau)}$ (dashed) of two multivariate normal distributions $\mathcal{N}((\pm\ell/2, 0, \ldots, 0, 0)^t, \sigma^2 I_n)$ in $\mathbb{R}^n$ with variations in (a) $\ell/\sigma = 0$, (b) $\ell/\sigma = 4$, and (c) $\ell/\sigma = 10$. For all three cases (a-c), EDT ($\tau > 0$) enlarged the difference between $d_{\max}^{(\tau)}/\bar{d}^{(\tau)}$ and $d_{\min}^{(\tau)}/\bar{d}^{(\tau)}$, and hence enhanced the contrast; when the initial inter-cluster distance $\ell \gg \sigma$, EDT with high index τ preserved contrast dramatically relative to initial Euclidean distance $d^{(0)}$, consistent with cluster condensation effect of EDT.

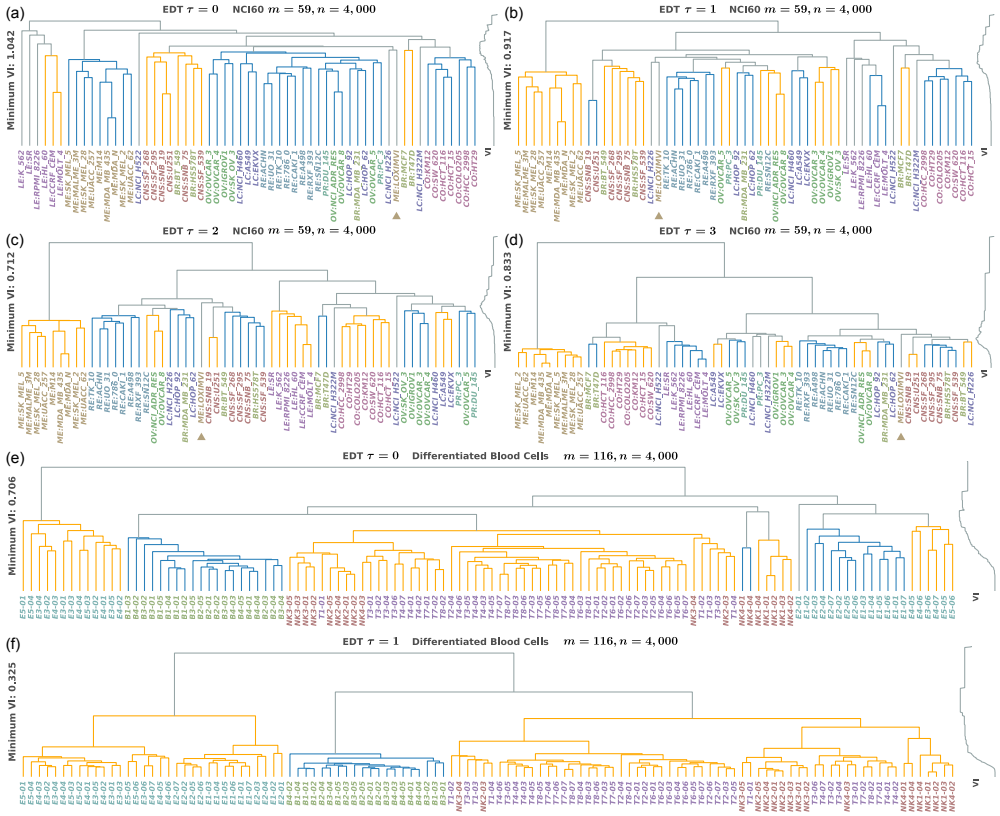

Figure 3.5.1: Hierarchical clustering of NCI60 cancer cell lines (a-d: $m = 59$ samples) and human differentiated blood cells (e,f: $m = 116$ samples) with $n = 4,000$ most variable genes (with largest standard deviations across all samples) in each each data set. For the NCI60 data, the original Euclidean distance (a: $\tau = 0$) gave minimum VI of 1.042; but, after two rounds of EDT (c: $\tau = 2$), the VI reduced by 31.7% to 0.712. The original Euclidean distance failed to combine all leukemia (LE) cell lines, but EDT (c,d: $\tau = 2, 3$) brought LE cell lines together into a single cluster. From the very beginning (a: $\tau = 0$), the melanoma cell lines were in a distinct single cluster except for one outlier LOXIM-VI, which is a desmoplastic melanoma cell line and is biologically similar to neurofibroma. Among the misclassified cell lines after two iterations of EDT, the LOXIM-VI found itself more similar to the mixture cluster of central nervous system (CNS) and breast cancer (BR) cell lines. (e) For the blood cell data, the original Euclidean distance split the erythrocyte (Ek, where larger values of k indicate latter stages of of maturity) samples into several small sub-clusters, and the VI was 0.706. (f) After one iteration of EDT, the VI reduced by 54.0% to 0.325, and all Ek samples were grouped into a single cluster with two branches – immature red blood cells (E1, E2) and more mature blood cells (E3, E4, E5) – well separated from the immune cells: T-cells, B-cells, and natural killer (NK) cells. These results support that the EDT can help improve clustering performance in real data analysis.

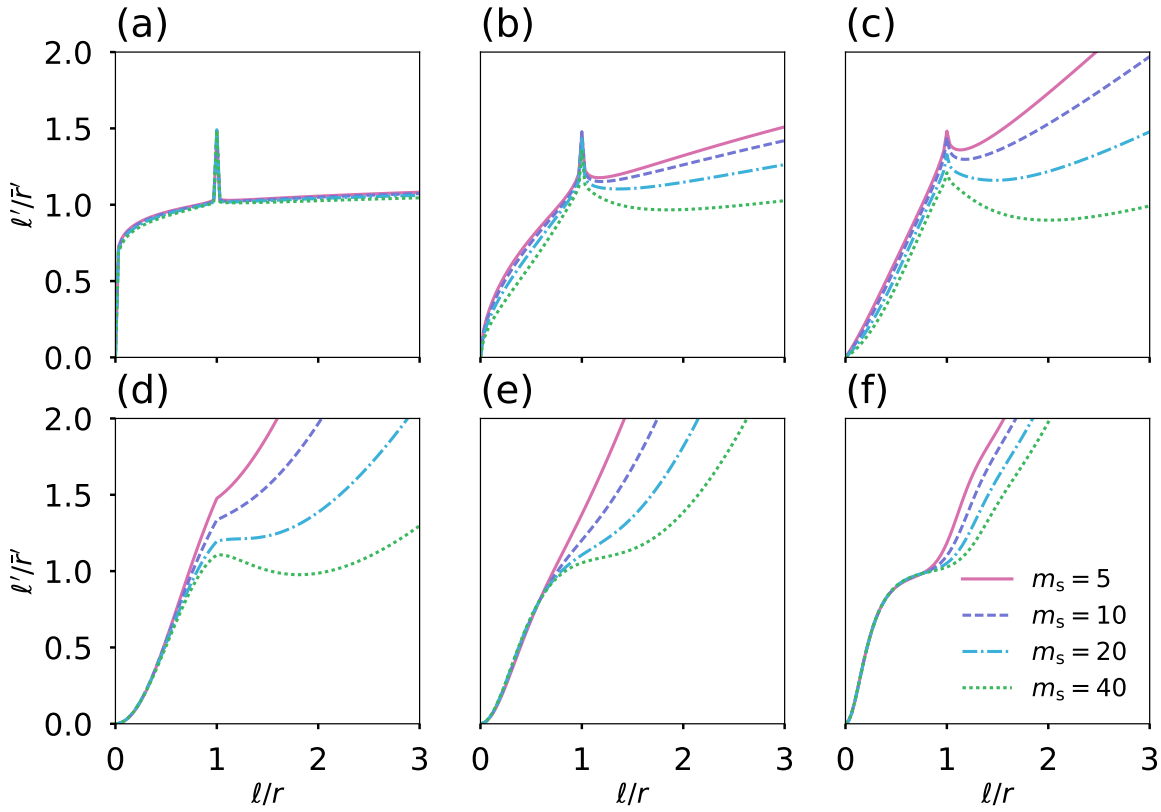

Figure 3.7.1: Outlier absorption effect under different α values. Figures (a-f) correspond to $\alpha = \frac{1}{20}, \frac{1}{4}, \frac{1}{2}, 1, 2, 5$, respectively. For $\alpha > \frac{1}{2}$, the singularity at $\ell/r = 1$ was attenuated, and the trapping window was shortened as α increased; by contrast, for $\alpha < \frac{1}{2}$, the singular peak grew sharper as α decreased, and the outlier far from the ideal cluster can still be absorbed into the cluster.

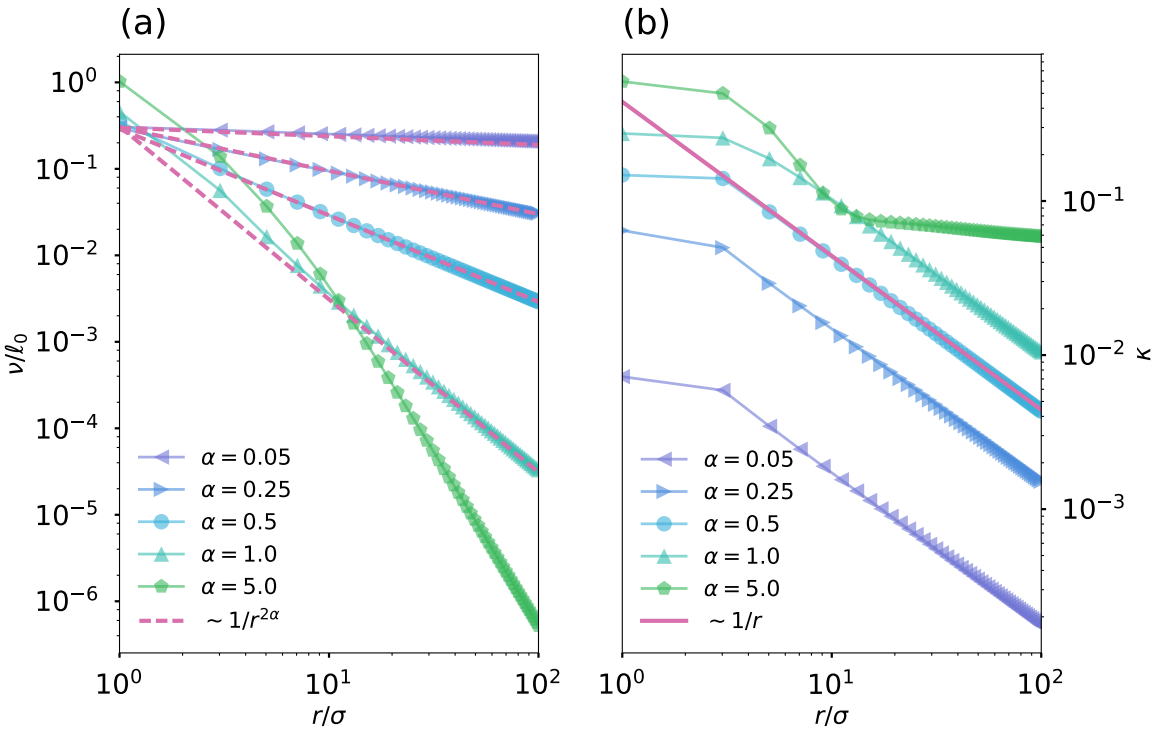

Figure 3.7.2: Effects of generalized EDT. We measured (a) the volume effect captured by ν and (b) anisotropy κ of a bivariate Gaussian cloud $\mathcal{N}((0,0)^\top, \sigma^2 = 0.01)$ of 50 samples in a plane, using the detector shown in Fig. 3.2.1(d). The original EDT ($\alpha = \frac{1}{2}$) showed $1/r$ power law decay, where r is the distance between the center of Gaussian cloud and center of the detector. For $\alpha \gg \frac{1}{2}$, both volume effect and anisotropy deviated from simple power laws.

Chapter 4

Clustering via quantum time evolution

4.1 Introduction

Grouping similar objects into sets is a fundamental task in modern data science. There have been several physics-inspired approaches based on classical spin-spin interaction models [39, 50] and Schrödinger equation [32]; however, the former usually requires computationally intensive Monte Carlo simulations which may get trapped in local optima, while the latter essentially amounts to Gaussian kernel density estimation. We here use the physics of quantum transport (QT) on data similarity networks to devise a simple and efficient clustering algorithm. This chapter is based on [69].

4.2 Schrödinger equation implies fluid dynamics

The Schrödinger equation for a free particle is, up to the Wick rotation $t \to \mathrm{i}t$, formally similar to the heat equation with heat conductance κ:

$$\partial_t u = \kappa \nabla^2 u. \tag{4.2.1}$$

Assuming that the heat conductance κ is constant in space, the heat equation can be rewritten as

$$\partial_t u = \kappa \nabla^2 u = -\nabla \cdot (-\kappa \nabla u)\,. \tag{4.2.2}$$

Defining the heat current as

$$\mathbf{j} = -\kappa \nabla u\,, \tag{4.2.3}$$

the heat equation then becomes the conservation law

$$\partial_t u + \nabla \cdot \mathbf{j} = 0. \tag{4.2.4}$$

The Schrödinger equation also embodies a conservation law. For example, consider the Schrödinger equation with a time-independent potential $V(x)$:

$$\mathrm{i}\partial_t \psi = -\frac{\nabla^2 \psi}{2m} + V(x)\psi, \tag{4.2.5}$$

in units where $\hbar = 1$. Writing its solution as

$$\psi(x,t) = \sqrt{\rho(x,t)}\mathrm{e}^{\mathrm{i}\theta(x,t)} \tag{4.2.6}$$

, where ρ is the probability density and θ the phase, we see that the Schrödinger equation is not one but two coupled equations for ρ and θ,

$$\dot{\rho} = -\nabla \cdot \left(\rho \frac{\nabla \theta}{m} \right) \equiv -\nabla \cdot (\rho \mathbf{v}) = -\nabla \cdot \mathbf{j}, \tag{4.2.7}$$

where $\mathbf{v} = \nabla\theta/m$ is the group velocity of a quantum mechanical particle, and $\mathbf{j} = \rho\mathbf{v}$ the current density; and

$$-\dot{\theta} = \frac{m}{2} \left(\frac{\nabla \theta}{m} \right)^2 + V - \frac{1}{2m} \left[\frac{\nabla^2 \sqrt{\rho}}{\sqrt{\rho}} \right] \tag{4.2.8}$$

$$\equiv \frac{1}{2} m \mathbf{v}^2 + V + Q \tag{4.2.9}$$

where

$$Q = -\frac{1}{2m} \left[\frac{\nabla^2 \sqrt{\rho}}{\sqrt{\rho}} \right] \tag{4.2.10}$$

is the "quantum potential."

Notice that the quantum current is proportional to $\nabla\theta$ instead of $\nabla\rho$. Thus, the phase gradient drives the propagation of the wave function, which encodes richer physics than classical heat density. This observation suggests that the phase information may be useful for devising quantum algorithms.

Heat diffusion has been applied to rank web page popularity [12], probe geometric features of data distribution [14], and measure similarity in classification problems [37, **?**]. By contrast, despite the formal resemblance between the heat equation and the Schrödinger equation, the time evolution of a quantum wave function has been largely ignored in machine learning. Both heat and Schrödinger equations have conserved currents; however, while the heat current is proportional to the negative gradient of

heat density itself, the velocity of quantum probability current is set by the phase gradient which satisfies the Naiver-Stokes equation, making quantum probability density an irrotational fluid. Thus, the Schrödinger equation embodies richer physics than heat diffusion and can capture spatiotemporal oscillations and wave interference. One promising observation has been that quantum time evolution can be faster in reaching faraway nodes compared with heat diffusion in ordered binary tree networks, suggesting the possibility of finding practical applications of quantum mechanics in network analysis [22, 13, 53, 21]. However, there are several outstanding challenges: e.g., unlike the heat kernel, the oscillatory quantum probability density is monotonic in neither time nor spatial distance; moreover, irregularities in either edge weights or network structure can severely restrict the propagation of a wave function through destructive interference, analogous to Anderson localization in disordered media [2]. We circumvent these difficulties associated with using the probability density itself and demonstrate the utility of the phase information for clustering network nodes.

4.3 Graph Laplacians

A generic undirected weighted network, e.g. a data similarity network of m samples in $\mathbb{R}^d$ represented as nodes, is encoded by an $m \times m$ symmetric adjacency matrix A. The row or column sum vector $\deg(i) = \sum_k A_{ik} = \sum_k A_{ki}$ gives rise to the diagonal degree matrix $D = \mathrm{diag}(\deg)$. Replacing the continuous Laplacian with the graph Laplacian $L = D - A$ then discretizes the heat and Schrödinger equations on data similarity networks. Enforcing the conservation of discrete heat current introduces the normalized graph Laplacian $Q = LD^{-1}$. The original graph Laplacian L of an

undirected network is automatically Hermitian, but we adopt the symmetrized version $H = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$ of Q as our Hamiltonian, since it has the same spectrum as Q. With this choice, H has a nontrivial ground state $\psi_0(i) \propto \sqrt{\deg(i)}$ [21].

For concreteness, we define the pairwise similarity or adjacency between sample $\mathbf{x}_i$ and sample $\mathbf{x}_j$ by the Gaussian function $A_{ij} = \exp(-r_{ij}^2/r_\varepsilon^2)$, where $r_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$ is the Euclidean distance and r_ε is the ε-quantile among $r_{ij} > 0$. Ideally, the proximity measure r_ε is chosen such that for samples i and j belonging to distinct clusters, we have $r_{ij} \gg r_\varepsilon$, but within any given cluster, a pair (i, j) of nearest neighbors has $r_{ij} \sim \mathcal{O}(r_\varepsilon)$.

4.4 Laplace transform of time evolution

The Laplace transform of a wave function $|\psi(t)\rangle$, evolved from an initial state $|\psi(0)\rangle$ via a time-independent Hamiltonian H, is given by

$$|\tilde{\psi}(s)\rangle \equiv \mathcal{L}[|\psi\rangle](s) = \int_0^\infty \mathrm{e}^{-st} \mathrm{e}^{-\mathrm{i}Ht} |\psi(0)\rangle \, dt. \tag{4.4.1}$$

Since H is time-independent, we have

$$|\tilde{\psi}(s)\rangle = \frac{1}{s + \mathrm{i}H} |\psi(0)\rangle = \mathrm{i}G(\mathrm{i}s)|\psi(0)\rangle, \tag{4.4.2}$$

where

$$G(z) \equiv (z - H)^{-1} \tag{4.4.3}$$

is the resolvent operator of H. We have interpreted $G(z)$ using an effective tight-binding model. Here, we study the Laplace-transformed wave function explicitly. The

inverse of the variable s sets the time scale within which the Schrödinger time evolution is averaged; i.e., this scale sets the extent to which oscillation in time is smoothed out and destructive interference that can potentially localize the transport gets ameliorated. Motivated by this observation, this paper demonstrates that taking the Laplace transform can resolve the issues of wave function oscillation and localization that have hindered the application of quantum mechanics to clustering problems.

Of note, recall that spectral clustering uses the j-th entries of the first few lowest-eigenvalue eigenvectors of the graph Laplacian to represent the j-th node. By contrast, one distinct advantage of QTC lies in utilizing the eigenvectors ψ_n twice when computing the phase of

$$\langle j|\tilde{\psi}(s)\rangle = \sum_n \frac{\langle \psi_n|\psi(0)\rangle}{s + \mathrm{i}E_n}\psi_n(j); \tag{4.4.4}$$

namely, both the j-th entries $\psi_n(j)$, just as in spectral clustering, and the projections $\langle \psi_n|\psi(0)\rangle$ onto the initialization node are used. In this way, as the initialization node varies during the random sampling step, the phase representations of two nodes within a cluster will stay close to each other, and this information is pooled together in the QTC algorithm.

4.5 Phase information of Laplace-transformed wave function

Defining the Laplace transform of a wave function initially localized at node j and evaluated at node i as [2]

$$\mathcal{L}[\psi(i|j)](s) = \int_0^\infty dt\, \langle i|\mathrm{e}^{-\mathrm{i}Ht-st}|j\rangle, \tag{4.5.1}$$

our clustering algorithm stems from the observation that the phase $\Theta(i|j)$ of this transformed function is essentially constant as i varies within a cluster, but jumps as i crosses clusters. The phase information thus provides a one-dimensional representation of data on S^1, such that distinct clusters populate separable regions on S^1; intuitively, the phase distribution $\Theta(\cdot|j)$ corresponds to a specific perspective on community structure sensed by the wave packet initialized at node j. In general, the phase distribution $\Theta(\cdot|j)$ changes with the initialization node j. Thus, if we randomly choose m' initialization nodes ($m' \approx 100$ for data sets in Fig. 4.5.1&4.5.2), for $1 < m' \leq m$, then we obtain an ensemble of m' phase distributions, in each of which the phase is almost constant within clusters; this ensemble ultimately provides a collection of perspectives on the underlying community structure, as sensed by the wave packets initialized at the chosen nodes.

In practice, we *a priori* specify the number q of clusters, and use the phase distribution of each wave function to partition the nodes into q subsets. We label each of the m'' distinct partitions by an integer α, where $m'' \leq m'$, and calculate the occurrence frequency $w_\alpha \in (0, 1]$ of each partition, such that the normalization condition

$\sum_\alpha w_\alpha = 1$ holds. Typically, we find that the frequencies are dominated by a single partition; other $m'' - 1$ less frequent partitions may arise from wave functions initialized at nodes of a small subnetwork isolated from the rest of the network. Hence, the minority predictions provide less holistic views of the network community structure, and we choose the majority prediction from the ensemble as our final clustering decision.

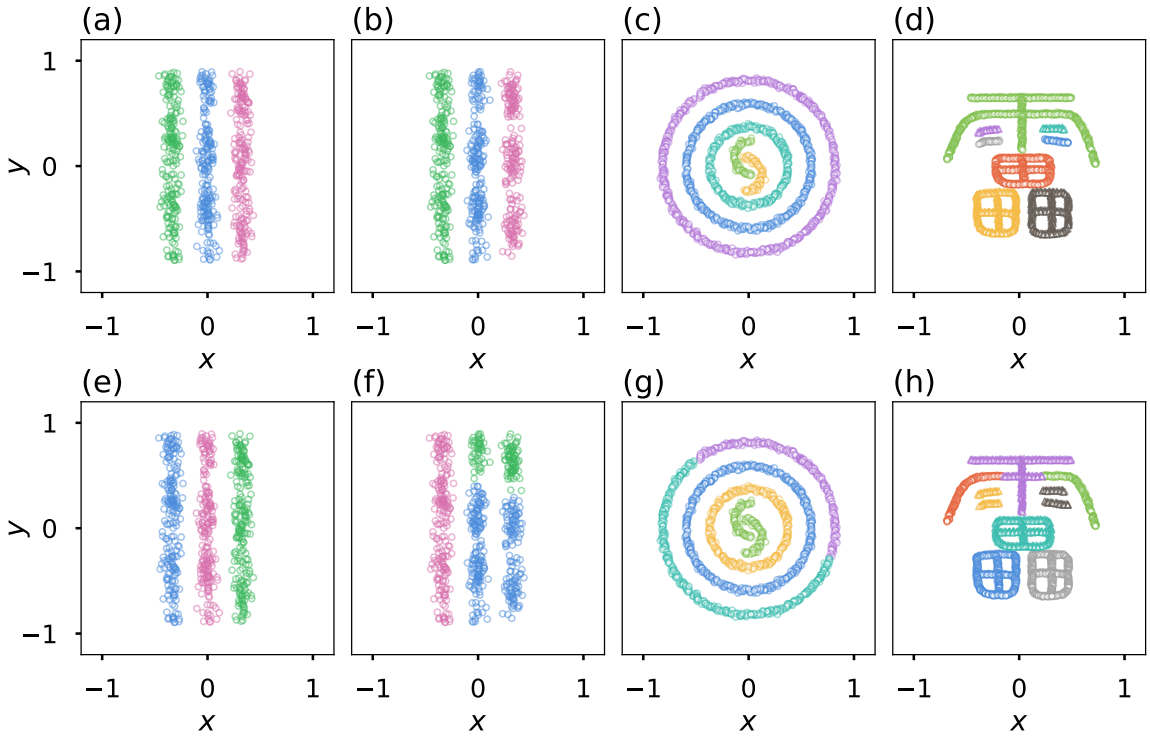

Figure 4.5.1: Comparison of (a-d) QTC and (e-h) spectral clustering using synthetic data. We specified three clusters for (a,b,e,f), five clusters for (c,g), and eight clusters of (d,h). We chose intermediate values of proximity measure r_ε in the Gaussian similarity function to demonstrate the robustness of QTC; spectral clustering was able to produce the correct clustering only when r_ε was tuned to be sufficiently small.

We compared the performance of QTC to spectral clustering [1] using four synthetic data sets having complex geometry (Fig. 4.5.1): (1) uniform sticks, (2) non-uniform

[1] When without an explicit specification, the affinity matrix used in spectral clustering is the same one used in QTC.

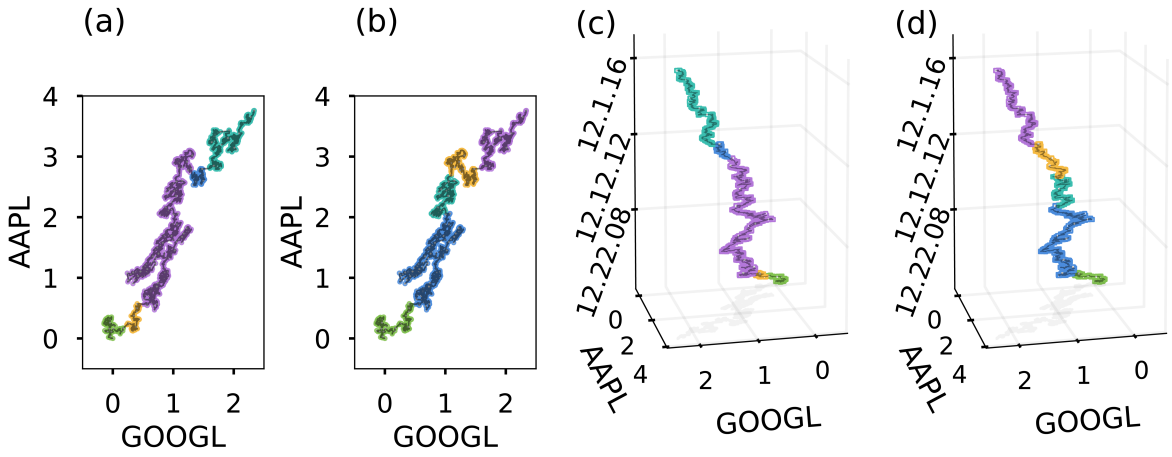

Figure 4.5.2: Comparison of (a) QTC and (b) spectral clustering using the time series data of log-prices of AAPL and GOOGL stocks from January 3, 2005 to November 7, 2017. Five clusters were specified, and the 1%-quantile $r_{1\%}$ was chosen as the proximity measure. The time evolution trajectories of data in (a) and (b) are displayed in (c) and (d), respectively, with an extra temporal dimension.

Table 4.1: Daily returns (%) at the identified jumps in Fig. 4.5.2

Date	2005		2010			2012	2013	
	5/23	10/21	4/16	4/20	4/21	2/8	1/24	10/18
GOOGL	+5.6	+11.4	−7.9	+0.9	−0.1	+0.5	+1.7	+13.0
AAPL	+5.7	−0.8	−0.6	−0.1	+5.8	+1.7	−13.2	+0.9
	Q	Q S	S	S	S	S	Q	Q S

sticks, (3) concentric annuli, and (4) the Chinese character for "thunder." Both algorithms performed equally well on the simple data set of uniformly sampled sticks (Fig. 4.5.1(a,e)) or when r_ε was chosen to be sufficiently small such that the clusters became almost disjoint subnetworks; as r_ε increased, however, QTC remained robust (Fig. 4.5.1(b-d)), while spectral clustering made mistakes (Fig. 4.5.1(f-h)). We further tested QTC on time-series stock price data (Data preparation section). The log-prices of a portfolio of stocks form a random walk in time with occasional jumps which are often triggered by important events such as the release of fiscal reports and sales records.

The jumps then separate the fractal-like trajectory of historical log-prices into several performance segments. Figure 4.5.2(a,b) shows the log-price distribution of two stocks, AAPL and GOOGL, from January 3, 2005 to November 7, 2017, where we removed the temporal information from the data set. When we specified five clusters, QTC cut the trajectory into five consecutive segments in the temporal space (Fig. 4.5.2(a,c)) with heterogeneous lengths, whereas spectral clustering partitioned the trajectory into clusters of similar sizes and mixed the temporal ordering near the boundary of blue and cyan clusters (Fig. 4.5.2(b,d)). The jumps identified by QTC (Q's in Table 4.1) coincided with major news events for the two stocks, whereas spectral clustering (S's in Table 4.1) failed to identify the large drop of AAPL on 1/24/2013 and instead included several less significant stock movements. These results showed that QTC was more robust than the conventional spectral embedding method on non-spherical data distributions with anisotropic density fluctuations (Fig. 4.5.1(b,f)) or complex geometric patterns exhibiting a hierarchy of cluster sizes (Fig. 4.5.1(c,g) and (d,h); Fig. 4.5.2).

When the clusters in data show strong mixing, no single partition may be clearly dominant, so using the partition corresponding to the highest occurrence frequency w_α may be unstable. In this scenario, we propose a "fuzzy" summary of the ensemble. Across m' different initializations, we count the number of times where two nodes, say i and k, are assigned to the same cluster, and then divide the count by m'. We thereby arrive at a symmetric consensus matrix C_{ik} with 1 along the diagonal and other entries in $[0, 1]$. The consensus matrix provides a useful visualization of processed clustering structure and also serves as a new input similarity measure suitable for many popular statistical learning algorithms, such as spectral clustering, hierarchical clustering, and SVM.

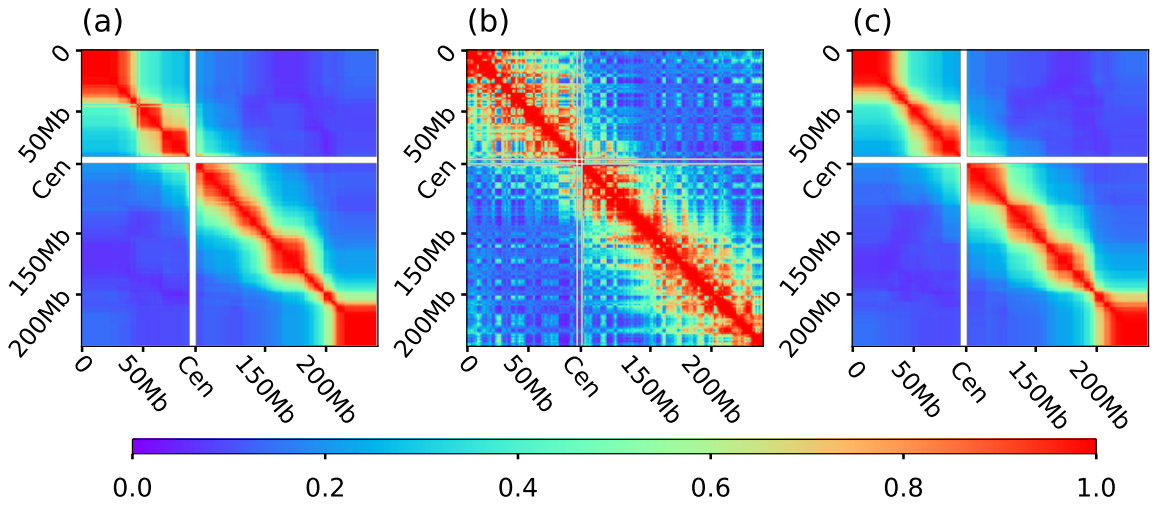

Figure 4.5.3: Similarity maps of genomic locations on human chromosome 2. (a) Averaged consensus matrix $\langle C_{\mathrm{LGG}} \rangle$ computed from SCNA data in LGG. (b) HiC contact map in normal glial cells [18]. (c) Averaged consensus matrix $\langle C_{\mathrm{GBM}} \rangle$.

For instance, we used the somatic copy number alteration (SCNA) data in low-grade glioma (LGG) and glioblastoma (GBM) patients from the Cancer Genome Atlas to construct an adjacency matrix of genomic locations, and performed QTC with the chosen number of clusters equal to $2, 3, 4$, or 5. We summarized the predicted similarity between genomic coordinates by averaging the consensus matrices $\{C(q)\}_{q=2}^{5}$ for LGG and GBM separately, yielding $\langle C_{\mathrm{LGG}} \rangle$ and $\langle C_{\mathrm{GBM}} \rangle$. The block structures in SCNA captured by QTC closely resembled the 3D chromatin interaction HiC contact matrix (Fig. 4.5.3) [18]; the Pearson correlation coefficients between $\langle C_{\mathrm{LGG/GBM}} \rangle$ and $\tanh((C_{\mathrm{HiC}})_{ij}/\bar{C}_{\mathrm{HiC}}) \in [0, 1)$ was 0.87, whereas the same correlation involving the raw SCNA data was less than 0.50 (Fig. 4.10.1). Our QTC consensus matrix thus denoises the SCNA data and helps support the previously observed phenomenon linking genomic alterations in cancer with the 3D organization of chromatin [25].

4.6 Spectrum of graph Laplacian reflects the number of clusters

If $q > 1$ clusters are well-separated, the Hamiltonian is approximately q-block diagonal. Fluctuations between the q macroscopic modes have lower kinetic energy, which mainly arises from inter-cluster tunneling, than microscopic fluctuations within each cluster. In this case, there exists an energy gap separating the low-energy macroscopic modes from the high-energy microscopic oscillations. Furthermore, the low-energy states can be approximated as linear combinations of cluster wave functions; thus, the number of low-energy states equals the number of putative clusters. For illustration, we generated well-separated $q = 2, 3,$ and 4 Gaussian clusters in three dimensions (Fig. 4.6.1(a,b,c)); the adjacency matrix was computed using the 10%-quantile of pairwise distance distribution as the proximity scale in Gaussian kernel. The first 6 eigenvalues of the Hamiltonian are plotted in Fig. 4.6.1(d,e,f).

4.7 Effective tight-binding model

Next, we provide a physical interpretation of the agglomeration phenomena observed in QTC using an effective tight-binding model. For this purpose, we rewrite the Laplace transform as $\mathcal{L}[\psi(i|j)](s) \equiv \mathrm{i}G(i, j; \mathrm{i}s)$, where

$$G(i, j; z) \equiv \langle i|(z - H)^{-1}|j\rangle = \sum_{n=0}^{m-1} \frac{\langle i|\psi_n\rangle\langle\psi_n|j\rangle}{z - E_n} \tag{4.7.1}$$

is the resolvent of H, and ψ_n and E_n are the eigenvectors and eigenvalues of H, respectively, for $n = 0, 1, \ldots, m-1$. We assume that E_n are ordered in a non-decreasing

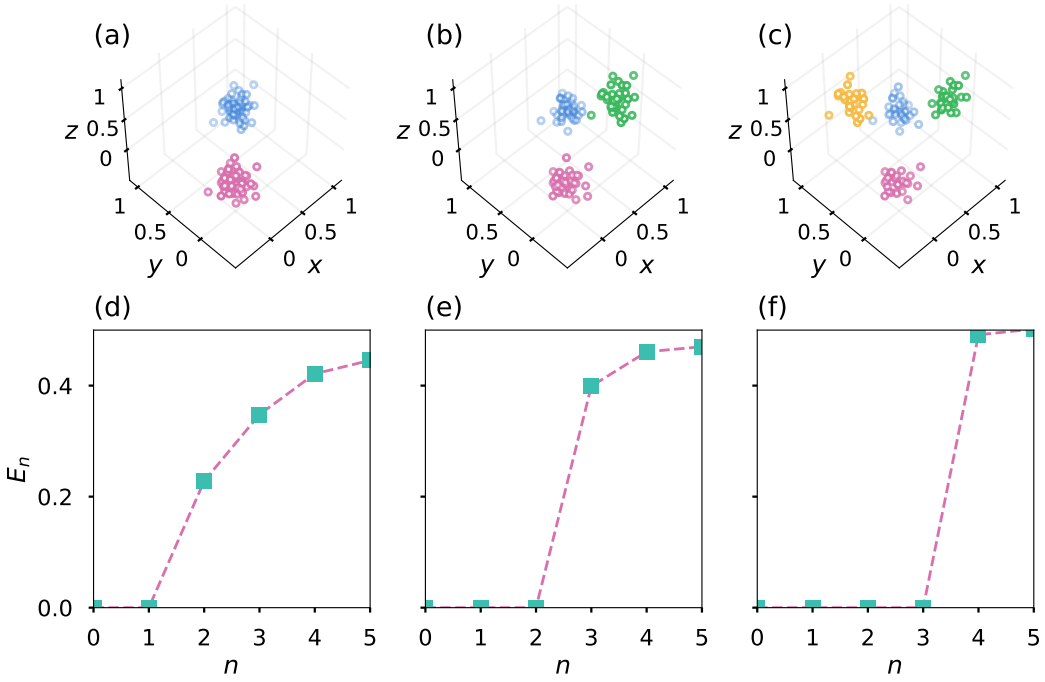

Figure 4.6.1: (a,b,c) Gaussian distributions ($\sigma = 0.1$) in $\mathbb{R}^3$, with the means located at the vertices of a regular tetrahedron of length 1. The inter-cluster distance is thus 10σ. (d,e,f) The spectrum of symmetric normalized graph Laplacian H corresponding to the data distributions in (a,b,c), respectively.

way. As a result of our choice of short-proximity adjacency measure, the largest contributions to $\mathrm{i}G(i,j;\mathrm{i}s)$ come from the low energy collective modes in the case of well-separated q clusters indexed by $\mu = 0, 1, \ldots, q-1$. In this case, the ground state density $|\psi_0(i)|^2 \propto \deg(i)$ will be accumulated around the hub nodes within each cluster. Furthermore, H is essentially q-block diagonal upon relabeling the nodes and exhibits a large energy gap separating the low energy collective modes $\{|\psi_n\rangle\}_{0\leq n<q}$ from the high energy eigenstates $\{|\psi_n\rangle\}_{q\leq n<m}$ capturing microscopic fluctuations within each cluster. Notice that the major contribution to the resolvent in Eq. 4.7.1 comes from terms with $n < q$, and that the number of low energy states equals the number of well-separated

clusters (Fig. 4.6.1). These observations thus motivate a q-dimensional coarse-grained Hamiltonian describing only the low energy collective modes.

In the extreme case where the clusters are completely separated from each other, the Hamiltonian H is strictly in q diagonal blocks; each block governs the dynamics within a cluster and has its own ground state wave function $\phi_\mu(i) = \langle i|\phi_\mu\rangle$, which is positive for node i belonging to the μ-th cluster and zero otherwise. We have $H|\phi_\mu\rangle = \xi_\mu|\phi_\mu\rangle$ and $\langle\phi_\mu|\phi_\nu\rangle = \delta_{\mu\nu}$ for all $\mu, \nu = 0, 1, \ldots, q-1$. As we gradually turn on off-diagonal couplings $v_{\mu\nu} = \langle\phi_\mu|H|\phi_\nu\rangle$ between clusters $\mu \neq \nu$, the wave functions ϕ_μ are no longer eigenstates of H. The effective tight-binding model assumes that in the weak coupling limit, we can project H onto the subspace spanned by $\{\phi_\mu\}_{\mu=0}^{q-1}$ and diagonalize the projected Hamiltonian $h_{\mu\nu} = \langle\phi_\mu|H|\phi_\nu\rangle$ to approximate the first q lowest energy eigenstates.

Let $\{\phi_\mu\}_{\mu=0}^{q-1}$ be the cluster wave functions, or "atomic orbitals," satisfying $\phi_\mu(i) > 0$ for i in cluster μ and zero elsewhere, and $\langle\phi_\mu|\phi_\nu\rangle = \delta_{\mu\nu}$. The effective tight-binding Hamiltonian is

$$\hat{H} \equiv \sum_{\mu,\nu=0}^{q-1} h_{\mu\nu}|\phi_\mu\rangle\langle\phi_\nu|, \text{ and } h_{\mu\nu} \equiv \xi_\mu\delta_{\mu\nu} + v_{\mu\nu}, \tag{4.7.2}$$

where $\xi_\mu = \langle\phi_\mu|H|\phi_\mu\rangle$ describes the ground state energy of each ϕ_μ, and the off-diagonal matrix $v_{\mu\nu} = \langle\phi_\mu|H|\phi_\nu\rangle$ for $\mu \neq \nu$, with $v_{\mu\mu} = 0$, couples the atomic orbitals ϕ_μ and ϕ_ν. Through the diagonalization of the tight-binding Hamiltonian $h_{\mu\nu}$, the q atomic orbitals are then linearly combined into q molecular orbitals.

To illustrate the effects of off-diagonal coupling, we split $\hat{H}$ into diagonal $\hat{H}_0$ and off-diagonal $\hat{V}$, and study the Born approximation of the Lippmann-Schwinger equation

$$\hat{G}(z) = \hat{G}_0(z) + \hat{G}_0(z)\hat{V}\hat{G}(z), \tag{4.7.3}$$

where $\hat{G}(z) = (z - \hat{H})^{-1}$ and $\hat{G}_0(z) = (z - \hat{H}_0)^{-1}$. The resolvent matrix $g_{\mu\nu}$ of $h_{\mu\nu}$ is defined through

$$g^{-1}(z)_{\mu\nu} = z\delta_{\mu\nu} - h_{\mu\nu}.$$

The resolvent matrix can be expanded if $|v_{\mu\nu}| < |z - \xi_\nu|$, for all $\mu, \nu = 0, 1, \ldots, q-1$, as

$$g_{\mu\nu}(z) = \frac{\delta_{\mu\nu}}{z - \xi_\mu} + \frac{v_{\mu\nu}}{(z - \xi_\mu)(z - \xi_\nu)} + \sum_\sigma \frac{v_{\mu\sigma} v_{\sigma\nu}}{(z - \xi_\mu)(z - \xi_\sigma)(z - \xi_\nu)} + \cdots \tag{4.7.4}$$

$$+ \sum_{\sigma,\rho} \frac{v_{\mu\sigma} v_{\sigma\rho} v_{\rho\nu}}{(z - \xi_\mu)(z - \xi_\sigma)(z - \xi_\rho)(z - \xi_\nu)} + \mathcal{O}(v^4). \tag{4.7.5}$$

Note that the resolvent matrix is thus a weighted sum over all possible tunneling paths between the q clusters.

The propagator from node j to i in the effective tight-binding theory, approximating Eq. 4.7.1, is directly related to $g_{\mu\nu}(z)$ as

$$g(i, j; z) = \sum_{\mu,\nu=0}^{q-1} \phi_\mu(i) g_{\mu\nu}(z) \phi^*_\nu(j). \tag{4.7.6}$$

If the nodes i and j belong to two non-overlapping clusters μ and ν, respectively, then the propagator reduces to $g(i, j; z) = \phi_\mu(i)\phi_\nu(j) g_{\mu\nu}(z)$ and $\arg g(i, j; z) = \arg g_{\mu\nu}(z)$, because of the disjoint support and the non-negativity of cluster wave functions. In other words, the propagator initiated at j has a constant phase at all nodes i within each cluster, and the phase associated with each cluster is completely determined by the phase of resolvent matrix $g_{\mu\nu}$, which in turn depends on the weak coupling $v_{\mu\nu}$ via Eq. 4.7.4.

As an example, consider two sets of m samples drawn from $\mathcal{N}((\pm\ell, 0)^\top, \sigma^2 \mathbf{1}_{2\times 2})$,

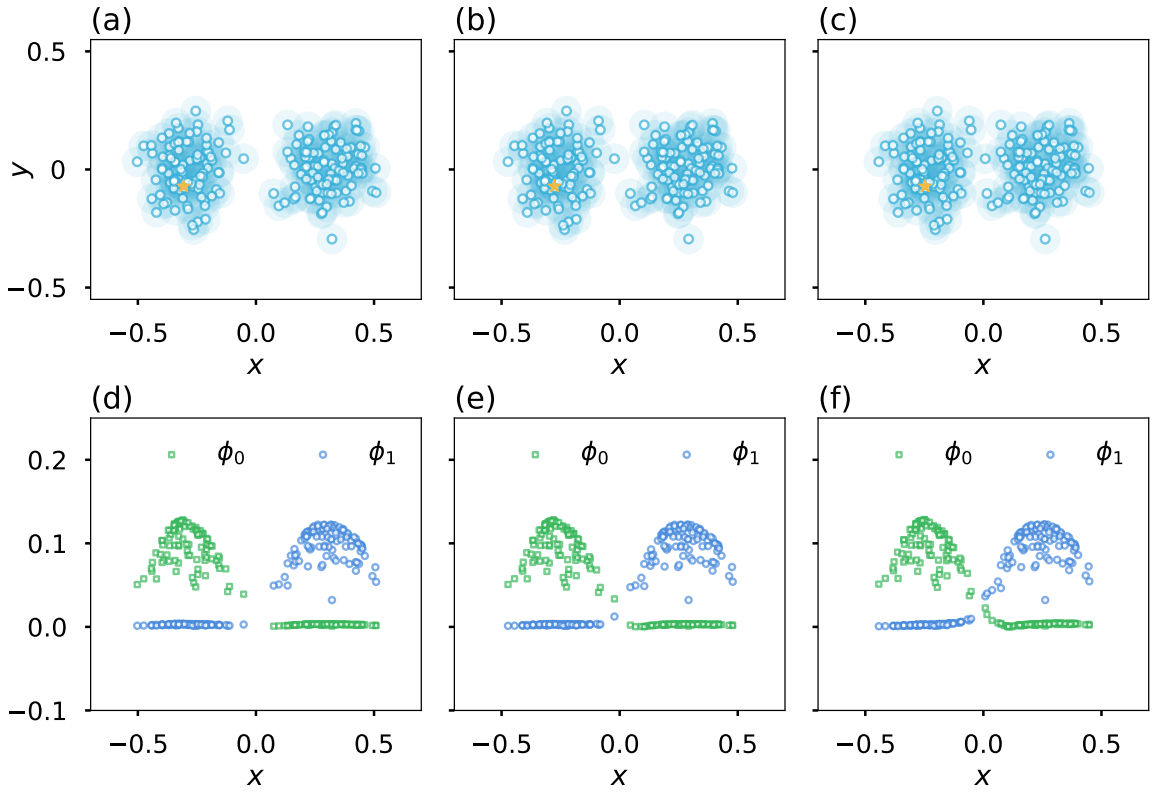

Figure 4.7.1: (a-c) Two-cloud distributions corresponding to Fig 4.7.2(a-c). (d-f) Cluster wave functions used to compute the theoretical predictions in Fig. 4.7.2(d-f).

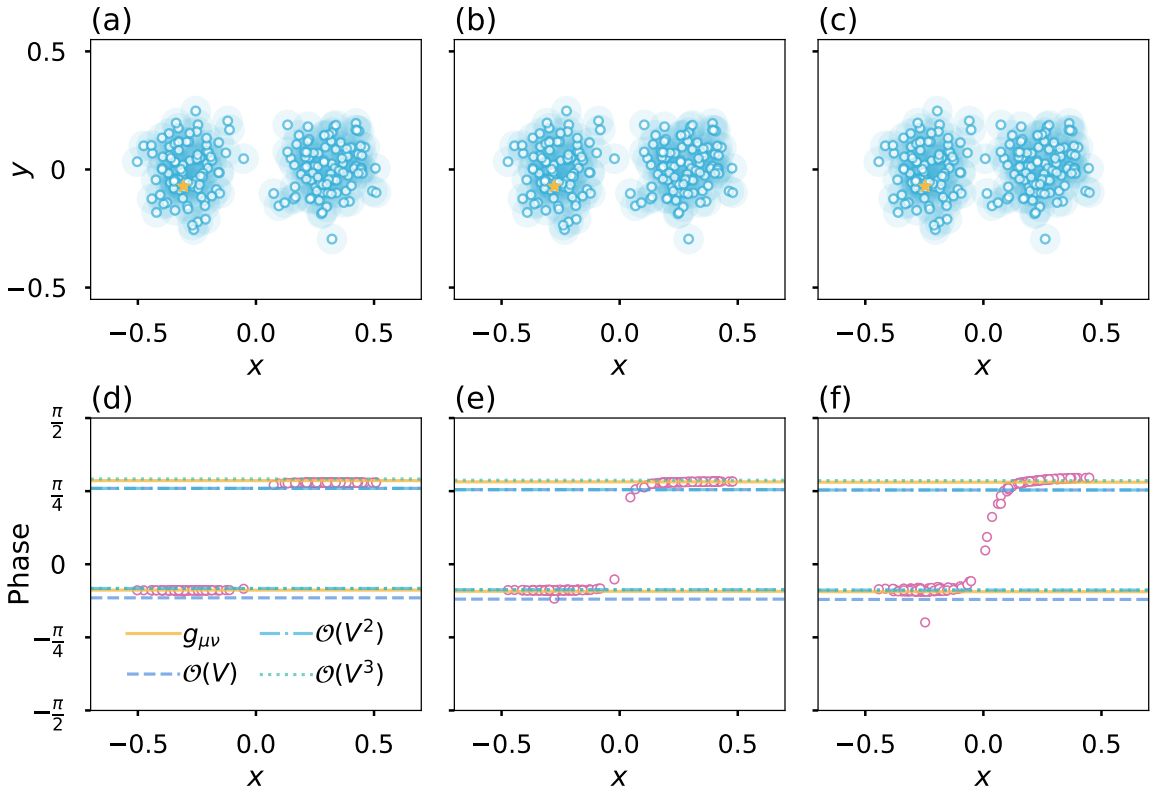

Figure 4.7.2: Two Gaussian clouds from $\mathcal{N}((\pm\ell, 0)^\top, \sigma^2 \mathbf{1}_{2\times 2})$ with variations in the center-to-center distance (a) $\ell = 3\sigma$, (b) $\ell = 2.7\sigma$, and (c) $\ell = 2.4\sigma$. Adjacency matrices were calculated using $r_\varepsilon = \sigma$. The radius of the faint large circle around each data point indicates $r_\varepsilon/2$. (d-f) The phase distributions (red circles) of all sample points from (a-c), respectively; exact theoretical predictions $\arg\{\mathrm{i}g_{\mu\nu}(\mathrm{i}s)\}$ from the low-energy effective model (solid line); the first, second, and third order perturbative approximations (dashed lines). The Laplace transform parameter was set to $s = 1.2(E_1 - E_0)$. The $\star$ in (a), (b), and (c) mark the initialization nodes.

respectively. The effective 2-level Hamiltonian and resolvent matrices are

$$h = \begin{pmatrix} \xi_0 & v \\ v & \xi_1 \end{pmatrix}, \text{ and } g(z) = \begin{pmatrix} z - \xi_0 & -v \\ -v & z - \xi_1 \end{pmatrix}^{-1}. \tag{4.7.7}$$

As we vary $\ell = 3\sigma, 2.7\sigma$, and 2.4σ, with a fixed proximity length scale $r_\varepsilon = \sigma$, the cluster configuration ranges from (a) well-separated, (b) in proximity, and (c) overlapping (Fig. 4.7.2; Fig. 4.7.1). For each case, Fig. 4.7.2(d-f) show the phase distribution of all samples when quantum transport is initialized at one of the nodes in the left cluster; it is seen that our theoretical prediction $\arg\{\mathrm{i}g_{\mu\nu}(\mathrm{i}s)\}$ and its perturbative approximations calculated from Eq. 4.7.4 agree well. Furthermore, if the two clusters are identical, i.e. $\xi_0 = \xi_1$, then the effective 2-level model can be mapped to the classic double-well tunneling model; in this case, the phase distribution of the Laplace transform of exact instanton solution matches that of our simulated Gaussian clouds (Fig. 4.8.1(a)). When the weak coupling assumption is not satisfied, the low-energy theoretical predictions serve only as asymptotic limits, and some ambiguous points in a strongly mixed region may have a phase that interpolates between the theoretical predictions (Fig. 4.7.2(c,f), Fig. 4.8.2, & Fig. 4.8.3).

4.8 Two-level toy model

Consider the case of two Gaussian clusters in $\mathbb{R}^2$ with mean at $(\pm\ell, 0)^\top$, as shown in Fig. 4.7.2(a-c) and Fig. 4.7.1(a-c). We expect two low energy states, i.e., the ground state and the first excited state (Fig. 4.8.1(b)). Let ϕ_0 and ϕ_1 denote the cluster wave functions for the left and right Gaussian clouds, respectively. Assuming that the two

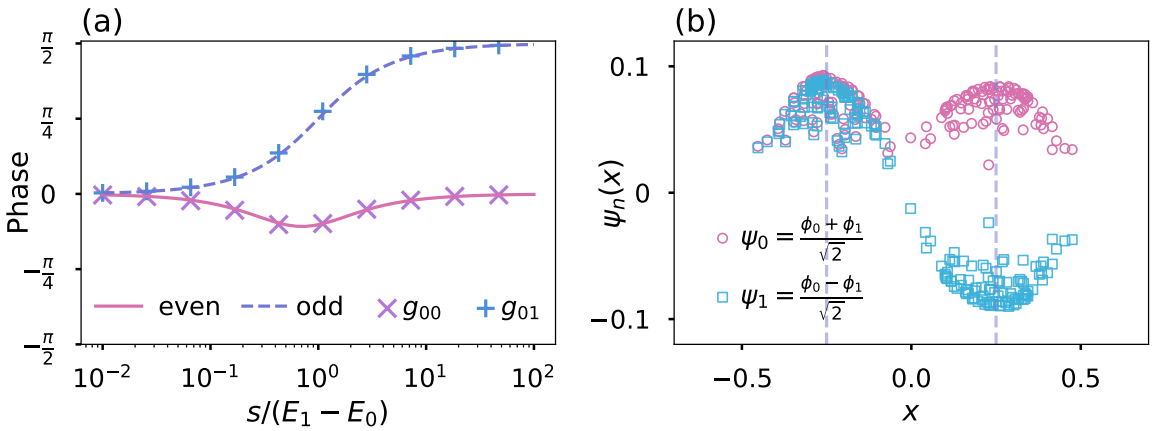

Figure 4.8.1: (a) The phase distribution of the Laplace transform of exact instanton solution (solid and dashed lines represent G_{00} and G_{01}, respectively). Also plotted are the phases calculated from our two simulated Gaussian clouds $\mathcal{N}((\pm\ell, 0)^\top, \sigma^2 \mathbf{1}_{2\times 2})$, with $\ell = 0.25$, $\sigma = 0.1$, and equal sample size $m = 100$ ($\times$ and $+$). (b) Plots of the ground state ψ_0 and the first excited state ψ_1 wave functions derived from the simulated data.

clusters have the same ground state energy, the ground state ψ_0 and the first excited state ψ_1 of the tight-binding Hamiltonian are

$$|\psi_0\rangle = \frac{|\phi_0\rangle + |\phi_1\rangle}{\sqrt{2}}, \quad |\psi_1\rangle = \frac{|\phi_0\rangle - |\phi_1\rangle}{\sqrt{2}}. \tag{4.8.1}$$

Setting the ground state energy $E_0 = 0$, and defining the first energy gap $E \equiv E_1 - E_0$, we have

$$|\tilde{\psi}(s)\rangle = \frac{1}{s + \mathrm{i}H}|\psi(0)\rangle \approx \frac{c_0|\psi_0\rangle}{s} + \frac{c_1|\psi_1\rangle}{s + \mathrm{i}E}, \tag{4.8.2}$$

where $c_j = \langle\psi_j|\psi(0)\rangle$. Thus,

$$|\psi(s)\rangle = \frac{\left(\frac{c_0}{s} + \frac{c_1}{s+\mathrm{i}E}\right)|\phi_0\rangle + \left(\frac{c_0}{s} - \frac{c_1}{s+\mathrm{i}E}\right)|\phi_1\rangle}{\sqrt{2}}, \tag{4.8.3}$$

from which we easily extract the phase in the left and right clusters to be

$$\Theta_0 = \arg\left(\frac{c_0}{s} + \frac{c_1}{s + \mathrm{i}E}\right) \tag{4.8.4}$$

$$= \arctan\frac{Ec_0}{(c_0 + c_1)s} - \arctan\frac{E}{s}, \tag{4.8.5}$$

$$\Theta_1 = \arg\left(\frac{c_0}{s} - \frac{c_1}{s + \mathrm{i}E}\right) \tag{4.8.6}$$

$$= \arctan\frac{Ec_0}{(c_0 - c_1)s} - \arctan\frac{E}{s}. \tag{4.8.7}$$

If the initial state $\psi(0)$ is a delta function located deep in the (1) left or (2) right cluster, then (1) $c_0 = c_1$ or (2) $c_0 = -c_1$, respectively. The phases of the left and right clusters in case (1) are

$$\Theta_{00} = \arctan\frac{E}{2s} - \arctan\frac{E}{s} \tag{4.8.8}$$

$$\Theta_{01} = \frac{\pi}{2} - \arctan\frac{E}{s}; \tag{4.8.9}$$

while in case (2), the phases are

$$\Theta_{10} = \frac{\pi}{2} - \arctan\frac{E}{s} \tag{4.8.10}$$

$$\Theta_{11} = \arctan\frac{E}{2s} - \arctan\frac{E}{s}. \tag{4.8.11}$$

Notice that $\Theta_{\mu\nu}$ is a constant diagonal symmetric matrix that preserves the left-right symmetry.

The two-cluster model can be mapped to the classic double-well instanton tunneling model which will be briefly summarized below; detailed derivations can be found in

[47]. The model Hamiltonian is

$$H = -\frac{1}{2}\partial_x^2 + \lambda(x^2 - \ell^2)^2, \tag{4.8.12}$$

where $\lambda > 0$. The potential $V(x) = \lambda(x^2-\ell^2)^2$ has two minima at $x = \pm\ell$ for $\ell > 0$ and one minimum at $x = 0$ for $\ell = 0$. The barrier height is $V(0) = \lambda\ell^4$ which grows rapidly with the separation distance ℓ. In the vicinity of minima, $V(\pm\ell+\varepsilon) = \lambda(\pm 2\varepsilon\ell+\varepsilon^2)^2 = 4\lambda\ell^2\varepsilon^2 + \mathcal{O}(\varepsilon^3)$; the local harmonic frequency is thus $\omega = 2\ell\sqrt{2\lambda}$ and $V(0) = \omega^4/64\lambda$.

In the limit $\lambda \downarrow 0$ while keeping ω constant, the barrier is infinite, and the ground state is two-fold degenerate with harmonic ground state energy $E_0 = \frac{1}{2}\omega$ and expected position $\langle x\rangle = \pm\ell$. For any finite barrier, however, we should have $\langle x\rangle = 0$, which is enforced by symmetry; the symmetric solution cannot be obtained via perturbation around either of the local minima.

Non-perturbative instanton solution splits the degeneracy:

$$E_0 = \frac{\omega}{2}\left(1 - 2\sqrt{\frac{\omega^3}{2\pi\lambda}}\mathrm{e}^{-\omega^3/12\lambda}\right), \tag{4.8.13}$$

$$E_1 = \frac{\omega}{2}\left(1 + 2\sqrt{\frac{\omega^3}{2\pi\lambda}}\mathrm{e}^{-\omega^3/12\lambda}\right). \tag{4.8.14}$$

The transition amplitudes are

$$\langle +\ell|\mathrm{e}^{-\mathrm{i}Ht}|-\ell\rangle = \mathrm{i}\sqrt{\frac{\omega}{\pi}}\mathrm{e}^{-\mathrm{i}\omega t/2}\sin(\omega\rho_{\mathrm{inst}}t) \tag{4.8.15}$$

$$\langle -\ell|\mathrm{e}^{-\mathrm{i}Ht}|-\ell\rangle = \sqrt{\frac{\omega}{\pi}}\mathrm{e}^{-\mathrm{i}\omega t/2}\cos(\omega\rho_{\mathrm{inst}}t), \tag{4.8.16}$$

where the instanton density $\rho_{\mathrm{inst}} = \sqrt{\frac{\omega^3}{2\pi\lambda}}\mathrm{e}^{-\omega^3/12\lambda}$. Notice that the energy gap is $E =$

$2\omega\rho_{\text{inst}}$; thus,

$$\langle \pm\ell | \mathrm{e}^{-\mathrm{i}Ht} | -\ell \rangle = \sqrt{\frac{\omega}{\pi}} \mathrm{e}^{-\mathrm{i}\omega t/2} \frac{\mathrm{e}^{\mathrm{i}Et/2} \mp \mathrm{e}^{-\mathrm{i}Et/2}}{2} \tag{4.8.17}$$

$$= \sqrt{\frac{\omega}{\pi}} \frac{\mathrm{e}^{-\mathrm{i}E_0 t} \mp \mathrm{e}^{-\mathrm{i}E_1 t}}{2} \tag{4.8.18}$$

$$= \sqrt{\frac{\omega}{\pi}} \mathrm{e}^{-\mathrm{i}E_0 t} \frac{1 \mp \mathrm{e}^{-\mathrm{i}Et}}{2}. \tag{4.8.19}$$

If we reset the ground state energy to zero, the Laplace transform of Eq. 4.8.19 yields the resolvent matrix elements

$$g_{00}(\mathrm{i}s) = \frac{1}{2}\sqrt{\frac{\omega}{\pi}} \left(\frac{1}{s} + \frac{1}{s + \mathrm{i}E} \right), \tag{4.8.20}$$

$$g_{01}(\mathrm{i}s) = \frac{1}{2}\sqrt{\frac{\omega}{\pi}} \left(\frac{1}{s} - \frac{1}{s + \mathrm{i}E} \right), \tag{4.8.21}$$

where 0 and 1 denote the states localized at $x = -\ell$ and $x = +\ell$, respectively. The phases are thus

$$\Theta_{00}(s) = \arctan \frac{E}{2s} - \arctan \frac{E}{s}, \tag{4.8.22}$$

$$\Theta_{01}(s) = \frac{\pi}{2} - \arctan \frac{E}{s}. \tag{4.8.23}$$

Note that the above phase distribution is exactly the same as that from the low-energy two-cluster model (Eq. 4.8.22) upon identifying the energy gaps.

The phase separation between the diagonal and off-diagonal elements of the resolvent is $\pi/2 - \arctan \frac{E}{2s}$, and this difference is thus controlled by the ratio s/E. In other words, the Laplace transform parameter s controls the separability between clusters in the QTC algorithm. For $s \ll E$, $s = E/2$, or $s \gg E$, the phase differences are 0, $\pi/4$,

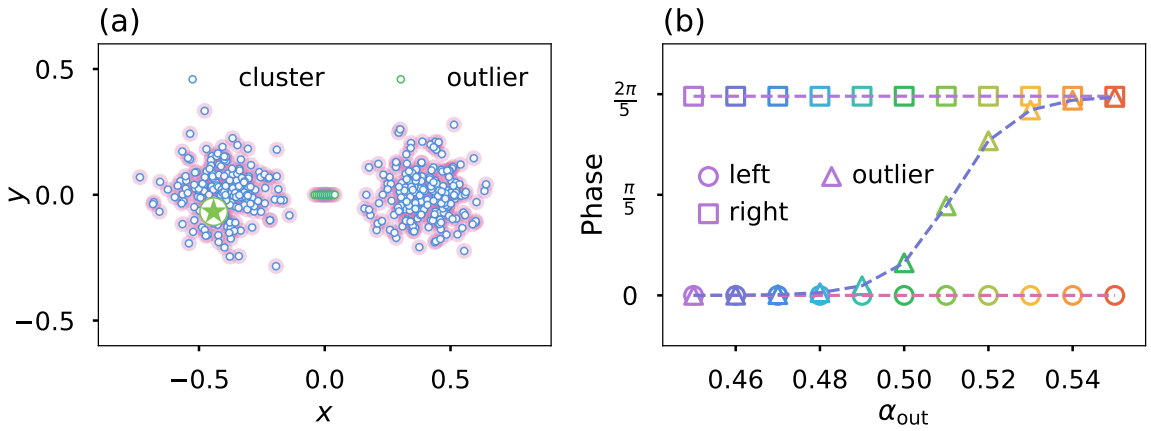

Figure 4.8.2: (a) Two Gaussian clusters were drawn from $\mathcal{N}((\pm\ell, 0)^\top, \sigma^2 \mathbf{1}_{2\times 2})$ with $\sigma = 0.1$, sample size $m = 100$, and $\ell = 0.4$ chosen to yield proximity $r_{5\%} \approx \sigma$; the outlier was located at $(-\ell(1-\alpha_{\text{out}}) + \ell\alpha_{\text{out}}), 0)^\top$ between the two clusters. (b) The quantum transport was initialized from a node in the left cluster (marked with $\star$). The phases of the left and right clusters, averaged over their respective nodes, and the phase of the outlier are plotted against α_{out}, with the left cluster phases set to zero.

or $\pi/2$, respectively. Fig. 4.8.1(a) shows the phases Θ_{00} and Θ_{01} for different values of s/E in the range $[10^{-2}, 10^2]$, suggesting that s should be chosen to be at least as large as the energy gap E.

In practice, for an ambiguous point located between two clusters, its phase interpolates smoothly between the cluster phases. Figure 4.8.2(b) shows the phases of the outlier for QTC initialized from a point deep in the left cluster. Moreover, Figure 4.8.3(b) shows the mean phases of the left and right clusters for QTC initialized at an outlier located at $(-\ell(1-\alpha_{\text{out}}) + \ell\alpha_{\text{out}}), 0)^\top$, and it demonstrates that a wave function initialized from an ambiguous point loses contrast between the two clusters.

Similarly, for cases involving more than two clusters, the full Θ-matrix for all nodes essentially amounts to the effective tight-binding matrix $\arg(\mathrm{i}g_{\mu\nu}(\mathrm{i}s))$. Our experience shows that choosing s based on the average gap, $E = (E_{q-1} - E_0)/(q-1)$, still provides

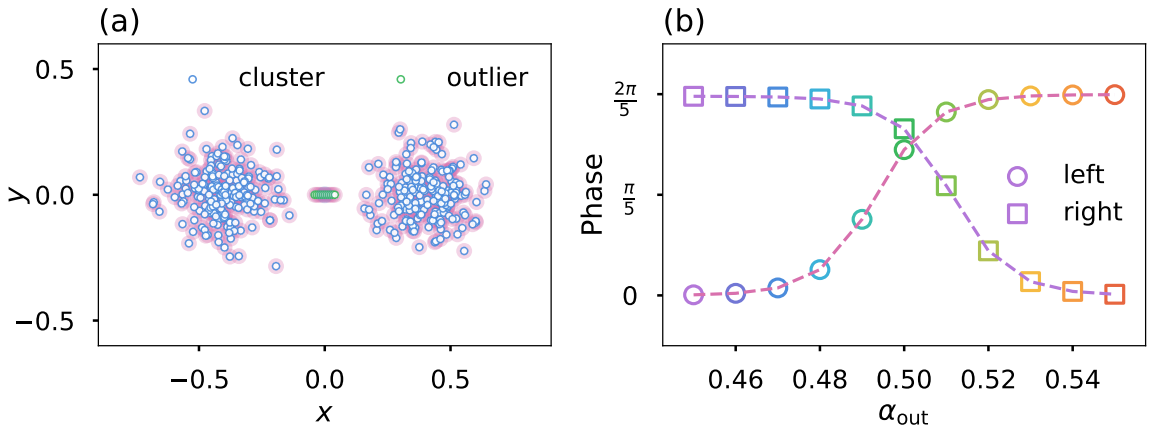

Figure 4.8.3: (a) Two Gaussian clusters were drawn from $\mathcal{N}((\pm\ell, 0)^\top, \sigma^2 \mathbf{1}_{2\times 2})$ with $\sigma = 0.1$, sample size $m = 100$, and $\ell = 0.4$ chosen to yield proximity $r_{5\%} \approx \sigma$; the outlier was located at $(-\ell(1-\alpha_{\text{out}}) + \ell\alpha_{\text{out}}), 0)^\top$ between the two clusters. (b) The quantum transport was initialized from the outlier, and the averaged phases of the left and right clusters are plotted against α_{out}.

a helpful guideline and yields good multi-class clustering results.

4.9 The algorithm

In applications, we numerically calculate the Laplace transform of a wave function initialized at a given node and then extract the phase distribution. As above, we will assume that the total number of nodes is m and the *a priori* determined number of clusters is q. The phases of nodes belonging to different clusters are typically separated by gaps, allowing us to assign discrete class labels to nodes. We propose two methods for converting the phases to class labels $0, 1, \ldots, q-1$: (Method 1) direct difference, and (Method 2) clustering. The steps in Method 1 are as follows:

Method 1

1. Sort the array $(\theta_0, \ldots, \theta_{m-1})$ of phases in ascending order. Let $\pi(i)$ denote the rank of the phase of node i in this sorted list.

2. Denote the j-th element in the sorted list as $\theta_{(j)}$ and compute $\hat{n}_j = (\cos\theta_{(j)}, \sin\theta_{(j)})^\top \in \mathbb{R}^2$, for $j = 0, \ldots, m-1$.

3. Compute the local difference $r_j = \|\hat{n}_{j+1} - \hat{n}_j\|$, for $j = 0, 1, \ldots, m-2$[2]

4. Locate the $q-1$ largest values in the array $(r_0, \ldots, r_{m-2})$ and return their indices $\{I_j\}_{j=1}^{q-1}$, where$I_j < I_{j+1}$.

5. Assign the class label j to node i iff $I_j < \pi(i) \leq I_{j+1}$, where $I_0 = -1$ and $I_q = m-1$.

The steps in Method 2 are as follows:

Method 2

1. Map each node i to $\hat{n}_i = (\cos\theta_i, \sin\theta_i)^\top \in \mathbb{R}^2$.

2. Apply a standard clustering algorithm in $\mathbb{R}^2$, e.g., k-means or k-medoids.

3. Return the class label for each node.

The first method is faster than the second method. However, when the clusters are not clearly separable it might recognize false cluster boundaries and produce fragmented clustering. We find that the second method is more robust.

[2]We did not use arc length, or the geodesic distance on S^1 in that arc length is sensitive to fluctuations in phase distribution; however, our goal is to pick out the largest jumps in phases and ignore small jumps which may arise around large jumps.

Using either Method 1 or Method 2, we are thus able to convert the phase distribution of a Laplace transformed wave function initialized at a single node to a set of discrete class labels. When we change the initialization node, some of the cluster boundaries can change. To improve clustering accuracy and reduce variation in clustering, we thus iterate QTC at multiple nodes; let m' denote this number of initialization nodes. The clustering results then form an ensemble of class labels, organized into a matrix (Ω_{ij}), where $i = 0, 1, \ldots, m-1$ runs through all nodes and $j = 0, 1, \ldots, m'-1$ indexes the iteration of initialization.

Notice that the class labels may get permuted across different initialization. We introduce two methods to handle this issue and summarize the Ω-matrix: (1) direct extraction, and (2) consensus matrix.

4.9.1 Direct Extraction

We want to count the multiplicity of the columns of Ω, up to permutation of class labels; i.e., two columns are considered equivalent if they are equal upon permuting the class labels. We will then choose the most frequent column vector as the desired partition of nodes. For this purpose, we first devise a scheme for testing whether a subset of columns are all equivalent. Let $\{p_i\} = \{2, 3, 5, 7, \cdots\}$ be the set of primes, then $\{\sqrt{p_i}\}$ is a set of irrational numbers serving as linearly independent vectors over the field $\mathbb{Q}$ of rational numbers. Let A be an index set containing at least two column indices of Ω. For each node i, we then compute the quantity $\xi_i = \sum_{k\in A} \Omega_{ik}\sqrt{p_k}$. For any two nodes i and j,

$$\xi_i - \xi_j = \sum_{k\in A} (\Omega_{ik} - \Omega_{jk}) \sqrt{p_k} \equiv \sum_{k\in A} b_k \sqrt{p_k}. \tag{4.9.1}$$

Suppose i and j are in the same cluster for all $k \in A$, then $b_k = 0$ for all k, and thus $\xi_i = \xi_j$; the converse is also true, because $\{\sqrt{p_i}\}$ are linearly independent over $\mathbb{Q}$. Thus, $\xi_i = \xi_j$ iff node i and node j are assigned to the same class by all columns indexed by A. The minimum number of distinct ξ_i is q, since any column of Ω partitions the nodes into q clusters. If the number of distinct ξ_i exceeds q, then there thus exists at least two columns that disagree on the partition, so the columns indexed by A are not all equivalent. Our algorithm including this scheme is as follows:

Ensemble Method 1

1. Let $K = \{0, 1, \cdots, m' - 1\}$ be the full index set indexing the columns of Ω. Denote any non-empty subset of K as K', and let k'_0 denote the first column index appearing in K'.

2. **Define** function $\mathsf{IsEquiv}(\{\Omega_{ik}\}_{k \in K'})$ to tell whether the columns of Ω indexed by K' yield an *equivalent clustering*:

   - **For** $i = 0, 1, \ldots, m - 1$:
   $$\xi_i = \sum_{k \in K'} \Omega_{ik} \sqrt{p_k}$$
   - Count the number q' of distinct ξ_i
   - **If** $q' = q$, then **Return** True
   - **Else Return** False

3. Let H be a hash table with non-negative integer keys α indexing the equivalence classes of columns of Ω and values H_α equal to the corresponding index sets of equivalent columns. Each key α is chosen from H_α to represent the class.

4. **Define** function Pigeonhole($\{\Omega_{ik}\}_{k\in K'}$, H):

    - **If** IsEquiv($\{\Omega_{ik}\}_{k\in K'}$) = True, then:
        - **For** α in H:
            - **If** IsEquiv($\{\Omega_{ik}\}_{k=\alpha,k'_0}$) = True:
                - IsExisting = True
                - Merge K' and H_α
                - **Break** for-loop
        - **If** IsExisting = False:
            - Create a new key α' and $\mathsf{H}_{\alpha'} = K'$
    - **Else**: Split K' in two halves K'_1 and K'_2
        - **Call** H = Pigeonhole($\{\Omega_{ik}\}_{k\in K'_1}$, H)
        - **Call** H = Pigeonhole($\{\Omega_{ik}\}_{k\in K'_2}$, H)
    - **Return** H

5. **Call** Pigeonhole($\{\Omega_{ik}\}_{k\in K}$, H^0), where H^0 is an empty hash table

4.9.2 Consensus matrix

Even though the class labels may get randomly permuted for different initializations, whether two nodes share the same class label within each initialization is independent of the labeling convention. Therefore, we define a consensus matrix C with elements

$$C_{ij} = \frac{\sum_{k=1}^{m'} \delta(\Omega_{ik} - \Omega_{jk})}{m'}, \tag{4.9.2}$$

where δ is the Kronecker delta or indicator function, and $m' \leq m$ is the number of the chosen initialization nodes. Notice that $C_{ij} = C_{ji} \in [0, 1]$, and $C_{ii} = 1$ for all nodes $i, j = 1, 2, \ldots, m$. The algorithm is sketched as follows:

Ensemble Method 2

1. Initialize C as an $m \times m$ identity matrix.
2. **For** $i = 0, 1, \ldots, m-1$:
    - **For** $j = i+1, \ldots, m-1$:
        - **For** $k = 0, 1, \ldots, m'-1$:
            - **If** $\Omega_{ik} = \Omega_{jk}$: $C_{ij}++$
                - $C_{ji} = C_{ij}$
3. $C_{ij} = C_{ij}/m'$ for $i \neq j$

The consensus matrix measures the similarity of node pairs and facilitates the visualization of network structure, e.g., chromatin interaction information between distal genomic loci, as in Fig. 4.5.3. It can also be used as a similarity measure or dissimilarity measure, e.g., $\delta_{ij} - C_{ij}$, in (dis)similarity-based algorithms such as spectral clustering and hierarchical clustering.

4.10 Data preparation

4.10.1 Synthetic data sets

In general, for sufficiently small proximity r_ε in the synthetic data sets in Fig. 4.5.1 (b-d & f-h), spectral clustering was able to produce the clustering by QTC at longer proximity; for sufficiently larger proximity, both spectral clustering and QTC failed to recognize the putative clusters. Thus, there was a finite interval of ε for each data set in which QTC outperformed spectral clustering. For data sets in Fig. 4.5.1 (b-d & f-h), the intervals are approximately $[3.1\%, 3.9\%]$, $[0.61\%, 0.85\%]$, and $[0.39\%, 0.46\%]$ respectively.

4.10.2 Time series stock price data

The stock price data consisted of the "adjusted close" prices of the AAPL and GOOGL stocks between January 3, 2005 and November 7, 2017, downloaded from Yahoo Finance. We log transformed the data and subtracted the two time series by the respective log-prices on the first day (1-3-2005). We computed the pairwise Euclidean distance in $\mathbb{R}^2$ and took 1%-quantile of the distance distribution as the proximity length $r_{1\%} = 0.05$. Next, we assembled the Gaussian similarity measure $A_{ij} = \exp[-(r_{ij}/r_{1\%})^2]$ and performed QTC and spectral clustering; the number of clusters was chosen to be five. Spectral clustering was able to produce the clustering by QTC at 1%-quantile only for shorter proximity lengths where $\varepsilon \sim [0.2\%, 0.5\%]$, and for $\varepsilon \lesssim 0.1\%$, clusters started to become disjoint.

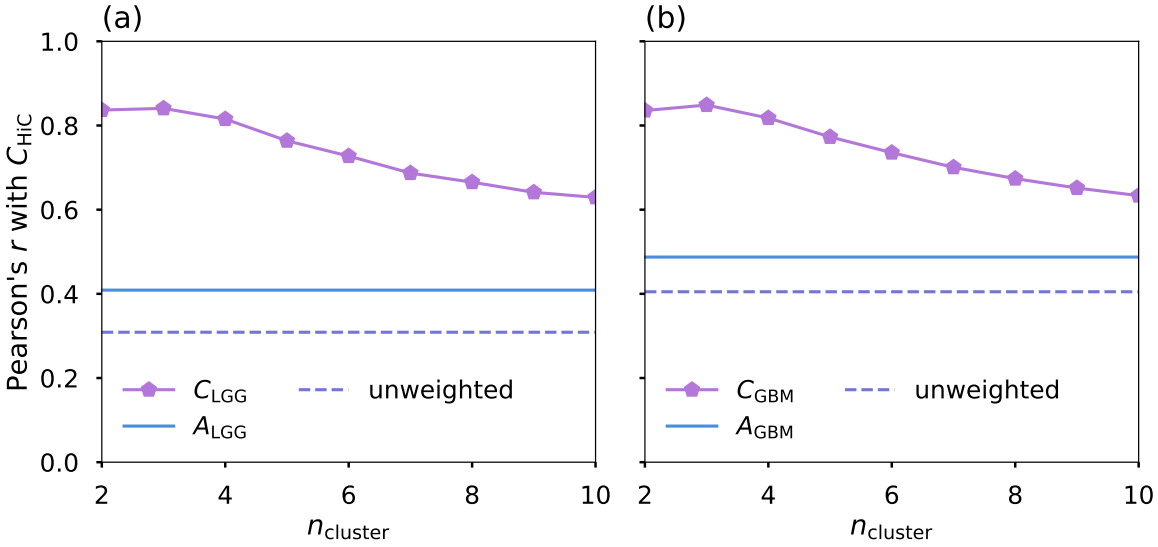

Figure 4.10.1: Pearson correlation coefficients between the tanh-normalized HiC matrix and various similarity measures. For (a) LGG and (b) GBM samples, respectively, correlations were computed using the "unweighted" raw counts N_{ij} of SCNA labeled by genomic location pair (i, j), the weighted adjacency $(A_{\mathrm{LGG/GBM}})_{ij} = N_{ij}w_{ij}$ with Gaussian weight $w_{ij} = \exp(-(r_{ij}/r_{\varepsilon})^2)$, and the QTC consensus matrix $C_{\mathrm{LGG/GBM}}$ calculated assuming a different number of clusters. Both weighted and unweighted similarity matrices were tanh-normalized.

4.10.3 Genomic data

The TCGA somatic copy number alteration (SCNA) data in low-grade glioma (LGG) and glioblastoma (GBM) patient samples were downloaded from the GDC Data Portal under the name "LGG/GBM somatic copy number alterations." To link these data to chromatin contact information, we followed the analysis described in [25]. We partitioned the genome into 1Mb bins and defined N to be a null square matrix of dimension equal to the total number of bins. For each amplified or deleted genomic segment starting at the i-th bin and ending at the j-th bin, we then incremented the (i, j)-th entry of N by 1. The main idea behind this analysis is that genomic amplification and deletion events are mediated by the physical co-location of the segment junctions. The raw count matrix N was thus to be compared with the HiC chromatin contact matrix. In cancer samples, however, an entire arm of a chromosome or even a whole chromosome can be duplicated or deleted, potentially leading to fictitious long-range off-diagonal elements in N. Therefore, we weighted the counts N_{ij} by $w_{ij} = \exp[-(r_{ij}/r_\varepsilon)^2]$ where r_{ij} is the genomic distance between the bins and $r_\varepsilon = 10\text{Mb}$. Using this weighted matrix as an adjacency matrix, we performed QTC with $s = 5(E_1 - E_0)$, assuming the number of clusters to be $q = 2, 3, 4, 5$, and computed the respective consensus matrices $C(q)$. Finally, we took the arithmetic mean $\langle C \rangle = \sum_{q=2}^{5} C(q)/4$.

The HiC data in normal human astrocytes of the cerebellum (glial cells) were downloaded from ENCODE under the name "ENCSR011GNI" [18]. We extracted the 3D interaction maps on chromosome 2 at 1Mb resolution. The distribution of HiC contact matrix entries was highly heavy-tailed. In order to compare C_{HiC} with $\langle C_{ij} \rangle \in [0, 1]$, we transformed C_{HiC} using $\tanh(C_{\text{HiC}}/\bar{C}_{\text{HiC}}) \in [0, 1)$, where $\bar{C}_{\text{HiC}}$ was the mean of all C_{HiC} entries. Next, we computed the Pearson correlation coefficients

between the transformed C_{HiC} and averaged $\langle C(q)_{ij} \rangle$.

4.11 Comparison with other methods

In this section, we first discuss spectral embedding and then derive three additional (dis)similarity measures using quantum mechanics. These measures can be combined with spectral clustering as well as other (dis)similarity-based learning algorithms.

4.11.1 Spectral embedding

The state-of-the art spectral clustering can be decomposed into three major steps: (1) assemble an affinity matrix A based on some similarity measure of sample points, (2) compute the symmetric normalized graph Laplacian H, and (3) map each sample point indexed by $i = 0, 1, \ldots, m-1$ to a Euclidean feature space using the corresponding elements of eigenvectors of the graph Laplacian; this mapping is called the spectral embedding. The first two steps are essentially the same as those of QTC; the key difference lies in the final usage of "spectral properties" of the data set. A single iteration of QTC succinctly represents the data on S^1, which we have shown is sufficient to separate distinct clusters.

By contrast, spectral embedding maps data samples to $\mathbb{R}^q$, where q is the number of putative clusters, or the number of low energy states if all putative clusters are clearly separable; then, the algorithm performs clustering, e.g., using k-means in the feature space $\mathbb{R}^q$. The feature vector $\mathbf{v}_i$ associated with the i-th sample has elements

$$(\mathbf{v}_i)_n = \psi_n(i) = \langle i|\psi_n\rangle, \quad n = 0, 1, \ldots, q-1, \tag{4.11.1}$$

where the ψ_n's are the first q lowest-eigenvalue eigenvectors of H. The L^2 Euclidean distance between nodes (i, j) is then

$$\begin{aligned}\mathcal{D}_{ij} =& \sqrt{\|\mathbf{v}_i - \mathbf{v}_j\|^2} \\ =& \sqrt{\sum_{n=0}^{q-1} |\psi_n(i) - \psi_n(j)|^2} \\ =& \sqrt{\sum_{n=0}^{q-1} (\langle i| - \langle j|)|n\rangle\langle n|(|i\rangle - |j\rangle)}\,. \end{aligned} \tag{4.11.2}$$

Note that if we actually used all eigenvectors of H, then $\mathcal{D}_{ij} = \sqrt{2(1-\delta_{ij})}$, i.e., each point is equally far away from any other node. Thus, the useful clustering information originates from the projection to low energy states,

$$\mathcal{D}_{ij} = \sqrt{(\langle i| - \langle j|)\mathcal{P}_{n<q}(|i\rangle - |j\rangle)} \tag{4.11.3}$$

$$\equiv \sqrt{\chi_{ii} + \chi_{jj} - \chi_{ij} - \chi_{ji}}, \tag{4.11.4}$$

where $\chi_{ij} = \langle i|\mathcal{P}_{n<q}|j\rangle \equiv \sum_{n<q} \psi_n(i)\psi_n^*(j)$.

In real data, the number of nodes as well as the distribution of node density could vary from one cluster to another. If a network is embedded in $\mathbb{R}^d$, then high density regions contain hub nodes, provided the adjacency A_{ij} is measured with a non-negative function that decreases with increasing distance r_{ij}, e.g., Gaussian function $A_{ij} = \exp(-r_{ij}^2/r_\varepsilon^2)$. For networks not embedded in $\mathbb{R}^d$, the "density" distribution should be interpreted as the degree distribution. We next illustrate how the spectral embedding distance $\mathcal{D}_{ij}$ responds to outliers in the presence of density variations using the simple two-cluster model.

Using the same notation as above, the ground state and first excited state, shown in Fig. 4.11.1(a,b), are $\psi_0 = \alpha\phi_0 + \beta\phi_1$ and $\psi_1 = \beta\phi_0 - \alpha\phi_1$, where $\alpha, \beta > 0$, and $\alpha^2 + \beta^2 = 1$. If we assume ϕ_0 and ϕ_1 are orthonormal, i.e., $\langle\phi_\mu|\phi_\nu\rangle = \delta_{\mu\nu}$ for $\mu, \nu = 0$, and 1, then $\langle\psi_n|\psi_{n'}\rangle = \delta_{nn'}$ for $n, n' = 0$, and 1. To simplify calculations, we further assume that ϕ_0 and ϕ_1 have identical shapes with the maximum value h located at node i and j, respectively; i.e., $\phi_0(i) = h = \phi_1(j)$. Then, $\psi_0(i) = \alpha h = -\psi_1(j)$ and $\psi_1(i) = \beta h = \psi_0(j)$. Let $\gamma \in (0, 1]$ such that $\phi_0(k) = \gamma\phi_0(i) = \gamma h$. Then, $\psi_0(k) = \gamma\alpha h$, and $\psi_1(k) = \gamma\beta h$ (Fig. 4.11.1(a,b)). Recall that $\psi_0(i) = \sqrt{\deg(i)}$ for a normalized symmetric Laplacian; hence, the differences in ψ_0 across nodes can be viewed as capturing the density variations in a network.

Simple calculations show that

$$\chi_{ii} = \chi_{jj} = h^2\left(\alpha^2 + \beta^2\right) = h^2 \tag{4.11.5}$$

$$\chi_{ij} = \chi_{ji} = h^2\left(\alpha\beta - \beta\alpha\right) = 0 \tag{4.11.6}$$

$$\chi_{kk} = (\gamma h)^2\left(\alpha^2 + \beta^2\right) = \gamma^2 h^2 \tag{4.11.7}$$

$$\chi_{ik} = \chi_{ki} = \gamma h^2\left(\alpha^2 + \beta^2\right) = \gamma h^2 \tag{4.11.8}$$

and

$$\chi_{jk} = \chi_{kj} = \gamma h^2\left(\alpha\beta - \beta\alpha\right) = 0. \tag{4.11.9}$$

Hence, we find

$$\mathcal{D}_{ij} = \sqrt{2}\, h \tag{4.11.10}$$

$$\mathcal{D}_{ik} = (1-\gamma)h \tag{4.11.11}$$

$$\mathcal{D}_{jk} = \sqrt{1+\gamma^2}\, h \tag{4.11.12}$$

with

$$\mathcal{D}_{ij} \geq \mathcal{D}_{jk} > \mathcal{D}_{ik} \text{ for } \gamma \in (0,1]. \tag{4.11.13}$$

In the limit k becomes an outlier of the left cluster ϕ_1, $\gamma \downarrow 0$ and $\mathcal{D}_{ik} \approx \mathcal{D}_{jk}$. Furthermore, although the inequalities $\mathcal{D}_{ij} > \mathcal{D}_{ik}$ and $\mathcal{D}_{jk} > \mathcal{D}_{ik}$ facilitate the task of grouping similar points, the inequality $\mathcal{D}_{jk} \leq \mathcal{D}_{ij}$ could potentially undermine the clustering accuracy. Notice that node k can be either close or far from the right cluster (Fig. 4.11.1(a,b), respectively), but yield the same $\mathcal{D}_{jk}$, as long as $\phi_\mu(k) = \gamma\phi_\mu(i)$. In other words, an outlier from the left cluster could be closer to the right cluster in spectral distance, even when the outlier has a negligible connection to the right cluster (Fig. 4.11.1(b)). By sharp contrast, in QTC, the phase at a node lying between two clusters interpolates monotonically between the phases of the two clusters (Fig. 4.8.2).

This undesirable behavior of spectral clustering may be avoided by renormalizing the eigenvectors. Two common approaches are (Fig. 4.11.1(c,d) and (e,f), respectively):

Approach 1

1. Compute $N(i) \equiv (\sum_{n=0}^{q-1} |\psi_n(i)|^2)^{\frac{1}{2}}$.

2. Divide each $\psi_n(i)$ by $N(i)$, i.e., $\psi_n \to \psi_n/N$.

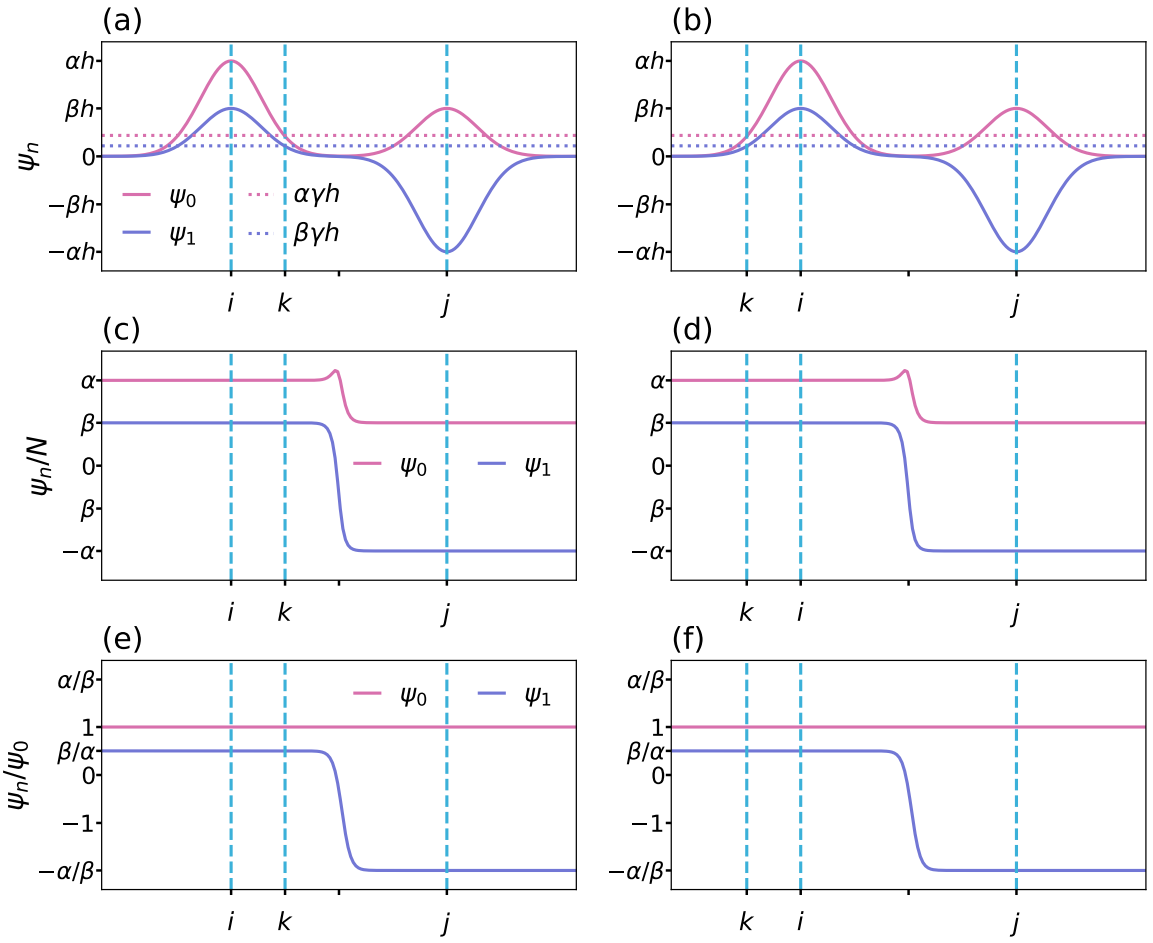

Figure 4.11.1: (a,b) Schematic illustrations of the ground state and the first excited state involving two clusters; i, j, and k are node indices. Node k is an outlier (a) lying between the two clusters or (b) far from both clusters. (c,d) The normalized ground state and first excited state eigenfunctions using Approach 1. (e,f) The modified ground state and first excited state eigenfunctions using Approach 2.

Approach 2

1. Divide each $\psi_n(i)$ by $\psi_0(i)$, i.e., $\psi_n \to \psi_n/\psi_0$.

Similar to the phase plateaus in QTC, ψ_n/N and ψ_n/ψ_0 are essentially flat within a cluster (Fig. 4.11.1(c,d) and (e,f), respectively).

In the first approach (Fig. 4.11.1(c,d)), the spectral embedding distances become

$$\mathcal{D}_{ij}^{(1)} = \sqrt{(\alpha-\beta)^2 + (\alpha+\beta)^2} = \sqrt{2} \tag{4.11.14}$$

$$\mathcal{D}_{ik}^{(1)} = 0 \tag{4.11.15}$$

$$\mathcal{D}_{jk}^{(1)} = \sqrt{(\alpha-\beta)^2 + (\alpha+\beta)^2} = \sqrt{2}. \tag{4.11.16}$$

In the second approach (Fig. 4.11.1(e,f)), the spectral embedding distances become

$$\mathcal{D}_{ij}^{(2)} = \sqrt{(\beta/\alpha + \alpha/\beta)^2} = 1/\alpha\beta \tag{4.11.17}$$

$$\mathcal{D}_{ik}^{(2)} = 0 \tag{4.11.18}$$

$$\mathcal{D}_{jk}^{(2)} = \sqrt{(\beta/\alpha + \alpha/\beta)^2} = 1/\alpha\beta. \tag{4.11.19}$$

In both cases, we have $\mathcal{D}_{jk}^{(1,2)} = \mathcal{D}_{ij}^{(1,2)}$; thus, the outlier node k is much more likely to be clustered with the left cluster. (Scikit-Learn, a very popular machine learning software package in Python, implements the second approach incorrectly as $\psi_n \to \psi_n \times \psi_0$ and sometimes yields counter-intuitive clustering results. In this paper, we use our own implementation of Approach 1.)

Finally, we note that spectral embedding has an intrinsic weakness stemming from ignoring potentially useful information from high-energy states. More precisely, recall that spectral embedding assumes that the most relevant information for clustering is

encoded in the first q low-energy eigenstates of H. However, this assumption could be invalid in some cases, e.g., our synthetic data sets in Fig. 4.5.1, and time series data in Fig. 4.5.2, where the information needed to separate some small clusters are stored in higher energy modes. In such a case, spectral clustering may not have the required information to separate the small clusters, but instead chop the large clusters into fragments at their weak edges in low density regions. By contrast, QTC does not require a manual cut-off in the spectrum and incorporates all eigenstates by naturally weighing the contribution from each eigenfunction ψ_n by $|s + \mathrm{i}E_n|^{-1}$. This difference may explain why QTC is more robust than spectral embedding when there exists a hierarchy of cluster sizes.

4.11.2 Time-averaged transition amplitude

The time-dependent transition amplitude $G_{ij}(t)$ from node j to i is complex-valued and oscillatory in time, i.e.

$$
\begin{aligned}
G_{ij}(t) &= \langle i|\mathrm{e}^{-\mathrm{i}Ht}|j\rangle && (4.11.20)\\
&= \sum_{m,n}\langle i|\psi_m\rangle\langle\psi_m|\mathrm{e}^{-\mathrm{i}Ht}|\psi_n\rangle\langle\psi_n|j\rangle && (4.11.21)\\
&= \sum_{n}\psi_n(i)\psi_n^*(j)\mathrm{e}^{-\mathrm{i}E_nt}. && (4.11.22)
\end{aligned}
$$

To obtain a real-valued matrix, we take the squared amplitude,

$$
\begin{aligned}
|G_{ij}(t)|^2 &= G_{ji}(-t)G_{ij}(t) && (4.11.23)\\
&= \sum_{m,n} \psi_m(j)\psi_m^*(i)\psi_n(i)\psi_n^*(j)\mathrm{e}^{\mathrm{i}(E_m-E_n)t} && (4.11.24)\\
&= \sum_{m,n} \rho_{mn}(i)\rho_{nm}(j)\mathrm{e}^{\mathrm{i}(E_m-E_n)t} && (4.11.25)
\end{aligned}
$$

where $\rho_{mn}(i) = \langle\psi_m|i\rangle\langle i|\psi_n\rangle$. The oscillation in time can be averaged as

$$
\begin{aligned}
P_{ij} &= \lim_{T\uparrow\infty}\frac{1}{T}\int_0^T dt\,|G_{ij}(t)|^2 && (4.11.26)\\
&= \sum_{m,n}\rho_{mn}(i)\rho_{nm}(j)\left[\lim_{T\uparrow\infty}\frac{1}{T}\int_0^T dt\,\mathrm{e}^{\mathrm{i}(E_m-E_n)t}\right] && (4.11.27)\\
&= \sum_{m,n}\delta_{E_m,E_n}\rho_{mn}(i)\rho_{nm}(j). && (4.11.28)
\end{aligned}
$$

If there is no degeneracy in the spectrum of H, then the time-averaged squared transition amplitude simplifies to

$$
P_{ij} = \sum_n \rho_{nn}(i)\rho_{nn}(j) = \sum_n |\psi_n(i)|^2|\psi_n(j)|^2, \qquad (4.11.29)
$$

which is a symmetric, non-negative matrix that can be used as a similarity measure.

The performance of P_{ij} as a spectral clustering affinity matrix was tested in four synthetic data sets (Fig. 4.11.2(a-d)) as well as the stock price time series data (Fig. 4.11.3(b)). The performance was similar to spectral clustering using Gaussian affinity.

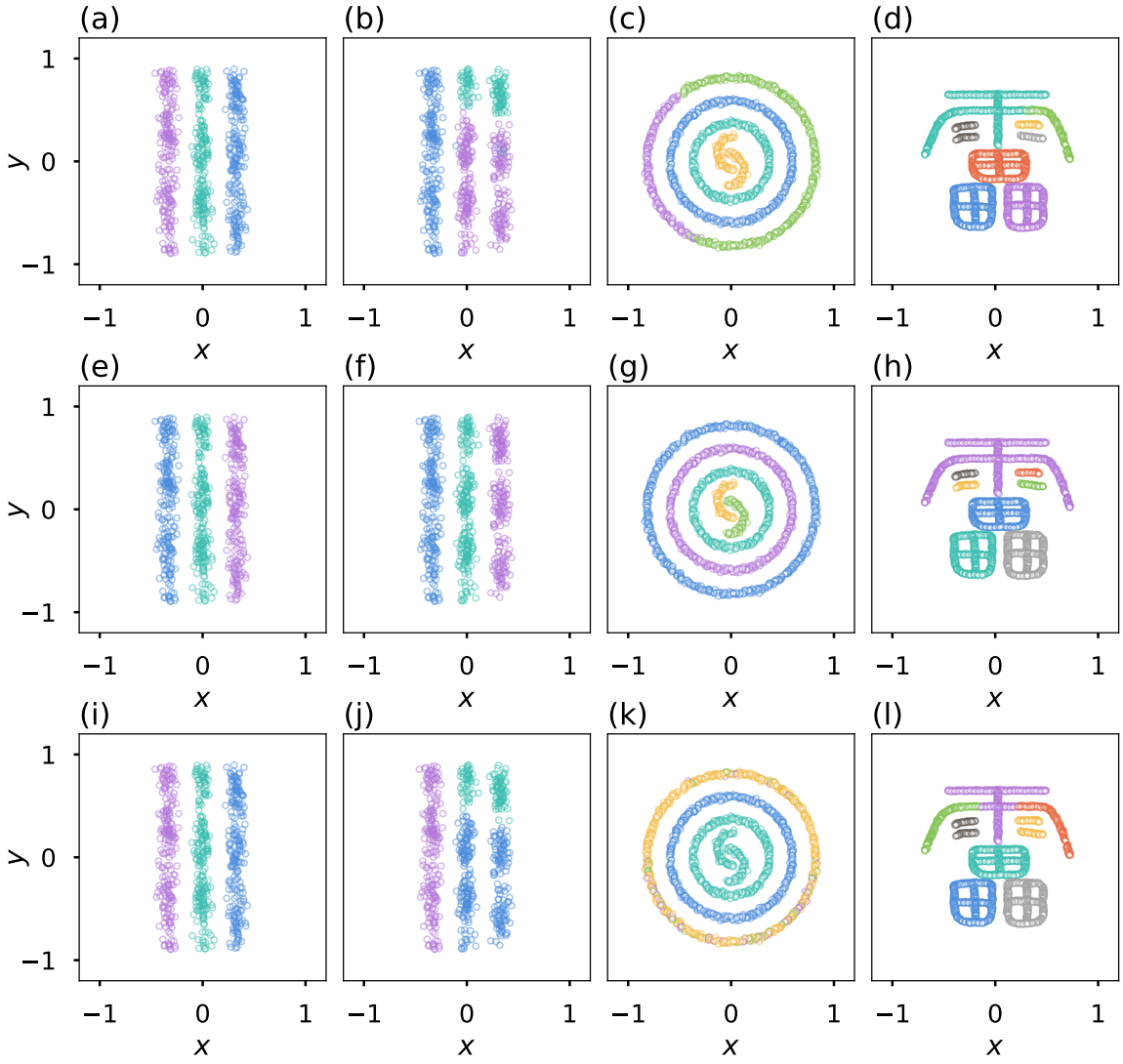

Figure 4.11.2: Synthetic data distributions plotted in Fig. 4.5.1. Spectral clustering was performed using as a similarity measure (a-d) the time-averaged squared transition amplitude, (e-h) the consensus matrices C produced by QTC, and (i-k) the similarity S of Laplace-transformed wave functions.

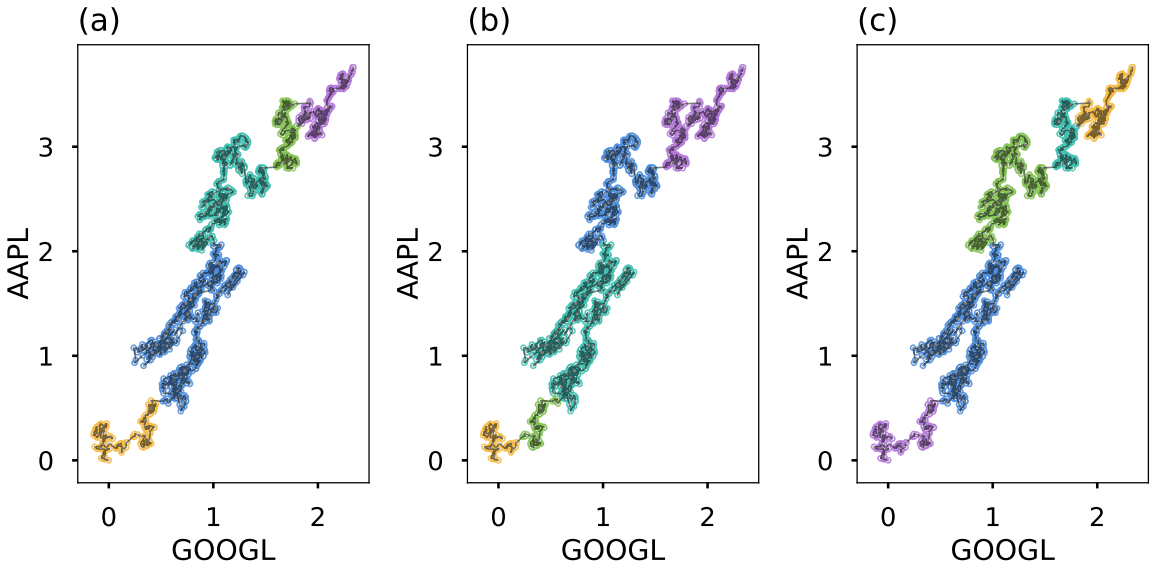

Figure 4.11.3: Time series data of the log-prices of AAPL and GOOGL stocks from January 1, 2005 to November 7, 2017. Spectral clustering was performed using as a similarity measure (a) the QTC consensus matrix C, (b) the time-averaged squared transition amplitude P, and (c) the similarity S of Laplace-transformed wave functions.

4.11.3 Density information of Laplace-transformed wave functions

As in QTC, given a time-independent Hamiltonian, we take the Laplace transform of two wave functions evolved from the states initialized at nodes i and j. Then, we take their inner product

$$\langle\tilde{\psi}_i(s)|\tilde{\psi}_j(s)\rangle = \langle i|(s - \mathrm{i}H)^{-1}(s + \mathrm{i}H)^{-1}|j\rangle \tag{4.11.30}$$

$$= \sum_n \frac{\psi_n(i)\psi_n^*(j)}{s^2 + E_n^2}. \tag{4.11.31}$$

Next, we define a similarity measure using the inner product

$$S_{ij} = \left[\frac{\left|\langle\tilde{\psi}_i(s)|\tilde{\psi}_j(s)\rangle\right|^2}{\left|\langle\tilde{\psi}_i(s)|\tilde{\psi}_i(s)\rangle\right| \left|\langle\tilde{\psi}_j(s)|\tilde{\psi}_j(s)\rangle\right|} \right]^{\frac{1}{2}}, \tag{4.11.32}$$

which is symmetric and non-negative. The performance of S_{ij} as a spectral clustering affinity matrix was also tested on four synthetic data sets (Fig. 4.11.2(i-l)) and the stock price time series data (Fig. 4.11.3(c)). The performance was similar to that of spectral clustering using Gaussian affinity (Fig. 4.11.2(i,j,l) and Fig. 4.11.3(c)), but gave sup-optimal clustering results on the annulus data set (Fig. 4.11.2(k)).

4.11.4 Jensen-Shannon divergence of density operators

The time evolution of the density operator $\rho(j) = |j\rangle\langle j|$ describing a pure state localized at node j at time $t = 0$ is

$$\begin{aligned}
\rho(j;t) &= \mathrm{e}^{-\mathrm{i}Ht}|j\rangle\langle j|\mathrm{e}^{\mathrm{i}Ht} && (4.11.33)\\
&= \sum_{m,n} \mathrm{e}^{-\mathrm{i}Ht}|\psi_m\rangle \left\{\langle\psi_m|j\rangle\langle j|\psi_n\rangle\right\} \langle\psi_n|\mathrm{e}^{\mathrm{i}Ht} && (4.11.34)\\
&= \sum_{m,n} \mathrm{e}^{-\mathrm{i}(E_m-E_n)t}\rho_{mn}(j)|\psi_m\rangle\langle\psi_n|\,, && (4.11.35)
\end{aligned}$$

where $\rho_{mn}(i) = \langle\psi_m|i\rangle\langle i|\psi_n\rangle$. If we again take the time average, then

$$\begin{aligned}
\bar{\rho}(j) &= \lim_{T\uparrow\infty}\int_0^T dt\,\rho(j;t) && (4.11.36)\\
&= \sum_{m,n} \delta_{E_m,E_n}\rho_{mn}(j)|m\rangle\langle n|; && (4.11.37)
\end{aligned}$$

and, in the absence of energy degeneracy, the time-averaged density operator initiated at node j simplifies to

$$\bar{\rho}(j) = \sum_n \rho_{nn}(j)|\psi_n\rangle\langle\psi_n| = \sum_n |\psi_n(j)|^2|\psi_n\rangle\langle\psi_n|. \qquad (4.11.38)$$

For two time-averaged density operators corresponding to pure states initialized at node i and j, respectively, we may measure the information-theoretic divergence between $\bar{\rho}(i)$ and $\bar{\rho}(j)$ using the Jensen-Shannon divergence (JSD),

$$\mathcal{D}_{\mathrm{JS}}[\bar{\rho}(i),\bar{\rho}(j)] = \mathcal{S}\left[\frac{\bar{\rho}(i)+\bar{\rho}(j)}{2}\right] - \frac{1}{2}\mathcal{S}[\bar{\rho}(i)] - \frac{1}{2}\mathcal{S}[\bar{\rho}(j)] \qquad (4.11.39)$$

where $\mathcal{S}[\rho] = -\mathrm{Tr}(\rho \log \rho)$ is the von Neumann entropy of ρ.

Using the eigenfunctions of H,

$$\mathcal{D}_{\mathrm{JS}}[\bar{\rho}(i), \bar{\rho}(j)] = \sum_n -\frac{|\psi_n(i)|^2 + |\psi_n(j)|^2}{2} \log \frac{|\psi_n(i)|^2 + |\psi_n(j)|^2}{2} \tag{4.11.40}$$

$$+ \frac{1}{2}|\psi_n(i)|^2 \log |\psi_n(i)|^2 + \frac{1}{2}|\psi_n(j)|^2 \log |\psi_n(j)|^2 \tag{4.11.41}$$

which is a non-linear function of $|\psi_n|^2$. The time-complexity for tabulating all elements in pairwise JSD matrix scales as $\mathcal{O}(m^3)$, where m is the total number of nodes, and the computation is very slow compared with the proposed QTC method. Using small synthetic data sets, we nevertheless implemented the JSD method and passed the JSD matrix to hierarchical clustering as a dissimilarity measure. The JSD measure did not show a significant performance improvement compared with the simple Euclidean distance.

4.12 Discussion

In addition to high dimensionality and strong mixing, geometric complexity remains an outstanding challenge; e.g., the cheese-stick distribution shown in Fig. 4.5.1(b) with several visually separable pieces confuses almost all clustering algorithms. But, we have demonstrated that the coherent phase information encoded in the Laplace-transformed wave functions are as powerful as the widely applied spectral clustering. Furthermore, the QTC shows more robustness when the data distribution contains density fluctuations or a hierarchy of cluster sizes. Using multiple initialization sites, QTC generates an ensemble of phase distributions, which in turn provide a collection

of discrete cluster labels. We may either select the most popular partition from the ensemble or encode the votes from the ensemble members into a consensus matrix. If most members favor a particular partition, it is an indication that the clusters are easily separable; conversely, split votes between several partitions may indicate suboptimal model parameters or strongly mixed clusters. Thus, QTC provides a useful self-consistency criterion absent in most clustering methods. Even in the case of spit votes, the consensus matrix can still be used in other clustering or supervised learning methods as an improved similarity measure. In addition to the consensus matrix, we have explored other ways of constructing a QT kernel that can be used as an input to numerous (dis)similarity-based algorithms. For example, we have tested the time-average of squared transition amplitude as a similarity measure in spectral clustering; the performance was slightly better than spectral clustering using Gaussian affinity, although some intrinsic weaknesses of spectral embedding persisted. These results provide evidence for potential benefits that may arise from studying data science using quantum physics.

Acknowledgements

I thank Alan Luu, Mohith Manjunath, and Yi Zhang for their help.

Chapter 5

Conclusions

In this thesis, I introduced three major machine learning methods based on the mathematical form and physical idea of diffusion. In Chapter 2, I generalized the Gaussian kernel in Euclidean space to a high-dimensional sphere using heat diffusion. The resulting hyperspherical heat kernel was expanded in eigenfunctions of high-dimensional angular momentum operator. The heat kernel was tested in SVM classifications of documents, cancer samples, and stocks; the hyperspherical kernel often outperformed Euclidean Gaussian RBF kernel and linear kernel. The advantage may arise from the hyperspherical transformation of feature space, and flexibility in decision boundary. In other words, the hyperspherical transformation removes less informative radial degree of freedom in a nonlinear fashion and compactifies the Euclidean feature space into a unit hypersphere where all data points are then enclosed within a finite radius. In Chapter 3, I introduced the effective dissimilarity transformation (EDT) based on all pairwise distances among the samples. The effects of the transformation were studied using thought experiments and tested using two gene expression data sets. The

transformation changes the topology of original Euclidean data space assisting the agglomeration of closely related points into dense clusters. Iteratively applying the EDT drives a static data distribution in a Euclidean space into a dynamical migration process on a hypersphere where the "curse of dimensionality" is adaptively ameliorated. In Chapter 4, I showed that a quantum mechanical wave function is dramatically different from a classical heat density, and so there was not a straight forward generalization of heat kernel to the case of quantum walks. But the coherent phase information encoded in the Laplace-transformed wave functions can be extracted to perform clustering. The resulting quantum transport clustering (QTC) algorithm is often more robust than traditional spectral embedding when there exist strong density fluctuations and heterogeneity in cluster sizes. The advantage of QTC may arise from the fact that it naturally weights the contribution of different graph Laplacian eigenmodes using their energies, whereas spectral clustering requires a fixed cut-off in the spectrum. More importantly, QTC using various initialization sites gives an ensemble of clusterings. The collection of distinct clusterings can be used to construct an empirical distribution of clustering decisions which contains much more information than a single clustering output. One may also find a new similarity measure by pooling together clustering results, and use the consensus matrix for further analysis. In summary, the three machine learning methods are based on three distinct diffusion processes. The dynamic diffusion processes often trace out hidden patterns in data sets, and thus, serve as a promising foundation for future development in machine learning methods.

Bibliography

[1] E Adinolfi, M Capece, A Franceschini, and S Falzoni. Accelerated tumor progression in mice lacking the ATP receptor P2X7. *Cancer research*, 75(4):635–644, 2015. 4

[2] P W Anderson. Absence of Diffusion in Certain Random Lattices. *Physical Review*, 109(5):1492–1505, March 1958. 4.2, 4.5

[3] P W Anderson. More is different. *Science*, 177(4047):393–396, 1972. 1

[4] J A Anderton and J C Lindsey. Global analysis of the medulloblastoma epigenome identifies disease-subgroup-specific inactivation of COL1A2. *Neuro-Oncology*, 2008. 5

[5] N Aronszajn. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337, 1950. 2.1

[6] David Arthur and Sergei Vassilvitskii. K-means++: The advantages of careful seeding. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1027–1035. Stanford University, Palo Alto, United States Stanford University, Palo Alto, United States Stanford University, Palo Alto, United States Stanford University, Palo Alto, United States, January 2007. 1

[7] Kendall Atkinson and Weimin Han. *Spherical Harmonics and Approximations on the Unit Sphere: An Introduction*. Springer Science & Business Media, February 2012. 2.1, 2.5.3, 2.5.3, 2.5.4

[8] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples. *Journal of Machine Learning Research*, 7(Nov):2399–2434, 2006. 2.1

[9] Kristin P Bennett, Usama Fayyad, and Dan Geiger. *Density-based indexing for approximate nearest-neighbor queries*. ACM, New York, New York, USA, August 1999. 1

[10] M Berger, P Gauduchon, and E Mazet. *Le spectre d'une variete riemannienne*. Springer, 1971. 2.1, 2.4

[11] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. *A training algorithm for optimal margin classifiers*. ACM, New York, New York, USA, July 1992. 2.1, 2.6

[12] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, April 1998. 4.2

[13] Andrew M Childs, Edward Farhi, and Sam Gutmann. An Example of the Difference Between Quantum and Classical Random Walks. *Quantum Information Processing*, 1(1-2):35–43, April 2002. 4.2

[14] Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, July 2006. 2.1, 3.1, 4.2

[15] Corinna Cortes and Vladimir Vapnik. Support-Vector Networks. *Machine learning*, 20(3):273–297, 1995. 2.1

[16] M Craven, A McCallum, D PiPasquo, and T Mitchell. Learning to extract symbolic knowledge from the World Wide Web. *Proceedings of the National Conference on Artificial Intelligence*, pages 509–516, 1998. 2.6

[17] D Defays. An efficient algorithm for a complete link method. *The computer journal*, 20(4):364–366, April 1977. 1

[18] Ian Dunham, Anshul Kundaje, Shelley F Aldred, and others. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57–74, 2012. 4.5.3, 4.5, 4.10.3

[19] M Ester, H P Kriegel, J Sander, and X Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. *Kdd*, 1996. 1

[20] Theodoros Evgeniou and Massimiliano Pontil. Support Vector Machines: Theory and Applications. In *Machine Learning and Its Applications*, pages 249–257. Springer Berlin Heidelberg, Berlin, Heidelberg, September 2001. 2.1

[21] Mauro Faccin, Tomi Johnson, Jacob Biamonte, Sabre Kais, and Piotr Migdał. Degree Distribution in Quantum Walks on Complex Networks. *Physical Review X*, 3(4):452–458, October 2013. 4.2, 4.3

[22] Edward Farhi and Sam Gutmann. Quantum computation and decision trees. *Physical Review A*, 58(2):915–928, August 1998. 4.2

[23] E W Forgy. *Cluster analysis of multivariate data: efficiency versus interpretability models*. Biometrics, 1965. 1

[24] Yoav Freund and Robert E Schapire. Large Margin Classification Using the Perceptron Algorithm. *Machine learning*, 37(3):277–296, 1999. 2.1

[25] Geoff Fudenberg, Gad Getz, Matthew Meyerson, and Leonid A Mirny. High order chromatin architecture shapes the landscape of chromosomal alterations in cancer. *Nature Biotechnology*, 29(12):1109–1113, November 2011. 4.5, 4.10.3

[26] Rosa Gómez-Villafuertes, Paula García-Huerta, Juan Ignacio Díaz-Hernández, and Mª Teresa Miras-Portugal. PI3K/Akt signaling pathway triggers P2X7 receptor expression as a pro-survival factor of neuroblastoma cells under limiting growth conditions. *Nature Publishing Group*, 5:1–15, December 2015. 4

[27] Alexander Grigor'yan. Analytic and geometric background of recurrence and non-explosion of the Brownian motion on Riemannian manifolds. *Bulletin of the American Mathematical Society*, 36(2):135–249, 1999. 2.5.3

[28] I Guyon, B Boser, and V Vapnik. Automatic Capacity Tuning of Very Large VC-dimension Classifiers. *Advances in Neural Information Processing Systems*, pages 147–155, 1993. 2.1, 2.6, 2.6.1

[29] Jörg Hamann, Gabriela Aust, Demet Araç, Felix B Engel, Caroline Formstone, Robert Fredriksson, Randy A Hall, Breanne L Harty, Christiane Kirchhoff, Barbara Knapp, Arunkumar Krishnan, Ines Liebscher, Hsi-Hsien Lin, David C Martinelli, Kelly R Monk, Miriam C Peeters, Xianhua Piao, Simone Prömel, Torsten Schöneberg, Thue W Schwartz, Kathleen Singer, Martin Stacey, Yuri A

Ushkaryov, Mario Vallon, Uwe Wolfrum, Mathew W Wright, Lei Xu, Tobias Langenhan, and Helgi B Schiöth. International Union of Basic and Clinical Pharmacology. XCIV. Adhesion G protein-coupled receptors. *Pharmacological Reviews*, 67(2):338–367, 2015. 2

[30] Douglas Hanahan and Robert A Weinberg. Hallmarks of Cancer: The Next Generation. *Cell*, 144(5):646–674, March 2011. 2.6

[31] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Data Mining, Inference, and Prediction. Springer Science & Business Media, November 2013. 1, 1, 2.1, 3.1

[32] David Horn and Assaf Gottlieb. Algorithm for Data Clustering in Pattern Recognition Problems Based on Quantum Mechanics. *Physical Review Letters*, 88(1):018702, December 2001. 1, 4.1

[33] Michael E Houle, Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek. Can Shared-Neighbor Distances Defeat the Curse of Dimensionality? In *Scientific and Statistical Database Management*, pages 482–500. Springer Berlin Heidelberg, Berlin, Heidelberg, June 2010. 1

[34] E P Hsu. *Stochastic analysis on manifolds, volume 38 of Graduate Studies in Mathematics*. American Mathematical Society, 2002. 2.1, 2.5.3

[35] Leonard Kaufman and Peter J Rousseeuw. *Finding Groups in Data*. An Introduction to Cluster Analysis. John Wiley & Sons, Hoboken, NJ, USA, September 2009. 1, 2.1, 3.1

[36] Balveen Kaur, Daniel J Brat, Cathárine C Calkins, and Erwin G Van Meir. Brain Angiogenesis Inhibitor 1 Is Differentially Expressed in Normal Brain and Glioblastoma Independently of p53 Expression. *The American Journal of Pathology*, 162(1):19–27, December 2010. 2

[37] John Lafferty and Guy Lebanon. Diffusion Kernels on Statistical Manifolds. *Journal of Machine Learning Research*, 6(Jan):129–163, 2005. (document), 2.1, 2.4, 4.2

[38] John Lafferty and Guy Lebanon. Diffusion Kernels on Statistical Manifolds. *Journal of Machine Learning Research*, 6(Jan):129–163, 2005. 1, 3.2, 3.7

[39] Hui-Jia Li, Yong Wang, Ling-Yun Wu, Junhua Zhang, and Xiang-Sun Zhang. Potts model based on a Markov process computation solves the community structure problem effectively. *Physical review. E*, 86(1):016109, July 2012. 1, 4.1

[40] Yu Liang, Maximilian Diehn, Andrew W Bollen, Mark A Israel, and Nalin Gupta. Type I collagen is overexpressed in medulloblastoma as a component of tumor microenvironment. *Journal of Neuro-Oncology*, 86(2):133–141, July 2007. 5

[41] Andromeda Liñán-Rico, Fabio Turco, Fernando Ochoa-Cortes, Alan Harzman, Bradley J Needleman, Razvan Arsenescu, Mahmoud Abdel-Rasoul, Paolo Fadda, Iveta Grants, Emmett Whitaker, Rosario Cuomo, and Fievos L Christofi. Molecular Signaling and Dysfunction of the Human Reactive Enteric Glial Cell Phenotype. *Inflammatory Bowel Diseases*, 22(8):1812–1834, August 2016. 4

[42] S Lloyd. Least squares quantization in PCM. *IEEE transactions on information theory*, 28(2):129–137, March 1982. 1

[43] L Lorch. Inequalities for ultraspherical polynomials and the gamma function. *Journal of Approximation Theory*, 40(2):115–120, 1984. 2.5.4

[44] Marina Meilă. Comparing clusterings—an information based distance. *Journal of Multivariate Analysis*, 98(5):873–895, May 2007. 3.1, 3.5, 3.6

[45] A Ng, M Jordan, Y Weiss, T Dietterich, and S Becker. Advances in Neural Information Processing Systems, 14, chapter On spectral clustering: analysis and an algorithm, 2002. 2.1, 3.1

[46] Noa Novershtern, Aravind Subramanian, Lee N Lawton, Raymond H Mak, W Nicholas Haining, Marie E McConkey, Naomi Habib, Nir Yosef, Cindy Y Chang, Tal Shay, Garrett M Frampton, Adam C B Drake, Ilya Leskov, Bjorn Nilsson, Fred Preffer, David Dombkowski, John W Evans, Ted Liefeld, John S Smutko, Jianzhu Chen, Nir Friedman, Richard A Young, Todd R Golub, Aviv Regev, and Benjamin L Ebert. Densely Interconnected Transcriptional Circuits Control Cell States in Human Hematopoiesis. *Cell*, 144(2):296–309, January 2011. 3.5, 3.6

[47] V A Novikov, M A Shifman, A I Vainshtein, and V I Zakharov. ABC of instantons. In *ITEP lectures on particle physics and field theory, Vol. I, II*, pages 201–299. World Sci. Publ., River Edge, NJ, 1999. 4.8

[48] Georgios Paliouras, Vangelis Karkaletsis, and Constantine D Spyropoulos. *Machine Learning and Its Applications*. Advanced Lectures. Springer, Berlin, Heidelberg, June 2003. 2.6, 2.6.1, 3.1

[49] Vern I Paulsen and Mrinal Raghupathi. *An Introduction to the Theory of Reproducing Kernel Hilbert Spaces (Cambridge Studies in Advanced Mathematics)*. Cambridge University Press, April 2016. 2.1

[50] Jörg Reichardt and Stefan Bornholdt. Detecting Fuzzy Community Structures in Complex Networks with a Potts Model. *Physical Review Letters*, 93(21):218701, November 2004. 1, 4.1

[51] Ed C Schwalbe, Janet C Lindsey, Debbie Straughton, Twala L Hogg, Michael Cole, Hisham Megahed, Sarra L Ryan, Meryl E Lusher, Michael D Taylor, Richard J Gilbertson, David W Ellison, Simon Bailey, and Steven C Clifford. Rapid diagnosis of medulloblastoma molecular subgroups. *Clinical Cancer Research*, 17(7):1883–1894, April 2011. 5

[52] Ksenya Shchors, Fruma Yehiely, Rupinder K Kular, Kumar U Kotlo, Gary Brewer, and Louis P Deiss. Cell death inhibiting RNA (CDIR) derived from a 3'-untranslated region binds AUF1 and heat shock protein 27. *Journal of Biological Chemistry*, 277(49):47061–47072, December 2002. 1

[53] Neil Shenvi, Julia Kempe, and K Birgitta Whaley. Quantum random-walk search algorithm. *Physical Review A*, 67(5):052307, May 2003. 4.2

[54] R Sibson. SLINK: An optimally efficient algorithm for the single-link cluster method. *The computer journal*, 16(1):30–34, January 1973. 1

[55] D et al Smedley. SHORT COMMUNICATION Cloning and Mapping of Members of the MYM Family. pages 1–4, September 1999. 1

[56] Z Song, S Krishna, D Thanos, J L Strominger, and S J Ono. A novel cysteine-rich sequence-specific DNA-binding protein interacts with the conserved X-box motif of the human major histocompatibility complex class II genes via a repeated Cys-His domain and functions as a transcriptional repressor. *Journal of Experimental Medicine*, 180(5):1763–1774, November 1994. 3

[57] M Stone and P Goldbart. *Mathematics for physics: a guided tour for graduate students*. Cambridge University Press, Cambridge, 2009. 2.2, 2.5.3

[58] Gábor J Székely, Maria L Rizzo, and Nail K Bakirov. Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35(6):2769–2794, December 2007. 3.7

[59] V Vapnik, E Levin, and Y Le Cun. Measuring the VC-dimension of a learning machine. *Neural Computation*, 6(5):851–876, 1994. 2.6, 2.6.1

[60] Vladimir N Vapnik. *The Nature of Statistical Learning Theory*. Springer Science & Business Media, April 2013. 2.6, 2.6.1

[61] N T Varopoulos. *Random walks and Brownian motion on manifolds*. Symposia Mathematica, 1987. 2.1, 2.5.3

[62] U Von Luxburg. A tutorial on spectral clustering. Max Planck Institute for Biological Cybernetics, 2006. 3.1

[63] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, August 2007. 1

[64] Zhen-Yi Wen and John Avery. Some properties of hyperspherical harmonics. *Journal of Mathematical Physics*, 26(3):396–399, 1985. 2.1, 2.5.1, 2.5.3, 2.5.3, 2.5.4

[65] Shuntaro Yamashita, Kaoru Fujii, Chong Zhao, Hiroshi Takagi, and Yoshinori Katakura. Involvement of the NFX1-repressor complex in PKC-δ-induced repression of hTERT transcription. *Journal of Biochemistry*, pages mvw038–5, June 2016. 3

[66] Iwei Yeh and Timothy H McCalmont. Distinguishing neurofibroma from desmoplastic melanoma: the value of the CD34 fingerprint. *Journal of Cutaneous Pathology*, 38(8):625–630, April 2011. 3.5

[67] Guoshen Yu, Guillermo Sapiro, and Stephane Mallat. Solving inverse problems with piecewise linear estimators: from Gaussian mixture models to structured sparsity. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 21(5):2481–2499, May 2012. 1

[68] Chenchao Zhao and Jun S Song. Emergent community agglomeration from data set geometry. *Physical review. E*, 95(4):042307, April 2017. 3.1

[69] Chenchao Zhao and Jun S Song. Quantum transport senses community structure in networks. *arXiv.org*, November 2017. 4.1

[70] Chenchao Zhao and Jun S Song. Exact Heat Kernel on a Hypersphere and Its Applications in Kernel SVM. *Frontiers in Applied Mathematics and Statistics*, 4:129, January 2018. 2.1, 3.2

[71] Arthur Zimek, Erich Schubert, and Hans-Peter Kriegel. A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 5(5):363–387, October 2012. 1

[72] V M Zohrabian, H Nandu, and N Gulati. Gene expression profiling of metastatic brain cancer. *Oncology Reports*, 2007. 2

Appendix A

Multidimensional scaling

Effective dissimilarity transformation (EDT) maps a distance matrix $d_{ij}^{(\tau)}$ to an adjusted distance matrix $d_{ij}^{(\tau+1)}$. The visualization is straight forward if the original distance matrix is calculated from a data distribution in $\mathbb{R}^n$ with $1 \leq n \leq 3$. However, EDT deforms the original distance matrix which can be alternatively viewed as the result of deformation of original data space, $\mathbb{R}^n$ for example, and thus, the new distances cannot be visualized in the original data space. The multidimensional scaling (MDS) provides approximate embeddings of distances in an Euclidean space of a given dimension n_{MDS}.

A.1 Multidimensional scaling

MDS takes the $m \times m$ distance matrix d_{ij} as an input, and then distance-squared matrix $D_{ij} = d_{ij}^2$ is computed. Next, using mean-centering matrix $J = I - \frac{1}{m}11^\top$, we get symmetric matrix $B = -\frac{1}{2}JDJ$ with spectrum $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$. If all eigenvalues are positive, then the distances can be exactly embedded in a Euclidean space, otherwise the distances d_{ij} can be approximately embedded in a Euclidean space

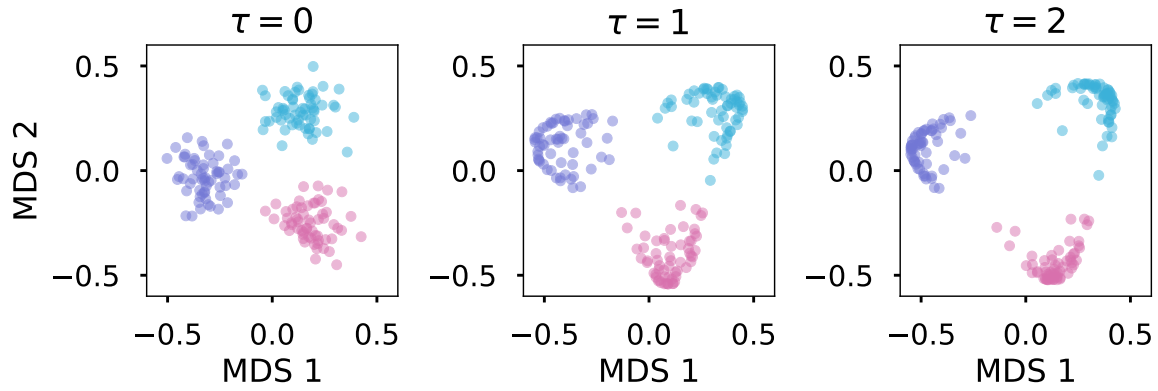

Figure A.2.1: Three Gaussian clouds were separated by EDT iterations.

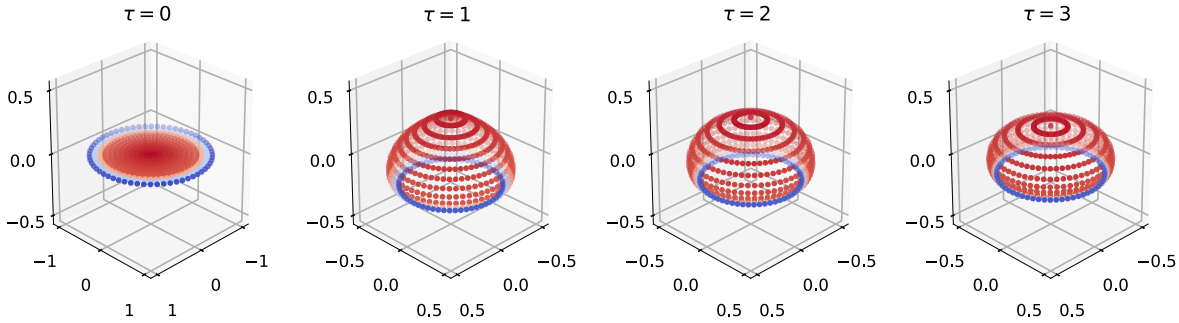

Figure A.2.2: Radially uniformly distributed concentric circles in $\mathbb{R}^2$ were deformed into a 2-sphere embedded in $\mathbb{R}^3$.

using the eigenvectors $\{\mathbf{v}_k\}_{k=1}^{n_{\text{MDS}}}$ of the first n_{MDS} largest positive eigenvalues. The MDS coordinates are then $(\mathbf{v}_1\lambda_1^{\frac{1}{2}}, \mathbf{v}_2\lambda_2^{\frac{1}{2}}, \cdots, \mathbf{v}_{n_{\text{MDS}}}\lambda_{n_{\text{MDS}}}^{\frac{1}{2}})_{m\times n}$.

A.2 Visualization of EDT iterations

Figure A.2.1 shows the drifting process of three clusters in $\mathbb{R}^2$. Figure A.2.2 and A.2.3 illustrate the global deformation effect described in Figure 3.3.5. Figure A.2.4 and A.2.5 illustrate the global deformation of annulus data sets in Figure 3.3.6.

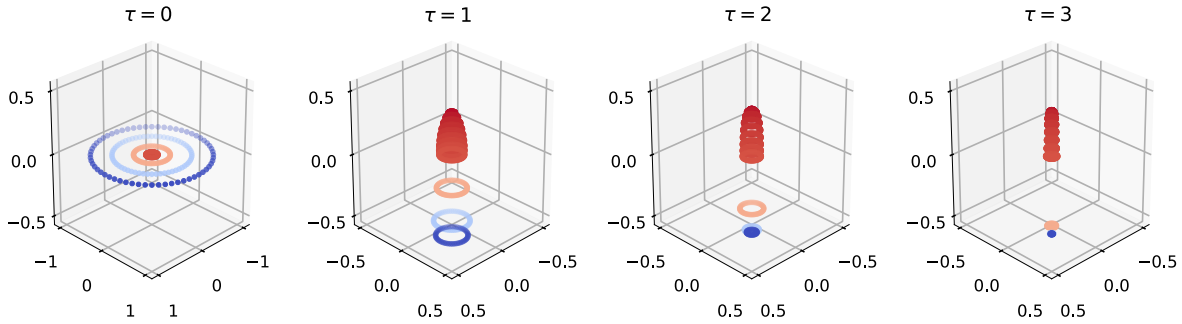

Figure A.2.3: Concentric circles concentrated at origin in $\mathbb{R}^2$ were deformed into a 2-sphere embedded in $\mathbb{R}^3$ where the outer rim circles were pushed to the south pole.

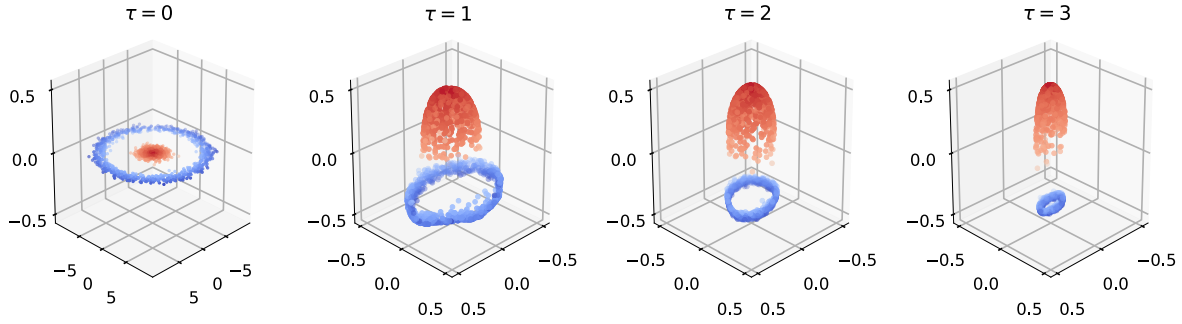

Figure A.2.4: Annulus data set consisting a central cluster and a concentric ring in $\mathbb{R}^2$ was deformed to a S^2 where the ring was pushed to the south pole.

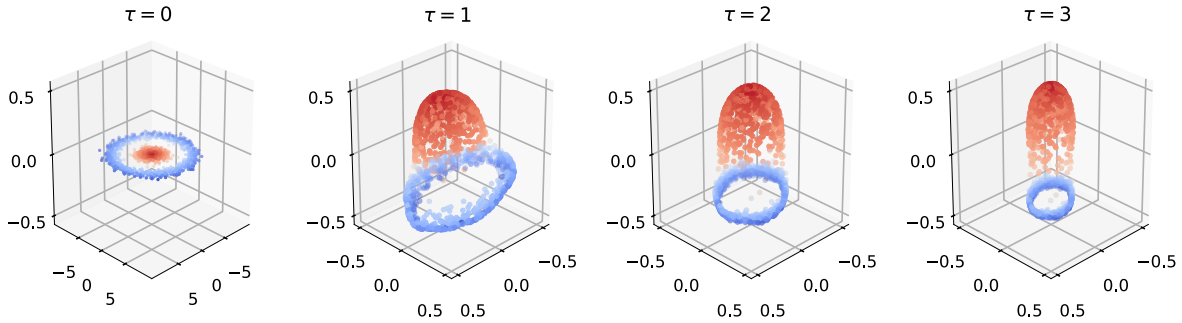

Figure A.2.5: Annulus data set consisting a central cluster and a concentric ring in $\mathbb{R}^2$ was deformed to a S^2 where the ring cluster was pushed to the south pole.

A.3 Visualization of similarity matrix

Under certain conditions, MDS is able to embed a collection of dissimilarities in some Euclidean space. Given a similarity matrix measured with some kernel function $G(x, y)$, we are able to construct a dissimilarity measure by identifying $G(x, y)$ as an inner product $\langle x, y\rangle$. In other words,

$$\|x - y\|^2 \equiv \langle x - y, x - y\rangle = G(x, x) - 2G(x, y) + G(y, y)$$

and then we can approximately embed data points in an Euclidean space with MDS, based on the metric induced by the kernel function.

For example, let the kernel function be the "propagator" of a massive particle in a network described by adjacency matrix A, or

$$G = \frac{1}{H + m^2}$$

where H is the symmetrically normalized graph Laplacian with non-negative eigenvalues. In Figure A.3.1, we have three well-separated Gaussian clusters with two of them being closer, then the first excited state ψ_1 with energy E_1 must corresponds to the separation of the third cluster from the two closer clusters. Then if $m^2 \gg E_1$ or the particle is very massive, the propagator is short ranged and we expect three blocks in plot of G matrix; if $m^2 \approx E_1$, the range of the propagator is extended and should be able to capture more global structures and thus, we expect two blocks in the plot G matrix. Figure A.3.2 shows the transition from three blocks to two blocks when m^2 decreased from $200E_1$ to E_1. We computed the induced dissimilarities from G and then

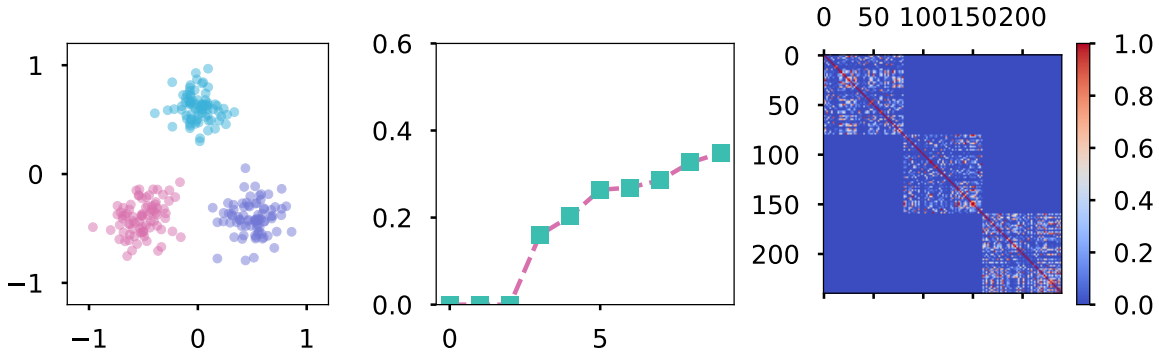

Figure A.3.1: Three Gaussian clusters (left) in $\mathbb{R}^2$ with two clusters being slightly closer. The adjacency (right) was generated using Gaussian RBF kernel with a short proximity length $\sim$ 1%-quantile of all pairwise distances. The symmetrically normalized graph Laplacian H was calculated and diagonalized; middle plot shows that there were three low energy modes in the spectrum of H.

embedded the dissimilarities in $\mathbb{R}^2$ using MDS. Figure A.3.3 shows the MDS embedding of G in $\mathbb{R}^2$ where each small blocks were represented as a stick; the two merged sticks at $m^2 \sim E_1$ corresponds to the slightly closer two clusters in the original distribution.

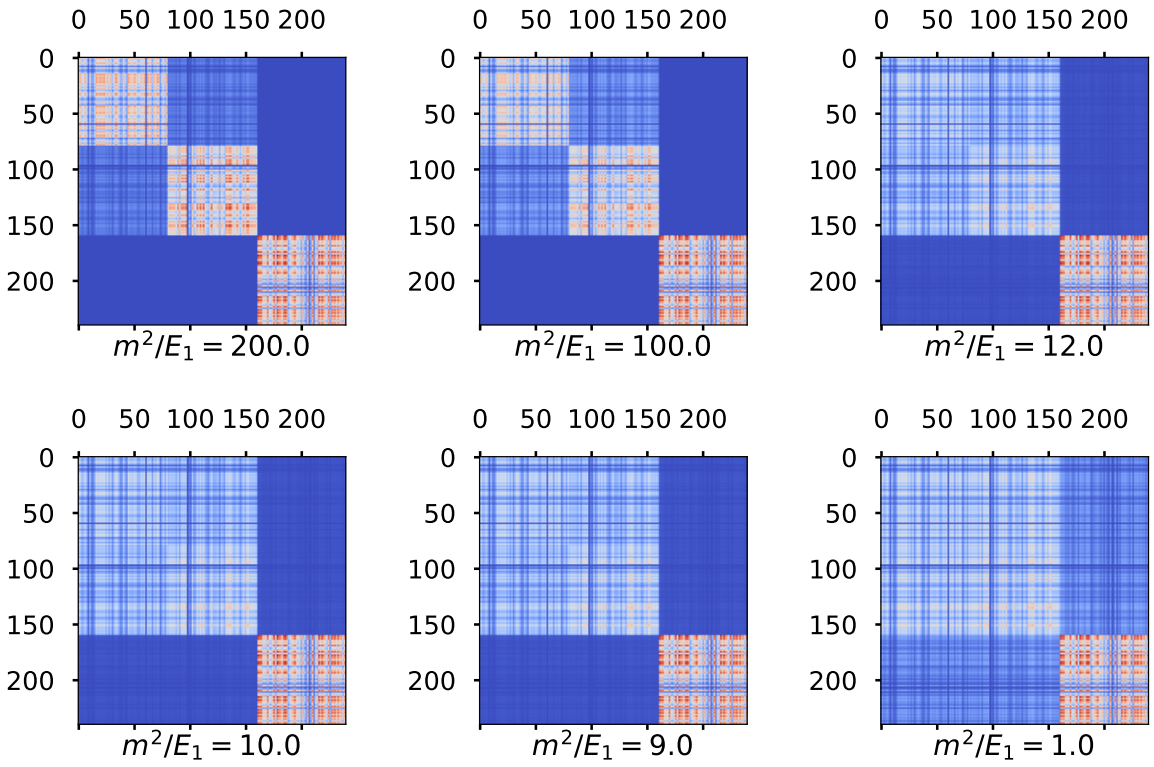

Figure A.3.2: Plots of $G = (H + m^2)^{-1}$ with variations in m^2. As the ratio m^2/E_1 decreased, the two block structure grew stronger than three block structure.

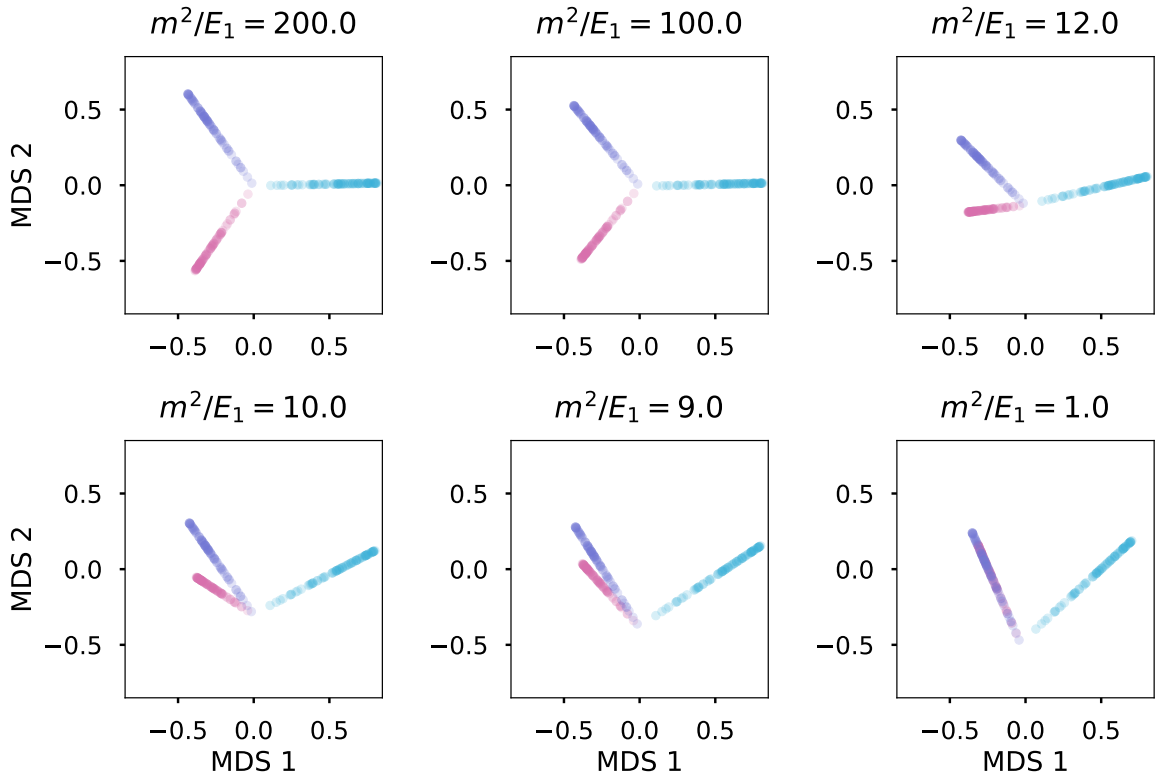

Figure A.3.3: Embeddings of dissimilarities induced by $G = (H + m^2)^{-1}$ with variations in m^2. As the ratio m^2/E_1 decreased, the three sticks gradually merged into two sticks in the embedded space where the merged sticks corresponds to the slightly closer two clusters.