A Volume-Based Heat-Diffusion Classifier

Haixuan Yang, Michael R. Lyu, *Fellow, IEEE*, and Irwin King, *Member, IEEE*

***Abstract*—Heat-diffusion models have been successfully applied to various domains such as classification and dimensionality-reduction tasks in manifold learning. One critical local approximation technique is employed to weigh the edges in the graph constructed from data points. This approximation technique is based on an implicit assumption that the data are distributed evenly. However, this assumption is not valid in most cases, so the approximation is not accurate in these cases. To solve this challenging problem, we propose a volume-based heat-diffusion model (VHDM). In VHDM, the volume is theoretically justified by handling the input data that are unevenly distributed on an unknown manifold. We also propose a novel volume-based heat-diffusion classifier (VHDC) based on VHDM. One of the advantages of VHDC is that the computational complexity is linear on the number of edges given a constructed graph. Moreover, we give an analysis on the stability of VHDC with respect to its three free parameters, and we demonstrate the connection between VHDC and some other classifiers. Experiments show that VHDC performs better than Parzen window approach, K nearest neighbor, and the HDC without volumes in prediction accuracy and outperforms some recently proposed transductive-learning algorithms. The enhanced performance of VHDC shows the validity of introducing the volume. The experiments also confirm the stability of VHDC with respect to its three free parameters.**

***Index Terms*—Heat diffusion, integral approximation, manifold learning, transductive learning.**

I. Introduction

RECENTLY, manifold learning has become a popular approach to nonlinear dimensionality reduction [1]–[4], density estimation [5], classification [6]–[10], regression [11], and ranking [12], [13]. Manifold learning is specially designed for the case that the data points are distributed on a low-dimensional nonlinear manifold, which is embedded into a high-dimensional Euclidean space. In such a case, the straight-line Euclidean distance may not be accurate because of the nonlinearity of the manifold. For example, on the surface of a sphere, the distance between two points is better measured by the geodesic path. Much recent work has captured the nonlinearity of the curved manifold. One common idea is that the local information in a nonlinear manifold is relatively accurate and can be used to construct the global information. This idea is reasonable because, in a manifold, every small area is equivalent to a Euclidean space and can be mapped to it by a smooth transformation. The local information appears in various types: local distance used in [2] and [4], local linearity used in [1], local covariance matrix used in [5], and local Laplacian approximation used in [3] and [11].

As an important technique in manifold learning, heat kernels have been successfully applied to various domains recently. In [3], a nonlinear dimensionality-reduction algorithm was proposed based on the graph Laplacian whose elements were induced by a local heat-kernel approximation. In [7], a discrete diffusion kernel on graphs and other discrete input spaces was proposed. When it was applied to a large-margin classifier, good performance for categorical data was demonstrated by employing the simple diffusion kernel on the hypercube. In [14], a general framework was proposed. The key idea was to begin with a statistical family that was natural for the data being analyzed and to represent data as points on the statistical manifold associated with the Fisher information metric of this family. When applied to the text classification, where the natural statistical family was multinomial, a closed-form approximation to the heat kernel for a multinomial family was proposed, which yielded significant improvements over the use of Gaussian or linear kernels. In [8], the solution of the heat-diffusion equation on a graph was employed to construct a classifier. In [13], heat-diffusion models were established on the Web graph, which was considered to lie on a manifold.

Despite the success of these heat-kernel applications, there are limitations. For example, when the manifold is unknown, the heat-kernel approximation method employed in [14] will not work. Although it is possible to extend the kernel-construction method employed in [7] to a data cloud, the proposed method is limited to a kernel-based algorithm. In [3], there is an implicit assumption in the local heat-kernel approximation: The data are distributed evenly. This assumption results in some errors when the data are not evenly distributed. Consequently, there exist errors in the heat-diffusion classifier (HDC) in [8], which adopted the same local heat-kernel approximation as that in [3].

When facing unevenly distributed data, we should use a more accurate approximation. Moreover, we also face the following problems where we cannot employ the traditional methods.

1) The density of data varies, which also results in unevenly distributed data.
2) The manifold is unknown.
3) The differential-equation expression is unknown even if the manifold is known.

We propose a volume-based heat-diffusion model (VHDM) on a graph by considering the earlier problems. In fact, we try to establish the heat-diffusion model on a graph by going back to the Fourier law, on which the original differential heat-diffusion equation is established. The novel heat-diffusion model on the graph is expected to incorporate the characteristics of the data.

Manuscript received February 28, 2008; revised June 27, 2008 and September 8, 2008. First published December 16, 2008; current version published March 19, 2009. This work was supported by the Research Grants Council of the Hong Kong Special Administrative Region, China, under Projects CUHK4150/07E and CUHK4125/07E. This paper was recommended by Associate Editor Q. Shen.

The authors are with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong (e-mail: hxyang@cse.cuhk.edu.hk; lyu@cse.cuhk.edu.hk; king@cse.cuhk.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCB.2008.2006279

In turn, it is expected to lead to an effective classifier called Volume-based HDC (VHDC).

The rest of this paper is organized as follows. In Section II, we show the basic concepts of heat diffusion and some related work about transductive learning. In Section III, we establish our VHDM model. Moreover, the VHDC is proposed in Section IV. Then, in Section V, we demonstrate the experimental results. Section VI provides the conclusions.

II. Heat-Diffusion Equations, Heat Kernels, and Transductive Learning

The heat equation describes the distribution of heat (or variation in temperature) in a given region over time. A heat kernel is a special solution to the heat equation, and it serves as a class of kernels in machine learning (for materials in learning with kernels, see [15] and [16]). In this section, we provide the basic concepts of heat diffusion and review some traditional discrete solutions. Moreover, we show some related work about transductive learning.

A. Heat-Diffusion Equations on a Manifold

Given a manifold M, let $f(\mathbf{x}, t)$ denote the temperature at location $\mathbf{x}$ at time t. Then, $f(\mathbf{x}, t)$ satisfies the following differential equation on manifold $\mathcal{M}$:

$$\begin{cases} \frac{\partial f}{\partial t} - \mathcal{L}f = 0 \\ f(\mathbf{x}, 0) = f_0(\mathbf{x}) \end{cases} \tag{1}$$

where $f_0(\mathbf{x})$ is an initial temperature distribution at time zero and $\mathcal{L}$ is the *Laplace–Beltrami operator*. If $f_0(\mathbf{x})$ is specialized as the delta function $\delta(\mathbf{x} - \mathbf{y})$, then the solution to (1) is specialized as the heat kernel $K_t(\mathbf{x}, \mathbf{y})$ [17]. More specifically, $\delta(\mathbf{x} - \mathbf{y})$ describes a unit heat source at position $\mathbf{y}$ while there is no heat in other positions. In other words, $\delta(\mathbf{x} - \mathbf{y}) = 0$ for $\mathbf{x} \neq \mathbf{y}$ and $\int_{-\infty}^{+\infty} \delta(\mathbf{x} - \mathbf{y}) d\mathbf{x} = 1$. If we let $f_0(\mathbf{x}, 0) = \delta(\mathbf{x} - \mathbf{y})$, then $K_t(\mathbf{x}, \mathbf{y})$ is the solution to (1). As a result, the heat kernel $K_t(\mathbf{x}, \mathbf{y})$ means the amount of heat that point $\mathbf{x}$ receives from the unit heat source at position $\mathbf{y}$ after a time period t. In machine learning, we can consider $K_t(\mathbf{x}, \mathbf{y})$ as a similarity measure between two points.

When the underlying manifold is the well-known m-dimensional Euclidean space, $\mathcal{L}f$ is simplified as $\sum_i \partial^2 f / \partial x_i^2$, and the heat kernel takes the Gaussian radial basis function (RBF) form

$$K_t(\mathbf{x}, \mathbf{y}) = (4\pi t)^{-\frac{m}{2}} e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{4t}}. \tag{2}$$

It is therefore observed that, when the underlying manifold is the Euclidean space, the Gaussian RBF kernel is a special case of the heat kernel. Note that the point-charge model in [10] also possesses this property.

If the manifold is the geometry of multinomial families, a closed-form approximation to the heat kernel is found in [14] and is applied to a kernel-based algorithm. In some situations where it is difficult to obtain the analytical solutions, it is natural to consider the numerical methods for differential equations.

B. Numerical Methods

Traditional numerical methods for solving differential equations are in fact established on a triangulation mesh or on a grid, and they have been classified into three main categories: finite-element (FE), boundary-element (BE), and finite-difference (FD) methods [18]. For the heat-diffusion equation, the situation is similar. The FE method for the heat-diffusion equation is used in surface smoothing (for example, see [19] and [20]).

First, we describe the FE method. If a simplicial surface S with vertex set V can be constructed from the data cloud, then, by the results in [21], the discrete Laplace–Beltrami operator $\mathcal{L}$ of the simplicial surface S can be established as follows.

Definition 1: For a function $f : V \to R^m$ on the vertices, the value of $\mathcal{L}f : V \to R^m$ at $x_i \in V$ is

$$\mathcal{L}f(x_i) = \sum_{x_j \in V : (x_i, x_j) \in E_D} \rho(x_i, x_j) \left(f(x_i) - f(x_j)\right) \tag{3}$$

where E_D is the edge set of a Delaunay triangulation of S, and the weights are given by

$$\rho(x_i, x_j) = \begin{cases} \frac{1}{2}(\cot \alpha_{ij} + \cot \alpha_{ji}), & \text{for interior edges} \\ \frac{1}{2} \cot \alpha_{ij}, & \text{for boundary edges} \end{cases} \tag{4}$$

where α_{ij} (and α_{ji} for interior edges) are the angels opposite the edge (x_i, x_j) in the adjacent triangles of the Delaunay triangulation. □

However, it is difficult to construct the mentioned simplicial surface S when facing a cloud of data points in an unknown geometry. For the same reason, we cannot construct the triangle mesh directly in our model. It is true that meshing algorithms exist and are widely employed in scientific computation, for example, see [22] and [23]. They are highly refined for low-dimensional point clouds. However, in situations where the data are quite high-dimensional and sparse, we are unaware of any effective meshing algorithm, and therefore, we cannot use the FE and BE methods.

Secondly, in the following, we illustrate the FD method for the heat-diffusion equation by considering the special case when the manifold is a 2-D Euclidean space. In such a case, the heat-diffusion equation in (1) becomes

$$\begin{cases} \frac{\partial f}{\partial t} - \frac{\partial^2 f}{\partial x^2} - \frac{\partial^2 f}{\partial y^2} = 0 \\ f(x, y, 0) = f_0(x, y). \end{cases} \tag{5}$$

The FD method begins with the discretization of space and time. For simplicity, we assume equal spacing among the points x_i in one dimension with interval $\Delta x = x_{i+1} - x_i$, equal spacing among the points y_j in another dimension with interval $\Delta y = y_{j+1} - y_j$ (assume $\Delta y = \Delta x = d$ for simplicity), and equal time of $\Delta t = t_{k+1} - t_k$. $f(i, j, k)$ is the heat at position (x_i, y_j) at time t_k. The grid on the plane is shown in Fig. 1(a). The grid creates a natural graph: The set of nodes is $\{(i, j)\}$, and node (i, j) is connected to node (i', j') if, and only if, $|i - i'| + |j - j'| = 1$. Note that each node (i, j) has four neighbors: $(i - 1, j)$, $(i + 1, j)$, $(i, j - 1)$, and $(i, j + 1)$.

Based on this discretization and approximation of the function, we then write the following approximations of its

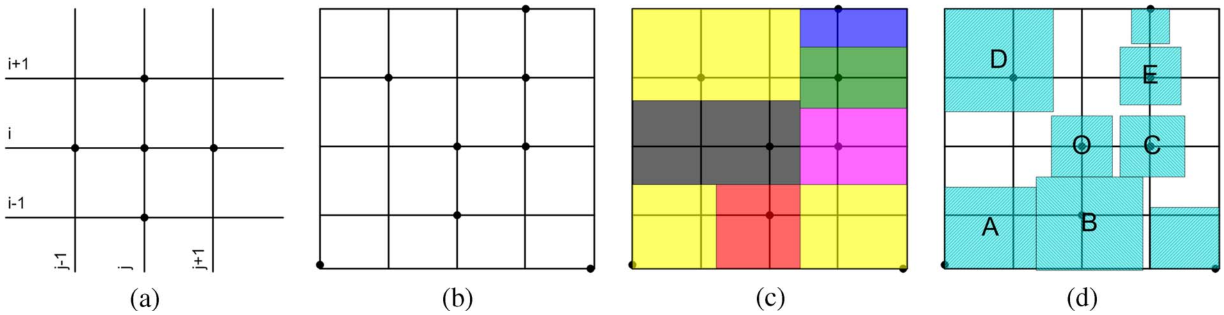


Fig. 1. (a) Grid on the 2-D space. (b) Eight irregularly positioned points. (c) Small patches around the irregular points. (d) Square approximations of the small patches.

derivatives in space and time:

$$\left.\frac{\partial f}{\partial t}\right|_{(i,j,k)} \approx \frac{f(i,j,k+1)-f(i,j,k)}{\Delta t}$$
$$\left.\frac{\partial^2 f}{\partial x^2}\right|_{(i,j,k)} \approx \frac{f(i-1,j,k)-2f(i,j,k)+f(i+1,j,k)}{(\Delta x)^2}$$
$$\left.\frac{\partial^2 f}{\partial y^2}\right|_{(i,j,k)} \approx \frac{f(i,j-1,k)-2f(i,j,k)+f(i,j+1,k)}{(\Delta y)^2}.$$

These approximations lead to a difference form of the heat equation as follows:

$$\begin{aligned}\frac{f(i,j,k+1)-f(i,j,k)}{\Delta t} &= \frac{f(i-1,j,k)-2f(i,j,k)+f(i+1,j,k)}{(\Delta x)^2} \\ &\quad + \frac{f(i,j-1,k)-2f(i,j,k)+f(i,j+1,k)}{(\Delta y)^2} \\ &= \frac{(f(i-1,j,k)-f(i,j,k))+(f(i+1,j,k)-f(i,j,k))}{d^2} \\ &\quad + \frac{(f(i,j-1,k)-f(i,j,k))+(f(i,j+1,k)-f(i,j,k))}{d^2}.\end{aligned} \tag{6}$$

The earlier two discretization methods are successful when the underlying triangulation mesh or the grid can be constructed successfully. However, in the real data analysis, the graph constructed from the data points is irregular, i.e., it is neither a triangulation mesh nor a grid. Even worse, in some situations, data points are distributed on an unknown manifold. For these cases, the heat equation on a graph is a good choice.

C. Heat Equations on a Graph

Consider an undirected unweighted graph $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$ and E is the set of all edges. Let d_i denote the degree of vertex i (number of edges emanating from vertex i). Heat kernels on G are defined as $e^{\gamma H}$, where

$$H_{ij} = \begin{cases} -d_i, & j = I \\ 1, & (j,i) \in E \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

is called the Laplacian of G. A heat kernel on a graph can be considered as a similarity measure between two nodes on the graph. Let $f(0)$ be the vector describing the initial temperature distribution on G and $f(t)$ be the vector describing the temperature distribution at time t. Then, the heat equation on G is $(d/dt)f(t) = \alpha H$, and its solution is $f(t) = e^{\alpha t H} f(0) = e^{\gamma H} f(0)$, where $\gamma = \alpha t$ (for a theoretical analysis of heat kernels on graphs, see [24]).

Heat kernels on graphs are applied to a large-margin classifier in [7] and [25]. Different from these work, the solution $f(t) = e^{\gamma H} f(0)$ to heat-diffusion equation (not the heat kernel) is employed directly to construct an HDC in [8].

D. Related Work in Transductive Learning

HDC is built on a graph, and it is actually a semisupervised algorithm: It needs access to the unlabeled data (for a systematic investigation on a semisupervised learning, refer to [26]). For transductive learning, the kernel matrix is important (for the kernel matrix learning, refer to [27]). Along the line of transductive learning, our method is related to [6], [28], and [29]. The models in [28] and [29] are mainly concerned with directed graphs such as the Web link, on which the cocitation is meaningful. This cocitation calculation, however, is not being considered in our model; hence, a comparison with [28] and [29] is inappropriate and is not provided empirically. We are interested in comparing with consistency method (CM) proposed in [6], which is most closely related to our proposed VHDC. CM consists of the following four steps.

Step 1) **Form the affinity matrix** W. Define $W_{ij} = e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\beta}$ if $i \neq j$ and $W_{ii} = 0$.

Step 2) **Construct the matrix** S. $S = D^{-1/2} W D^{-1/2}$ in which D is a diagonal matrix with its (i, i) element equal to the sum of the ith row of W.

Step 3) **Iterate**. $F(t+1) = \alpha S F(t) + (1-\alpha) Y$ until it converges to F^*, where α is a parameter in (0, 1) and Y is a $n \times c$ matrix with $Y_{ij} = 1$ if $\mathbf{x}_i$ is labeled as $y_i = j$ and $Y_{ij} = 0$, otherwise. Note that $F^* = (1-\alpha)(I - \alpha S)^{-1} Y$.

Step 4) **Label the unlabeled data**. Label each point $\mathbf{x}_i$ as a label $y_i = \arg\max_{j \leq c} F^*_{ij}$.

Along the line of SVM, transductive SVM algorithms (UniverSVM (USVM) [30] and SVMLight [31]) are popular. Employed as baselines, the recent one, USVM, will be compared to our method in the experimental section.

E. Connections Between the Heat-Diffusion Model on Graphs and That on Manifolds

Since we approximate the unknown manifold by a neighborhood graph, it is interesting to show the similarity between heat diffusion on a manifold and that on a neighborhood graph. In the following, we list some correspondences between the heat-diffusion model on graphs and that on manifolds.

1) The heat-diffusion equation on a graph is $(d/dt)f(t) = \alpha H f(t)$; the heat-diffusion equation on a manifold is, from (1)
$$\begin{cases} \frac{\partial f}{\partial t} = \mathcal{L} f \\ f(\mathbf{x}, 0) = f_0(\mathbf{x}). \end{cases}$$
2) The solution to the heat-diffusion equation on a graph is $f(t) = e^{\alpha t H} f(0) = e^{\gamma H} f(0)$; the solution to the heat-diffusion equation on a manifold is $f(\mathbf{x}, t) = \int_M K_t(\mathbf{x}, \mathbf{y}) f_0(\mathbf{y}) d\mathbf{y}$.
3) The delta function $\delta(\mathbf{x} - \mathbf{y})$ is used to represent a unit heat source at position $\mathbf{y}$; the vector $\mathbf{e}_j$, whose jth element is one while other elements are zero, is used to represent a unit heat source at node j.

F. Problem in the Integral Approximation

In [3], an integral approximation to the heat-diffusion equation is employed to weigh the edges of the graph constructed from data points. Then, the Laplacian of the weighted graph is applied to a nonlinear dimensionality-reduction algorithm. This weighing method is also used in [6], [11], [12], and [25].

We will inherit the integral-approximation technique in [3], but our focus in this paper is not the dimensionality-reduction problem. In [3], there is an implicit assumption in the local heat-kernel approximation: The data are distributed evenly. For example, the integral $\int_0^1 f(x)dx$ can be approximated by

$$\int\limits_0^1 f(x)dx \approx \sum_1^n f(\xi_i)\Delta x_i \tag{8}$$

where Δx_i is a partition of the interval [0, 1] and $\xi_i \in \Delta x_i$. If $\Delta x_i = \Delta x_j$ for any pair of i and j, i.e., when the data are evenly distributed, then

$$\int\limits_0^1 f(x)dx \approx \frac{1}{n}\sum_1^n f(\xi_i). \tag{9}$$

The integral approximation in [3] uses (9) and, thus, implicitly uses the assumption of evenly distributed data. This results in some errors when the data are not evenly distributed, and consequently, there exist errors in the HDC in [8], which adopts the same local heat-kernel approximation as that in [3].

We consider to introduce the concept of volumes to solve the problem of the unevenly distributed data. In this example, the volume of point i is Δx_i. In a manifold setting, the volume of point i is the surrounding hypervolume. In Section III-B1, we will provide more mathematical justifications.

In the next section, we will propose a VHDM on a graph by preserving the common features in (3) and (6) such that the neighbor x_j of x_i affects x_i in proportion to the difference $f(x_j) - f(x_i)$.

III. VHDM on a Graph

First, we give our notation for the heat-diffusion model. Consider a directed weighted graph $G = (V, E, W)$, where $V = \{v_1, v_2, \ldots, v_n\}$, $E = \{(v_i, v_j) | \text{there is an edge from } v_i \text{ to } v_j\}$ is the set of all edges, and $W = (w_{ij})$ is the weight matrix. In contrast to the normal undirected weighed graph, the edge (v_i, v_j) is considered as a pipe that connects nodes i and j, and the weight w_{ij} is considered as the length of the pipe (v_i, v_j). The value $f_i(t)$ describes the temperature of node i at time t, beginning with an initial distribution of temperature given by $f_i(0)$ at time zero.

Next, we consider the representation ability of each node. In a manifold, there are infinite nodes on it, but only a finite number n of nodes are known and form the graph. We can assume that there is a small patch $P(j)$ of space containing node j and many other nodes around node j; node j is seen by the observer, but the small patch is unseen to the observer. The volume of the small patch $P(j)$ is $V(j)$.

A. Establishment of VHDM

In this section, we try to establish the heat-diffusion model by employing Fourier's law, which states that the rate of heat flow through a homogenous solid is directly proportional to the area of the section at right angles to the direction of heat flow and proportional to the temperature difference along the path of heat flow. Heat always conducts from warmer objects to cooler objects. If there is no temperature difference, no heat will diffuse. The larger the temperature difference, the more quickly the heat diffuses. The relation between the rate of heat conduction and the contact area is the same as that between the rate of heat conduction and the temperature difference. According to Fourier's law, both the temperature difference and the contact area affect the conduction rate linearly.

Suppose, at time t, the unit volume containing i receives an amount $HM(i, j, t, \Delta t)$ of heat from its neighbor j during a period of Δt. Then, according to Fourier's law, we assume that we have the following conditions: 1) The heat $HM(i, j, t, \Delta t)$ should be proportional to the time period Δt and the temperature difference $f_j(t) - f_i(t)$; and 2) the amount of heat that patch $P(j)$ diffuses to the unit volume containing i is proportional to the surface area $S(i)$ of the unit volume. Moreover, the heat flows from node j to node i through the pipe that connects nodes, and therefore, the heat diffuses in the pipe in the same way as it does in the 1-D Euclidean space, as described in (2). Consequently, we further assume that $HM(i, j, t, \Delta t)$ is proportional to $e^{-w_{ij}^2}$, the amount of heat that a unit heat source at node j transferred to node i, which is a fact in 1-D Euclidean space. In addition, the temperature in the small patch $P(j)$ at time t is almost equal to $f_j(t)$ because every unseen node in the small patch is near to node j, and so, the amount of heat in patch $P(j)$ is proportional to $V(j)$. As a result

$$HM(i, j, t, \Delta t) = \alpha S(i) e^{-w_{ij}^2/\beta} \left(f_j(t) - f_i(t)\right) V(j) \Delta t.$$

The amount of heat in the unit volume containing i is equal to $f_i \cdot 1$. The heat difference in this unit volume should be $f_i(t + \Delta t) - f_i(t)$, which is caused by the sum of the heat that it receives from all its neighbors and the small patches around these neighbors. This is formulated as

$$f_i(t+\Delta t)-f_i(t)=\alpha \sum_{j:(j,i)\in E} S(i)e^{-w_{ij}^2/\beta}(f_j(t)-f_i(t))V(j)\Delta t \tag{10}$$

which can be further expressed as $\alpha\Delta t(A-B)$, where $A=\sum_{j:(j,i)\in E} S(i)e^{-w_{ij}^2/\beta}f_j(t)V(j)$ and $B=f_i(t)\sum_{j:(j,i)\in E}\times S(i)e^{-w_{ij}^2/\beta}V(j)$. To find a closed-form solution to (10), we express it as a matrix form: $(f(t+\Delta t)-f(t)/\Delta t)=\alpha Hf(t)$, where $f(t)$ is the vector $(f_1(t),f_2(t),\ldots,f_{M+N}(t))^{\mathrm{T}}$, $f(t+\Delta t)$ is the vector $(f_1(t+\Delta t),f_2(t+\Delta t),\ldots,f_{M+N}(t+\Delta t))^{\mathrm{T}}$, $H=(H_{ij})$, and

$$H_{ij}=\begin{cases}-\sum_{k:(k,i)\in E} S(i)e^{-w_{ik}^2/\beta}V(k), & j=i\\ S(i)e^{-w_{ij}^2/\beta}V(j), & (j,i)\in E\\ 0, & \text{otherwise.}\end{cases} \tag{11}$$

In the limit $\Delta t\to 0$, we have $(d/dt)f(t)=\alpha Hf(t)$. Solving it, we obtain a closed-form expression

$$f(t)=e^{\alpha tH}f(0)=e^{\gamma H}f(0) \tag{12}$$

where $\gamma=\alpha t$, and $e^{\gamma H}$ is defined as

$$e^{\gamma H}=I+\gamma H+\frac{\gamma^2}{2!}H^2+\frac{\gamma^3}{3!}H^3+\cdots\approx\left(I+\frac{\gamma}{s}H\right)^s. \tag{13}$$

The matrix $e^{\gamma H}$ is called the *diffusion kernel* in the sense that the heat diffusion between nodes from time zero to t is completely described by the elements in the matrix. For the sake of computational considerations, $e^{\gamma H}f(0)$ can be approximated as $(I+\gamma/sH)^s f(0)$, where s is a large integer. The latter can be calculated by iteratively applying the operator $(I+\gamma/sH)$ to $f(0)$.

In the model, $V(i)$ is used to estimate the volume of the small patch around node i. Intuitively, if the data density is high around node i, the nodes around node i will have a high probability of being selected, and thus, there are fewer unseen nodes around node i. In this paper, we define $V(i)$ to be mean value of $1/n$ and the normalized volume of the hypercube whose side length is the distance between node i and its nearest neighbor. Formally

$$V(i)=\eta \min_{j:(j,i)\in E} w_{ij}^{\nu}/2+1/2n \tag{14}$$

where ν is the dimension of the space in which graph G lies and η is a normalized parameter such that $\sum_{i\in V}V(i)=1$.

In the earlier discussions, we established VHDM by physical intuitions. Next, we will show a mathematical justification for the introduction of volumes, and as a by-product, we find a way to calculate the contact area $S(i)$.

B. Why Introducing Volumes: Mathematical Foundation

In this section, we show that it is necessary to introduce the concept of volumes from three aspects.

1) Justification by Integral Approximations: In this section, except for volumes, we follow the approximation techniques employed in [3]. Note that, when all the volumes are equal, the last approximation in (16) becomes the case in [3].

It turns out that in an appropriate coordinate system $K_t(\mathbf{x},\mathbf{y})$ on a manifold is approximately the Gaussian

$$K_t(\mathbf{x},\mathbf{y})=(4\pi t)^{\frac{-m}{2}}e^{-\|\mathbf{x}-\mathbf{y}\|^2/4t}\left(\phi(\mathbf{x},\mathbf{y})+O(t)\right) \tag{15}$$

where $\phi(\mathbf{x},\mathbf{y})$ is a smooth function with $\phi(\mathbf{x},\mathbf{x})=1$ and, when t is small, $O(t)$ can be neglected. Therefore, when $\mathbf{x}$ and $\mathbf{y}$ are close and t is small, we have $K_t(\mathbf{x},\mathbf{y})\approx(4\pi t)^{-m/2}e^{-\|\mathbf{x}-\mathbf{y}\|^2/4t}$ (for more details, see [3] and [17]).

It is well known that the solution to (1) can be expressed as $f(\mathbf{x},t)=\int_M K_t(\mathbf{x},\mathbf{y})f_0(\mathbf{y})$. From $\mathcal{L}f(\mathbf{x},t)=-\partial f(\mathbf{x},t)/\partial t$, we have

$$\begin{aligned}\mathcal{L}f(\mathbf{x}_i,t)&\approx\left(f(\mathbf{x}_i,t)-f(\mathbf{x}_i,t+\Delta t)\right)/\Delta t\\ &\approx\left(f(\mathbf{x}_i,t)-\int\limits_M K_{\Delta t}(\mathbf{x}_i,\mathbf{y})f(\mathbf{y},t)\right)/\Delta t\\ &\approx\Bigg(f(\mathbf{x}_i,t)-(4\pi\Delta t)^{-\frac{m}{2}}\\ &\qquad\times\int\limits_M e^{-\|\mathbf{x}_i-\mathbf{y}\|^2/4\Delta t}f(\mathbf{y},t)\Bigg)/\Delta t\\ &\approx\frac{1}{\Delta t}\Bigg(f(\mathbf{x}_i,t)-(4\pi\Delta t)^{-\frac{m}{2}}\\ &\qquad\times\sum_{j:(j,i)\in E}e^{-\|\mathbf{x}_i-\mathbf{x}_j\|^2/4\Delta t}f(\mathbf{x}_j,t)V(j)\Bigg)\end{aligned} \tag{16}$$

where volumes are considered as the partition of M and the last approximation is based on the definition of the integral, which will become an equality if $\bigcup_{\{j:(j,i)\in E\}}P(j)=M$, $P(j)\cap P(k)=\emptyset$, and $\max V(j)\to 0$. To satisfy the earlier conditions, the volume of a patch should occupy the manifold as much as possible while every two patches are not intersected. When the number of data is large enough, $\max V(j)$ will be small enough. This motivates us to define the volume in (14). Note that, when data points are evenly distributed, $V(i)$ will be a constant, and thus, the effect of volumes will disappear.

However, in practice, the earlier three conditions cannot be satisfied, so errors may arise in the last approximation. For example, (16) may not possess the property that a constant temperature distribution at time t will also result in a constant temperature distribution at time $t+\Delta t$. To satisfy such a property, we change the constant $(4\pi\Delta t)^{-m/2}$ to a variable $S(i)$. This idea is similar to the constant adaptation method in solving some ordinary differential equations. From the knowledge that a constant temperature distribution at time t will also result in a constant temperature distribution at time $t+\Delta t$, we have $1\approx S(i)\sum_{j:(j,i)\in E}e^{-\|\mathbf{x}_i-\mathbf{x}_j\|^2/4\Delta t}V(j)$, and so

$$S(i)\approx 1\Bigg/\left(\sum_{j:(j,i)\in E}e^{-\|\mathbf{x}_i-\mathbf{x}_j\|^2/4\Delta t}V(j)\right). \tag{17}$$

This is the definition of the surface area of node i. The intuition is that the larger the volumes of its neighbors, the less the surface that is left to i. This intuition comes from the observation that, in Fig. 1(d), larger volumes of squares A, B, C, D, and E force the surface of O to be smaller.

2) Justification by the Definition of a Manifold: Volumes are theoretically important because heat diffuses throughout the whole space of any given volume in a physical system, and the concept of the volume is crucial in its ability to represent the whole space, including both known points and other points between them. Moreover, the idea of volume can be explained further by the definition of local charts in a differential manifold as shown in [14].

Definition 2: An m-dimensional differential manifold $\mathcal{M}$ is a set of points that is locally equivalent to the m-dimensional Euclidean space $\mathcal{R}^m$ by smooth transformations, supporting operations such as differentiation. Formally, a differentiable manifold is a set $\mathcal{M}$ together with a collection of local charts $\{(U_i, \phi_i)\}$, where $U_i \subset \mathcal{M}$ with $\cup_i U_i = \mathcal{M}$, and $\phi_i : U_i \subset \mathcal{M} \to \mathcal{R}^m$ is a bijection from U_i to $\phi_i(U_i)$. For each pair of local charts (U_i, ϕ_i) and (U_j, ϕ_j), it is required that $\phi_j(U_i \cap U_j)$ is open and $\phi_{ij} = \phi_i \circ \phi_j^{-1}$ is a diffeomorphism. □

The small patch around each point i can be considered as a local charts U_i, and the volume of i is the volume of U_i. Consequently, the whole manifold $\mathcal{M}$ is formed by joining the small patches together.

3) Justification by Practice Considerations: In VHDM, if $\beta \to +\infty$, the graph is in the form as shown in Fig. 1(a), which means that each node has four neighbors, and if the volume of each node is set to one, then (10) becomes (6). Therefore, we can say that VHDM generalizes the FD method from a Euclidean space to an unknown space. The generalization is interesting for its ability to solve the following problems.

1) **Irregularity of the graph**. By setting β to be finite, we actually soften the neighborhood relation between the data points, and thus, we avoid the difficulty in handling the irregularity of the graph constructed by the data points. For example, in Fig. 1(b), the central data point has four neighbors, which are not regularly positioned on nodes in the grid. The FD method has difficulty in handling such a case. Even worse, in real data sets, each data point has many neighbors, which are positioned in a space with an unknown dimension.
2) **Variation of density**. The data points are not drawn uniformly, and we use the volume of the hypercube around a node to perform the local density estimation around the node. In Fig. 1(c), the whole space is covered by small patches, and in Fig. 1(d), each small patch is approximated by a small square. In this way, we actually consider the unseen points so that the concept of heat diffusion on a graph can be treated as an approximation of heat diffusion in a space. There is no such consideration in the FD method.
3) **Unknown manifold and unknown differential-equation expression**. In most cases, we do not know the true manifold that the data points lie in, or we cannot find the exact expression for the *Laplace–Beltrami operator*; therefore, we cannot employ the FD method. In contrast, our model has the advantage of not depending on the manifold expression and the differential-equation expression. Moreover, volumes serve as patches that are connected together to form the underlying unknown manifold, while each volume is a local Euclidean space. The idea of volume fits the definition of local charts in differential manifold.

C. Determining ν Using Existing Methods

In the definition of volumes, we introduce the parameter ν describing the dimension of the space in which graph G lies. From the definition of a differential manifold, ν corresponds to the unknown dimension m of the local Euclidean space. In the following, we consider how to determine the value of this parameter.

PCA is a traditional method for dimension estimation. In this method, the intrinsic dimension is determined by the number of eigenvalues greater than a given threshold. Both global and local PCAs have the disadvantage of introducing another parameter—the threshold. To avoid such a new parameter, we choose the maximum-likelihood estimation method proposed in [32].

In our proposed classifier, the graph is constructed by K nearest neighbors (KNNs), which will be described in Section IV. This parameter K is the same as the one employed in dimension estimation by the maximum-likelihood estimation. Thus, we avoid in introducing another new parameter. In addition, this method helps to reduce the complexity of searching the parameter K in that we can discard those K's by which the estimated dimensions are greater than the number of attributes or are less than one.

Although the parameter K could be selected automatically in the proposed method, it is still a parameter. We note that the selection used in this paper could be suboptimal.

1) Maximum-Likelihood Estimation of Intrinsic Dimension: If $T_j(x)$ is the Euclidean distance from a fixed point x to its jth nearest neighbor in the sample, then the local dimension $\hat{m}_K(x)$ at point x can be estimated by a maximum-likelihood estimation, as described in [32], as follows:

$$\hat{m}_K(x) = \left[\frac{1}{K-1} \sum_{j=1}^{K-1} \log \frac{T_K(x)}{T_j(x)} \right]^{-1}. \tag{18}$$

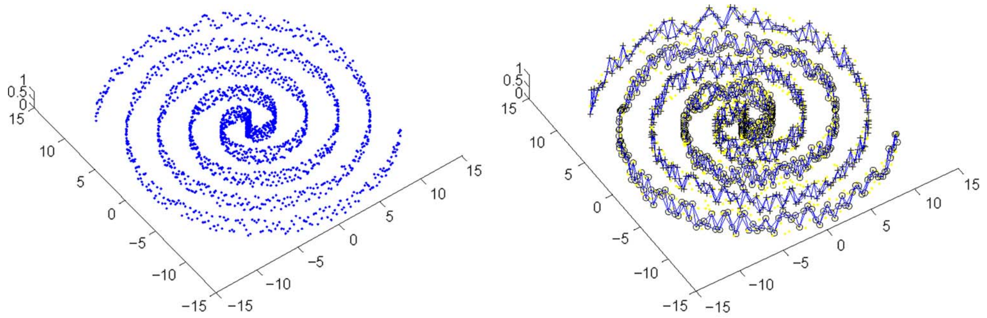
To avoid overflowing during calculations when $T_j(x)$ is very small, we slightly change (18) to the following:

$$\hat{m}_K(x) = \left[\frac{1}{K-1} \sum_{j=1}^{K-1} \log \frac{T_K(x)+\epsilon}{T_j(x)+\epsilon} \right]^{-1}. \tag{19}$$

ϵ is set to be 0.0000001 in this paper. To obtain a unique dimension ν as required by Definition 2, we need to average all these estimated dimensions. Then, we have $\nu = 1/n \sum_{i=1}^{n} \hat{m}_K(x_i)$. In fact, we observe that an arbitrary selection of the parameter ϵ in the interval [0.0000001, 0.001] cannot produce much difference on the estimation of the dimension, and so, we need not pay much care on the selection of ϵ.

IV. VHDC

In the case that the underlying geometry is unknown or its heat kernel cannot be approximated in the same way as used



(a) (b)

Fig. 2. Illustration of the spiral manifold and its graph approximation. (a) Two thousand data points on a spiral manifold. (b) Neighborhood graph of the 1000 data points on the spiral manifold.

by [14], it is natural to approximate the unseen manifold by the graph created by the KNNs in our model and to establish a heat-diffusion model on the neighborhood graph rather than on the underlying geometry. The graph embodies the discrete structure of the nonlinear manifold. By doing so, we can imitate the way that heat flows through a nonlinear manifold. In this section, we first establish the algorithm based on the closed-form solution in (12), then we discuss the free parameters, and finally we justify the necessity of introducing the heat-diffusion model in the field of classification.

A. Establishment of the Algorithm

We assume that there are c classes, namely, $1, 2, \dots, c$. Let the labeled data set contain M samples, $(\mathbf{x}_i, k_i)(i = 1, 2, \dots, M)$, which means that the data point $\mathbf{x}_i$ belongs to class k_i. Suppose the labeled data set contains M_k points in class k so that $\sum_k M_k = M$. Moreover, the unlabeled data set containing N unlabeled samples is represented by $\mathbf{x}_i(i = M+1, M+2, \dots, M+N)$.

We first employ the neighborhood-construction algorithm commonly used in the literatures, for example, in [1]–[4], and [33], to form a graph for all the data. Then, we apply the heat-diffusion kernel to the graphs. For the purpose of classification, for each class k in turn, we set the initial heat at the labeled data point in class k to be one and all other data points to be zero, then calculate the amount of heat that each unlabeled data point receives from the labeled data points in class k. Finally, we assign the unlabeled data point to the class from which it receives most heat. More specifically, we describe the resulting VHDC as follows.

Step 1) **Construct neighborhood graph**. Define graph G over all data points both in the training data set and in the unlabeled data set by connecting points $\mathbf{x}_j$ and $\mathbf{x}_i$ from $\mathbf{x}_j$ to $\mathbf{x}_i$ if $\mathbf{x}_j$ is one of the KNNs of $\mathbf{x}_i$, measured by the Euclidean distance. Let $d(i, j)$ be the Euclidean distance between point $\mathbf{x}_i$ and point $\mathbf{x}_j$. Set edge weight w_{ij} to be equal to $d(i, j)$ if $\mathbf{x}_i$ is one of the KNNs of $\mathbf{x}_j$, and set $n = M + N$.

Step 2) **Compute the Heat Distribution**. Using (13), we obtain c results for $f(t)$, namely, $f^k(t) = e^{\gamma H} f^k(0)$, $k = 1, 2, \dots, c$, where $f^k(0) = (x_1^k, x_2^k, \dots, x_M^k, 0, 0, \dots, 0)^{\mathrm{T}}$, $k = 1, 2, \dots, c$, $x_i^k = 1$ if $k_i = k$, and $x_i^k = 0$, otherwise. Here, $f^k(0)$ means that all the data points in class k have unit heat at the initial time, while other data points have no heat, and the corresponding result $f^k(t)$ means that the heat distribution at time t is caused by the initial temperature distribution $f^k(0)$.

Step 3) **Classify the data**. For $l = 1, 2, \dots, N$, compare the pth $(p = M + l)$ component of $f^1(t), f^2(t), \dots, f^c(t)$, and choose class k such that $f_p^k(t) = \max_{q=1}^{c} f_p^q(t)$, i.e., choose the class that distributes the most heat to the unlabeled point $\mathbf{x}_p$, then classify the unlabeled point $\mathbf{x}_p$ to class k.

As an example of step 1), in Fig. 2(a), we show 2000 points on a 2-D spiral manifold which is embedded into 3-D space. In Fig. 2(b), we show the neighborhood-graph approximation of the spiral manifold, which contains 1000 points drawn from the 2000 ones in Fig. 2(a) and in which each node has three neighbors. In step 2), the heat diffuses from the labeled data to the unlabeled data along the graph, and consequently, the heat flows along the spiral manifold. In step 3), if the unlabeled data point is closer to one class in the sense that it receives more heat in total from this class of data, then the unlabeled data point is classified into this class; otherwise, it is classified into the other class.

The outside appearance of $e^{\gamma H}$ is the same as that in [7] and [25]; however, the numerical value of $e^{\gamma H}$ in our paper is quite different from [7] and [25] [refer to (11)]. The heat kernel in [7], [14], and [25] is applied to a large-margin classifier; in contrast, the heat kernel is employed directly to construct a classifier in our model.

Here, we analyze the computational complexity of VHDC given a constructed KNN graph (or another kind of graph). In VHDC, for each class, we need to compute the multiplication between a vector and a sparse matrix $I + \gamma/sHs$ times. Since $I + \gamma/sH$ has Kn nonzero elements $(n = N + M)$, we need s multiplications in total. Usually, $c \ll n$, and s is set to be 100 in practice. Therefore, the complexity of VHDC is $O(Kn)$, given a constructed KNN graph. Similarly, it is linear on the number of edges in other constructed graphs. Considering the straightforward KNN graph-construction method, which needs $O(Kn^2)$ comparisons, the overall complexity of VHDC is $O(Kn^2)$ if the KNN graph needs to be constructed in VHDC.

B. Roles of the Parameters

It is easy to find out that K is used to control the manifold approximation, and ν is used to model the true dimensionality of the manifold that the data lie in.

1) Local Heat Diffusion Controlled by β: In Section III-A, we assumed that the heat diffuses along the pipe in the same way as it does in the 1-D Euclidean space. Next, we will justify this assumption. In VHDM in Section III, heat flows in a small time period Δt, and the pipe length between node i and node j is small (recall that we create an edge from j to i only when j is one of the KNNs). Therefore, the approximation in (15) can be used in our model, and we rewrite it as $K_{\Delta t}(i,j) \approx (4\pi\Delta t)^{-m/2}e^{-w_{ij}^2/4\Delta t}$. According to the mean-value theorem and the fact that $K_0(i,j)=0$, we have $K_{\Delta t}(i,j) = K_{\Delta t}(i, j) - K_0(i,j) = dK_{\Delta t}(i,j)/d\Delta t|_{\Delta t=\beta}\Delta t \approx \alpha \cdot e^{-w_{ij}^2/4\beta}\Delta t$, where β is a parameter that depends on Δt, and $\alpha = 1/4w_{ij}^2\beta^{-2-m/2} - 1/2m\beta^{-1-m/2}$. To make our model concise, α and β simply serve as free parameters because the relation between Δt and β is unknown. This explains the statement that β controls the local heat diffusion from time t to $t+\Delta t$ and the reason why we assume that, at time t, the amount of heat that node i receives from its neighbor j is proportional to $e^{-w_{ij}^2/\beta}$.

2) Global Heat Diffusion Controlled by γ: From $\gamma = \alpha t$, we can see that γ controls the global heat diffusion from time zero to t. Another interesting finding is that γ can be explained as a regularization parameter: When $\gamma = 0$, we have $e^{\gamma H}f(0) = If(0) = f(0)$, resulting in a classifier which has zero error on the training set. When $\gamma \rightarrow +\infty$, the system will stop diffusing heat, and the heat at each node are equal. This means that the function on the graph becomes smoothest in the sense that the variance between values on neighbors is smallest. The best γ is a tradeoff between the training error and the smoothness and should not be zero or infinity.

Finally, we investigate the singular behavior of VHDC in the limit $\gamma \rightarrow 0$. If we simply let $\gamma = 0$ in the equation $e^{\gamma H}f(0)$, then we only get a trivial classifier as shown earlier. From a different viewpoint, we observe the following interesting phenomenon.

Subtracting I from $e^{\gamma H}$ then dividing by γ changes the values of the testing data in the same scale, and so, it does not change the performance of the classifier, i.e., $(e^{\gamma H} - I)/\gamma f(0)$ behaves the same as $e^{\gamma H}f(0)$ as a classifier. Then, we can take the limit over $(e^{\gamma H} - I)/\gamma f(0)$ and obtain

$$\begin{aligned}\lim_{\gamma\rightarrow 0}\frac{(e^{\gamma H}-I)}{\gamma}f(0) &= \lim_{\gamma\rightarrow 0}\frac{I+\gamma H+\frac{\gamma^2}{2!}H^2+\cdots-I}{\gamma}f(0)\\ &= \lim_{\gamma\rightarrow 0}\left(H+\frac{\gamma}{2!}H+\cdots\right)f(0)\\ &= Hf(0).\end{aligned}$$

We consider $Hf(0)$ as the singular behavior of VHDC in the limit $\gamma \rightarrow 0$.

3) Stability of VHDC With Respect to Parameters: There are three free parameters in VHDC. If the parameters in the model is not stable, then a small deviation from the best value of a parameter may result in a totally different performance. This instability of the parameters is not desirable. In this section, we will show that the classifier VHDC is not sensitive to the parameters β, γ, and K.

Since $e^{\gamma H}$ is continuous on β and γ in the sense that small changes in these parameters result in a small change in $e^{\gamma H}$, VHDC is not sensitive to these three parameters if they change slightly.

For easy analysis, in the following, we ignore the volumes and surfaces. The existence of the derivatives of $e^{\gamma H}$ with respect to β and γ can be seen in the following:

$$\frac{de^{\gamma H}}{d\gamma} = e^{\gamma H}H \tag{20}$$

$$\frac{de^{\gamma H}}{d\beta} = \gamma e^{\gamma H}\frac{dH}{d\beta} \quad \frac{dH}{d\beta} = \left(\frac{dH_{ij}}{d\beta}\right) \tag{21}$$

$$\frac{dH_{ij}}{d\beta} = \begin{cases} -\sum_{k:(k,i)\in E} e^{-w_{ik}^2/\beta}w_{ik}^2\beta^{-2}, & j=i\\ e^{-w_{ij}^2/\beta}w_{ij}^2\beta^{-2}, & (j,i)\in E\\ 0, & \text{otherwise.}\end{cases} \tag{22}$$

It is well known that $\Delta \mathbf{f} \approx d\mathbf{f}/dt\Delta t$. Since there exist derivatives of $e^{\gamma H}$ with respect to β and γ, we can say that $e^{\gamma H}$ is stable with respect to these parameters and so is $e^{\gamma H}f(0)$.

If H is symmetric, then we can estimate an upper bound for these derivatives. First of all, we claim that the i-row and j-column elements in $e^{\gamma H}$ mean the amount of heat that i receives from a unit heat source at j. Therefore, physically, we claim that $e^{\gamma H}$ is a nonnegative matrix. Next, we show that the sum of each row in $e^{\gamma H}$ is equal to one. Let $\mathbf{1}$ and $\mathbf{0}$ be the vector of all ones and the vector of all zeros, respectively. Then, $H\mathbf{1} = \mathbf{0}$. According to (13), we have

$$e^{\gamma H}\mathbf{1} = I\mathbf{1} + \gamma H\mathbf{1} + \frac{\gamma^2}{2!}H^2\mathbf{1} + \cdots = \mathbf{1}$$

which means that the sum of each row in $e^{\gamma H}$ is equal to one. Consequently, we can assume that each row in $e^{\gamma H}$ is a vector $(a_1, a_2, \ldots, a_n)$ satisfying $a_i \geq 0$ and $\sum_i a_i = 1$. Let $(b_1, b_2, \ldots, b_n)^{\mathrm{T}}$ be a column in H. Then, each element in $de^{\gamma H}/d\gamma$ is of the form $(a_1, a_2, \ldots, a_n)(b_1, b_2, \ldots, b_n)^{\mathrm{T}}$ by (20). By Hölder's inequality $(p=1, q=\infty)$, we have $(a_1, a_2, \ldots, a_n)(b_1, b_2, \ldots, b_n)^{\mathrm{T}} \leq (\sum_i |a_i|)\max_i |b_i| = \max_i |b_i| \leq K$, which means that each element in $\frac{de^{\gamma H}}{d\gamma}$ is not greater than K. Similarly, if $\gamma e^{-w_{ij}^2/\beta}w_{ij}^2\beta^{-2} \leq 1$ for all i and j, then each element in $de^{\gamma H}/d\beta$ is not greater than K.

For the parameter K, it has an unstable effect on the constructed neighborhood graph. Increasing K by one will result in a structural change in the underlying KNN graph. However, the numerical values in the matrices H do not change much. We analyze this here. Let i_j denote the jth nearest neighbor of i. If we increase K by one, then only the ii_{K+1}th element $H_{ii_{K+1}}$ in the ith row of H in (11) is changed from zero to nonzero. More specifically

$$H_{ii_{K+1}} = S(i)e^{-w_{ii_{K+1}}^2/\beta}V(i_{K+1}) = \frac{e^{-w_{ii_{K+1}}^2/\beta}V(i_{K+1})}{\sum_{j=1}^{K+1}e^{-w_{ii_j}^2/\beta}V(i_j)}.$$

If volumes are ignored, then $H_{ii_{K+1}}$ is less than $1/K+1$, since $w_{ii_j} \leq w_{ii_{K+1}}$ for $j = 1, 2, \ldots, K+1$.

For $j \leq K$, if we increase K by one, then the ii_jth element H_{ii_j} in the ith row of H in (11) is changed from a nonzero number to another. The difference is

$$\frac{e^{-w_{ii_j}^2/\beta}V(i_j)}{\sum\limits_{j=1}^{K} e^{-w_{ii_j}^2/\beta}V(i_j)} - \frac{e^{-w_{ii_j}^2/\beta}V(i_j)}{\sum\limits_{j=1}^{K+1} e^{-w_{ii_j}^2/\beta}V(i_j)}$$

$$= \frac{e^{-w_{ii_j}^2/\beta}V(i_j) \cdot e^{-w_{ii_{K+1}}^2/\beta}V(i_{K+1})}{\sum\limits_{j=1}^{K} e^{-w_{ii_j}^2/\beta}V(i_j) \cdot \sum\limits_{j=1}^{K+1} e^{-w_{ii_j}^2/\beta}V(i_j)}$$

which is less than $1/K+1$ if volumes are ignored. In any case, the elements in H do not change much if K is increased by one. So do the elements in $e^{\gamma H}$, since $e^{\gamma H}$ is continuous on H.

C. Need for Heat-Diffusion Model in Classification

It is not absolutely necessary to have a physical model behind a learning algorithm. However, the situation is different for the heat-diffusion model, since many learning algorithms can be interpreted by the heat-diffusion model theoretically, although not empirically.

We first show that KNN can be considered as a special case of VHDC (when $\beta \to +\infty$, $N = 1$, and γ is small and when volumes are constant and thus the contact surfaces are constant, VHDC becomes KNN); and when the window function is a multivariate normal kernel, the Parzen window approach (PWA) [34] can be considered as a special case of HDC (when $K = n - 1$ and γ is small and when volumes are constant, VHDC becomes the PWA).

When the parameter γ is small, we can approximate $e^{\gamma H}$ in (13) by its first two items, i.e., $e^{\gamma H} \approx I + \gamma H$, then, in VHDC, $f^k(t) = e^{\gamma H} f^k(0) \approx f^k(0) + \gamma H f^k(0)$. As the constant γ and the first item $f^k(0)$ have no effect on the classifier, VHDC possesses a similar classification ability to that determined by the equation $f^k(t) = H f^k(0)$. As a classifier, $H f^k(0)$ will not be affected by an arbitrary scaling on each row of H, and so, the surface factor can be ignored in $H f^k(0)$. This result will be used in the next two sections.

1) VHDC and PWA: First, we review the Parzen window nonparametric method for density estimation using Gaussian kernels. When the kernel function $H(u)$ is a multivariate normal kernel, a common choice for the window function, by the estimate of the class-conditional densities for class C_k and Bayes's theorem, we have [34] as follows: The density at the point x is

$$\widetilde{p}(\mathbf{x}) = \frac{1}{M} \sum_{i=1}^{M} \frac{1}{(2\pi h^2)^{d/2}} e^{-\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{2h^2}}. \tag{23}$$

When applying it for classification, we need to construct the classifier through the use of Bayes's theorem. This involves modeling the class-conditional densities for each class separately and then combining them with priors to give models for the posterior probabilities which can then be engaged to make classification decisions [34]. The class-conditional densities for class C_k can be obtained by extending (23)

$$\widetilde{p}(\mathbf{x}|C_k) = \frac{1}{M_k} \sum_{i:C_{k_i}=C_k} \frac{1}{(2\pi h^2)^{d/2}} e^{-\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{2h^2}} \tag{24}$$

where the priors can be estimated by $\widetilde{p}(C_k) = M_k/M$. By Bayes's theorem, we get

$$\widetilde{p}(C_k|\mathbf{x}) = \frac{\sum_{i:C_{k_i}=C_k} e^{-\|\mathbf{x}-\mathbf{x}_i\|^2/2h^2}}{Mp(\mathbf{x})(2\pi h^2)^{d/2}}. \tag{25}$$

Without considering the volume, if we set $K = n - 1$ and if γ is small, then the graph constructed in step 1) will be a complete graph, and the matrix H in (11) becomes

$$H_{ij} = \begin{cases} -\sum_{k \neq i} e^{-w_{ik}^2/\beta}, & j = i \\ e^{-w_{ij}^2/\beta}, & j \neq i. \end{cases} \tag{26}$$

Then, in VHDC, the heat $f_p^k(t)$ that unlabeled data $\mathbf{x}_p$ receives from the data points in class C_k will be equal to $\sum_{i:C_{k_i}=C_k} e^{-\|\mathbf{x}_p-\mathbf{x}_i\|^2/\beta}$, which is the same as (25) if we let $\gamma = 1/Mp(\mathbf{x})(2\pi h^2)^{d/2}$, and $\beta = 2h^2$. This means that, when the window function is a multivariate normal kernel, the PWA can be considered as a special case of VHDC.

2) VHDC and KNN: If β tends to infinity and, again, the volume is not considered, then $-w_{ij}^2/\beta$ will tend to zero, and the matrix H in (11) becomes

$$H_{ij} = \begin{cases} -O_i, & j = i \\ 1, & \mathbf{x}_j \text{ is one neighbor of } \mathbf{x}_i \\ 0, & \text{otherwise} \end{cases} \tag{27}$$

where O_i is the outdegree of the point $\mathbf{x}_i$ (note that the indegree of the point $\mathbf{x}_i$ is K). Then, in VHDC when γ is small, the heat $f_p^q(t)$ that unlabeled data x_p receives from the data points in class C_q will be equal to $f_p^q(t) = \sum_{i:l_i=C_q} 1 = K_q$, where K_q is the number of the labeled data points from class C_q, which are the KNNs of the unlabeled data point $\mathbf{x}_p$. Note that when $N = 1$, i.e., when the number of unlabeled data is equal to one, $\sum_{q=1}^{c} K_q = K$. According to step 3), we will classify the unlabeled data $\mathbf{x}_p$ to the class C_k such that $f_p^k(t) = K_k$ is the maximal among all $f_p^q(t) = K_q$. This is exactly what KNN does, and so, KNN can be considered as a special case of VHDC.

We show one advantage of the generalization of KNN. It is well known that expected error rate of KNN is between P and $2P$ when N tends to infinity, where P is the Bayes error rate. Therefore, the upper bound of the expected error rate of VHDC is less than $2P$ if β is infinity and volumes are constant. It should be tighter if appropriate parameters for VHDC are found.

3) VHDC and CM: Although our model VHDC adopts a different approach to CM in [6], there is an overlap between our solution and CM. The overlap happens when the volume is not considered and γ is small in our model, while α is small and the normalization is not performed in [6]. This can be perceived from the approximation $(I - \alpha S)^{-1} \approx I + \alpha S$ when α is small. Subtracting I from $(I - \alpha S)^{-1}$ then dividing by α changes the values of the testing data in the same scale, and so, it does not change the performance of the classifier CM. As a result, $(I - \alpha S)^{-1}Y$ has similar performance to SY when α is small. Similarly, by the approximation $e^{\gamma H} \approx I + \gamma H$ when γ is small, $e^{\gamma H}Y$ has the similar performance as HY. Consequently, when the normalization in CM is not performed and when the volume is not considered in VHDC, S and H are equal except for the diagonal elements, which have no effect on the classifiers SY and HY. Another interesting point is that the classifier $(I - \alpha S)^{-1}Y$ is supported by a regularization framework. It is true that, currently, we cannot find a similar regularization approach that can output the proposed classifier $e^{\gamma H}Y$, but we can interpret it in another way: γ plays a role like the regularization parameter as shown in Section IV-B2.

4) VHDC and Some Other Popular Algorithms: As explained in [35] and [36], a number of popular algorithms such as SVM, Ridge regression, and splines may be broadly interpreted as regularization algorithms with different empirical cost functions and complexity measures in an appropriately chosen reproducing kernel Hilbert space (RKHS). For a Mercer kernel $K : X \times X \to R$, there is an associated RKHS $\mathcal{H}_K$ of functions $X \to R$ with the corresponding norm $\| \|_K$. Given a set of labeled examples $(\mathbf{x}_i; \mathbf{y}_i)$, $i = 1, \ldots, l$, the standard framework estimates an unknown function by minimizing

$$f^* = \arg \min_{f \in \mathcal{H}_K} \frac{1}{l} \sum_{i=1}^{l} V(\mathbf{x}_i, y_i, f) + \alpha \|f\|_K^2$$

where V is some loss function, such as squared loss $(y_i - f(\mathbf{x}_i))^2$ for regularized least square or the hinge loss function $\max[0, 1 - y_i f(\mathbf{x}_i)]$ for SVM. Penalizing the RKHS norm imposes smoothness conditions on possible solutions. The classical representer theorem states that the solution to this minimization problem exists in $\mathcal{H}_K$ and can be written as

$$f^*(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i K(\mathbf{x}, \mathbf{x}_i). \tag{28}$$

If K takes the Gaussian RBF $e^{-\|\mathbf{x}-\mathbf{y}\|^2/\sigma^2}$, then $f^*(\mathbf{x})$ in (28) is the solution of the following heat-diffusion equation on a $m-$dimensional Euclidean space:

$$\begin{cases} \frac{\partial f}{\partial t} - \mathcal{L}f = 0 \\ f(\mathbf{x}, 0) = f_0(\mathbf{x}) \end{cases} \tag{29}$$

where $f_0(\mathbf{x}) = (4\pi t)^{m/2} \sum_{i=1}^{l} \alpha_i \delta(\mathbf{x} - \mathbf{x}_i)$ and $4t = \sigma^2$. This amounts to the solution in (28) can be obtained by solving a heat-diffusion equation with a special initial temperature setting.

It is interesting to mention that the representer theorems for the Laplacian regularized least squares and the Laplacian SVM (manifold regularization) are similar

$$f^*(\mathbf{x}) = \sum_{i=1}^{l+u} \alpha_i K(\mathbf{x}, \mathbf{x}_i) \tag{30}$$

where $\{\mathbf{x}_j\}_{j=l+1}^{l+u}$ denotes the u unlabeled examples. This also means that the solution for Gaussian RBF kernel can be obtained by solving a heat-diffusion equation with a special initial temperature setting.

Therefore, we can say that many learning algorithms can be interpreted by the heat-diffusion model theoretically. This shows the necessity of introducing the heat-diffusion model. However, the problem is where and how we set the initial conditions in the heat-diffusion equation in what kind of space. This paper shows what we can achieve by a simple setting of the initial conditions—set the temperature to be one at the training data points (before we can find an optimization method to find the best initial setting, we have to adopt this "simple and stupid" setting). Although such an attempt may not achieve much accuracy improvement, it inspires a broad research space for future investigations on the heat-diffusion equation, since we feel we have still not fully fulfilled its potential for its applications in classification tasks.

V. Experiments

The PWA, KNN, USVM, CM, and VHDC and HDC (the special version of VHDC when all volumes are one) are applied here to 3 synthetic data sets and 16 data sets from the UCI Repository [37]. Since discrete attributes and the problem of missing values are out of the scope of this paper, we simply remove all the discrete attributes and all the cases that contain missing values. The first four columns in Table I describe the corresponding data sets we use. Note that the data set Zoo is a special example, by which we want to show a problem of VHDC. There are 17 features in the original data. Here, we remove the feature "animal name" that is no use for the classification task, and all the Boolean values are considered as numeric values. We preprocess the data set Zoo with PCA such that the dimensionality is reduced from the original 16 to 8, which is shown in the last row.

We employ the Gaussian RBF kernel for USVM. We obtain the free parameters in PWA, KNN, USVM, CM, HDC, and VHDC via ninefold crossvalidations on the training data set including the testing data without labels. The figures shown in the last six columns in Table I are the mean accuracy of ten runs by dividing the data into 10% for training and 90% for testing and their variances. Note that the results are quite different if we choose the best values in each run in hindsight, i.e., the testing data with label are given when we choose the parameters. However, such a setting has a bias against algorithms with less parameters. If this setting is employed, then VHDC will perform, at least, equally well as either of PWA and KNN does on every data set since VHDC generalizes them.

There are three synthetic data sets with manifold structures. Spiral-1000 is a synthetic data set, which is shown in Fig. 2(b). In the spiral data set, the data points in one class are distributed on a spiral rotated clockwise while the data points in another class are distributed on a spiral rotated counterclockwise. The data set concentric circles shown in Fig. 3(a) consists of two circles that are concentric, and the data set tangent circles in Fig. 3(b) consists of two circles that are tangent at the point (1, 0).

The better results of VHDC over HDC show the necessity of introducing the volume representation of a node in a graph. From the results, we also observe that both HDC and VHDC outperforms PWA and KNN in accuracy, as we expected. The better results on the synthetic data sets show that VHDC fits problems with a manifold structure particularly well. The

TABLE I
DATA SETS DESCRIPTION AND MEAN ACCURACY ON THREE SYNTHETIC DATA SETS, THE 15 BENCHMARK DATA SETS, AND DATA SET ZOO. ACHIEVED BY TEN RUNS BY DIVIDING THE DATA INTO 10% FOR TRAINING AND 90% FOR TESTING

Dataset	Cases	Classes	Features	**PWA**	**KNN**	**USVM**	**CM**	**HDC**	**VHDC**
Spiral-1000	1000	2	3	81.2± 0.56	78.2±0.92	66.6±1.73	80.5±0.69	92.7±0.61	94.1±0.65
Concentric-circles	100	2	2	52.8±0.69	53.3±1.32	50.0±0.00	81.0±0.90	100±0.00	100±0.00
Tangent-circles	100	2	2	52.4±0.70	52.0±1.35	50.0±0.00	52.4±0.81	66.8±0.88	67.2±1.114
Anneal	898	5	6	76.2±0.00	75.8±0.54	45.8±1.34	76.2±0.00	75.6±0.50	75.3±0.68
Auto	195	7	14	33.1±0.00	33.2±1.92	33.2±0.19	33.1±0.00	30.4±0.94	33.7±1.01
Breast-w	683	2	9	95.3±0.26	95.7±0.12	65.1±0.06	96.3±0.15	95.7±0.21	96.0±0.15
Credit-a	666	2	6	52.3±0.96	64.4±1.00	54.9±0.15	55.1±0.00	61.6±1.53	63.8±0.93
Diabetes	768	2	8	65.1±0.71	67.8±0.56	65.1±0.09	65.6±0.32	67.1±0.88	67.2±0.88
Glass	214	6	9	54.3±1.16	51.2±1.12	49.9±3.79	54.7±1.71	55.5±1.36	56.4±1.18
Heart-c	303	2	5	55.0±0.59	60.5±0.52	54.6±0.32	52.1±0.44	59.3±1.32	61.5±1.12
Heart-h	293	2	4	62.2±0.71	63.5±0.90	64.1±0.14	62.6±0.77	63.5±0.90	63.5±0.90
Hepati	148	2	3	79.7±0.00	77.1±1.68	70.3±1.14	79.7±0.00	79.0±2.34	79.4±2.23
Iono	351	2	34	67.5±1.73	79.7±1.38	85.6±1.66	71.4±2.02	80.3±1.67	80.2±1.19
Iris	150	3	4	94.3±0.89	91.1±2.18	93.6±1.09	93.5±1.08	91.7±2.18	92.4±2.15
Sonar	208	2	60	55.5±0.92	61.3±1.53	67.4±1.74	54.7±0.53	60.0±1.31	62.3±1.50
Vehicle	846	4	18	53.6±0.80	49.5±0.92	54.1±0.78	55.0±0.65	52.1±0.82	53.6±1.02
Waveform	300	3	21	74.7±1.31	72.0±1.52	69.0±2.21	76.4±1.18	74.4±1.23	73.9±1.17
Wine	178	3	13	61.6±2.76	66.5±2.68	66.6±0.79	63±2.70	63.6±2.05	63.4±2.40
Zoo	101	7	16	**40.6±0.00**	**40.6±0.00**	**97.2±0.00**	**40.6±0.97**	**40.6±0.00**	**40.6±0.00**
Zoo+PCA	101	7	8	87.0±3.30	90.1±2.98	**97.1±1.44**	96.0±0.70	97.1±0.81	**97.1±0.81**

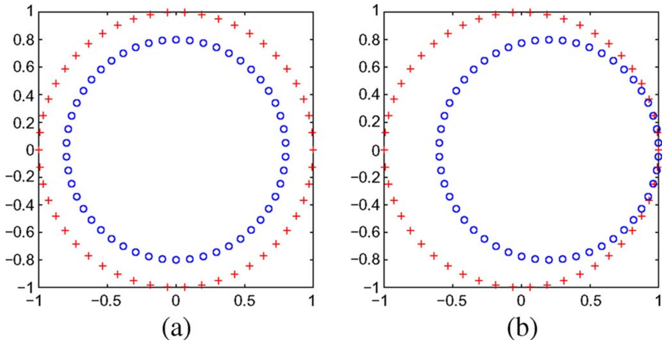


Fig. 3. Two toy data sets.

TABLE II
PAIRED LEFT-TAILED T-TEST ON THE MEAN ACCURACY ON THE 15 BENCHMARK DATA SETS. THE NUMBERS ARE THE P-VALUES. A MARK < IS SHOWN IF THE NULL HYPOTHESIS CAN BE REJECTED AT THE 5% LEVEL

Algorithm	**PWA**	**KNN**	**USVM**	**CM**	**HDC**	**VHDC**
PWA		0.09	0.09	0.08	0.05 (<)	0.01(<)
KNN	0.91		0.93	0.84	0.48	0.05
USVM	0.21	0.07		0.16	0.07	0.04 (<)
CM	0.92	0.16	0.84		0.09	0.03(<)
HDC	0.95	0.52	0.93	0.91		0.01(<)
VHDC	0.99	0.95	0.96	0.97	0.99	

overall results on the 15 benchmark data sets indicate that our approach VHDC outperforms both CM and USVM on problems without any *a priori* knowledge. To give a solid support for the earlier conclusions, we perform a paired left-tailed T-test on each pair of the six algorithms. The results about the 15 benchmark data sets are shown in Table II. The results including both the 15 benchmark data sets and the 3 synthetic data sets are shown in Table III.

To verify the stability of VHDC with respect to its three parameters K, β, and γ, we perform more experiments. After obtaining these parameters by the crossvalidation on the training data sets, we adjust one of them a little while fixing the others; then, we observe how the resulting accuracy changes. From Figs. 4–6, we can see that stability of VHDC with respect to the three parameters: The mean accuracy does not change much when each of them is increased a little.

TABLE III
PAIRED LEFT-TAILED T-TEST ON THE MEAN ACCURACY ON THE THREE SYNTHETIC DATA SETS AND THE 15 BENCHMARK DATA SETS. THE NUMBERS ARE THE P-VALUES. A MARK < IS SHOWN IF THE NULL HYPOTHESIS CAN BE REJECTED AT THE 5% LEVEL

Algorithm	**PWA**	**KNN**	**USVM**	**CM**	**HDC**	**VHDC**
PWA		0.12	0.87	0.11	0.03 (<)	0.01 (<)
KNN	0.88		0.97	0.38	0.07	0.04 (<)
USVM	0.13	0.03 (<)		0.05 (<)	0.01 (<)	0.01 (<)
CM	0.89	0.62	0.95		0.01 (<)	0.005 (<)
HDC	0.97	0.93	0.99	0.99		0.003 (<)
VHDC	0.99	0.96	0.99	0.995	0.997	

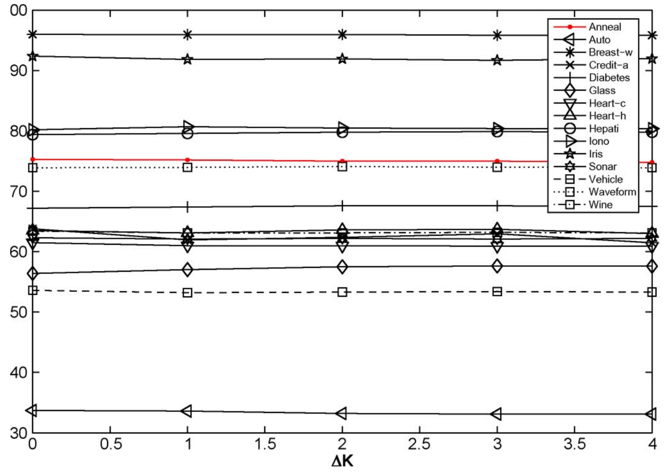


Fig. 4. Variation of the mean accuracy on the 15 data sets as the parameter K found by the crossvalidation is increased by one, two, three, and four. Mean accuracy is achieved by ten runs by dividing the data into 10% for training and 90% for testing.

Despite its success, VHDC is still not perfect. Next, we discuss it from four aspects.

1) **Neighborhood Construction**. Although the KNN neighborhood-construction algorithm is commonly used in the literature, there may exist better neighborhood-construction methods. For example, the idea of using

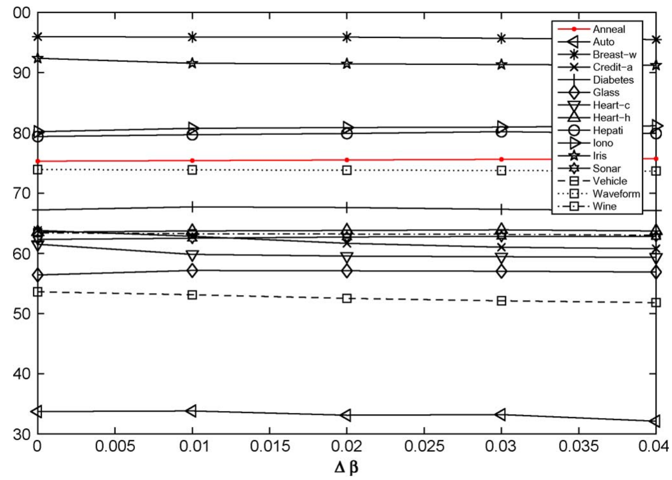


Fig. 5. Variation of the mean accuracy on the 15 data sets as the parameter β found by the crossvalidation is increased by 0.01, 0.02, 0.03, and 0.04. Mean accuracy is achieved by ten runs by dividing the data into 10% for training and 90% for testing.

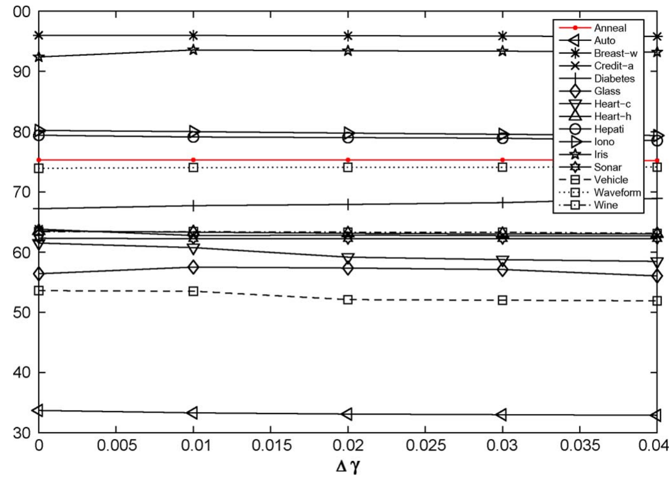


Fig. 6. Variation of the mean accuracy on the 15 data sets as the parameter γ found by the crossvalidation is increased by 0.01, 0.02, 0.03, and 0.04. Mean accuracy is achieved by ten runs by dividing the data into 10% for training and 90% for testing.

mutual information to find neighborhoods [38] can be adopted, and the method of linear neighborhood construction [9] can be inherited.

2) **Dependence on Distance Measure**. The Boolean attributes in the Zoo data set are considered as continuous attributes. We observe that PWA, KNN, CM, HDC, and VHDC achieve 40.6% mean accuracy and perform more poorly than USVM (with 97.2% mean accuracy) on data set Zoo; indeed, the difference is as high as 46.6%. This can be explained by the fact that all these methods depend heavily on the distance measure, and as a consequence, if the direct Euclidean distance is not accurate, these methods will perform poorly. We argue that the noises in the Zoo data set causes inaccurate distance measurement between data points. To find the performance of these algorithms on data set Zoo with less noise, we preprocess it with PCA such that the dimensionality is reduced from the original 16 to 8. The results are encouraging: VHDC achieves 97.1% mean accuracy, the same as what USVM achieves. This example shows that VHDC relies heavily on the local distance, so a suitable feature-extraction method may help to increase its accuracy. A possible solution is to employ the semisupervised metric-learning method proposed in [39].

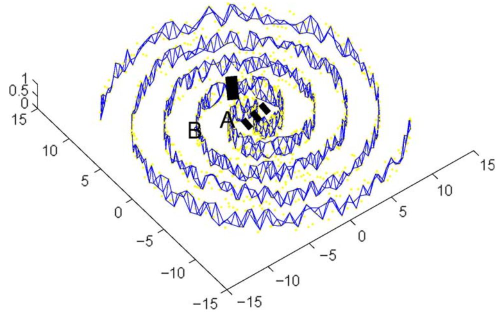


Fig. 7. Illustration showing that the equal setting of initial temperatures is not perfect. Only two data points A and B are labeled, the equal initial temperature setting on these two points will result in classification errors. The decision boundary will be the bar while the dashed line should be the ideal decision boundary.

3) **Local Minimum**. Note that VHDC generalizes both PWA and KNN. However, it is observed that, on data set Iris, VHDC performs worse than PWA, and on data set Wine, VHDC performs worse than KNN. We argue that there exist local minimum problems hidden in the crossvalidation search for the best parameters in VHDC. A possible way to solve this kind of problem is to comprehend the initial temperature distribution as a random field, to estimate the covariance of the random field at time t, and, then, to minimize the appropriately defined error measure including both the fitting error and the variance.

4) **Initial Temperature Setting**. According to the discussions in Section IV-C4, by appropriately setting the initial temperatures, the heat-diffusion model can interpret many learning algorithms. In this paper, the initial temperature is set to be one, and this simple setting will result in some errors. For example, in Fig. 7, a higher initial temperature in point B than that in A is expected in order to achieve the best decision boundary, as indicated by the dashed line. This problem is quite open now.

VI. Conclusion

We have shown how to employ a thermophysical system to achieve promising performance in accuracy in transductive learning. The proposed VHDM has the following advantages: It can model the effect of unseen points by introducing the volume of a node; it avoids the difficulty of finding the explicit expression for the unknown geometry by approximating the manifold by a finite neighborhood graph; and it has a closed-form solution that describes the heat diffusion on a manifold. As an application of VHDM, a classifier VHDC is successfully constructed. We have investigated a number of properties of VHDC, including the computational complexity, the roles of its free parameters, its stability with respect to these parameters, and the connection between VHDC and other algorithms. Intensive experiments have demonstrated that VHDC gives more accurate results in the classification tasks. Experimental results also confirmed the stability of VHDC with respect to its three parameters.

In order to capitalize on these promising achievements, further study is needed on the following problems: How to find a graph that better approximates the manifold instead of the KNN graph; how to utilize some feature-extraction methods in order to make the local distance more accurate; how to find out the parameters which avoid the local minimum problem; how to design a criterion for better setting the initial temperatures; how to construct a better volume representation of the unseen points; and how to apply VHDC to inductive learning. We have discussed some possible solutions to the first four problems in the previous section. Next, we discuss the last two problems. For the problem of applying VHDC to inductive learning, we have to find an inductive function that is not only smooth on the underlying space but also is smooth on the underlying graph in the sense that the value difference between two adjacent nodes is small. A possible way to find such an inductive function is to inherit the idea of manifold regularization and modify the term related to the graph Laplacian by including the volumes. For the problem of constructing a better volume representation of the unseen points, we point out that, ideally, the volume of a patch should occupy the manifold as much as possible while every two patches are not intersected. The current definition of the volume of a node does not satisfy such a property. A possible solution to this problem is to slightly increase the volume to achieve a balance between space occupation and nonintersection.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable comments. They would also like to thank X. Li and Y. Zhou for their help.

References

[1] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 22, 2000.

[2] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, Dec. 22, 2000.

[3] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, Jun. 2003.

[4] X. Geng, D.-C. Zhan, and Z.-H. Zhou, "Supervised nonlinear dimensionality reduction for visualization and classification," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 6, pp. 1098–1107, Dec. 2005.

[5] P. Vincent and Y. Bengio, "Manifold Parzen windows," in *Proc. NIPS*, S. Becker, S. Thrun, and K. Obermayer, Eds., 2003, vol. 15, pp. 825–832.

[6] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Proc. NIPS*, S. Thrun, L. K. Saul, and B. Schölkopf, Eds., 2004, vol. 16, pp. 321–328.

[7] R. I. Kondor and J. D. Lafferty, "Diffusion kernels on graphs and other discrete input spaces," in *Proc. 19th ICML*, C. Sammut and A. G. Hoffmann, Eds., 2002, pp. 315–322.

[8] H. Yang, I. King, and M. R. Lyu, "NHDC and PHDC: Non-propagating and propagating heat diffusion classifiers," in *Proc. 12th ICONIP*, 2005, pp. 394–399.

[9] F. Wang and C. Zhang, "Label propagation through linear neighborhoods," in *Proc. 23rd ICML*, 2006, pp. 985–992.

[10] F. Wang and C. Zhang, "Semisupervised learning based on generalized point charge models," *IEEE Trans. Neural Netw.*, vol. 19, no. 6, pp. 1307–1311, Jul. 2008.

[11] J. Nilsson, F. Sha, and M. I. Jordan, "Regression on manifolds using kernel dimension reduction," in *Proc. 24th ICML*, 2007, pp. 697–704.

[12] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf, "Ranking on data manifolds," in *Proc. NIPS*, S. Thrun, L. Saul, and B. Schölkopf, Eds., 2004, vol. 16, pp. 169–176.

[13] H. Yang, I. King, and M. R. Lyu, "DiffusionRank: A possible penicillin for Web spamming," in *Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retrieval (SIGIR)*, W. Kraaij, A. P. de Vries, C. L. A. Clarke, N. Fuhr, and N. Kando, Eds., 2007, pp. 431–438.

[14] J. Lafferty and G. Lebanon, "Diffusion kernels on statistical manifolds," *J. Mach. Learn. Res.*, vol. 6, pp. 129–163, Dec. 2005.

[15] B. Scholkopf and A. J. Smola, *Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA: MIT Press, 2001.

[16] K. R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 181–201, Mar. 2001.

[17] S. Rosenberg, *The Laplacian on a Riemmannian Manifold*. Cambridge, U.K.: Cambridge Univ. Press, 1997.

[18] A. A. Becker, *An Introductory Guide to Finite Element Analysis*. London, U.K.: Professional Eng. Publ., 2004.

[19] M. Chung and J. Taylor, "Diffusion smoothing on brain surface via finite element method," in *Proc. 2004 IEEE Int. Symp. Biomed. Imag.—From Nano to Macro*, 2004, pp. 432–435.

[20] B. Tang, G. Sapiro, and V. Caselles, "Direction diffusion," in *Proc. ICCV*, 1999, pp. 1245–1252.

[21] A. I. Bobenko and B. A. Springborn, "A discrete Laplace–Beltrami operator for simplicial surfaces," *Discrete Comput. Geom.*, vol. 38, no. 4, pp. 740–756, Dec. 2007.

[22] R. E. Bank and M. Holst, "A new paradigm for parallel adaptive meshing algorithms," *SIAM Rev.*, vol. 45, no. 2, pp. 291–323, 2003.

[23] V. Cristini, J. Blawzdziewicz, and M. Loewenberg, "An adaptive mesh algorithm for evolving surfaces: Simulation of drop breakup and coalescence," *J. Comput. Phys.*, vol. 168, no. 2, pp. 445–463, Apr. 2001.

[24] F. R. K. Chung, *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92)*. Providence, RI: Amer. Math. Soc., Feb. 1997.

[25] V. Sindhwani, P. Niyogi, and M. Belkin, "Beyond the point cloud: From transductive to semi-supervised learning," in *Proc. 22nd ICML*, L. D. Raedt and S. Wrobel, Eds., 2005, pp. 824–831.

[26] X. Zhu, "Semi-supervised learning with graphs," Ph.D. dissertation, Carnegie Mellon Univ., Pittsburgh, PA, 2005. CMU-LTI-05-192.

[27] G. R. G. Lanckriet, N. Cristianini, P. L. Bartlett, L. E. Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *J. Mach. Learn. Res.*, vol. 5, pp. 27–72, Dec. 2004.

[28] D. Zhou, J. Huang, and B. Schölkopf, "Learning from labeled and unlabeled data on a directed graph," in *Proc. 22nd ICML*, L. D. Raedt and S. Wrobel, Eds., 2005, pp. 1036–1043.

[29] D. Zhou, B. Schölkopf, and T. Hofmann, "Semi-supervised learning on directed graphs," in *Proc. NIPS*, L. K. Saul, Y. Weiss, and L. Bottou, Eds., 2004, vol. 17, pp. 1633–1640.

[30] R. Collobert, F. H. Sinz, J. Weston, and L. Bottou, "Large scale transductive SVMs," *J. Mach. Learn. Res.*, vol. 7, pp. 1687–1712, Dec. 2006.

[31] T. Joachims, "Making large-scale SVM learning practical," in *Advances in Kernel Methods: Support Vector Learning*, B. Schölkopf, C. Burges, and A. Smola, Eds. Cambridge, MA: MIT Press, 1999, ch. 11, pp. 169–184.

[32] E. Levina and P. J. Bickel, "Maximum likelihood estimation of intrinsic dimension," in *Proc. NIPS*, S. Thrun, L. K. Saul, and B. Schölkopf, Eds., 2004, vol. 16, pp. 777–784.

[33] L. K. Saul and S. T. Roweis, "Think globally, fit locally: Unsupervised learning of low dimensional manifolds," *J. Mach. Learn. Res.*, vol. 4, pp. 119–155, Jun. 2003.

[34] C. M. Bishop, *Neural Networks for Pattern Recognition*. London, U.K.: Oxford Univ. Press, 1995.

[35] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, pp. 2399–2434, Dec. 2006.

[36] T. Evgeniou, M. Pontil, and T. Poggio, "Regularization networks and support vector machines," *Adv. Comput. Math.*, vol. 13, no. 1, pp. 1–50, Apr. 2000.

[37] S. Hettich, C. Blake, and C. Merz, *UCI Repository of Machine Learning Databases*, 1998. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html

[38] Y. Zhao and S. A. Billings, "Neighborhood detection using mutual information for the identification of cellular automata," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 2, pp. 473–479, Apr. 2006.

[39] D.-Y. Yeung and H. Chang, "A kernel approach for semisupervised metric learning," *IEEE Trans. Neural Netw.*, vol. 18, no. 1, pp. 141–149, Jan. 2007.

Haixuan Yang received the B.S. degree in mathematics from Lanzhou University, Lanzhou, China, in 1990, the M.S. degree in mathematics from Qufu Normal University, Qufu, China, in 1993, the Ph.D. degree in mathematics from Lanzhou University in 1996, and the Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong, Shatin, N.T., in 2007.

From 1998 to 2003, he was an Associate Professor with Tianjin University of Commerce, Tianjin, China. He is currently a Postdoctoral Fellow with the Department of Computer Science and Engineering, The Chinese University of Hong Kong. His research interests include machine learning, information retrieval, and information systems.

Dr. Yang served as a Program Cochair with the Second Beijing Hong Kong International Doctoral Forum in 2006.

Michael R. Lyu (S'84–M'88–SM'97–F'04) received the B.S. degree in electrical engineering from the National Taiwan University, Taipei, Taiwan, in 1981, the M.S. degree in computer engineering from the University of California, Santa Barbara, in 1985, and the Ph.D. degree in computer science from the University of California at Los Angeles, Los Angeles, in 1988.

From 1988 to 1990, he was a Technical Staff Member with the Jet Propulsion Laboratory. From 1990 to 1992, he was an Assistant Professor with the Department of Electrical and Computer Engineering, The University of Iowa, Iowa City. From 1992 to 1995, he was a Member of the Technical Staff in the applied research area of Bell Communications Research (Bellcore), Morristown, NJ. From 1995 to 1997, he was a Research Member of the technical staff with Bell Laboratories, Murray Hill, NJ. He is currently a Professor with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, N.T. He is also the Director of the Video over InternEt and Wireless (VIEW) Technologies Laboratory. He has published over 300 refereed journal and conference papers in these areas. He has participated in more than 30 industrial projects and helped to develop many commercial systems and software tools. He was the Editor of two book volumes: *Software Fault Tolerance* (New York: Wiley, 1995) and *The Handbook of Software Reliability Engineering* (Piscataway, NJ: IEEE, and New York: McGraw-Hill, 1996). His research interests include software-reliability engineering, distributed systems, fault-tolerant computing, mobile networks, Web technologies, multimedia information processing, and E-commerce systems.

Dr. Lyu initiated the First International Symposium on Software Reliability Engineering (ISSRE) in 1990. He was the Program Chair for ISSRE96 and General Chair for ISSRE2001. He was also PRDC99 Program Cochair, WWW10 Program Cochair, SRDS2005 Program Cochair, and PRDC2005 General Cocair. He has served in program committees for many other conferences including HASE, ICECCS, ISIT, FTCS, DSN, ICDSN, EUROMICRO, APSEC, PRDC, PSAM, ICCCN, ISESE, and WI. He has been frequently invited as a keynote or tutorial speaker to conferences and workshops in U.S., Europe, and Asia. He served on the Editorial Board of IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING and has been an Associate Editor of the IEEE TRANSACTIONS ON RELIABILITY and *Journal of Information Science and Engineering*. He is a Fellow of IEEE and a Fellow of AAAS for his contribution to software-reliability engineering and software fault tolerance. He was the recipient of the Best Paper Awards in ISSRE98 and ISSRE2003.

Irwin King (S'91–M'93) received the B.Sc. degree in engineering and applied science from California Institute of Technology, Pasadena, in 1984 and the M.S. and Ph.D. degrees in computer science from the University of Southern California, Los Angeles, in 1988 and 1993, respectively.

Since 1993, he has been with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, N.T. His research interests include machine learning, multimedia processing, and web intelligence.

Dr. King is a member of the Association for Computing Machinery (ACM), IEEE Computer Society, International Neural Network Society, and Asian Pacific Neural Network Assembly (APNNA). He is an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS (TNN). He is a member of the Editorial Board of the *Open Information Systems Journal*, *Journal of Nonlinear Analysis and Applied Mathematics*, and *Neural Information Processing—Letters and Reviews Journal (NIP-LR)*. He has also served as Special Issue Guest Editor for *Neurocomputing* and *Journal of Computational Intelligent Research*. Currently, he is serving the Neural Network Technical Committee and the Data Mining Technical Committee under the IEEE Computational Intelligence Society (formerly, the IEEE Neural Network Society). He is also a governing board member of the APNNA. He is a founding member of the Neural Computing and Engineering Laboratory and the Multimedia Information Processing Laboratory (MIP Lab). He has served as program and/or organizing member in international conferences and workshops, e.g., WWW, ACM MM, ICME, ICASSP, IJCNN, ICONIP, ICPR, etc. He has also served as Reviewer for international conferences as well as journals, e.g., *Information Fusion*, IEEE TCAS, SIGMOD, IEEE PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON SYSTEM, MAN, AND CYBERNETICS, *Machine Vision and Applications*, *International Journal of Computer Vision*, *Real-Time Imaging*, *SPIE Journal of Electronic Imaging*, *International Journal of Pattern Recognition and Artificial Intelligence*, etc.