An Integrated Cluster Detection, Optimization, and Interpretation Approach for Financial Data

Tie Li, Gang Kou, Yi Peng, and Philip S. Yu, *Life Fellow, IEEE*

***Abstract*—In many financial applications, such as fraud detection, reject inference, and credit evaluation, detecting clusters automatically is critical because it helps to understand the subpatterns of the data that can be used to infer user's behaviors and identify potential risks. Due to the complexity of human behaviors and changing social environments, the distributions of financial data are usually complex and it is challenging to find clusters and give reasonable interpretations. The goal of this study is to develop an integrated approach to detect clusters in financial data, and optimize the scope of the clusters such that the clusters can be easily interpreted. Specifically, we first proposed a new cluster quality evaluation criterion, which is free from large-scale computation and can guide base clustering algorithms such as *k*-Means to detect hyperellipsoidal clusters adaptively. Then, we designed a new solver for a revised support vector data description model, which efficiently refines the centroids and scopes of the detected clusters to make the clusters tighter such that the data in the clusters share greater similarities, and thus, the clusters can be easily interpreted with eigenvectors. Using ten financial datasets, the experiments showed that the proposed algorithm can efficiently find reasonable number of clusters. The proposed approach is suitable for large-scale financial datasets whose features are meaningful, and also applicable to financial mining tasks, such as data distribution interpretation and anomaly detection.**

***Index Terms*—Clustering methods, data mining, financial management, spectral analysis.**

I. Introduction

IN MANY financial applications, such as credit evaluation, fraud detection, and reject inference, labeled data of ground truth are highly scarce. Hence, unsupervised models are used intensively to infer the patterns behind the data. Compared to supervised learning, unsupervised learning is more challenging because of lacking objective criteria to guide the learning process and evaluate the results. Since most industrial data are unsupervised, unsupervised models, such as frequent itemset mining and clustering, all play important roles in certain domains [1]. This article focuses on clustering for two reasons: first, one purpose is to analyze the unknown subpatterns that usually represent users' changing behaviors and potential risks in financial data. Clustering can detect groups of similar data points aggregating in local areas, and thus, it is an appropriate unsupervised tool for this purpose. Second, this study concerns about financial data, which are normally numerical or tabular, and clustering has been widely used to deal with such data types. This work is motivated by the following observations.

1) It is difficult for most clustering algorithms to determine the number of clusters in advance due to the complicated distributions of financial data. Besides, correlations among features are ubiquitous and result in various shapes of data distributions. Many clustering algorithms cannot adapt to these varying shapes. For example, *k*-Means can only separate space with hyperspheres, which are not suitable for rotated strip-shaped distributions derived from linear correlations [2].
2) The interpretability of a model is crucial in financial applications. For instance, in credit evaluation, the data mining results have important impacts on customers; thus, they must be interpretable to customers and managers. Although there exist many cluster interpretation approaches [3], few focus on the interpretation of financial data.
3) The size of financial data is normally huge and the calculation speed is critical. Although some clustering techniques are theoretically powerful, they are slow and infeasible in financial applications. For instance, spectral clustering involves the eigen decomposition of the Laplacian matrix, and it is very difficult to perform large-scale matrix decomposition in practice.

Based on the above observations, we need an integrated clustering approach to address the aforementioned issues derived from complex distributions and large volumes of financial data. While traditional clustering algorithms pursue overall space separation, clustering analysis in this study tries to capture the landmark characteristics of the data distribution, which is important for financial applications, such as new pattern analysis, reject inference, and fraud detection. This work also considers the computation time and memory cost such that the models can apply to large-scale financial datasets.

The novel contributions of this work are twofold. First, we proposed a new criterion to evaluate the cluster quality

Manuscript received 11 June 2021; accepted 20 August 2021. Date of publication 22 September 2021; date of current version 18 November 2022. This work was supported in part by the Ministry of Education Project of Humanities and Social Science under Grant 20YJC630064; in part by the China Postdoctoral Science Foundation under Grant 2019M653388; and in part by the National Natural Science Foundation of China under Grant U1811462, Grant 71771037, Grant 71725001, and Grant 71971042. This article was recommended by Associate Editor Z. Xu. *(Corresponding author: Yi Peng.)*

Tie Li and Yi Peng are with the School of Management and Economics, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: lteb2002@163.com; pengyi@uestc.edu.cn).

Gang Kou is with the School of Business Administration, Southwestern University of Finance and Economics, Chengdu 610074, China (e-mail: kougang@swufe.edu.cn).

Philip S. Yu is with the Department of Computer Science, University of Illinois at Chicago, Chicago, IL 60607 USA (e-mail: psyu@uic.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2021.3109066.

Digital Object Identifier 10.1109/TCYB.2021.3109066

based on the spectral graph theory, which guides base clustering algorithms to detect hyperellipsoidal clusters iteratively. Compared to traditional methods, the criterion is free from large-scale computation, and the automatic clustering framework is well suited to the analysis of large-scale financial data with complex distributions. Second, we developed a new penalty-function-based solver for a revised support vector data description (SVDD) model [4], which conducts the optimization in the Euclidean space, such that SVDD can be used to optimize the centroids and the scopes of the detected clusters more smoothly and efficiently.

The remainder of this article is organized as follows. Section II reviews the related works. Section III analyzes the characteristics and major types of financial data distributions. Section IV describes the proposed approach. Section V presents an experimental study using ten financial datasets. Finally, Section VI concludes the article.

II. Related Works

This section reviewed the related works in correlation analysis, adaptive clustering, and cluster interpretation.

A. Disentanglement of Features

A hot topic recently is to detect clusters of varying shapes. Irregularly shaped clusters are usually caused by the interactions between features [5]. Disentanglement learning focuses on analyzing feature interactions [6]. The efforts to disentangle the interactions can be divided into two categories.

1) *Nonlinear Models:* Nowadays, the most state-of-the-art approaches for nonlinear disentanglement learning are based on a particular type of deep neural networks, that is, variational autoencoders (VAE) [7]. The key idea of VAE is that the high-dimensional entangled features can be explained by the lower dimensional and simply distributed latent variables. However, this notion has been challenged recently. Locatello *et al.* [8] proved that features of unknown and complex distributions are very difficult to be transformed to latent variables of simple distributions in an unsupervised manner.
2) *Linear Models:* Different from the difficulties of nonlinear disentanglement of features, linear disentanglement analysis is straightforward. Many well-known tools, such as singular value decomposition (SVD) [9] and principal component analysis (PCA) [10], can be used for this purpose. SVD-based methods eliminate the linear entanglement by transforming the original dataset with left eigenmatrix, and keep all features orthogonal [11]. PCA eliminates the entanglement through a transformation with the eigenmatrix of the inverse of the covariance matrix, and keeps the correlation coefficients zero [12]. Besides, Sim *et al.* [13] proposed to find real-world profitable stocks through clustering in a transformed subspace. Jung and Chang [14] used partial correlation to study the market structure and conduct clustering in the disentangled space.

One implicit assumption of correlation analysis is that data have a uniform pattern, which is not true because of complicated distributions [15]. Therefore, it is necessary to develop models that can handle heterogeneous data distributions.

B. Automatic Clustering Algorithms

Many works suggested that practical datasets follow multimodal multivariate distributions [16]. In this case, there may be several distinct peaks in the probability density function of every single variable (feature). Such multimodal distributions generate clusters located in several areas of the Euclidean space. Consequently, the analysis of social data distributions relies on clustering algorithms to conduct spatial division [17]. In financial data, the cluster distribution could be very complex due to the noncooperative behaviors of game partners [18].

The most frequently used clustering algorithm is k-Means and its variants, which group samples into k hyperspherical scopes. For instance, Liu *et al.* [19] suggested that in social stream data distributions may differ as time passes, and thus, proposed a heuristic space partitioning method based on k-Means to improve the concept drift detection. One limitation of k-Means is that hyperspheres are only appropriate when the variance of each feature is identical and no linear correlation exists [20]. In many cases, hyperellipsoids are more suitable than hyperspheres because correlations and varying variances of features result in rotated hyperellipsoidal [21]. A frequently used model that overcomes the limitation of k-Means is the Gaussian mixed model (GMM), which is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters [22]. The main merit of GMM is that they generate clusters of hyperellipsoids and take the linear correlations into consideration [22]. That is why this article uses hyperellipsoids rather than hyperspheres.

A common difficulty in GMM and k-Means is that it is hard to determine the appropriate number of clusters in advance. Many efforts have been made to solve this problem. The naivest approach is to evaluate the clustering results iteratively with different cluster numbers using evaluation criteria, such as the Akaike information criterion (AIC) [23], Bayesian information criterion (BIC) [24], and Silhouette score [25], and then choose the best performed number of clusters. AIC and BIC usually rely on an elbow test, that is, choose the cluster number at the biggest inflection of the score curve [26]. However, in some cases, the score curve is smooth and it is unclear which cluster number is the best [27]. The silhouette test is easier to use because the principle is to choose the cluster number with the highest score [25]. The limitation of the Silhouette test is that its time complexity is high, and sometimes inclined to high clustering resolution and lose distribution details [28]. Another well-known theory is spectral analysis, which builds a graph with data points as nodes and similarities between points as edges, and then studies the eigenvalues of the normalized graph Laplacian matrix [29]. The graph cut theory suggests that small eigenvalues of the Laplacian matrix reflect weak connections of components, and a large eigen gap indicates a proper cluster number [30].

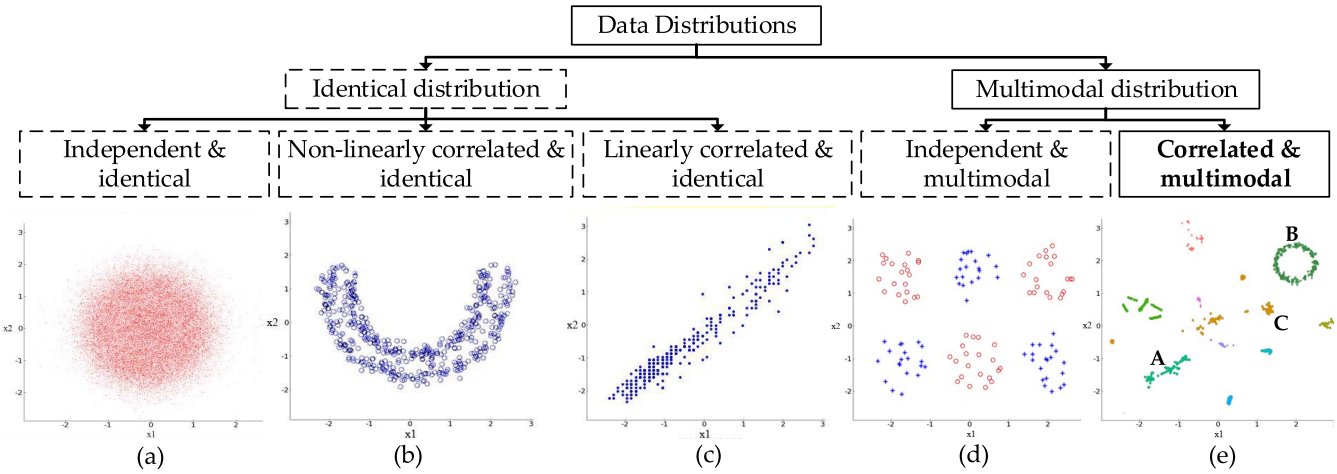

Fig. 1. Typical types of data distributions. (a) Independent and identical distribution. (b) Non-linearly correlated and identical distribution. (c) Linearly correlated and identical distribution. (d) Independent and multimodal distribution. (e) Correlated and multimodal distribution.

The advantages of spectral clustering include the detection of various cluster shapes, utilization of kernel functions, and cluster number estimation [29]. Chen *et al.* [17] proposed a two-Level subspace weighting method and then conducted spectral clustering to group customers from the commercial transaction data. Türkmen *et al.* [31] proposed a method to recover the accurate representation of the underlying market graph structure, and then conducted automatic spectral clustering toward the recovered data. The main limitation of spectral clustering is that the graph Laplacian matrix's dimension is the same as the sample number, which makes it hard to apply to large-scale datasets [29].

Besides these well-known canonical methods above, Song *et al.* [32] used correlation-based agglomerative hierarchical clustering to analyze the structure of the cryptocurrency market. Kingrani *et al.* [33] suggested that the difference between the global diversity of clusters and the sum of each cluster's local diversity can be used as an effective indicator of the optimality of the number of clusters. Guo *et al.* [34] proposed an approach to achieve low-rank clusters through a unified scheme for distance metric learning and clustering, and the clusters can be recognized easily in the low-dimensional embedding.

However, these cluster number estimation methods, which were based on connected components, density, hierarchy, diversity, or low-rank structure, were not designed for space division methods used in this article (such as GMM and k-Means). There is no standard solution to this problem yet.

C. Cluster Interpretation

One explanatory clustering method is to keep clusters tight and use instances or centroids to interpret clusters. Davidson *et al.* [35] proposed a method to find a compact and distinct explanation of each cluster using instance-level descriptors from a common dictionary. Tightness of clusters indeed makes great sense because it implies robust underlying patterns, which are critical to data analysis and interpretation. However, the tightness of clusters is difficult to guarantee and thus, centroids are not necessarily representative.

Other methods suggest that density within clusters, noises, and outliers can be used to interpret clusters. Sakai *et al.* [36] argued that the density-based adaptive spatial clustering algorithm extracts areas which mean local topics and, thus, it can be used as geo-tagged documents topics explanation. Some researchers suggested that outliers can be expressed as vital contextual information, which improves the interpretability of local distribution [37]. Davidson *et al.* [38] proposed that noises can also be used to interpret the clusters and enhance clustering stability.

Although understanding the subpatterns of financial data is important, studies dedicated toward the integrated cluster detection, optimization, and interpretation are rare. The clustering analysis can provide important information, such as data density, outliers, numeric characteristics, and number of potential modals. Therefore, this work focuses on the automatic cluster detection and the interpretation of the results.

III. Problem Formulation

A. Characteristics of Financial Data

Distributions of financial data are inherently complex, due to the following reasons.

1) Financial data are social data, which are usually dominated by multiple complicated latent factors, and these factors can be easily influenced by external changeful social environments, and even evolving over time.
2) In fraud-related financial data mining tasks, criminals take countermeasures based on the anti-fraud actions of financial institutions. Such a game alike phenomenon has been frequently observed in financial projects [18].
3) Latent factors of financial data are usually correlated and even autocorrelative, that is, these factors influence each other mutually and result in complex patterns.

B. Typical Data Distributions of Financial Data

Fig. 1 illustrates typical data distributions using 2-D synthetic datasets.

1) Identical Distribution: In this category, all samples follow the same multivariate distribution. According to the

interactions between features, this category can be further divided into independent and identical distribution, nonlinearly correlated and identical distribution, and linearly correlated and identical distribution. As shown in Fig. 1(a)–(c), data points in each figure follow an identical distribution, but their correlation types vary.

2) Multimodal Distribution: In this category, multimodality indicates that the samples are not homogenous and they come from several different distributions. Again, according to the interactions between features, this distribution can be further divided into independent and multimodal distribution, and correlated and multimodal distribution. As shown in Fig. 1(d) and (e), datasets in this category have multimodals, that is, several clusters. In Fig. 1(d), each cluster follows a Gaussian distribution. In Fig. 1(e), samples in different clusters have different correlation types. Take clusters A, B, and C in Fig. 1(e) for examples. Samples in A are linearly correlated, samples in B are nonlinearly distributed, while samples in C follow the Gaussian distribution. Different linear and nonlinear correlations may exist simultaneously among different parts of a single dataset. In practice, correlated and multimodal distribution is common among financial data.

Due to space limitation, this work focuses on linearly correlated types. If the features are nonlinearly correlated or autocorrelated, they should be preprocessed with disentangled representation or differencing techniques [39].

Based on the above analysis, the proposed approach is intended to deal with the following issues.

1) Due to the unknown distributions of data, we need automatic techniques to detect clusters representing subpatterns without losing the details of the distribution.
2) Since correlations are ubiquitous, we also need a mechanism to handle linear correlations automatically.
3) Due to the sensibility of financial applications, we need an interpretability approach to convince financial managers and customers of the detection results.

IV. Proposed Approach for Automatic Cluster Detection, Optimization and Interpretation

This section introduces the ideas, theoretical background, and technical details of the proposed approach. The entire research framework is outlined in Fig. 2.

A. Theretical Background

We first introduce a kernel matrix and explain why it is used to analyze cluster distribution. A kernel matrix is defined as

$$K = \begin{bmatrix} \kappa_{11} & \kappa_{12} & \dots & \kappa_{1m} \\ \kappa_{21} & \kappa_{22} & \dots & \kappa_{2m} \\ \dots & \dots & \dots & \dots \\ \kappa_{m1} & \kappa_{m2} & \dots & \kappa_{mm} \end{bmatrix} \tag{1}$$

where κ is the kernel function of pairwise data points. It is used to measure the similarities between pairwise data points, and is defined as

$$\kappa(x_i, x_j) = \phi(x_i)^T \cdot \phi(x_j) \tag{2}$$

where x_i and x_j are two pairwise data points, and ϕ is a transformation function that projects the data points into a

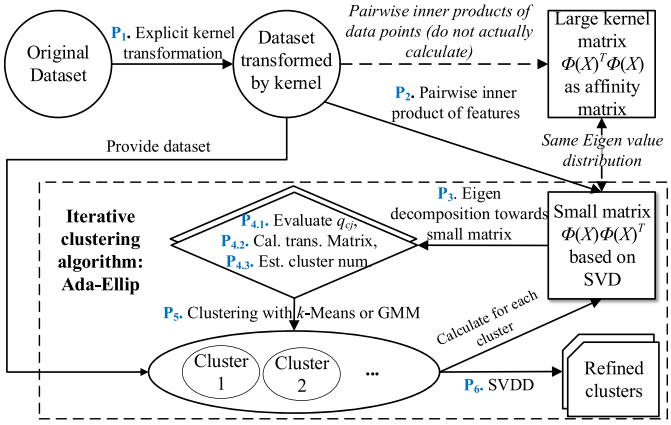

Fig. 2. Research framework of the adaptive clustering.

reproducing kernel Hilbert space [40]. In most kernel functions, transformation with ϕ is implicit, that is, we can calculate $\kappa(x_i, x_j)$ with data points directly, but an explicit form of ϕ cannot be given [40]. However, in recent years, the random fourier (RF) transform, Nyström method, and random kitchen sinks (RKS) have been proposed to approximate an explicit transformation of ϕ [41]. The subsequent sections depend on such an explicit kernel transformation, that is, ϕ, and it is marked as **P1** in Fig. 2.

In the last two decades, the kernel matrix has been utilized intensively by spectral clustering, which uses the kernel matrix to represent a weighted adjacency matrix [42]. The spectral graph theory relates the eigenvalues of the adjacency matrix or Laplacian matrix, which is the minus of the degree matrix, to structural properties of graphs [29]. A well-known property is that the number of eigenvalue 0 of the Laplacian matrix is equal to the number of connected components, and the corresponding eigenvectors are the indicator vectors of the connected components [42]. Then, the trick of spectral clustering is that the connected components can be viewed as clusters, and we can perform clustering on the eigenvectors of the Laplacian matrix corresponding to the 0 or small eigenvalues with a base clustering algorithm such as k-Means. Besides, the position of the biggest eigen gap can be used to infer the possible cluster number when the eigenvalues are ordered increasingly [29]. Although this approach has a strong theoretical background, it is usually found impractical in large-scale financial applications. Because the dimension of the square Laplacian matrix is n, which is the sample size, and the eigen decomposition toward it is infeasible when the scales of the datasets are large, as discussed previously.

B. Criterion for Cluster Quality Evaluation

Due to the above limitation, we utilize SVD to avoid the computation of the eigen decomposition toward the kernel matrix, which represents the adjacency matrix, and propose a criterion to evaluate the quality of a cluster.

Suppose that there is an explicit kernel transform $\phi(x_i)$, Φ is a kernel transformation toward the entire dataset X, where each column of X represents a data point x_i, and each column of $\Phi(X)$ represents a kernel transformation of x_i, that is, $\phi(x_i)$.

Now, the kernel matrix can be denoted in such a form

$$K = \Phi(X)^T \cdot \Phi(X) \tag{3}$$

whose dimension is $n \times n$, and n is the sample size.

Solving the eigenvalues of $\Phi(X)^T \cdot \Phi(X)$ involves large-scale computation. But there exists a well-known property of SVD that can facilitate the computation. Suppose an SVD toward $\Phi(X)$ is

$$\Phi(X)_{m' \times n} = U_{m' \times m'} \Sigma_{m' \times n} V_{n \times n} \tag{4}$$

where m' is the dimension of a transformed data point $\phi(x_i)$. $U_{m' \times m'}$ contains the eigenvectors of $\Phi(X) \cdot \Phi(X)^T$, $V_{n \times n}$ contains the eigenvectors of $\Phi(X)^T \cdot \Phi(X)$, and $\Sigma_{m' \times n}$ is a diagonal matrix containing the square root of the eigenvalues of both $\Phi(X) \cdot \Phi(X)^T$ and $\Phi(X)^T \cdot \Phi(X)$. $\Phi(X) \cdot \Phi(X)^T$ and $\Phi(X)^T \cdot \Phi(X)$ have the same nonzero eigenvalues. Since the dimension of the matrix $\Phi(X) \cdot \Phi(X)^T$, which is $m' \times m'$, is significantly smaller than the dimension of the matrix $\Phi(X)^T \cdot \Phi(X)$, which is $n \times n$, in practice, we can obtain the eigenvalue of $\Phi(X)^T \cdot \Phi(X)$, that is, K , by performing a much faster and lightweight eigen decomposition toward $\Phi(X) \cdot \Phi(X)^T$. This trick makes the spectral analysis on the large-scale kernel matrix feasible. The above idea is outlined in Fig. 2, the calculation of $\Phi(X)^T \cdot \Phi(X)$ is marked as **P2**, and the eigen decomposition of $\Phi(X) \cdot \Phi(X)^T$ is marked as **P3**.

We denote the singular value of $\Phi(X)$ as s_i, and thus, the eigenvalue of K is s_i^2. With the eigenvalues of the kernel matrix that represents the adjacency matrix, we can estimate the eigenvalue distribution of the Laplacian matrix and then infer the possible cluster number with the position of the biggest eigen gap [43]. Nevertheless, there are the following limitations with this approach.

1) The canonical spectral clustering uses the eigenvectors of the Laplacian matrix as the data representation [29]. However, we cannot obtain the eigenvectors with the above method.
2) It is hard to control the clustering resolution just with the position of the biggest eigen gap.
3) It is unavailable when the eigenvalue curve is smooth and there is no evident big gap between the eigenvalues.

Due to the limitations above, the eigen gap approach is not well suited to the data distribution analysis tasks proposed in Section III. Therefore, we take another way to evaluate the cluster quality and then design an adaptive clustering framework with the eigenvalues of the kernel matrix.

In the spectral graph theory, the largest eigenvalue of the adjacency matrix is called "spectral radius," and it reflects the bounded degrees of the nodes [29]. A dominantly large eigenvalue indicates that the nodes are densely connected, and the data points share great similarities mutually. Based on the analysis above, we propose the following criterion to evaluate the quality of a cluster:

$$q_{ci} = s_1^2 \Big/ \sum_{j=1}^{m'} s_j^2 \tag{5}$$

where s_j refers to the singular values of $\Phi(X_i)$ arranged in a descending order, X_i are the data points within cluster c_i, m' is the dimension of the transformed space by Φ, and $q_{ci} \in (0, 1)$. Higher q_{ci} value indicates that data points within the cluster share greater similarities measured by $\kappa(x_i, x_j)$. This criterion helps to control the clustering resolution and capture landmark characteristics of the data distribution. This process is marked as **P4.1** in Fig. 2.

As for computational complexity, eigen decomposition requires $O(m^{2.376})$ time, where m is the feature number of the dataset [44]. In practice, we can solve singular values efficiently with a power method such as the Lanczos algorithm [45].

C. Cluster Shape Analysis and Orthogonal Transformation

In an Euclidean space transformed by the explicit kernel Φ, a high q_{ci} indicates two types of possible distributions.

1) The samples squeeze in a small area, which means that the samples are determined by some dominant factors.
2) The samples distribute in a large area, but they are linearly correlated. In this case, the samples usually distribute along a hyperplane and are low rank [46].

In the first case, the samples can be clustered with a hypersphere. In the second case, the linear correlations usually mean that the features are not well presented, which may be inevitable in practice.

Since a high q_{ci} value is not capable of determining whether there are linear correlations or the degree of aggregation of the data is high, it should work with clustering algorithms to generate hyperellipsoidal clusters, or eliminate linear correlations before applying clustering algorithms using hyperspheres. Without loss of generality, we use a matrix deduced from (4) to revise the space. Let $P_i = \Sigma_i^{-1} U_i^T$, we have

$$P_i \cdot \Phi(X_i) = \Sigma_i^{-1} U_i^T \Phi(X_i) = V_i^T. \tag{6}$$

Due to V_i^T's orthogonality, now we can conduct a linear transformation toward $\Phi(X_i)$

$$X_i \to P_i \cdot \Phi(X_i) \tag{7}$$

such that the data features are kept orthogonal and the data points locate in a hyperspherical area of unit radius. The transformation with (7), which is marked as **P4.2** in Fig. 2, can be omitted when the base clustering algorithm uses hyperellipsoids, such as GMM.

D. Iterative Clustering Algorithm: Ada-Ellip

Since a larger q_{ci} indicates that the data points aggregate better in the hyperellipsoidal scope, we can set a threshold T_q and split the cluster into k_s if $q_{ci} < T_q$. The value of k_s can be 2 or estimated in the following way:

$$k_s = \arg\min_k \left(\sum_{i=1}^{k} s_i^2 \Big/ \sum_{i=1}^{m'} s_i^2 > T_q \right), \quad 0 < T_q < 1. \tag{8}$$

The above solution is marked as **P4.3** in Fig. 2. Next, like spectral clustering, we perform clustering using base algorithms such as GMM with k_s as the preset cluster number. The clustering process carries on with the q_{ci} evaluation recursively until all q_{ci} values of clusters satisfy the preset condition:

Algorithm 1 Ada-Ellip

Input:

X: A normalized or standardized dataset with columns representing data points.

T_q: A threshold of q_{cj} to check if the data points in cluster aggregate well.

BCA: Base clustering algorithm, the feasible value is 'GMM' or 'k-Means'.

T_{noise}: A threshold to resist noises, default value: 0.05;

Output:

$\{CL_i\}$: a list of k cluster structs CL_i, which has fields of c_i as a centroid in the transformed space, X_i as sub-dataset allocated to it, P_i as a linear projection matrix, and q_{ci} value of $\Phi(X_i)$, where $i = 1, \ldots, k$.

Method:

1: Calculate P_i if $BCA != 'GMM'$, otherwise $P = I$;
2: Calculate eigenvalues s_i^2 of $P_i\Phi(X) \cdot (P_i\Phi(X))^T$;
3: Calculate k_s with Eq. (8);
4: Conduct clustering with *BCM* on $P_i\Phi(X)$ to find k_s centroids: $\{c_i\}$; //marked as **P5** in Fig.2.
5: Allocate each data points in $P_i\Phi(X)$ to the nearest c_i as $P_i\Phi(X_i)$;
6: **for** c_i in $\{c_i\}$ **do** // Parallel loop is recommended.
7: Calculate q_{ci} with dataset $P_i\Phi(X_i)$ in c_i;
8: **if** $q_{ci} > T_q$ **then**
9: Add $CL_i(c_i, X_i, P_i, q_{ci})$ to $\{CL_i\}$;
10: **else**
11: Go to step (1) with X_i as the inputted dataset;
//The following codes are resisting noises.
12: Sort $\{CL_i\}$ based on their inner data points X_i's size in descending order, and calculate $k = \arg\min_t(\sum_{i=1}^{t} size(CL_i.X_i)/\sum_{i=1}^{total} size(CL_i.X_i) > 1 - T_{noise})$; //only use dense clusters covering $(1 - T_{noise})$ proportion of data points in total.
13: Allocate each data points in $\{CL_{k+1}, \ldots, CL_{total}\}$ to CL_i, whose centroid c_i is the nearest to the data point, and $CL_i \in \{CL_1, \ldots, CL_k\}$;
14: **return** $\{CL_1, \ldots, CL_k\}$;

$q_{ci} < T_q$. The idea above is outlined in algorithm Ada-Ellip, which highlights its characteristic of adaptive hyperellipsoids.

Intrinsically, Ada-Ellip is a framework to add adaptivity to base clustering algorithms such that automatic clustering can be achieved. Besides, it is also notable that we have a noise resisting mechanism in the algorithm.

E. Cluster Optimization Based on Revised SVDD

Ada-Ellip only generates rough scopes of clusters, and the borders of clusters are ambiguous and easy to be influenced by outliers near the borders. As suggested by [35], keeping a cluster tighter makes the data inside share greater similarities and the cluster easier to be interpreted.

Given the data points in a hyperellipsoidal cluster, the optimal centroid and scope of the cluster can be solved with the following optimization model, which is a revised version of SVDD [4]:

$$\begin{aligned} &\min \ R^2 + C\sum_{i=1}^{n_i}\xi_j \\ &\text{s.t.} \ \left\|P_i\big(\phi(x_j)\big) - \alpha\right\|^2 \le R^2 + \xi_j, \quad \xi_j \ge 0 \end{aligned} \tag{9}$$

where x_j is the data points in cluster c_i, α is the centroid, P_i is a projection matrix (it can be solved with (6) using the data points inside the cluster) to overcome linear correlation in local area, R is the radius of the cluster c_i in the transformed space, ξ_j are slack variables, and C is a punishment parameter. The main difference between the model defined in (9) and the original SVDD is that (9) learns a hyperellipsoidal scope rather than a hyperspherical one. Thus, we name it Ellip-SVDD. It can result in tighter scopes for the clusters learned by Ada-Ellip. This process is marked as **P6** in Fig. 2.

Original SVDD is solved by maximizing the dual Lagrange problem, that is, utilizing Karush–Kuhn–Tucker (KKT) conditions [4]. This solving approach was used by the original version partially due to the pursuit of "inner product" forms of data points such that *implicit* kernel functions can be adopted smoothly. In contrast, this study uses an *explicit* kernel to transform the datasets into another Euclidean space, and thus, the original solver of SVDD cannot be used for this study. In addition, the time complexity of the pairwise "inner product" computation is $O(n^2)$, which is not applicable for large-scale financial datasets.

Thus, we propose a new solver for SVDD based on the penalty function method. We first rewrite the objective function and constraints as follows:

$$\begin{aligned} &\min \ f_c(\alpha, R, \xi) = R^2 + C\sum_{j=1}^{n_i}\xi_j^2 \\ &\text{s.t.} \ h_j(\alpha, R, \xi) = \left\|P_i\big(\phi(x_j)\big) - \alpha\right\|^2 - R^2 - \xi_j^2 \le 0. \end{aligned} \tag{10}$$

To transform (10) into an unconstrained problem, we formulate the penalty function of the constraint function as

$$\theta(\alpha, R, \xi) = \sum_{j=1}^{n_i}\max\big\{h_j(\alpha, R, \xi), 0\big\}. \tag{11}$$

The principle of the penalty function is that when a constraint h_j is violated, the function will give a punishment. The transformed unconstrained problem, which is formulated by adding the smooth penalty function term to its objective function, is as follows:

$$L_{C,\beta}(\alpha, R, \xi) = f_c(\alpha, R, \xi) + \beta\theta(\alpha, R, \xi) \tag{12}$$

where β is a penalty coefficient. During the optimization process, we need to increase β iteratively by: $\beta(T+1) = \delta\beta(T)$, where $\delta > 1$. As β increases, the problem converges to the optima. Now, $L_{C,\beta}(\alpha, R, \xi)$ is a quadratic function and strictly convex. The partial derivatives of α, R, and ξ_i are

$$\frac{\partial L}{\partial \alpha} = -2\beta\sum_{j=1}^{n_i}\begin{cases} P_i\phi(x_j) - \alpha, & h_j(\alpha, R, \xi) > 0 \\ 0, & h_j(\alpha, R, \xi) \le 0 \end{cases} \tag{13}$$

$$\frac{\partial L}{\partial R} = 2R - 2\beta\sum_{j=1}^{n_i}\begin{cases} R, & h_j(\alpha, R, \xi) > 0 \\ 0, & h_j(\alpha, R, \xi) \le 0 \end{cases} \tag{14}$$

$$\frac{\partial L}{\partial \xi_j} = 2C\xi_j - 2\beta \begin{cases} \xi_j, & h_j(\alpha, R, \xi) > 0 \\ 0, & h_j(\alpha, R, \xi) \leq 0. \end{cases} \tag{15}$$

Thus, the gradient of $L_{C,\beta}$ is: $g_{C,\beta}(\alpha, R, \xi) = ((\partial L/\partial \alpha)^T, (\partial L/\partial R)^T, ([\partial L/\partial \xi_1], \ldots, [\partial L/\partial \xi_{n_i}])^T)^T$, and the problem can be solved with the quasi-Newton method. We can also use the centroids generated by the Ada-Ellip as the initial value of α when solving the optimization problem.

The proposed SVDD solver works in the Euclidean space and, thus, it is compatible with the theoretical framework of this study and the base clustering algorithms. Besides, it is free from kernel matrix computation and, thus, fast on large-scale financial datasets.

After solving the optimal centroid and radius of a cluster with (12) in the transformed space, we can mark the data points outside the scope of the hyperellipsoid as "outliers." Eliminating outliers in each cluster can make the cluster tighter and data points share higher similarities, which facilitates the interpretability of the clusters described in the next section.

F. Cluster Interpretation

Based on the clusters detected and refined by the previous steps, this section aims to give interpretation of the results.

1) Robust Subpattern Analysis: After the refinement with Ellip-SVDD, data points inside each cluster are very similar to each other. We can view these tight clusters as robust subpatterns, and use the centroids as the landmark data points. It is easy to attach financial meanings to these clusters and analyze financial implications behind these landmark samples and their scopes, which are expressed by the radii and the revised projection matrices $\hat{P}_i$. Such subpatterns are frequently used for distribution analysis and reject inference.

2) Each Feature's Role in a Cluster: A large q_{ci} means that there is only one large singular value in the transformed data, and the dataset can be reduced to one dimension. SVD in (4) can be rewritten as an orthogonal projection form. The projection eigen vector, which performs dimension reduction, is the one in the first column of the left eigen matrix U corresponding to the largest singular value of $\Phi(X^-_{m\times n_i})$, where $X^-_{m\times n_i}$ denotes samples in cluster c_i without outliers. Now, the values of the elements in the projection vector can represent the roles of the features transformed by ϕ

$$W_i = U_{i\_1} \tag{16}$$

where each element of vector W_i interprets the importance of the corresponding transformed feature and the direction of its influence, that is, positive or negative. Since we use an explicit kernel transformation ϕ, the transformed features are generally a combination of the original features, and it is easy to interpret the meanings. Furthermore, if we use a linear kernel, the features transformed by ϕ are the original features themselves.

3) Outliers as Potential Financial Anomalies: In a tight hyperellipsoidal cluster refined by Ellip-SVDD, many outliers can be excluded from the cluster scopes. These outliers are important because they contain anomalous financial activities hiding beneath the data, and investigating the reasons behind them may provide valuable information to financial institutions.

TABLE I
DETAILS OF THE BENCHMARK DATASETS

Dataset	Inst.	Attr.	Max/Avg. Corr.	Max/Avg. VIF
CC-GENERAL	8,590	16	0.9168/0.2180	5.8767E4/6.3767E3
FinancialDistress	3,672	83	0.9999/0.0955	5.5927E10/3.7642E9
BankruptcyPredict	6,819	95	1.0/0.0780	$+\infty/+\infty$
FinanceWellBeing	6,394	215	0.9822/0.0881	$+\infty/+\infty$
CreditCustomer	10,127	37	0.9999/0.0564	$+\infty/+\infty$
CreditCard	30,000	23	0.9515/0.1839	2.3564E2/4.2505E1
BankMarketing	45,211	48	0.8699/0.0539	1.395E10/7.0498E8
GiveMeCredit	150,000	10	0.9927/0.0965	9.1182E1/2.2068E1
ChineseBank	224,858	37	1.0/0.1053	$+\infty/+\infty$
CreditRisk	500,000	25	0.9690/0.0711	1.4194E3/1.2969E2

V. EXPERIMENTS AND EVALUATION

We implemented the proposed approach using the Julia language,[1] and the comparative algorithms were invoked from ScikitLearn.jl[2] and Smile.[3] All of the experiments were conducted on a Lenovo server of 6 CPU cores (12 threads) and 32-GB RAM.

A. Benchmark Datasets

We used ten financial benchmark datasets for the experiments. CreditCard is a dataset collected from the UCI's machine learning repository.[4] ChineseBank is a real financial dataset from a Chinese commercial bank. The others are collected from Kaggle's data repertory.[5] The three largest datasets (GiveMeCredit, ChineseBank, and CreditRisk) are also used to test the models' performance on large-scale datasets. Table I summarizes the details of the datasets. To reflect the correlations among the features, the maximal and average Pearson correlation coefficient, which indicates pairwise linear correlations, and variance inflation factor (VIF), which indicates multicollinearity, are also recorded in Table I.

As shown in Table I, there are severe linear correlations in most datasets, and handing correlations using (7) or GMM is necessary. To demonstrate the distributions of these datasets, we reduced their dimensions to 2 using t-SNE [47], and illustrated them in Fig. 3. The colors and transparent scopes are manually estimated clusters for reference only.

B. Explicit Kernel Transformation

As introduced in Section IV-A, we have a built-in explicit kernel transformation mechanism $\phi(x_i)$. The transformation is needed when the data in the original space cannot be separated well. As shown in Fig. 3, data points in CreditCard and GiveMeCredit are not clustered well. We applied an RKS [48] kernel transformation to CreditCard and an RF [49] kernel to GiveMeCredit, to explore the effects of kernel transformation. The dimensions of the two datasets in the target space were both set to 100, which are much higher than their original dimensions. After the transformation, we used t-SNE to visualize them in Fig. 4.

[1] https://julialang.org/
[2] https://scikitlearnjl.readthedocs.io
[3] https://haifengl.github.io/
[4] http://archive.ics.uci.edu/ml/index.php
[5] https://www.kaggle.com/datasets

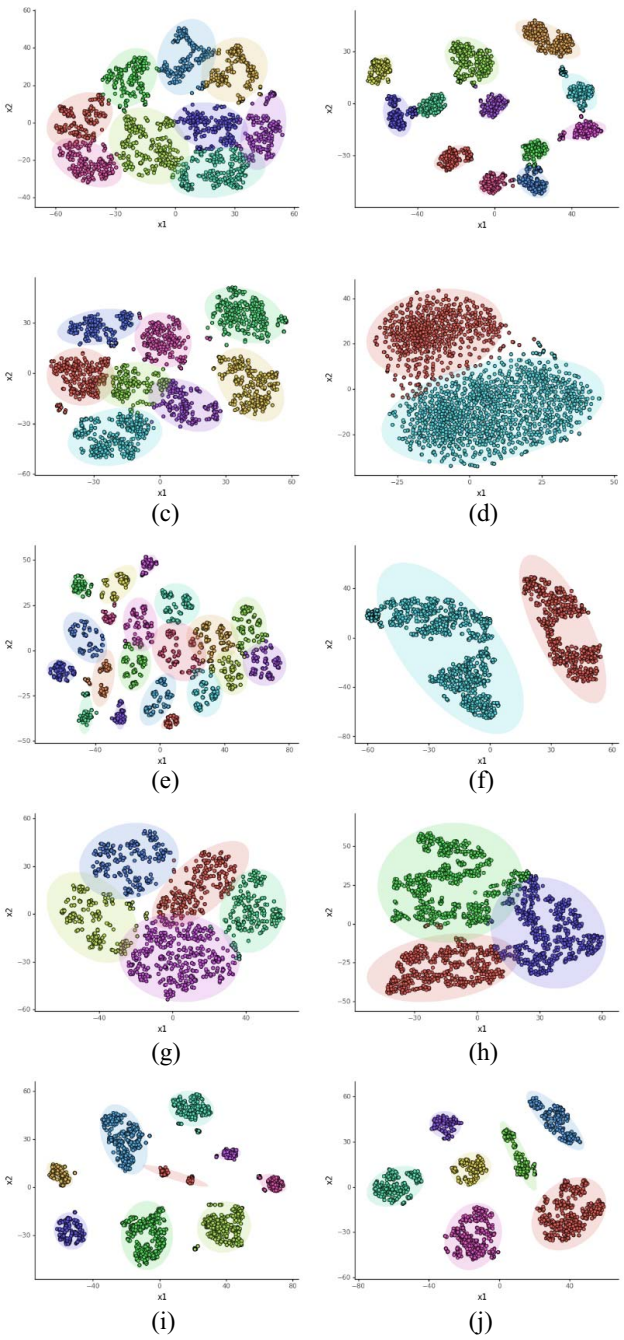

Fig. 3. Distributions of the benchmark datasets and the estimated clusters. (a) CC-GENERAL. (b) FinancialDistress. (c) BankruptcyPredict. (d) FinanceWellBeing. (e) CreditCustomer. (f) CreditCard. (g) BankMarketing. (h) GiveMeCredit. (i) ChineseBank. (j) CreditRisk.

As shown in Fig. 4, after the kernel transformation, both of the datasets can be easily clustered. Explicit kernel transformations are important techniques for clustering when the datasets are inseparable in the original space. When we do not apply any kernel function toward a dataset, it is actually a linear kernel, that is, inner products of the pairwise data points. It is a particular case of the research framework of the study.

C. Cluster Number Estimation

This section compares the cluster number estimated by our approach with other well-known methods, including the Elbow test [26] and Silhouette test [25], and then explained why the proposed approach is well suited to financial data.

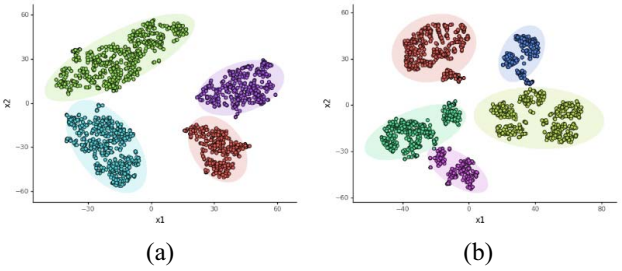

Fig. 4. Distributions of datasets transformed by explicit kernel methods. (a) CreditCard trans. by RKS. (b) GiveMeCredit trans. by RF.

TABLE II
ESTIMATED CLUSTER NUMBER AND TIME COST (SECONDS)

Dataset	Ada-Ellip		Elbow-AIC		Elbow-BIC		SI Test	
	Cl. num	Time	Cl. num	Time	Cl. num	Time	Cl. num	Time
CC-GENERAL	12	1.4	3	0.3	3	0.3	2	19.8
FinancialDistress	13	0.8	8	0.5	5	0.5	2	3
BankruptcyPredict	8	1.0	9	0.8	9	0.8	2	23.6
FinanceWellBeing	2	0.6	5	0.8	5	0.9	2	10
CreditCustomer	26	1.5	11	0.6	11	0.6	6	10.9
CreditCard	2	2	8	1.4	8	1.4	2	168.4
BankMarketing	9	5	7	9.6	7	9.6	7	317.1
GiveMeCredit	3	5.8	4	3.7	4	3.7	2	4259.9
ChineseBank	9	19.9	9	16.6	9	16.5	9	9776.4
CreditRisk	7	22.3	9	25.5	9	25.4	2	46351.9
CreditCard-RKS	4	1.7	7	6.5	7	6.5	2	182.2
GiveMeCredit-RF	7	20.5	8	31.5	8	31.9	2	4397

**The cluster number estimated by Elbow-BIC was used for CLARANS, GMM and BayesGMM in section 5.5.*

1) Proposed Approach: Ada-Ellip used GMM as the base clustering algorithm to estimate the cluster number of the ten datasets, along with the two datasets transformed by kernel methods. The estimation results and the time cost were recorded in Table II, and the parameters of T_q in Ada-Ellip were set to 0.9, 0.95, 0.95, 0.85, 0.75, 0.8, 0.6, 0.9, 0.9, 0.88, 0.9, and 0.94 for the 12 datasets, respectively.

2) Elbow Test: An Elbow test uses score curves of AIC or BIC to estimate the cluster number, and chooses the cluster number generating the biggest inflexion on the score curve [26]. AIC is defined as [23]

$$\mathrm{AIC}(k) = 2mk + 2\ln(\mathrm{SSE}). \tag{17}$$

BIC replaces the constant coefficient of 2 in AIC with a logarithm of the sample number n, and is defined as [24]

$$\mathrm{BIC}(K) = mk\ln(n) + 2\ln(\mathrm{SSE}). \tag{18}$$

The smaller of the above two criteria, the better of the clustering results.

3) Silhouette Test: A Silhouette test evaluates the Silhouette coefficient of clustering results iteratively with different preset cluster numbers, and then chooses the cluster number that maximizes the Silhouette (SI) score, which is

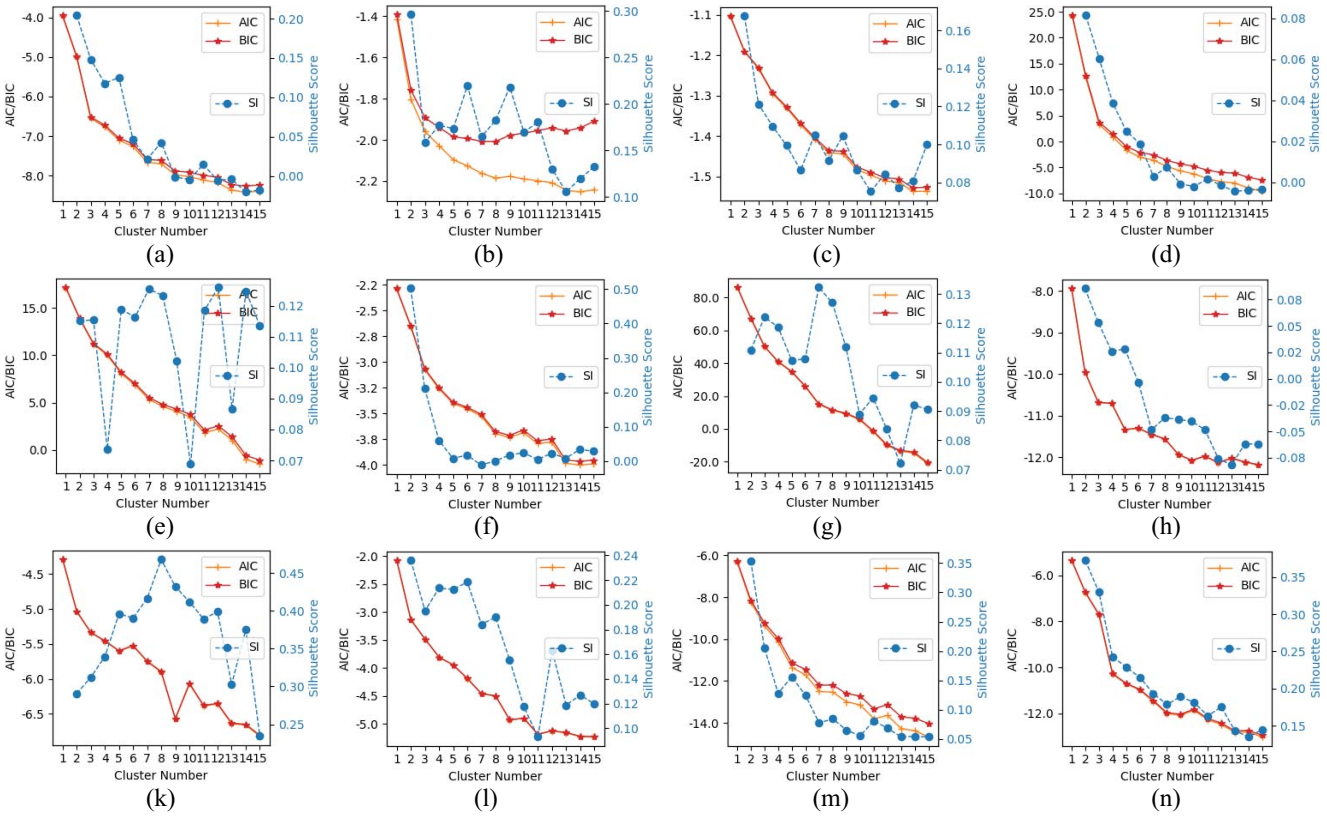

Fig. 5. Distributions of the benchmark datasets and the estimated clusters. (a) CC-GENERAL. (b) FinancialDistress. (c) BankruptcyPredict. (d) FinanceWellBeing. (e) CreditCustomer. (f) CreditCard. (g) BankMarketing. (h) GiveMeCredit. (i) ChineseBank. (j) CreditRisk. (k) ChineseBank. (l) CreditRisk.

defined as [25]

$$\mathrm{SI} = \sum_{i=1}^{n} \frac{b_i - a_i}{\max(a_i, b_i)} \tag{19}$$

where a_i is the average distance between a data point i and all the other points in the same cluster, and b_i is the average distance between a data point i and all the other points in the next nearest cluster. A higher SI score indicates that the model generates better defined clusters.

To compare the results with our approach fairly, GMM was used as the base clustering algorithm for all three methods. The AIC score, BIC score, and Silhouette score were illustrated in Fig. 5. The Silhouette test starts from two clusters because its score is unavailable for one cluster. The estimation results and time costs of them were recorded in Table II.

As shown in Fig. 5, AIC and BIC curves were almost overlapped on all the datasets except FinancialDistress. The Elbow test is sometimes unreliable because the AIC/BIC score curves may be smooth and it is hard to find an elbow. For instance, no evident inflexions were found on the datasets CreditCard, ChineseBank, and CreditCard-RKS, and thus, it is unclear which was the best cluster number.

The Silhouette test estimated that most datasets, except CreditCustomer, BankMarketing, and ChineseBank, have two clusters. The results showed that it was prone to estimate small cluster numbers, which result in large-scope clusters. With a higher Silhouette score, the clustering algorithms mainly keep the maximal variances or the best space divisions, and such a global mechanism was inclined to overlook the small clusters located in the sparse areas and lose landmark information. Besides, the time costs recorded in Table II showed that Silhouette was significantly slower than the other two estimation methods.

Compared with the Elbow test and Silhouette test, Ada-Ellip is more suitable for evaluating the distribution characteristics of a dataset, rather than just a cluster number estimation method. By filtering q_{ci} criteria for each cluster, the clustering resolution can be controlled and no major local distribution information will be lost. It is well suited for the financial data distribution analysis.

D. Evaluation of Clustering Results

We used the following three unsupervised criteria to evaluate the clustering results.

1) *Weighted q (WQ)* is proposed here based on q_{ci} to evaluate the overall aggregation degree of the clustering results

$$\text{weighted } q = \sum_{i=1}^{k} \frac{n_i}{n} q_{cj} \tag{20}$$

where n_i refers to the sample number in cluster c_i, and n refers to the total number of data points. The larger the weighted q, the better the aggregation of data points insides clusters.

TABLE III
COMPARISON OF WQ AND CLUSTER NUMBER

Dataset	Ada-Ellip(*k*-Means)	Ada-Ellip (GMM)	DBSCAN	Hier.	Optics	Spectral	Mean Shift	G-Means	DENC.	BIC-CLAR.	BIC-GMM	BIC-Bayes GMM
CC-GENERAL	**0.9115/12**	**0.9211/12**	0.7746/2	0.8614/2	0.8282/226	0.8923/8	0.8917/15	**0.9103/11**	0.7728/1	0.8760/3	0.8624/3	0.8654/3
Financial-Distress	**0.9514/13**	**0.9531/13**	0.9067/29	0.9008/2	0.8692/36	0.9459/8	0.8989/8	**0.9360/6**	0.8552/1	0.9257/5	0.9256/5	0.9224/5
BankruptcyPredict	**0.9677/8**	**0.9646/8**	0.9496/16	0.9535/2	0.9514/75	0.9638/8	0.9485/29	**0.9640/8**	0.9477/1	0.9589/9	0.9574/9	0.9550/9
FinanceWellBeing	**0.8583/2**	0.8578/2	0.8463/1	0.8577/2	0.8465/2	0.8470/8	0.8502/67	**0.8708/9**	0.8578/3	0.8540/5	0.8567/5	**0.8585/5**
CreditCustomer	**0.7569/26**	**0.7532/26**	0.6430/120	0.5927/2	0.7459/338	0.6920/8	0.5480/1	0.7237/13	0.5480/1	0.6958/11	0.7187/11	**0.7261/11**
CreditCard	0.8856/2	0.8856/2	**0.9087/17**	0.8843/2	0.8475/945	*	0.8856/2	**0.9205/7**	0.7939/1	**0.9233/8**	0.9044/8	0.8967/8
BankMarketing	0.6291/9	**0.6306/9**	**0.8495/1499**	0.4792/2	**0.7363/2767**	*	0.4217/1	0.6214/15	0.4217/1	0.5781/7	0.6032/7	0.6016/7
GiveMeCredit	0.9576/3	**0.9585/3**	*	*	**0.9696/10356**	*	**0.9689/31**	0.9331/4	0.9408/1	0.9347/4	0.9581/4	0.9476/4
ChineseBank	0.8970/9	**0.9050/9**	*	*	0.2127/43548	*	0.7187/18	0.8686/5	0.7089/1	0.9130/9	**0.9050/9**	0.9020/9
CreditRisk	**0.8970/7**	**0.8971/7**	0.8636/165	*	0.8766/17780	*	*	**0.9164/12**	0.8196/1	0.8883/9	0.8943/9	0.8965/9
CreditCard-RKS	**0.9123/4**	0.9089/4	*	*	0.6524/925	*	0.9018/30	0.9174/8	0.8990/3	**0.9146/7**	**0.9208/7**	0.9119/7
GiveMeCredit-RF	**0.9447/6**	**0.9480/7**	*	*	0.7827/7993	*	*	0.9415/4	0.9244/3	0.9455/8	0.9470/8	**0.9473 /8**

** The number of clusters detected by the algorithm is list after "/". "*" means the model ran out of memory. The three best performances of the q value are denoted in bold type.*

TABLE IV
COMPARISON OF SILHOUETTE COEFFICIENT (SI)

Dataset	Ada-Ellip(*k*-Means)	Ada-Ellip (GMM)	DBSCAN	Hier.	Optics	Spectral	Mean Shift	G-Means	DENC.	BIC-CLAR.	BIC-GMM	BIC-Bayes GMM
CC-GENERAL	0.2303	0.2154	**0.3706**	**0.3759**	-0.4939	0.3483	0.3451	0.2529	/	**0.3981**	0.1561	0.1411
FinancialDistress	0.22	0.2195	-0.1768	**0.2999**	-0.4839	**0.285**	0.2271	**0.2205**	/	0.2123	0.2094	0.1892
BankruptcyPredict	**0.1382**	0.0624	-0.135	0.1231	-0.4528	**0.1289**	0.1163	**0.1292**	/	0.1119	0.0275	0.0053
FinanceWellBeing	0.0701	0.0655	/	0.067	**0.3743**	**0.1492**	0.0515	0.0119	0.0633	0.0302	0.0171	0.0126
CreditCustomer	**0.1444**	**0.117**	-0.2617	0.0988	-0.1574	0.1035	/	0.1161	/	**0.1846**	0.1162	0.1053
CreditCard	**0.5027**	**0.5027**	0.1305	0.502	-0.4778	*	**0.5028**	0.1938	/	0.1873	0.0345	0.034
BankMarketing	**0.1362**	**0.138**	0.4326	0.1012	-0.0263	*	/	**0.112**	/	0.0793	0.113	0.1117
GiveMeCredit	**0.3708**	**0.3673**	*	*	0.0224	*	-0.0207	**0.2896**	/	0.2738	0.0258	0.0483
ChineseBank	0.4184	**0.4288**	*	*	-0.9534	*	0.1284	0.3816	/	0.416	**0.4283**	**0.4446**
CreditRisk	**0.206**	**0.2075**	-0.251	*	-0.4539	*	*	0.1865	/	**0.197**	0.1321	0.1957
CreditCard-RKS	**0.2792**	**0.2644**	*	*	-0.4516	*	0.2341	0.1969	**0.3535**	0.1919	0.1113	0.0692
GiveMeCredit-RF	**0.2161**	0.1743	*	*	-0.2958	*	*	**0.2619**	**0.3749**	0.194	0.1735	0.1792

"" means the model ran out of memory. The three best performances are denoted in bold type. "/" means the criterion does not apply to the model since there is only one cluster.*

2) *Silhouette coefficient (SI)* is as it is introduced in (19).
3) The *Davies-Bouldin Index (DBI)* is defined as the average similarity between each cluster c_i and its most similar one c_j [50]

$$\mathrm{DBI} = \frac{1}{k}\sum_{i=1}^{k}\frac{s_i + s_j}{n} q_{d_{ij}} \tag{21}$$

where s_i is the average distance between each point in cluster c_i and the centroid of the cluster c_j, and d_{ij} is the distance between the centroids of cluster c_i and c_j. Lower *DBI* indicates that a model has better separation between clusters.

Ada-Ellip can guild the clustering process for both *k*-Means and GMM. Therefore, we examined two versions of it: 1) Ada-Ellip (*k*-Means) and 2) Ada-Ellip (GMM). To evaluate the performance of the proposed approach, we selected eight well-known automatic (without preset *k*) clustering algorithms [51], including DBSCAN, Ward Hierarchical, Optics, Spectral, MeanShift, *G*-Means, DENCLUE, and CLARANS, for comparison. Besides, we also evaluated the GMM and its variational Bayesian version: BayesGMM, with the estimated cluster numbers by BIC, which were recorded in Table II. All of the comparative algorithms used their default parameters in ScikitLearn.jl and Smile. The performances of these algorithms on WQ, SI, DBI, and time cost were list in Tables III–VI, respectively.

As shown in Table III, DBSCAN and Optics tended to generate too many clusters. Hierarchical always generated two clusters. *G*-Means detected too many clusters in some datasets, which is partially due to *G*-Means's assumption that the samples in each cluster follow a strict normal distribution, yet, in practice, most clusters cannot hold the assumption and the clusters are split unduly. The cluster number detected by MeanShift is unreasonably large in many datasets. DENCLUE was not effective because it did not detect most clusters in the datasets. Above all, only Ada-Ellip and Spectral estimated reasonable cluster numbers automatically, but Spectral ran out of memory on the datasets with more than 30 000 samples.

Ada-Ellip performed well on most datasets in terms of WQ. Although Hierarchical and Optics also had the best WQ on some datasets, they generated too many clusters, which indicates that WQ does not discriminate the situation when an algorithm achieves high score by generating excessive number of clusters.

TABLE V
COMPARISON OF THE DBI

Dataset	Ada-Ellip(*k*-Means)	Ada-Ellip (GMM)	DBSCAN	Hier.	Optics	Spectral	Mean Shift	G-Means	DENC.	BIC-CLAR.	BIC-GMM	BIC-Bayes GMM
CC-GENERAL	1.4149	1.3813	1.6936	**1.091**	1.1039	**0.9864**	**1.1756**	1.2074	/	1.2531	3.3576	3.4367
FinancialDistress	1.7215	1.7746	1.6232	1.4776	**1.4247**	**1.3724**	**1.1978**	1.549	/	1.7216	1.8896	1.8524
BankruptcyPredict	**1.9875**	2.4303	2.0881	2.7737	**1.3745**	2.015	**0.8091**	1.9259	/	2.0425	4.4543	4.61
FinanceWellBeing	3.1921	**3.1598**	/	3.2884	**0.9947**	1.5867	**0.9969**	4.5082	4.3033	4.914	5.2749	5.3285
CreditCustomer	2.0657	**2.004**	**1.3848**	2.5869	**1.3177**	2.2711	/	2.3197	/	2.6389	2.1529	2.4027
CreditCard	**0.8308**	**0.8308**	1.4087	**0.8308**	1.1464	*	0.8309	1.6879	/	1.6646	3.4357	2.7705
BankMarketing	2.3647	**2.2003**	**1.1583**	3.0136	**1.1131**	*	/	2.3691	/	2.9015	2.2607	2.4043
GiveMeCredit	**0.7378**	**0.7437**	*	*	1.4145	*	**0.9911**	1.1649	/	1.229	3.5409	4.1387
ChineseBank	**1.2224**	1.5412	*	*	1.4982	*	**1.0914**	1.3046	/	**0.9037**	1.5264	1.5385
CreditRisk	1.8187	1.8227	**1.4342**	*	**1.1028**	*	*	1.8998	/	**1.7573**	4.9204	2.7198
CreditCard-RKS	**1.7278**	2.2281	*	*	**1.1473**	*	1.7875	1.8852	**1.3528**	1.7665	2.9193	3.154
GiveMeCredit-RF	**1.7485**	1.7821	*	*	**1.1012**	*	*	1.7493	**0.9868**	1.7213	2.3182	2.267

"" means the model ran out of memory. The three best performances are denoted in bold type. "/" means the criterion does not apply to the model since there is only one cluster.*

TABLE VI
COMPARISON OF THE TIME COST (SECONDS)

Dataset	Ada-Ellip(*k*-Means)	Ada-Ellip (GMM)	DBSCAN	Hier.	Optics	Spectral	Mean Shift	G-Means	DENC.	BIC-CLAR.	BIC-GMM	BIC-Bayes GMM
CC-GENERAL	**1**	1.4	2.8	5	30.6	13	80.7	68.2	4.1	**0.8**	**0.5**	1.1
FinancialDistress	**0.8**	**0.8**	1	1.4	50.2	2.4	38.1	4.7	4.2	**0.9**	1.3	1.2
BankruptcyPredict	**1**	**3.5**	11.6	8.2	96.8	8.1	180.2	41.5	14.3	**1.2**	5.1	15.2
FinanceWellBeing	**0.6**	**1.3**	13.3	7.7	140	9.7	297.9	20	79.3	**2.2**	4.3	7.9
CreditCustomer	**0.8**	1.6	5.8	6.6	60.7	18.2	125.6	24.7	9	**1.1**	**0.6**	1.6
CreditCard	**1.7**	**2**	19.1	61.4	439	*	124.7	76.2	63.9	**1.9**	4.1	26.3
BankMarketing	**0.6**	**5.6**	75.4	139.4	1595.9	*	4455.6	130.7	269.6	**3.3**	7.1	8.7
GiveMeCredit	**3.4**	**5.8**	*	*	5264	*	45522.5	434.5	1214.9	5.9	**4.9**	16.3
ChineseBank	**14**	**19.9**	*	*	34446	*	133809.2	904.8	7177.2	**31.2**	43.8	168.4
CreditRisk	**14.7**	**22.3**	8613	*	168597	*	*	3903.7	41044.7	**49.1**	103.8	285
CreditCard-RKS	**0.2**	**2.5**	*	*	2507.3	*	3309.2	155.6	171.8	**5.1**	66.2	43.8
GiveMeCredit-RF	**1.5**	**46.6**	*	*	55308.4	*	*	775.7	3592.5	**34.8**	228.9	399

"" means the model ran out of memory. The three best performances are denoted in bold type.*

SI prefers clusters far apart from each other. Take dataset CC-GENERAL for example. Both DBSCAN and Hierarchical detected two clusters, and fewer clusters certainly make data points in a cluster more apart from samples in other clusters and result in higher SI scores. In contrast, Optics tended to generate too many clusters and result in lower SI scores, indicating that the clusters were not well apart from each other. Ada-Ellip generated a reasonable number of clusters and guaranteed that data points inside tight clusters shared numerical similarities. Its SI scores (*k*-means and GMM) were among the best on six datasets, including all the three large-scale datasets.

DBI reflects the quality of space separation by clusters. Ada-Ellip's performances were among the best DBI on five datasets. The reason Ada-Ellip did not achieve the best DBI on all the datasets is that space separation is not its most important objective. Take the two clusters located at the right of FinancialDistress in Fig. 3(b) as an example. The two clusters were not well separated in the space. If we merge them into one cluster, its DBI can be improved to 1.2441 by lowering the T_q value to 0.9, which will result in three clusters. But the data points in the two clusters do not squeeze tightly and share lower similarities. Ada-Ellip treated them as two clusters for the sake of interpretabilit. Thus, Ada-Ellip does not necessarily beat other models on DBI while it pursues high tightness.

Table VI showed that Ada-Ellip was faster than all the other models, which were especially notable on the last three large-scale datasets. As discussed in Section IV, eigen decomposition of the $m-$dimension matrix is not computationally extensive, and the experimental results proved that the time costs of Ada-Ellip were trivial. This merit of Ada-Ellip is significant in large-scale financial applications.

E. Case Study

Because the proposed models are designed for unsupervised learning, it is inherently hard to evaluate the performance of SVDD and numerical interpretation of the clusters. This section uses the dataset CreditRisk to further explain how these models work.

As recorded in Table II, Ada-Ellip detected seven clusters on the dataset CreditRisk. These clusters can be considered as seven subpatterns of the dataset, the centroids can be viewed as the representatives of the clusters, and the large q_{cj} guaranteed the similarities of data points inside each cluster. The linear correlations of features can be interpreted by the projection matrix $\hat{P}_i$ attached in each cluster.

TABLE VII
NORMALIZED SINGULAR VALUES IN EACH CLUSTER OF "CREDITRISK" AND OUTLIERS

Clu. no.	Inside num.	Outliers num.	1st S.V.	2nd S.V.	3rd S.V.	4th S.V.
1	108614	983	0.908	0.0218	0.0158	0.0113
2	153034	1529	0.8953	0.0246	0.0178	0.0134
3	122101	1219	0.8923	0.0217	0.0193	0.0173
4	15791	158	0.8893	0.0252	0.0198	0.0142
5	32653	325	0.9042	0.0203	0.0159	0.0136
6	46304	462	0.8878	0.029	0.0149	0.0129
7	21503	217	0.8956	0.0311	0.0165	0.012

After performing SVD with the data points inside each cluster, the top four normalized squared singular values were recorded in Table VII, along with the number of outliers found by SVDD for each cluster.

As shown in Table VII, the largest squared singular value accounts for the major proportion in each cluster. It means that the dataset inside each cluster was inherently dominated by one "principal component," and it can be interpreted by the first eigenvector, which corresponds to the largest singular value. The elements of the eigenvector, which can be interpreted as features' weights in each cluster, were illustrated in Fig. 6.

As shown in Fig. 6(a), in cluster 1, the 4th, 6th, 12th, 14th–16th, and 18th features were the most important ones and took negative effects, while the 5th feature took no effect. Cluster 2 was almost the same as cluster 1 except the weight of the 15th feature was zero. Clusters 3–7 were slightly different from each other with varying weights of certain features.

Plus, it can be deduced from the above distributions of feature weights in U_1 of each cluster that which features resulted in the divergence of the clusters. It can be observed from Fig. 6 that the 15th feature played an important role in differentiating clusters 2, 3, 6, and 7 from clusters 1, 4, and 5, and the 19th feature was the key in distinguishing clusters 1, 2, 3, and 5 from clusters 4, 6, and 7. It was also notable that the 12th and 16th features were only prominent in clusters 1 and 2.

Finally, the number of outliers was also recorded in Table VII. These outliers are important clues for detecting potential anomalous activities. The analysis of them depends on domain knowledge, and it is out of the scope of this article.

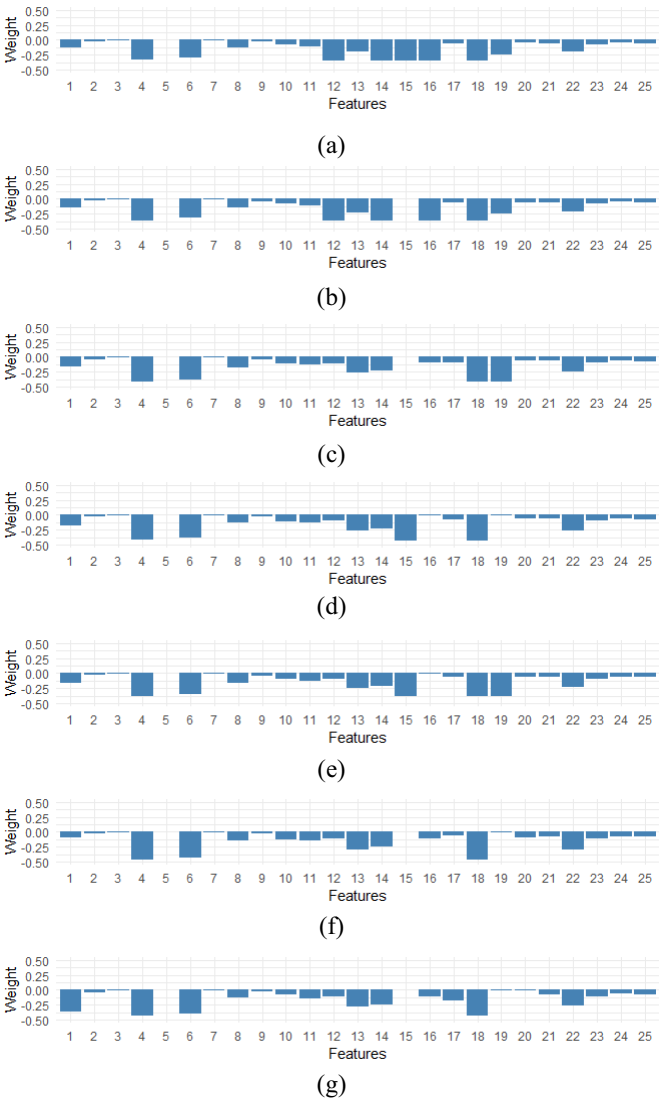

Fig. 6. Distributions of features' weights in clusters. (a) Distribution of features' weights in cluster 1. (b) Distribution of features' weights in cluster 2. (c) Distribution of features' weights in cluster 3. (d) Distribution of features' weights in cluster 4. (e) Distribution of features' weights in cluster 5. (f) Distribution of features' weights in cluster 6. (g) Distribution of features' weights in cluster 7.

F. Further Discussion

1) Setting of the T_q Value: T_q is the threshold of q_{cj}, and a large T_q value in Ada-Ellip tends to generate more clusters, which also means that samples in the clusters share greater similarities, and increases the interpretability of the clusters. Generally speaking, clusters of wider hyperellipsoidal scopes require smaller T_q values, such as CreditCard, because data points in those clusters distribute in a larger area. On the other hand, clusters of narrower scopes require larger T_q values, such as "FinancialDistress," because data points in these clusters squeeze in a small area. Our experiences showed that an interval $[0.85, 0.95]$ of T_q would be appropriate in most cases, and the setting of T_q depends on the specific distribution of the data. Too small T_q value indicates that the dataset (such as BankMarketing) cannot be clustered well.

2) Impacts of Data Features: The efficiency of the proposed approach depends on an assumption that the feature number of a financial dataset is far less than the sample number. Besides, the interpretation function requires that the features themselves are interpretable. The proposed approaches were motivated by financial applications where most features have explainable management or economical meanings, and the number of them is finite. If the feature number of a dataset is too large, the computation of the proposed models, such as q_{cj}, will no longer be efficient. If the features are uninterpretable, such as text or graph embedded features and image channel features, the clusters are then uninterpretable. When the dimension of a dataset is extremely high, the Euclidean distance is not reliable anymore because of the dimension curse, not to mention the algorithms based on it. Therefore,

the proposed approaches are not intended to be applied to data, such as texts, voices, images, and videos.

VI. Conclusion

Distributions of financial datasets are always complex due to changing social environments and human activities. Interpretability, adaptivity, and speed are of great importance to financial data mining. To conduct cluster analysis for unlabeled financial datasets and give reasonable interpretations, we proposed a criterion q_{ci} to evaluate the quality of a cluster by measuring the aggregation degree of data points inside the cluster. Then, we designed an adaptive algorithm Ada-Ellip to detect hyperellipsoidal clusters automatically with the help of q_{ci}. We also proposed a revised SVDD model with a new solver based on a penalty function to refine the centroids and hyperellipsoidal scopes of the clusters. As a result, the clusters are made tighter and easier to be interpreted. The adaptively detected clusters can be used to analyze the subpatterns of financial datasets, and the first vector in the left eigen matrix after SVD can be used to interpret the roles of the features.

Experiments on ten financial benchmark datasets, along with two datasets transformed by kernel functions, showed that the proposed Ada-Ellip estimated reasonable cluster numbers, and generated tight clusters containing data points that share great similarities. Most importantly, it was fast and, thus, highly applicable to large-scale financial datasets. Besides, a case study on the dataset CreditRisk explained how the dataset can be interpreted using the detected clusters. Experiments showed that Ada-Ellip was fast, reliable, free from sophisticated parameter tuning techniques, and well suited for unsupervised financial data mining tasks, such as fraud detection, reject inference, and credit evaluation.

Theoretically, the proposed clustering framework also has the potential to be used for other types of data, as long as the applied domain faces similar circumstances and challenges as we discussed in Section III. Finally, the clustering evaluation criterion proposed in (20) cannot discriminate the situation when an algorithm achieves high score by generating too many clusters. Our future works will focus on the improvement of this criterion and other clustering evaluation methods.

References

[1] J. María Luna, P. Fournier-Viger, and S. Ventura, "Frequent itemset mining: A 25 years review," *Wiley Interdiscipl. Rev. Data Min. Knowl. Discov.*, vol. 9, no. 6, p. e1329, 2019.

[2] J. VanderPlas, *Python Data Science Handbook*, O'Reilly Media, Inc., Sebastopol, CA, USA, 2019, pp. 433–479.

[3] R. Guan, H. Zhang, Y. Liang, F. Giunchiglia, L. Huang, and X. Feng, "Deep feature-based text clustering and its explanation," *IEEE Trans. Knowl. Data Eng.*, early access, Oct. 6, 2020, doi: 10.1109/TKDE.2020.3028943.

[4] D. M. J. Tax, and R. P. W. Duin, "Support vector data description," *Mach. Learn.*, vol. 54, no. 1, pp. 45–66, 2004.

[5] Y. Pang, J. Xie, F. Nie, and X. Li, "Spectral clustering by joint spectral embedding and spectral rotation," *IEEE Trans. Cybern.*, vol. 50, no. 1, pp. 247–258, Jan. 2020.

[6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[7] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2013. [Online]. Available: arXiv:1312.6114.

[8] F. Locatello *et al.*, "Challenging common assumptions in the unsupervised learning of disentangled representations," in *Proc. ICML*, 2019, pp. 4114–4124.

[9] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1253–1278, 2000.

[10] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics Intell. Lab. Syst.*, vol. 2, nos. 1–3, pp. 37–52, 1987.

[11] X. Cheng and T. J. Dunkerton, "Orthogonal rotation of spatial patterns derived from singular value decomposition analysis," *J. Climate*, vol. 8, no. 11, pp. 2631–2643, 1995.

[12] M. E. Wall, A. Rechtsteiner, and L. M. Rocha, "Singular value decomposition and principal component analysis," *A Practical Approach to Microarray Data Analysis*. Boston, MA, USA: Springer, 2003, pp. 91–109.

[13] K. Sim, G.-E. Yap, D. R. Hardoon, V. Gopalkrishnan, G. Cong, and S. Lukman, "Centroid-based actionable 3D subspace clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 6, pp. 1213–1226, Jun. 2013.

[14] S. S. Jung and W. Chang, "Clustering stocks using partial correlation coefficients," *Physica A, Stat. Mech. Appl.*, vol. 462, pp. 410–20, Nov. 2016.

[15] D. Hübner and M. Tangermann, "Challenging the assumption that auditory event-related potentials are independent and identically distributed," in *Proc. 7th Int. Brain Comput. Interface Meeting*, 2017, pp. 192–197.

[16] Q. Yang, W.-N. Chen, Y. Li, C. L. P. Chen, X.-M. Xu, and J. Zhang, "Multimodal estimation of distribution algorithms," *IEEE Trans. Cybern.*, vol 47, no. 3, pp. 636–650, Mar. 2017.

[17] X. Chen, W. Sun, B. Wang, Z. Li, X. Wang, and Y. Ye, "Spectral clustering of customer transaction data with a two-level subspace weighting method," *IEEE Trans. Cybern.*, vol. 49, no. 9, pp. 3230–3241, Sep. 2019.

[18] B.-S. Chen, W.-Y. Chen, C.-T. Yang, and Z. Yan, "Noncooperative game strategy in cyber-financial systems with Wiener and poisson random fluctuations: LMIs-constrained MOEA approach," *IEEE Trans. Cybern.*, vol. 48, no. 12, pp. 3323–3336, Dec. 2018.

[19] A. Liu, J. Lu, and G. Zhang, "Concept drift detection via equal intensity k-means space partitioning," *IEEE Trans. Cybern.*, vol. 51, no. 6, pp. 3198–3211, Jun. 2021.

[20] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 7, no. 7, pp. 881–892, Jul. 2002.

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

[22] C. E. Rasmussen, "The infinite Gaussian mixture model," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2000.

[23] Y. Sakamoto, M. Ishiguro, and G. Kitagawa, *Akaike Information Criterion Statistics*, vol. 81. Dordrecht, The Netherlands: D. Reidel, 1986.

[24] R. E. Kass and A. E. Raftery, "Bayes factors," *J. Amer. Stat. Assoc.*, vol. 90, no. 430, pp. 773–795, 1995.

[25] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, Nov. 1987.

[26] J.-P. Baudry, A. E. Raftery, G. Celeux, K. Lo, and R. Gottardo, "Combining mixture components for clustering," *J. Comput. Graph. Stat.*, vol. 19, no. 2, pp. 332–353, 2010.

[27] Y. Huang, Y. Zhang, P. Shi, Z. Wu, J. Qian, and J. A. Chambers, "Robust Kalman filters based on Gaussian scale mixture distributions with application to target tracking," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 10, pp. 2082–2096, Oct. 2019.

[28] L. C. Matioli, S. R. Santos, M. Kleina, and E. A. Leite, "A new algorithm for clustering based on kernel density estimation," *J. Appl. Stat.*, vol. 45, no. 2, pp. 347–366, 2018.

[29] U. Von Luxburg, "A tutorial on spectral clustering," *Stat. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.

[30] Z. Wu, S. Liu, C. Ding, Z. Ren, and S. Xie, "Learning graph similarity with large spectral gap," *IEEE Trans. Syst., Man, Cybern., Syst*, vol. 51, no. 3, pp. 1590–1600, Mar. 2021.

[31] A. C. Türkmen, G. Çapan, and A. T. Cemgil, "Clustering event streams with low rank Hawkes processes," *IEEE Signal Process. Lett.*, vol. 27, pp. 1575–1579, Aug. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9178953

[32] J. Y. Song, W. Chang, and J. W. Song, "Cluster analysis on the structure of the cryptocurrency market via Bitcoin–Ethereum filtering," *Physica A, Stat. Mech. Appl.*, vol. 527, Aug. 2019, Art. no. 121339.

[33] S. K. Kingrani, M. Levene, and D. Zhang, "Estimating the number of clusters using diversity," *Artif. Intell. Res.*, vol. 7, no. 1, pp. 15–22, 2018.
[34] W. Guo, Y. Shi, and S. Wang, "A unified scheme for distance metric learning and clustering via rank-reduced regression," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 8, pp. 5218–5229, Aug. 2021.
[35] I. Davidson, A. Gourru, and S. Ravi, "The cluster description problem-complexity results, formulations and approximations," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran, 2018.
[36] T. Sakai, K. Tamura, and H. Kitakami, "Identifying main topics in density-based spatial clusters using network-based representative document extraction," in *Proc. IEEE 8th Int. Workshop Comput. Intell. Appl. (IWCIA)*, Hiroshima, Japan, 2015, pp. 77–82.
[37] J. Zhang, X. Yu, Y. Xun, S. Zhang, and X. Qin, "Scalable mining of contextual outliers using relevant subspace," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 3, pp. 988–1002, Mar. 2020.
[38] G. S. Davidson, B. N. Wylie, and K. W. Boyack, "Cluster stability and the use of noise in interpretation of clustering," in *Proc. INFOVIS*, San Diego, CA, USA, 2001, pp. 23–30.
[39] G. He, Y. Pan, X. Xia, J. He, R. Peng, and N. N. Xiong, "A fast semi-supervised clustering framework for large-scale time series data," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 7, pp. 4201–4216, Jul. 2021.
[40] P. Jain, B. Kulis, J. V. Davis, and I. S. Dhillon, "Metric and kernel learning using a linear transformation," *J. Mach. Learn. Res.*, vol. 13, pp. 519–547, Mar. 2012.
[41] S. Si, C.-J. Hsieh, and I. S. Dhillon, "Memory efficient kernel approximation," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 682–713, 2017.
[42] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2002.
[43] M. Honarkhah and J. Caers, "Stochastic simulation of patterns using distance-based pattern modeling," *Math. Geosci.*, vol. 42, no. 5, pp. 487–517, 2010.
[44] S. Shalev-Shwartz, A. Gonen, and O. Shamir, "Large-scale convex minimization with a low-rank constraint," in *Proc. ICML*, 2011, pp. 329–336.
[45] K. Wu and H. Simon, "Thick-restart Lanczos method for large symmetric Eigenvalue problems," *SIAM J. Matrix Anal. Appl.*, vol. 22, no. 2, pp. 602–616, 2000.
[46] X. B. Zhu, P. Witold, and Z. Li, "Granular data description: Designing ellipsoidal information granules," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4475–4484, Dec. 2017.
[47] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.
[48] R. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proc. 20th Int. Conf. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2007, pp. 1177–1184. [Online]. Available: https://dl.acm.org/doi/10.5555/2981562.2981710
[49] F. Li, C. Ionescu, and C. Sminchisescu, "Random Fourier approximations for skewed multiplicative histogram kernels," in *Proc. Joint Pattern Recognit. Symp.*, 2010, pp. 262–271.
[50] H. Maria, B. Yannis, and M. Vazirgiannis, "On clustering validation techniques," *J. Intell. Inf. Syst.*, vol. 17, nos. 2–3, pp. 107–145, 2001.
[51] P.-N. Tan, M. Steinbach, A. Karpatne, and V. Kumar, *Introduction to Data Mining*, 2nd ed. New York, NY, USA: Pearson, 2018.

Tie Li received the Ph.D. degree in management science and engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2018.

He was a Visiting Scholar with the University of Illinois at Chicago, Chicago, IL, USA. He is currently a Lecturer with the School of Management and Economics, University of Electronic Science and Technology of China. He has authored four software and 13 papers. His research interests include big data mining, distributed computing, and information management.

Gang Kou received the B.S. degree in physics from Tsinghua University, Beijing, China, in 1997, and the M.S. degree in computer science and the Ph.D. degree in information technology from the University of Nebraska at Omaha, Omaha, NE, USA, in 2003 and 2006, respectively.

He is a Professor and the Executive Dean of the School of Business Administration, Southwestern University of Finance and Economics, Chengdu, China. His research interests include data mining, multiple criteria decision making, and optimization.

Prof. Kou has been selected to the list Highly Cited Researchers 2016 in the field of computer science published by Clarivate Analytics.

Yi Peng received the B.S. degree in management information systems from Sichuan University, Chengdu, China, in 1997, and the M.S. degree in management information systems and the Ph.D. degree in information technology from the University of Nebraska at Omaha, Omaha, NE, USA, in 2003 and 2007, respectively.

She is currently a Professor with the School of Management and Economy, University of Electronic Science and Technology of China, Chengdu. Her research interests include multiple criteria decision making, mathematical modeling, and data mining techniques and applications.

Prof. Peng has been selected to the list Highly Cited Researchers 2016 in the field of computer science published by Clarivate Analytics (formerly, Thomson Reuters).

Philip S. Yu (Life Fellow, IEEE) received the M.B.A. degree from New York University, New York, NY, USA, in 1982, and the Ph.D. degree in EE from Stanford University, Stanford, CA, USA, in 1978.

He is a Distinguished Professor with the Department of Computer Science, University of Illinois at Chicago, Chicago, IL, USA, and also holds the Wexler Chair in Information and Technology. He spent most of his career with IBM Thomas J. Watson Research Center, where he was the Manager of the Software Tools and Techniques Department. He has published more than 970 papers in refereed journals and conferences with more than 74 500 citations and an H-Index of 127. He holds or has applied for more than 300 U.S. patents. His main research interests include big data, data mining, social network, privacy-preserving data publishing, data stream, database systems, and Internet applications and technologies.

Dr. Yu is on the Steering Committee of ACM Conference on Information and Knowledge Management and was a Steering Committee Member of the IEEE Conference on Data Mining and the IEEE Conference on Data Engineering. In addition to serving as program committee member on various conferences, he was the Program Chair or Co-Chairs of the 2009 IEEE International Conference on Service-Oriented Computing and Applications, the IEEE Workshop of Scalable Stream Processing Systems (SSPS'2007), the IEEE Workshop on Mining Evolving and Streaming Data in 2006, the 2006 joint conferences of the 8th IEEE Conference on E-Commerce Technology (CEC'2006), the 3rd IEEE Conference on Enterprise Computing, E-Commerce and E-Services (EEE'2006), the 11th IEEE International Conference on Data Engineering, the 6th Pacific Area Conference on Knowledge Discovery and Data Mining, the 9th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, the 2nd IEEE International Workshop on Research Issues on Data Engineering: Transaction and Query Processing, the PAKDD Workshop on Knowledge Discovery from Advanced Databases, and the 2nd IEEE International Workshop on Advanced Issues of E-Commerce and Web-based Information Systems. He is the Editor-in-Chief of *ACM Transactions on Knowledge Discovery from Data*. He was the Editor-in-Chief for IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING from 2001 to 2004. He had also served as an Associate Editor for *ACM Transactions on the Internet Technology* from 2000 to 2010 and *Knowledge and Information Systems* from 1998 to 2004. He is a Fellow of ACM.