Support vector machine based clustering: A review

Abou Bakr Seddik Drid, Djeffal Abdelhamid, Abdelmalik Taleb-Ahmed

Support vector machine based clustering: A review

DRID Abou Bakr Seddik
Computer science departement
Mohamed Khider university
LESIA Laboratory
Biskra, Algeria
drid.abs@univ-biskra.dz

Dr. Djeffal Abdelhamid
Computer science departement
Mohamed Khider university
LESIA Laboratory
Biskra, Algeria
abdelhamid.djeffal@univ-biskra.dz

Pr. TALEB-AHMED Abdelmalik
Opto-Acoustic-Electronics Department
Polytechnic University of Hauts-de-France
IEMN Laboratory
Valenciennes, France
abdelmalik.taleb-ahmed@univ-valenciennes.fr

***Abstract*—Clustering or cluster analysis is one of the most important data mining techniques, its objective is to regroup similar objects (data points) into groups, with the aim of maximizing the similarity between objects in the same group (intra-cluster similarity) and minimizing it between the objects of different groups (inter-cluster similarity) in unsupervised way. Recently, support-based clustering methods attracted a lot of attention, especially Support Vector Clustering (SVC) due to its capability to overcome the main hardships of classical clustering methods. SVC can easily handle complex shape clusters and easily identify the number of clusters without initialization. SVC undergoes on two main steps, training step and labelling step, the first one consist of solving a quadratic programming problem (QPP) to obtain a decision mathematics function, the next step uses the decision function to label all objects with their appropriate cluster by constructing the adjacency matrix known as complete graph (CG). However, training an SVC model (solving a QPP) and labelling objects using huge data sets can lead to a high computation burden, in order to surmount this main issue and trying to improve the SVC performance, many methods and techniques was proposed in literature. In this paper, we aim to highlight and classify some of the most insightful works works proposed by researchers according to their targeted SVC step.**

***Index Terms*—Clustering, Support vector clustering, support vector machine, sequential minimal optimization.**

I. Introduction

The last few decades have witnessed the rise and the advance of the information age, which is due to the evolution on the information technologies and computing infrastructures. With these technologies empowering all sectors, there has been a surge of data available in digital form. Meanwhile, improving data collection and storage has proving to be quite difficult on developers. However, the challenge is not in collecting neither storing data, but the dilemma is how to deal with this new generation data in order to maximize information and insights extraction, deliver accurate results faster, even in real time, for the fundamental aim which is helping organizations make more-informed decisions. Many fields and disciplines were emerged to optimize the benefit extracted from acquired data. One of these advantageous disciplines is Data Mining, which is a computational process of discovering interesting and useful patterns and relationships in large volume of data. The field combines tools from mathematics, statistics and artificial intelligence with database management to analyze large digital collections, known as data sets. Since the nature of the analyzed data and the purpose of use of the results is widely diverse, data mining has incorporated many tasks as clustering, classification, regression, association rule learning, etc. Clustering, also known as cluster analysis, has been identified as a core task in data mining [3], its objective is to form natural groups of patterns (e.g. objects, data points, or feature vectors) in a supervised way [4]. The aims are to maximize intra-cluster similarity and minimize inter-cluster similarity [4]. Since the apparition of K-means in 1955, the simplest and most known clustering algorithm, thousands of algorithms have been published such as k-Medoids, DBSCAN, BIRCH, STING, etc. [1]. During the course of developing new clustering algorithms and optimizing existing ones, they have been widely and successfully used in many domains such as business intelligence, image pattern recognition, Web search, biology and security. In spite of this success, these methods often have an unstable performance when extracting appropriate cluster boundaries and identifying the exact number of clusters [4]. Support Vector Clustering (SVC), is a relatively new kernel-based algorithm, proposed in 2000 by Ben-Hur et al. [6], inspired by the Support Vector Machine (SVM) method, it accurately groups data point into clusters based on two main steps training and labeling. The SVC training step uses the kernel trick to form the minimum sphere that enclose most of the data points, then, by mapping back to the input space, the obtained sphere generates the cluster boundaries. Labeling step compute the adjacency matrix based on the direct connection test between data point, then it labels the whole data points in terms of the adjacency matrix [6]. Due to its ability of generating complex cluster boundaries, and its smoothness on dealing with outliers, also, the no necessity of predefining the number of clusters, the SVC method outperforms clustering conventional methods. However, dealing with large data sets shows a significant time consumption, both in training and labeling step, and it presents great challenges. In addition, the resulting clustering is very sensitive to the selection of some parameters, which are basically done in a supervised way . Trying to solve these bottlenecks, many works have been proposed in the literature with various optimization concepts, Li et al. [4] presented a survey, in which they summarized and classified a large number of works done until mid-2014 into theory or application works. In this work, we will spotlight

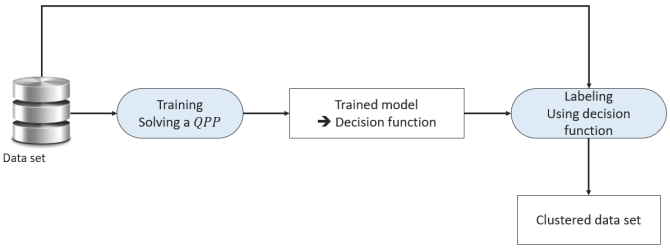


Fig. 1: SVC main steps.

and discuss the innovative theories in both, the most insightful works presented in [4], and the latest significant researches proposed after [4], in the aims of making simple and fast to get familiar with the SVC method and its latest research trends. Finally, we give our vision of optimizing the SVC method.

II. A REVIEW OF SUPPORT VECTOR CLUSTERING

Based on support vector domain description (SVDD) by Tax and Duin [8] and Support Vector Machine by Vapnik [7], Ben-Hurr proposed a robust mathematical kernel-based method called support vector clustering (SVC) [6]. The method task is to accurately label a set of data points in an unsupervised way, throw two main steps. Training step to construct a trained kernel radius function, and the label step to assign a cluster index for each data point. In this section, we present an overview of SVC method principals, and show why it is considered as an advantageous method compared to conventional clustering methods, and the main upgrade possibilities to overcome its drawbacks.

A. SVC training

Also known as the estimation of a trained support function. Two major approaches are proposed in the literature to define the domain of novelty in this step, the large margin hyperplane (LMH) [9], [10], [27] and the minimum englobing sphere (MES) [6], [8] Fig.2.

1) LMH: From a given data set $X = x_1, x_2, \ldots, x_N$, the LMH method tries to define the domain of novelty by learning an optimal hyperplane that can separate the data samples and the origin such that the margin, i.e., the distance from the origin to the hyperplane, is maximized. This optimization problem is formulated as follows [9]:

$$\max_{w,\rho}\left(\frac{|\rho|}{\|w\|^2}\right)$$

subject to

$$y_i(w^T\phi(x_i) - \rho) \geq 0, i = 1, 2, ..., N$$

Using Lagrangian and some transformations, we get the following dual problem [10]:

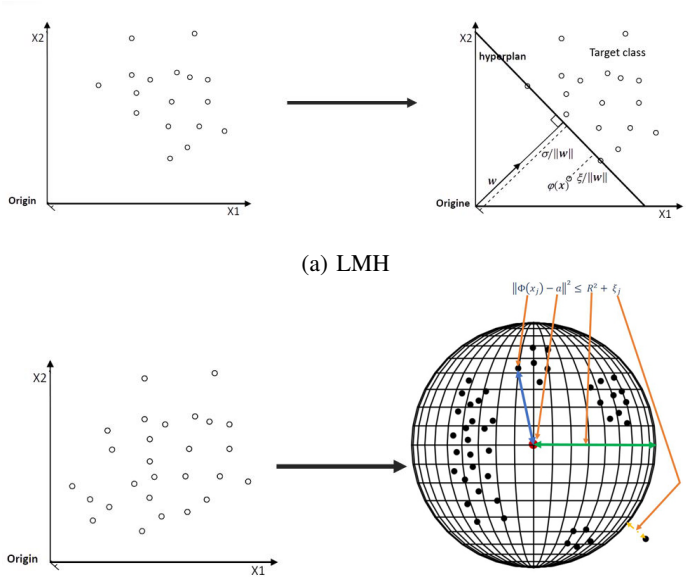


Fig. 2: LMH vs MES.

$$\min_{\alpha} \quad \sum_{i,j}\alpha_i\alpha_j k(x_i, x_j) - \sum_i \alpha_i k(x_i, x_j)$$

subject to

$$0 \leq \alpha_i \leq \frac{1}{vl}\sum_i \alpha_i = 1$$

.

2) MES: However, the MES method, starts by defining a hypersphere as following:
Let $\{x_i/i = 1, 2, \ldots, N\} \subset X$ represent the set of N data points, with $X \subset R^n$, the data space. And let Φ be the nonlinear transformation from X to high-dimensional feature-space. The goal is to find the smallest hypersphere with radius R, and a center a that enclose almost of the data set, so we can write:

$$\min_{R,\alpha\xi_j} R^2 + C\sum_j \xi_j \tag{1}$$

$$\text{s.t} \begin{cases} \left\|\Phi(x_i) - a\right\|^2 \leq R^2 + \xi_j, j = 1, 2, ...N \\ \xi_j \geq 0, \qquad\qquad\qquad\quad j = 1, 2, ...N \end{cases}$$

Where $\xi_j \geq 0$ are slack variables that relax the constraints, and $C \in [0, 1]$ is a constant parameter allows controlling the penalty of noise, are introduced to allow dealing with the outliers. In order to solve the primal problem (1), we need to solve the following Wolfe dual problem -obtained

by introducing the Lagrangian and the KKT complementary condition of Fletcher (1987) in (1)- [6], [11]:

$$\max \sum_{i=1}^{N} \beta_i K(x_i, x_i) - \sum_{i=1}^{N}\sum_{j=1}^{N} \beta_i \beta_j K(x_i, x_j) \qquad (2)$$

$$\text{s.t} \begin{cases} \sum_{i=1}^{N} \beta_i = 1 \\ 0 \leq \beta_i \leq C \quad \forall i = 1, 2, ..., N \end{cases}$$

Where β_i are Lagrange multipliers, and $K(x_i, x_j) = \Phi(x_i).\Phi(x_j)$ is the kernel function. According to [6], only the points with $0 \leq \beta_i \leq C$ lies on the hypersphere's surface, they are known as the support vectors (SVs), and they can define the cluster boundaries. Points with $\beta_i = 0$ lies inside the hypersphere and are called inner data points (ID). Data points with $\beta_i = C$ lies outside the hypersphere and they are called bounded support vectors (BSVs). Many kernel functions can be used as polynomial and sigmoid, but the widely used one is the Gaussian kernel with the following form:

$$K(x_i, x_j) = \exp^{-q\left\|x_i - x_j\right\|^2}$$

The decision function is given by:

$$f(x) = R^2 = \left\|\phi(x) - a\right\|^2$$

$$= K(x, x) - 2\sum_{i=1}^{N} \beta_i K(x_i, x) + \sum_{i=1}^{N}\sum_{j=1}^{N} \beta_i \beta_j K(x_i, x_j) \quad (3)$$

B. SVC labeling (cluster assignment):

The obtained decision function 3 indicates only if the tested data point is inside one of the clusters or not, it does not differentiate between points that belong to different clusters [6]. To do so, a simple graphical direct connection test can be used to assign each sample to the appropriate cluster. For any two points x_i, x_j, and using the function (3), we check the m segmers on the line segment that connect their images in the hyperspace, if all the m segmers lies in the hypersphere, x_i, x_j should be labeled with the same cluster, otherwise, they will be assigned to two different clusters.

III. Most important current works on SVC

The decision function $f(x)$ (3) is considered as the backbone of the SVC method, since the whole clustering operation is strongly depending on its results. For that, almost of the optimization works on the literature are turning around it. Optimizing the computational time required to formulate $f(x)$ depends on solving its quadratic programming problem (QP). As well as, $f(x)$ is influenced by many parameters. Getting high accurate results is strongly depending on tuning those parameters. In addition, labeling computational time and connectivity graph storage requirements are with high dependency of the function $f(x)$.

As it is mentioned in the introduction, the main drawback of SVC method is the computational time requirement in both, training and labeling step. Thus, almost of the theoretical research works on the literature are focusing on minimizing time consumption with the preservation -or even improvement in some propositions- of the method's accuracy.

Ping et al. [4] have classified the theoretical contributions into three main classes: parameter selection and optimization, solving dual problem and improving cluster labeling methods. In what following, we will follow this classification on describing and discussing the most important works sited with some adaptations. We will use parameter tuning instead of approaches to parameter selection and optimization, and we will add the fourth class which is data space reduction. In addition, a miscellaneous class is proposed to group the researches that propose an uncommon ideas or reformulating the original proposed SVD method principals.

A. Data space reduction

Given that we are dealing with huge amount of data, reducing it or selecting the most relevant data points shown that it has a great effect on reducing computational time requirement. In fact, there is two possible position where we can reduce the data points used in the SVC method process.

1) Reducing data before training: According to [4], [8], [12] only a few data points are needed to define $f(x)$, these points are the SVs. So, the aim is to pinpoint these SVs and use only them to train $f(x)$. Many algorithms proposed to eliminate a large data point proportion and use only the remaining small part –which implicitly contains the SVs- to train the model.

Based on local geometrical and statistical information method proposed Y. Li et al. [13], Y. Ping et al. [14] proposed a border-edge pattern selection (BEPS) method to identify the boundaries points. Their algorithm tries to localize points with all of their neighbors or almost of them are locating on one side (upside or downside) of the tangent plane passing throw that points, depending on the curvature of the surface Fig. 3.

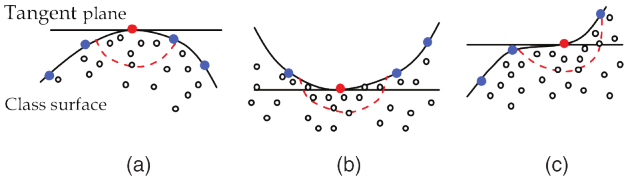


Fig. 3: border-edge pattern selection (BEPS). [14]

Applying BEPS result on a reduced data set, which significantly reduce the amount of training time. However, the necessity to set some parameters as: the thresh hold γ which is used to control the curvature of the surface, and test if a data point is a boundary point, and the number of neighbors *k*, can affect the accuracy of the method.

Authors in [15], suggested another method to eliminate unnecessary data, they proposed, as first step, to eliminate the noise data points using the shared nearest neighbor (SNN) algorithm, then identify and eliminate the core data points using the unit vectors concept and only keep the boundary

points to use in the training step. However, this method can't effectively detect and eliminate all core points. In addition, they need to define a thresh hold δ test the similarity between data points, and a constant *k* for minimum number of neighbors.

2) Reducing data before labeling: To avoid testing every pairwise data points and label them one by one, most of remarkable works in this axe of research are based on separating the data set into groups, usually known as convex hulls, then with each group, locating the most representative points that attract all remaining data points, generally named stable equilibrium points (SEPs) or stable equilibrium vectors (SEVs), Label the SEVs, and then each inner data point is labeled according to its SEVs.

Based on the topological property of the trained kernel radius function, J. Lee et al. [16] proposed a novel method to reduce computational time required in the labeling step in two phases. Firstly, they used a generalized gradient descent process to decompose data set into a small number of disjoint groups and locate the representative point (SEP) of each of these groups. Then in second step, label the obtained SEPs using the classical method but with a reduced graph (RG) instead of complete graph (CG), thus, label all data points of each group according to their SEP. Their method shows a great reduction in time consumption storage requirement. However, it suffers from the cluster convexity problem as shown in Fig.4a. This problem has been solved by J. Lee et al. [17] by introducing the basin cells and adjacency points notion. The authors proved that the original space can be decomposed to regions named basin cells (BS), and every basin cell contains a SEP and all the data point that converge to that SEP. The of intersection of these BS are adjacency points. These adjacency points are used in the labeling step as following: if d_i is an adjacency point between two SEPs s_j, s_k, and if $f(d_i) \leq r$, then $s_j and s_k$ are in the same cluster. Despite this improvement, the method still leads to a relatively high error on irregular-shaped data [12].

B. Solving dual problem

Although reducing data space can allow a great performance improvement, finding an alternative problem to the original dual problem attracted the main researchers' attention due its high computational complexity.

The first attempt to find an alternative problem to the SVC dual problem was in [6], authors proposed to use the sequential minimal optimization (SMO) algorithm. It decomposes the original dual problem into a series of small-scale convex quadratic programming problems. In each problem, only two samples are required as the working set. Thus, in order to obtain a globally optimal solution, the algorithm starts with two heuristically selected factors β while the others are fixed [4]. Despite the improvement that SMO guarantied, it still requires a high time complexity.

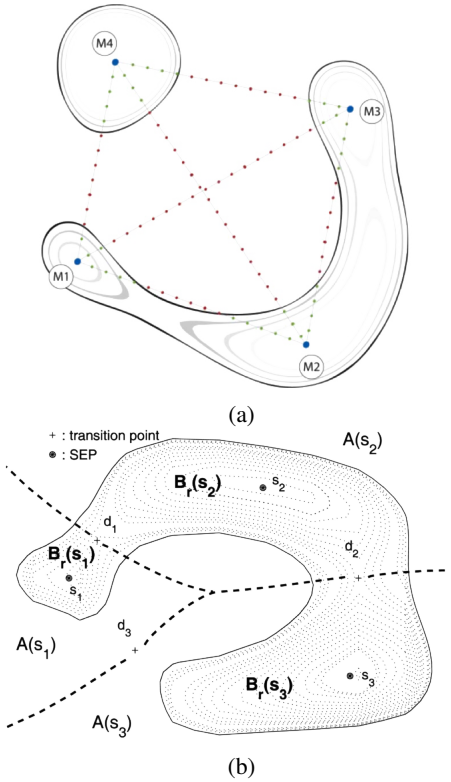


Fig. 4: (a):SEP and convexity problem [16], (b): BC, the solution of convexity problem [17]

C. Guo et al. [14] followed the Jaynes' maximum entropy principle and replaced the QP by an Iterative problem. Based on the problem constraints:

$$\sum_{j=1}^{N} \beta_j = 1$$

$$0 \leq \beta_j \leq 1$$

It meets the probability definition. Based on the probability interpretation that: the center of the sphere represents the mean vector of the images of all data points, and that β_i represents the probability that x_i is a SV so, the search of the MES can be considered as the probability assignment of β_i.

Another remarkable work done by J. H. Chiang et al. [18]. The authors used a fuzzy clustering principal which consider that a data point can belong to many clusters at the same time, with a certain membership grade to every cluster, instead of belonging to only one cluster. This definition is used in addition to some other principals (such as the nonlinear mapping, cell growing principal and points density region) to define a new algorithm (named MSV-clustering algorithm) which aims to map original data points into many hyper-spheres instead of a single one. When mapped back, every hyper-sphere forms a single cluster with a specified decision function. However, in addition to the need to set many parameters (cluster compactness, vigilance threshold,

etc.), the algorithm still shows high computation and storage complexity.

In the aim to make SVC consume less, Y. Ping et al. [22] proposed to use the dual coordinate descent method (DCD), which was firstly proposed by C.-J. Hsieh et al. [23] to solve large scale linear SVM. Firstly, they reformulated the original dual problem to a linear one similar to SVM, however it still an unsupervised model. Thus, they extended it to an iterative algorithm to compute the β_i coefficients. The resulting model shows a flexibility when dealing with storage resources. It offers two ways to compute and store kernel values, search on demand or calculate on demand, depending on the used platform capabilities. If the platform has sufficient storage resources, one can calculate and store the whole kernel matrix in the memory, then use the search on demand policy. Otherwise, we compute only the needed block of the kernel matrix on demand. This strategy can reduce the storage complexity of the SVC and allow individuals to use their limited platform resources to deal with large-scale data.

T. Pham et al. [9] applied the stochastic gradient descent (SGD) to the LMH version of the SVC. Their SGD-LMSVC algorithm iteratively train an optimal hyperplane. In each iteration t, it uniformly samples a single data point from the training data set to form a new hyperplane based on the updated information from the previous one.

$$w_{t+1} = (1 - \frac{1}{t})w_t + \frac{C}{t}\mathbb{I}_{[w_t^T \phi(x_{n_t}) \leq 1]}\phi(x_{n_t})$$

Where $\mathbb{I}_A$ is an indicator function, it returns 1 if *A* is true and 0 otherwise. This kind of solution allows its use in a dynamic mode and optimize memory usage. However, the use of the kernels can lead to a considerable model growth, which can slower the computation rate and cause a potential memory overflow. To overcome this issue, the same authors proposed in [27] to fix the size of the current model using the budget approach. If the current model size exceeds the budget, a maintenance procedure which includes two strategies is applied. A removal strategy, consist of removing the most redundant vector, and a projection strategy, which projects the most redundant vector onto the linear span of the remaining vectors in the support set in the feature space before removing it. The proposed update can limit the model growth, however, another parameter to optimize is added, which is the size of the budget. If the budget is too small a precious learning information can be lost, and a rise of computation rate and memory usage is expected if the budget is too big.

C. Improving cluster labeling

Cluster label assignment operation is based on pair-wise testing and graph construction using the decision function $f(x)$, which shown a considerable time and storage complexity. Thus, most of researches in the literature, who are dealing with this step, are interested on how reducing data points used in both, pair-wise tests and graph construction.

Methods concerned by data points reduction are presented in the section: reducing data points before labeling. In addition, some other methods proposed to solve dual problem have a transitive effect on the improvement of the cluster labeling step. J. H. Chiang et al. [18] method, discussed in solving dual problem section, leads to train many functions, one for each cluster, so, assigning a cluster label to a data point simply done by testing which function gives a positive result.

Ben-Hur et al. [6] proposed to use SVs in the graph construction (named SVG) instead of using whole original data points complete graph (CG), which led to computation and storage space improvement, but they need another step to assign the remaining inner data points into their appropriate clusters. L. Ping et al. [19] presented a novel method named NSVC, in which they proposed to label the remaining inner points using spectrum analysis (SA), and a weighted-voting kNN (WkNN). However, according to [4], the spectrum analysis is time consuming, especially in high dimension data sets. Y. Ping et al. [12] used the SVs to decompose the data space into separated and non-overlapped subsets which contains most of the inner data points, every subset represents a so-called convex hull, which is considered as a prototype for connectivity analysis. Contrary to J. Lee et al. [17] which used the representative data points of every basin cell to construct the graph, authors in [12] used the line segment connecting two nearest neighboring convex hulls, (from the vertex of one convex hull which is the nearest to the other convex hull to any point on the corresponding border [12]). Y. Ping et al. [22] also used the convex hull decomposition and the most representative point of the sub-cluster named stable equilibrium vectors (SEVs), however, their algorithm checks whether any two convex hulls belongs to same cluster by checking the connectivity between the SEV of one sub cluster and the nearest SV of the other sub cluster. The decision is made using sample once for connection checking first strategy, explained in sample rate tuning section.

D. Parameter tuning

The aim of training $f(x)$ step is to estimate the β_i coefficients. However, and as seen in the previous section, $f(x)$ results depends also on other parameters, such as: the kernel width q, the penalty factor C and the sample rate m -which has an influence on the labeling step-. Thus, many researches in the literature are interested on tuning those parameters due to their great influence on the clustering results. Following, we will discuss the influence of every parameter, and some important researchers work done about that parameter.

1) Kernel width q: The selected kernel function defines how the data points will be mapped to the new hyperspace. As well as, the most used one is the Gaussian kernel, the clusters boundary shape depends on the selected q value. In addition, and as we have seen previously, some of the boundary data points known as SVs, are with direct responsibility of defining the function $f(x)$, thus, increasing or decreasing the number of the SVs will obviously affect the cluster's boundary shape. Ben-Hur et al. [6] showed that the kernel width q is on direct

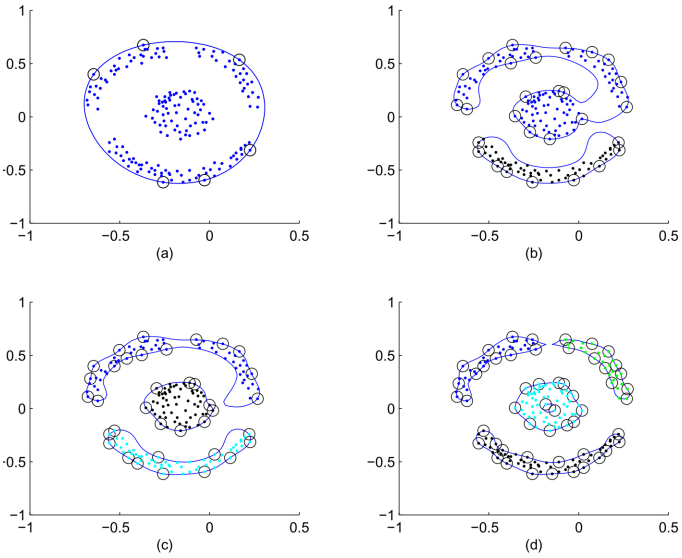


Fig. 5: Parameters tuning influence on clustering results [6].

relation with the SVs. The authors proposed to begin with small q value, equal to $1/\max_{i,j}\left\|x_i - x_j\right\|^2$, which leads to generate one cluster englobing all of the data points, then incrementally increasing it into obtaining the desired cluster split, Fig.5. S.-H. Lee et al. [20] characterized R^2 as a function of q, and established that: $0 \leq R^2 \leq 1 - 1/N$ for $0 \leq q \leq \infty$, $R^2 = 0$ for $q = 0$, $R^2 = 1 - 1/N$ if and only if $q = \infty$. Also, they provided that R^2 is monotonically increasing function of q. With a fixed C value, their secant-like algorithm generates a number of q values. The previous methods show an efficient way to set the kernel width q. However, it is too difficult to apply these technics, and define when to stop increasing q on an unsupervised clustering.

D. Huand et al. [24] discussed the problem of setting the kernel and the trade-off parameters (q and C) manually using trial and error strategy, and how to properly select them. They proposed an algorithm named ensemble-driven support vector clustering (EDSVC) which can automatically compute the parameters values based on ensemble learning strategy. Authors proposed to construct an ensemble of m clustering using k-means (they used $m = 10$ and $k \in [2, \sqrt[3]{N}]$). Then they proposed the following steps: set an ensemble of candidate values for each $q(n_q)$ and $C(n_c)$ (they used 100 candidates for each), vary one parameter while fixing the value of the other parameter and train an SVC model, and vice versa. Compute the average normalized mutual information (ANMI) between each trained SVC model and the constructed ensemble of m clustering. The parameters of the SVC cluster with maximum ANMI are adopted. EDSVC showed high performance on selecting parameter values. However, it shows huge computation consumption and still depends on other parameters like k, m, n_q and n_c. In addition, and according to [26], the algorithms based on k-means unstable and strongly

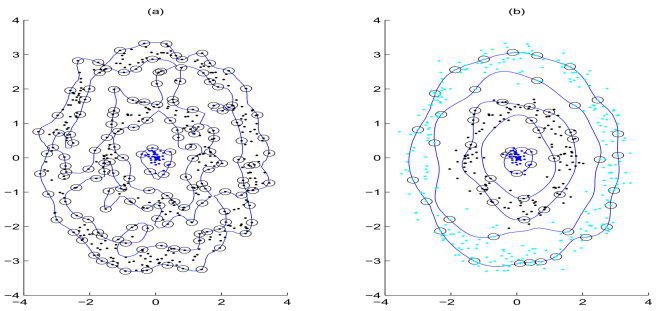


Fig. 6: Penalty factor influence [6].

depend on the initial labels.

2) Penalty factor C (trade-off parameter): From the fact that working on real data sets, and trying to divide them into separated clusters is usually not allowed even with different q values, Ben-Hur et al [6] confirmed the necessity of allowing the BSVs for cluster separation, they concluded that the number of outliers is monotonically decreasing function of C as following (where n_{bsv}):

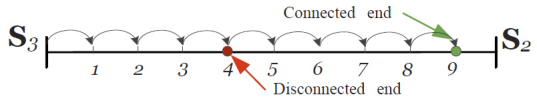
$$n_{bsv} \leq 1/C$$

In case of a large portion of data points turns into SVs or singleton clusters appears, authors in [6] warns that is the indication to consider BSVs, to allow points turns into outliers allowing contour separation. Also, authors noted that increasing or decreasing the penalty factor C does not affect only the number of BSVs, but also affects the number of SVs, and as result it affects the shape of the cluster. Same issues with defining optimal q value, it is difficult to set the optimal value of the factor C when dealing with an unsupervised clustering. Thus, the number of support vectors and the shape of the clusters depends on both q and m.

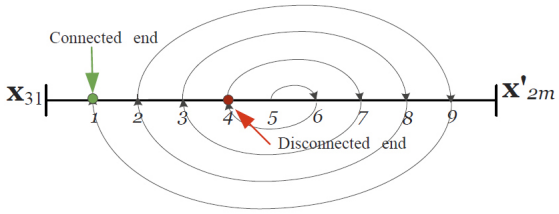
To tackle manual affectation of the penalty factor C, F. Pu [25], and in order to reflect the within class importance, he assigned an exponential local weight factor to each data point based on its image on the feature space and the center of the feature space images. The author reformulated the original dual problem to a new one, which suppress the effect of the outliers, represent by the penalty factor C, and replace them by weight factor.

3) Sample rate m: Another important parameter to tune, the sample rate m. It does not affect the cluster shape as the previous two parameters, but it has a great influence on the labeling step. However, not a lot of interest in the literature about this parameter.

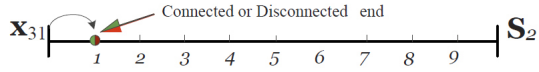
The sample rate m is crucial on determining the connection between components. Increasing m allows high labeling accuracy but leads to the increase on computational complexity. Decreasing m leads to incorrect labeling results but minimize the computational time. According to the literature's, the preferred sample rate m range is from 10 to 20 [4], [6]. In addition, we differentiate three strategies on how checking the m segmers. The first one is a linear sampling, it stops



(a) Linear sample sequence of traditional methods



(b) Nonlinear sample sequence for disconnection first



(c) Sample once for connection checking first

Fig. 7: Sampling strategies [22].

when either the number of tested points reach m or a negative check result is obtained [6]. Authors in [12], [21], used the disconnected checking first strategy, which uses a nonlinear sample sequence. The later one shows a reduction on the sample rate -less than two-.

As well as they used the convex hull decomposition, and specific connection checking between sub clusters, described in section III-C, Y. Ping et al. [22] proposed a new and very reduced sampling method, they only test the first segmer which connect the SEV of the first sub cluster to the closest SV of the second sub-cluster Fig.7.c. The decision is made if and only if $f(x_{(tested\_segmer)}) \leq R^2$ which means the two sub clusters are in the same cluster. Otherwise, a second inversed test is done between SV of the first sub cluster and the SEV of the second sub cluster. If $f(x_{(tested\_segmer)}) \leq R^2$ then the two sub cluster are in the same cluster, else they are a different clusters. Although the proposed method showed a great reduction in time consumption, it can fail when dealing with a very complicated cluster shapes.

E. Miscellaneous

In this section, we will present some works which applied a novel ideas or propose extensions to the original SVC algorithm, these ideas can't be classified in any of the previous sections. Almost all aforementioned researches have totally omitted the outlier data points, R. Saltos et al. [11] consider the closest BSVs to the formed clusters can contain some insights which can be exploited. They proposed the rough fuzzy support vector clustering (RFSVC) algorithm, which undergoes three steps: training, labeling and fuzzification. The last step, uses the fuzzy set theory to construct a fuzzy matrix. Each element of the matrix is a Gaussian distance from the i^{th} BSV to either the closest SV, given by:

$$\mu_{i,j} = \mu(x_i, SV_j) = k(x_i, SV_j) = \exp^{q\left\|x_i, SV_j\right\|^2}$$

or the mean distance to all SVs of the j^th cluster, given by:

$$\mu_{i,j} = \frac{1}{|SV_j|} \sum_{x \in SV_j} k(BSV_i, x_k)$$

Based on the fact that almost all real-world phenomena are characterized by dynamicity, where their data structures changes over time, and believing that changes leads to uncertainty, R. Saltos et al. [28] introduced a dynamic aspect to the original static RFSVC algorithm [11] to dynamically cluster moving data sets. As well as the technics for uncertainty modeling have been integrated successfully into dynamic clustering algorithms, authors in [28] combined the fuzzy logic and rough sets with the SVC to obtain the dynamic rough-fuzzy support vector clustering (D-RFSVC) algorithm. The fuzzy logic and rough sets are integrated to model the uncertainty aspect of the dynamic data, by providing a membership degree of data points to the found clusters, in the aim to trace their evolution over time. However, the SVC is chosen because of its ability to deal with outliers and represent clusters with complex shapes as in real world. The D-RFSVC undergoes three more steps in addition to the basic steps of RFSVC algorithm, which are training update, labeling update and fuzziffication update when a new data point is present. Training update, step consist of determine a feasible solution to the new system, then optimize the obtained feasible solution. In this step, some of inner data points can become SVs or the inverse. Labeling update step, updates the adjacency matrix and data points labels. If some changes occurred on the SVs set, the algorithm updates the membership matrix in the last step. According to [28], the application of the Dynamic RFSVC can occur the following changes on each update cycle of the model: creation, deletion, movement, merging, splitting, change of shape, dilatation, and contraction of the cluster, in addition to the change of uncertainty level and outlier traceability.

R. Khemchandani et al [29] proposed a novel plan-based binary classifier, the twin support vector machine (TWSVM), it solves two related smaller-size QPPs instead of a single large one as in a classical SVM to construct two nonparallel separating hyperplane. Z. Wang et al. [26] proposed the unsupervised version of TWSVM for clustering purpose named twin support vector clustering (TWSVC). The new proposed clustering algorithm iteratively train a plane for each cluster, the obtained plan is close to data points of its own cluster and far away from the data points of the other clusters from both sides as showed in Fig.8. The TWSVC uses a reduced data set in the training step, which results on a considerable computational complexity reduction. However, it omits one of the important advantages of the SVC method, which is the automatic detection of the cluster's number. In addition, as TWSVC began the training step by an initial cluster labels, the final results may depend on the first cluster initialization.

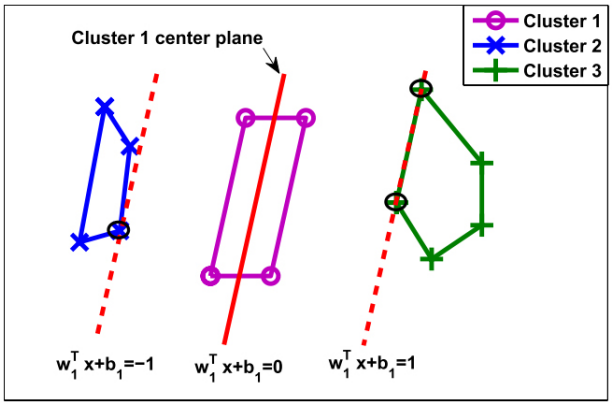


Fig. 8: Geometric interpretation of TWSVC [26].

Several works are proposed based on the new TWSVC algorithm. R. Khemchandani et al. [30] introduced their fuzzy least square version of the TWSVC algorithm named F-LS-TWSVC. The least squares principle is used to redefine the solution of TWSVC, which requires solving QPP and a system of linear equations, to solve a series of system of linear equations [30]. The fuzzy principal is applied by defining a membership degree matrix of every data point to different available clusters. The matrix is initialized using the fuzzy nearest neighbor algorithm, and it is updated in each iteration. Experiment results in [30] showed a considerable computational time reduction. The accuracy was maintained or even enhanced in some cases compared with the original TWSVC or the classical plane-based clustering (*PC*) methods (*k-PC*, *proximal PC*).

In order to alleviate the effect of the outliers, Q. Ye et al. [31] suggested to use the L1-norm in the distances computation instead of using L2-norm, which make their first algorithm (robust TWSVC) more robust to outliers' effect. They also proposed an effective iterative algorithm to accelerate their first version algorithm and upgrade it to be fast robust TWSVC (FRTWSVC). The experiments showed that booth RTWSVC and FRTWSVC are generally more accurate against TWSVC and classical plane-based clustering (k-PC, PPC). However, RTWSVC is a high time consuming algorithm compared to other methods, which let the FRTWSVC on the lead followed by TWSVC.

Motivated by the idea of the TWSVM, and based on maximum margin clustering (MMC) algorithm [33]. J. Fang et al. [32] developed their version of TWSVC. They proposed to reformulate the original MMC optimization problem (which is a non-convex) to a semi-definite (SDP) programming problem, then decompose it into several smaller ones and performing an alternative optimization. To improve the algorithm generalization ability, authors integrated the structural risk minimization (SRM). Authors proved that the obtained algorithm until this phase (alternating twin bounded SVC (ATBSVC)) can suffer from the premature convergence problem. Hence, they relaxed it by replacing the hinge loss by the Laplacian loss, the algorithm becomes: alternating relaxed twin bounded SVC (ARTBSVC).

Another algorithm based non-parallel hyperplanes is proposed by J. Fang et al. [34]. They designed a synchronized feature selection process based on the non-parallel hyperplane SVM (NHSVM) instead of TWSVC, because the first is a single QPP which is more adequate. However, an iterative optimization strategy is also adopted to allow the NHSVM dealing with the unsupervised learning (clustering). In this alternating work, authors also replaced the hinge loss with Laplacian one to avoid premature convergence, and the L-infinite penalization norm is imposed to both hyperplanes for the feature elimination. It's important to note that the proposed algorithm in [34] (Iterative tighter non-parallel hyperplane support vector clustering with simultaneous feature selection (IT-NHSVC-SFS)) is applied for binary clustering only, and generalization to multi-class clustering is suggested by the authors.

IV. Conclusion

We highlighted several SCV research papers, and we classified them according to the studied SVC section onto four classes, data space reduction, solving dual problem, improving cluster labelling and parameter tuning. As we mentioned in the introduction, the main issue of the SVC method is when dealing with huge data sets, thus the most important classes are data space reduction, solving dual problem. Also, we can notice that almost of the works are based on mathematics, in another hand the AI methods and some of newly computer science methods are ignored as cloud computing.

References

[1] Han, Jiawei, Jian Pei, and Micheline Kamber. Data mining: concepts and techniques. Elsevier, 2011.
[2] Yongjian, Fu. "Data mining: tasks, techniques and applications." IEEE Potentials 16.4 (1997): 18-20.
[3] Estivill-Castro, Vladimir. "Why so many clustering algorithms: a position paper." ACM SIGKDD explorations newsletter 4.1 (2002): 65-75.
[4] H. Li and Y. Ping, "Recent advances in support vector clustering: Theory and applications," Int. J. Pattern Recognit. Artif. Intell., vol. 29, no. 01, p. 1550002, 2015.
[5] A. K. Jain, "Data clustering: 50 years beyond K-means," Pattern Recognit. Lett., vol. 31, no. 8, pp. 651–666, 2010.
[6] Ben-Hur, Asa, et al. "Support vector clustering." Journal of machine learning research 2.Dec (2001): 125-137.
[7] Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." Machine learning 20.3 (1995): 273-297.
[8] D. M. J. Tax and R. P. W. Duin, "Support vector domain description," Pattern Recognit. Lett., vol. 20, no. 11–13, pp. 1191–1199, 1999.
[9] T. Pham, H. Dang, T. Le, and H. Le, "Stochastic Gradient Descent Support Vector Clustering," pp. 88–93, 2015.
[10] B. Sch, J. C. Platt, J. Shawe-taylor, A. J. Smola, and R. C. Williamson, "Estimating the Support of a High-Dimensional Distribution," vol. 1471, pp. 1443–1471, 2001.
[11] R. Saltos and R. Weber, "A Rough-Fuzzy approach for Support Vector Clustering," Inf. Sci. (Ny)., vol. 339, pp. 353–368, 2016.
[12] Y. Ping, Y. F. Chang, Y. Zhou, Y. J. Tian, Y. X. Yang, and Z. Zhang, "Fast and scalable support vector clustering for large-scale data analysis," Knowl. Inf. Syst., vol. 43, no. 2, pp. 281–310, 2015.
[13] Y. Li, S. Member, and L. Maguire, "Selecting Critical Patterns Based on Local Geometrical and Statistical Information," vol. 33, no. 6, pp. 1189–1201, 2011.
[14] C. Guo and F. Li, "An improved algorithm for support vector clustering based on maximum entropy principle and kernel matrix," Expert Syst. Appl., vol. 38, no. 7, pp. 8138–8143, 2011.

[15] Wang, Jeen-Shing, and Jen-Chieh Chiang. "An Efficient Data Preprocessing Procedure for Support Vector Clustering." J. UCS 15.4 (2009): 705-721.
[16] J. Lee and D. Lee, "An improved cluster labeling method for support vector clustering," IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 3, pp. 461–464, 2005.
[17] J. Lee and D. Lee, "Dynamic characterization of cluster structures for robust and inductive support vector clustering," IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 11, pp. 1869–1874, 2006.
[18] J. H. Chiang and P. Y. Hao, "A new kernel-based fuzzy clustering approach: Support vector clustering with cell growing," IEEE Trans. Fuzzy Syst., vol. 11, no. 4, pp. 518–527, 2003.
[19] L. Ping, G. Dajin, H. Fujiang, R. Xiangsheng, and Y. Xiangyang, "Novel support vector clustering with label assignment in enriched neighborhood," 6th Int. Conf. Fuzzy Syst. Knowl. Discov. FSKD 2009, vol. 1, no. 2, pp. 500–504, 2009.
[20] S.-H. Lee and K. Daniels, "Gaussian Kernel Width Generator for Support Vector Clustering," Adv. Bioinforma. Its Appl. - Proc. Int. Conf., pp. 151–162, 2005.
[21] Y. Ping, Y. J. Tian, Y. J. Zhou, and Y. X. Yang, "Convex decomposition based cluster labeling method for support vector clustering," J. Comput. Sci. Technol., vol. 27, no. 2, pp. 428–442, 2012.
[22] Y. Ping, Y. Tian, C. Guo, B. Wang, and Y. Yang, "FRSVC: Towards making support vector clustering consume less," Pattern Recognit., vol. 69, pp. 286–298, 2017.
[23] C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. S. Keerthi, S. Sundararajan, A dual coordinate descent method for large-scale linear svm, in: Proceedings of the 25th International Conference on Machine Learning (ICML '08), ACM, 2008, pp. 408–415.
[24] D. Huang, C. Wang, J. Lai, Y. Liang, S. Bian, and Y. Chen, "Ensemble-Driven Support Vector Clustering: From Ensemble Learning to Automatic Parameter Estimation," no. 978, pp. 444–449, 2016.
[25] F. Pu, "Locally Weighted Support Vector Clustering," 2017.
[26] Z. Wang, Y. Shao, L. Bai, and N. Deng, "Twin Support Vector Machine for Clustering," no. 4, pp. 1–6, 2015.
[27] T. Pham, T. Le, and H. Dang, "Scalable Support Vector Clustering Using Budget," arXiv Prepr. arXiv1709.06444, 2017.
[28] R. Saltos, R. Weber, and S. M. Ieee, "Dynamic Rough-Fuzzy Support Vector Clustering," vol. 6706, no. c, pp. 1–14, 2017.
[29] R. Khemchandani, S. Member, and S. Chandra, "Twin Support Vector Machines for Pattern Classification," vol. 29, no. 5, pp. 905–910, 2007.
[30] R. Khemchandani, A. Pal, and S. Chandra, "Fuzzy least squares twin support vector clustering," Neural Comput. Appl., 2016.
[31] Q. Ye et al., "L1-Norm Distance Minimization-Based Fast Robust Twin Support Vector k -Plane Clustering," pp. 1–10, 2017.
[32] J. Fang, Q. Liu, and Z. Qin, "Alternating Relaxed Twin Bounded Support Vector Clustering," Wirel. Pers. Commun., 2017.
[33] Xu, L., Neufeld, J., Larson, B., & Schuurmans, D. (2004). Maximum margin clustering. In NIPS. 7. Valizadegan, H., & Jin, R. (2007). Generalized maximum margin clustering and unsupervised kernel learning. In NIPS, pp. 1417–1424.
[34] J. Fang, Q. Liu, and Z. Qin, "Iterative tighter nonparallel hyperplane support vector clustering with simultaneous feature selection," Cluster Comput., 2017.