2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Finding Convex Hull Vertices in Metric Space

Jinhong Zhong
USTC-Birmingham Joint Research Institute in Intelligent Computation and Its Applications (UBRI)
School of Computer Science and Technology of USTC
Hefei, China
Email: jinhong@mail.ustc.edu.cn

Ke Tang
USTC-Birmingham Joint Research Institute in Intelligent Computation and Its Applications (UBRI)
School of Computer Science and Technology of USTC
Hefei, China
Email: ketang@ustc.edu.cn

A. K. Qin
School of Computer Science and Information Technology
Royal Melbourne Institute of Technology
Melbourne, Australia
Email: kai.qin@rmit.edu.au

***Abstract*—The convex hull has been extensively studied in computational geometry and its applications have spread over an impressive number of fields. How to find the convex hull is an important and challenging problem. Although many algorithms had been proposed for that, most of them can only tackle the problem in two or three dimensions and the biggest issue is that those algorithms rely on the samples' coordinates to find the convex hull. In this paper, we propose an approximation algorithm named FVDM, which only utilizes the information of the samples' distance matrix to find the convex hull. Experiments demonstrate that FVDM can effectively identify the vertices of the convex hull.**

Keywords—convex hull; metric space

I. INTRODUCTION

Convex hull plays a crucial role in many mathematics and computational geometry problems [3]. Its practical applications can be widely found in pattern recognition, image processing, statistics and static code analysis by abstract interpretation [15][19-21]. It also serves as a major building block for a number of computational-geometric algorithms such as the rotating calipers method for computing the width and diameter of a point-set. In particular, it is well known that there is a strong link between Support Vector Machine (SVM) and convex hull. Training an SVM on a separable data set is equivalent to the problem of computing the nearest points between the convex hulls formed by the positive and negative samples [5-7]. Hence, quite a few of competitive SVM training algorithms had been developed based on such equivalence [16-17].

Over the past decades, many efforts have been devoted to develop convex hull algorithms for the planar point-set. In 1970, Chand and Kapur [8] initially proposed a convex hull algorithm with O(n2) time complexity by constructing the borders of convex hull according to the geometric properties of a point-set. Another algorithm with O(mn) time complexity was developed by Jarvis [9], where m is the number of convex hull vertices. Graham [4] provided a solution to compute the convex hull of a finite linear set in a 2D plane. Adopting the dichotomy method, many convex hull algorithms had been proposed. In [11], a divide-and-conquer method was proposed for convex hull extraction. In this method, the point-set was divided into two roughly equal-sized subsets. Their convex hulls were recursively computed respectively, and the entire convex hull was obtained by merging those two convex hulls. In another study, Chan [12] used pairs of points to calculate the line slopes and determine the median values of these slopes, then divided the point-set into two parts by median values and recursively computed the convex hull. He proposed another algorithm [22] that partitions the point-set and then computes the convex hull of each group, respectively. The entire convex hull was finally obtained by computing the union of the polygons. Similarly, Quickhull [3] used a divide and conquer approach similar to that of the quick sort. Because few convex hull algorithms remain effective in the high dimensional space and all of these algorithms are computationally expensive, many approximate algorithms had been proposed to address these issues. For example, Convex Hull Vertices Selection (CHVS) Algorithm proposed by Wang et al. [14]. To reduce the intermediate storage used in the computation of planar convex hulls, Brönnimann et al. [13] investigated the storage space of some convex hull algorithms.

Although many algorithms have been proposed for identifying the convex hull of a dataset (point-set). All those algorithms implicitly assume that the samples in the dataset lie in a finite dimensional space, and the samples' values in each dimension are known and will be fed into the algorithms as inputs. However, in some situation, information about the samples' values in each dimension is unknown, or the dataset may not even be provided as a set of points in a multi-dimensional space. For example, in most kernel-based methods, e.g., SVM, the features of the kernel space are not really known when a nonlinear kernel function is used. Moreover, only similarities/dissimilarities between data points are available in some real-world applications, e.g., Cooper et al. extracted summary excerpts from audio and video using the similarity matrix [25]. To the best of our knowledge, no existing algorithm can effectively identify the convex hull of a dataset in these situations.

In this paper, we propose an efficient approximation algorithm, called Finding Vertices with Distance Matrix (FVDM), to seek for the convex hull using merely the similarity information (e.g., a similarity matrix or distance matrix) between data points. The proposed method is based on two theorems about the convex hull. Each theorem provides a necessary and sufficient condition of a point being a convex hull vertex. The effectiveness of FVDM is demonstrated by comparing it against a naïve approach having high computational complexity on both synthetic and real-world data.

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

The rest of this paper is organized as follows. Section II presents the definition of the convex hull and proves two necessary and sufficient conditions of a point being a convex hull vertex. The FVDM algorithm is introduced in Section III. In Section IV, experiments are conducted on synthetic and real-world data to illustrate the effectiveness and efficiency of the proposed method. Conclusions and discussions are given in Section V.

II. ANALYZING CONVEX HULL IN METRIC SPACE

The convex hull of a set of points is the smallest convex set that contains all the points [1]. From a computational geometry's point of view, an object in the Euclidean space is convex if every pair of points falls inside the object, i.e., any point along the straight line segment connecting every point pair is within the object [2]. A set S is convex if for any $\boldsymbol{x}, \boldsymbol{y} \in S$ and any $t \in [0,1]$, the point $(1-t)\boldsymbol{x} + t\boldsymbol{y}$ belongs to S. Moreover, if S is a convex set, for any $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_r \in S$ and any non-negative numbers $\{\lambda_1, \lambda_2, \ldots \lambda_r\}$ with $\sum_{i=1}^{r} \lambda_i = 1$, the vector $\sum_{i=1}^{r} \lambda_i \boldsymbol{x}_i$ is called the convex combination of $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_r$ and $\sum_{i=1}^{r} \lambda_i \boldsymbol{x}_i$ belongs to S. According to the above definitions, the convex hull can be defined in terms of convex sets or convex combinations, so that the convex hull S_X of a point-set X in the Euclidean space can be defined as:

- the minimal convex set containing X, or
- the intersection of all convex sets containing X. or
- the set of all convex combinations of points in X.

From the definition of the convex hull, the following two theorems can be derived.

Theorem 1: *In R^n, for any convex hull E and any point $\boldsymbol{x} \in E$, there exists a point $\boldsymbol{y} \in R^n$ such that $\|\boldsymbol{x}-\boldsymbol{y}\|^2 = \max_{\boldsymbol{z}\in E} \|\boldsymbol{z}-\boldsymbol{y}\|^2$ is a necessary and sufficient condition for $\boldsymbol{x} \in \partial E$. ∂E is the boundary of E.*

Proof:

a. Necessity

As shown in Fig 1, for any boundary point $\boldsymbol{x}$ of E, we shall prove that $\exists \boldsymbol{y} \in R^n, s.t. \|\boldsymbol{x}-\boldsymbol{y}\|^2 = \max_{\boldsymbol{z}\in E} \|\boldsymbol{z}-\boldsymbol{y}\|^2$.

We construct a hyperplane P at $\boldsymbol{x}$, $s.t. P \cap E = \boldsymbol{x}$. Let $\vec{n}$ denote the hyperplane's normal vector that points to the inside of convex hull E and let $\boldsymbol{y} = \boldsymbol{x} + b\vec{n}$ $(b > 0)$. Any point $\boldsymbol{z}$ in E can be represent by $\boldsymbol{z} = \boldsymbol{X}_0\boldsymbol{\alpha}$, here $\boldsymbol{X}_0 = [\boldsymbol{x}_1\ \boldsymbol{x}_2\ \cdots\ \boldsymbol{x}_m]$, and $\boldsymbol{x}_j$ ($1 \le j \le m$) are all the boundary points. And $\boldsymbol{\alpha} = [\alpha_1\ \alpha_2\ \cdots\ \alpha_m]^T$, in which $\boldsymbol{\alpha}$ satisfies $\sum_{j=1}^{m} \alpha_j = 1$, $\alpha_j \ge 0$ $(1 \le j \le m)$.

So,

$$\begin{aligned}\|\boldsymbol{z}-\boldsymbol{y}\|^2 &= \|\boldsymbol{X}_0\boldsymbol{\alpha} - \boldsymbol{x} - b\vec{n}\|^2 \\ &= b^2 - 2b(\boldsymbol{X}_0\boldsymbol{\alpha} - \boldsymbol{x})^T \cdot \vec{n} + \boldsymbol{\alpha}^T\boldsymbol{X}_0^T\boldsymbol{X}_0\boldsymbol{\alpha} \qquad (1)\end{aligned}$$

$\because$ $\overrightarrow{\boldsymbol{xz}}$ and $\vec{n}$ point to the same side of P.

$\therefore$ $(X_0\boldsymbol{\alpha} - \boldsymbol{x})^T \cdot \vec{n} \ge 0$, the equality holds if and only if $\boldsymbol{X}_0\boldsymbol{\alpha} - \boldsymbol{x} = 0$.

From (1), we can know that if b is large enough, for all $\boldsymbol{\alpha}$, $\|\boldsymbol{z}-\boldsymbol{y}\|^2$ would reach its maximum if and only if $\vec{n} \cdot (\boldsymbol{X}_0\boldsymbol{\alpha} - \boldsymbol{x}) = 0$.

So when $\|\boldsymbol{z}-\boldsymbol{y}\|^2$ reach its maximum, $\boldsymbol{X}_0\boldsymbol{\alpha} - \boldsymbol{x} = 0$.

That is $\boldsymbol{X}_0\boldsymbol{\alpha} = \boldsymbol{x}$, and that is $\|\boldsymbol{x}-\boldsymbol{y}\|^2 = \max_{\boldsymbol{z}\in E} \|\boldsymbol{z}-\boldsymbol{y}\|^2$.

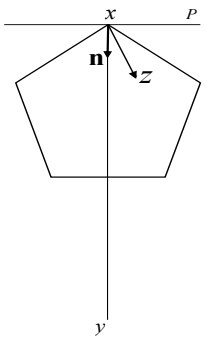


Fig 1

b. Sufficiency

Proof by contradiction, we suppose exist points $\boldsymbol{x}$ and $\boldsymbol{y}$ subjected to $\|\boldsymbol{x}-\boldsymbol{y}\|^2 = \max_{\boldsymbol{z}\in E} \|\boldsymbol{z}-\boldsymbol{y}\|^2$ while $\boldsymbol{x} \notin \partial E$. As shown in Fig 2, we could connect $\boldsymbol{y}$ and $\boldsymbol{x}$ and extent to any point $\boldsymbol{w}$ in E. Clearly, $\|\boldsymbol{x}-\boldsymbol{y}\|^2 < \|\boldsymbol{w}-b\|^2$, which contradicts with $\|\boldsymbol{x}-\boldsymbol{y}\|^2 = \max_{\boldsymbol{z}\in E} \|\boldsymbol{z}-\boldsymbol{y}\|^2$, so the assumption does not hold.

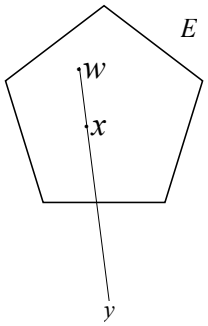


Fig 2

The theorem is thus proved.

□

Theorem 2. *In R^n, for any convex hull E and any point $\boldsymbol{x} \in E$, there exists a point $\boldsymbol{y} \in R^n \backslash E$, subjected to $\|\boldsymbol{x}-\boldsymbol{y}\|^2 = \min_{\boldsymbol{z}\in E} \|\boldsymbol{z}-\boldsymbol{y}\|^2$ is a necessary and sufficient condition for $\boldsymbol{x} \in \partial E$.*

Proof:

a. Necessity

As shown in Fig 3, for any boundary point $\boldsymbol{x}$ of E, then, we shall prove that $\exists \boldsymbol{y} \in R^n \backslash E, s.t. \|\boldsymbol{x}-\boldsymbol{y}\|^2 = \min_{\boldsymbol{z}\in E} \|\boldsymbol{z}-\boldsymbol{y}\|^2$.

We construct a hyperplane P at $\boldsymbol{x}$, $s.t. P \cap E = \boldsymbol{x}$. $\vec{n}$ is the hyperplane's normal vector which points to the inside of convex hull E. Set $\boldsymbol{y} = \boldsymbol{x} - b\vec{n}$ (b>0) . For any point $\boldsymbol{z}$ in E,

$$\begin{aligned}\|\boldsymbol{z}-\boldsymbol{y}\|^2 &= (\mathbf{z}-\mathbf{y}) \cdot (\mathbf{z}-\mathbf{y}) \\ &= (\mathbf{z}-\mathbf{x}+\mathrm{b}\vec{n}) \cdot (\mathbf{z}-\mathbf{x}+\mathrm{b}\vec{n}) \\ &= \|\boldsymbol{z}-\boldsymbol{x}\|^2 + b^2 + 2b(\boldsymbol{z}-\boldsymbol{x}) \cdot \vec{n}\end{aligned}$$

$\because$ $\overrightarrow{\boldsymbol{xz}}$ and $\vec{n}$ point to the same side of P.

$\therefore$ $(\boldsymbol{z}-\boldsymbol{x}) \cdot \vec{n} \ge 0$

$\therefore$ $\|\boldsymbol{z}-\boldsymbol{y}\|^2 \ge \|\boldsymbol{z}-\boldsymbol{x}\|^2 + b^2 \ge b^2 = \|\boldsymbol{x}-\boldsymbol{y}\|^2$

$\therefore$ $\exists \boldsymbol{y} \in R^n, s.t. \|\boldsymbol{x}-\boldsymbol{y}\|^2 = \min_{\boldsymbol{z}\in E} \|\boldsymbol{z}-\boldsymbol{y}\|^2$

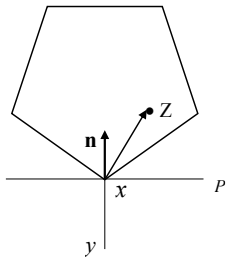


Fig 3

b. Sufficiency

Proof by contradiction. Suppose exist points $\boldsymbol{x}$ and $\boldsymbol{y}$ subjected to $\|\boldsymbol{x}-\boldsymbol{y}\|^2 = \min_{\boldsymbol{z}\in E}\|\boldsymbol{z}-\boldsymbol{y}\|^2$ but $\boldsymbol{x} \notin \partial E$.As shown in Fig 4. we could connect $\boldsymbol{y}$ and $\boldsymbol{x}$. As $\boldsymbol{x} \in E\backslash\partial E$, then the line $\overline{\boldsymbol{xy}}$ and ∂E would intersect at **W**. Clearly. $\|\boldsymbol{x}-\boldsymbol{y}\|^2 > \|\boldsymbol{w}-\boldsymbol{y}\|^2$, which contradicts with $\|\boldsymbol{x}-\boldsymbol{y}\|^2 = \min_{\boldsymbol{z}\in E}\|\boldsymbol{z}-\boldsymbol{y}\|^2$, so the assumption does not hold.

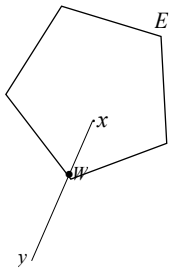


Fig 4

The theorem is thus proved.

□

III. ALGORITHMS OF FINDING THE CONVEX HULL VERTICES

In this section, two algorithms for finding the convex hull vertices based on the similarity information between data points are presented. Firstly, the FVDM algorithm will be presented. After that, an intuitive approach, namely FVSVM will be described. FVSVM can be used to identify the convex hull using a special case of the similarity information, i.e., the kernel matrix, and will be used to gauge the effectiveness of FVDM.

A. Finding the convex hull vertices with distance matrix

Suppose we need to identify the vertices of the convex hull of a given dataset X. The two theorems presented in Section 2 can be used to determine whether a data point **x** is a vertex of the convex hull. For each data point **x** in X, if a reference data point **y**, which does not necessarily belong to X, satisfies the condition given by either Theorem 1 or Theorem 2, **x** can be identified as a vertex of the convex hull of X. Theoretically, it is impossible to enumerate all possible **y** to check whether either of the two theorems holds. In practice, however, we can use a finite number of reference points to assess how likely the two theorems hold for a given **x**. In this case, the set of reference points may consist of not only all data in X but also any other data that are provided together with X or even some randomly generated synthetic data. Based on these considerations, we proposed the FVDM algorithm, and its pseudo-code is given in Algorithm 1.

Algorithm 1

Algorithm FVDM

Input: D , distance matrix; X, the point-set; O, the point set out*side* the convex hull

Output: V, the set of convex hull vertices

1. Initial $V=\emptyset$;
2. for every point $\boldsymbol{x} \in X$
3. if there is $\boldsymbol{y} \in X \cup O$,s.t. d(**x**, **y**)=$\max_{\boldsymbol{z}\in I} d(\boldsymbol{z},\boldsymbol{y})$
4. $V = V \cup \{\boldsymbol{x}\}$;
5. elseif there is $\boldsymbol{y}\in O$ s.t. d(**x**, **y**)=$\min_{\boldsymbol{z}\in I} d(\boldsymbol{z},\boldsymbol{y})$
6. $V = V \cup \{\boldsymbol{x}\}$;
7. endif
8. endfor

In algorithm 1, line 3 and line 4 correspond to Theorem 1. In these two lines, we use dataset $X \cup O$ as reference points. Line 5 and line 6 correspond to Theorem 2, where only dataset O is used as reference points, because Theorem 2 requires **y** to be outside the convex hull of X. It should be noted that the dataset O is not mandatory for FVDM, albeit O will increase the probability of FVDM for making the correct identification of convex hull vertices. Fortunately, additional data that can be used as O are natural available for some applications. For example, in a classification problem, when one needs to identify the convex hull of the data from one class, all the data from the other classes can be used as O. In case the data for O is not provided, one may either set $O = \emptyset$, or generate some synthetic data to form O.

B. Finding the convex hull vertices with SVM

As discussed in Section 1, no existing approach can find the convex hull of a dataset only based on the similarity information (e.g., a similarity matrix) between data points. In other words, there is no way to know whether a data point **x** is on the convex hull for sure, and thus it is difficult to assess whether FVDM works properly as we expect. Fortunately, if we consider the case of SVM, for which the kernel matrix can be viewed as a special form of the similarity matrix, there is a naïve approach with which we can identify the true vertices of the convex hull. To be specific, if a point **x** is a vertex of the convex hull of X in the kernel space, it should be linearly separable (in the kernel space) from all the other points in X, as depicted in Fig. 3. Therefore, for each point **x** in X, one may just apply a SVM classifier to check whether it is linearly separable from the other points. This intuition leads to the approach named finding Vertices with SVM (FVSVM). The pseudo-code of FVSVM is given in Algorithm 2. It should be noted that the FVSVM algorithm might not be appropriate for practical use. First, it involves training SVM classifier for n times, where n is the size of the dataset, and thus is very time consuming. Second, it is only applicable for methods where a kernel matrix is available, and may not well extended to problems with other forms of similarity matrices. We present it here because it can evaluate the correctness of FVDM.

a b

Fig 3: The green points and red points represent the point-set and the red points represent the convex hull vertices of the point-set.

Algorithm 2

Algorithm FVSVM

Input: X, the point-set; K ,the kernel matrix of X

Output: V, the convex hull vertices set

1. Initial $V=\emptyset$;
2. for every point $\boldsymbol{x} \in X$
3. if there is a SVM classifier which can separate $\boldsymbol{x}$ from $X/\boldsymbol{x}$
4. $V = V \cup \{\boldsymbol{x}\}$;
5. endif
6. endfor

C. Complexity analysis

FVDM needs to find the minimum distance and the maximum distance for every point. So, for every point, FVDM should compare all the distances from it to the others n-1 points. The complexity for every point is O(n-1). There are total n points. So the total complexity of FVDM is $O(n^2)$,

IV. EXPERIMENT

To assess the effectiveness of FVDM, we applied it to two synthetic datasets as well as one real-world dataset. The major aim of this empirical study is to evaluate how accurately FVDM could identify the convex hull. For this reason, FVSVM was applied to each point which is not identified as the convex hull vertex by FVDM to confirm whether it is in the convex hull identified by FVDM or not.

For FVSVM, the linear kernel is utilized, because the other two commonly used kernel functions, i.e., RBF kernel and polynomial kernel, map all data into the feature space in which all data points become the convex hull vertices of the convex hull. The proof for this issue is presented in the Appendix. Furthermore, the experimental study only requires a kernel matrix. Linear kernel is sufficient for this purpose.

Therefore, all our experiments will use the linear kernel.

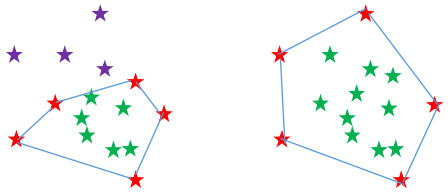
A. Performance measure

Since the empirical study mainly concerns whether FVDM can accurately find the convex hull. And the most important characteristic of an approximate convex hull is how many points in the dataset can be covered, a measure called convex hull coverage was used for our evaluation, as given in (2).

$$R_{cover} = \frac{|I_E|}{|X \backslash V|} \quad (2)$$

where R_{cover} is the coverage of the convex hull formed by convex hull vertices V, X is a point-set, V is the convex hull vertices of X found by FVDM and $I_E = (X \backslash V) \cap (E_v)$. Here, E_v is the convex hull of V and the point-set I_E is all the points of $X \backslash V$ in the convex hull E_v. In other words, I_E is the set of points that lie in E_v. It is worth mentioning that a point **x** being in E_v is the sufficient and necessary condition for satisfying that there exists no hyperplane that can separate the point $\boldsymbol{x}$ from the convex hull vertices V.

Intuitively, (2) measures how many points in X fall inside the convex hull formed by V. In two or three dimensions, it is intuitive. For example, Fig 5 represents two example results. It is obvious that Fig 5(b) corresponds to a better case because the convex hull covers all data points while Fig 5(a) have some data points falling outside the convex hull (purple points). In fact, the convex hull coverage is $\frac{6}{10}$ for in Fig 5(a) and $\frac{10}{10}$ for Fig 5(b).



(a) (b)

Fig 5: (a) and (b) represents two example results of finding the convex hull vertices. The red points are the identified convex hull vertices.

B. Experiments on low dimensional data

In the first experiment, we generated a 2-dimensional synthetic data set based on normal distributions. The data set consists of two classes, denoted as the "+" and "-" classes. The probability density functions (PDF) of the two classes are denoted by $f_+(x)$ and $f_-(x)$, respectively. The two PDFs are with different means but share the same covariance matrix, which is $\begin{bmatrix} 0.4 & 0 \\ 0 & 0.4 \end{bmatrix}$. The PDFs are as follow:

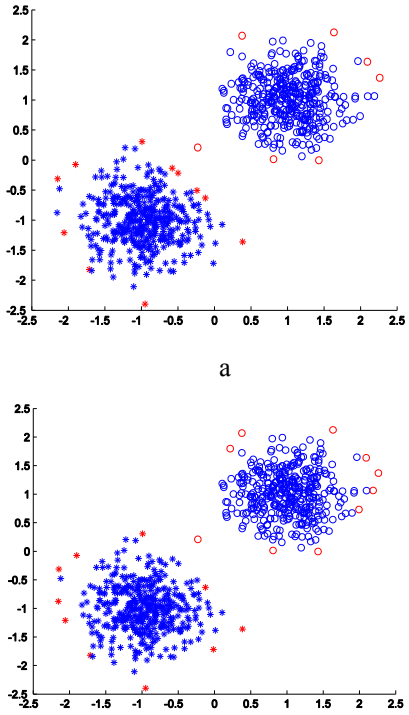
$$f_+(x) = \frac{1}{2\pi\sqrt{\det(\Sigma)}} \exp\left(-\frac{1}{2}(x-\mu_+)^T \Sigma^{-1}(x-\mu_+)\right)$$

$$f_-(x) = \frac{1}{2\pi\sqrt{\det(\Sigma)}} \exp\left(-\frac{1}{2}(x-\mu_-)^T \Sigma^{-1}(x-\mu_-)\right)$$

where, $\Sigma = \begin{bmatrix} 0.4 & 0 \\ 0 & 0.4 \end{bmatrix}$, and $\mu_+ = [1 \;\; 1]^T$, $\mu_- = [-1 \;\; -1]^T$

In the experiment, 400 positive points and 400 negative points were first randomly generated according to $f_+(x)$ *and* $f_-(x)$ respectively. Then, FVDM and FVSVM were employed to find the convex hull vertices of the positive and negative point-sets, respectively. After that, the obtained convex hull coverage, the number of identified CHVs (convex hull vertices) and the computation time were calculated. This experiment was repeated for ten times. The average coverage, the number of CHVs and the computation time are reported in Table I. The result of one example run of experiments is

depicted in Fig 6.



a

b

Fig 6: (a) and (b) represent the results of FVDM and FVSVM respectively. The star points and circle points represent data points belonging to different classes, and the red points stands for the convex hull vertices found by FVDM and FVSVM.

TABLE I
THE COMPARISON OF FVDM AND FVSVM ON LOW DIMENSIONAL DATA

algorithm	FVDM		FVSVM	
class	+	-	+	-
No. of instances	400	400	400	400
No. of CHVs	9.5	7	9.5	9.5
coverage	99.8%	99.78%	100%	100%
run time(seconds)	6.2134		3969.1	

From the coverage values given in Table I, we can see that FVDM satisfactorily identified the convex hull to cover almost all data points. Similar observation is illustrated in Fig. 6, where the convex hull vertices found by FVDM are very similar to those found by FVSVM. However, from the computation time reported in Table I, we can see that FVDM is far more time-efficient than FVSVM.

C. Experiments on high dimensional data

To assess the effectiveness of FVDM on the high dimensional data, we generated a 6-dimensional synthetic data set of two classes. The PDFs of the positive and negative classes are also normal distributions, Their covariance matrices are both $0.4\mathbf{I}_{6\times6}$, where $\mathbf{I}_{6\times6}$ is an identity matrix, and their means are $\mu_+ = [1\ 1\ 1\ 1\ 1\ 1]^T$ and $\mu_- = [-1\ -1\ -1\ -1\ -1\ -1]^T$ respectively.

In the experiment, 10000 points were generated for each class. Since the FVSVM will be too time-consuming in this case, only FVDM was applied to this dataset. The results are reported in Table II

TABLE II
THE RESULT OF FVDM ON HIGH DIMENSIONAL DATA

class	+	-
No. of instances	10000	10000
No. of CHVs	79	85
coverage	87.09%	89.02%
run time(seconds)	7489.4	

From the coverage in Table II, we can see that the performance of FVDM, though still acceptable, is worse than that of the 2-dimensional case. A possible explanation is that in the high dimensional space, it is less likely that the set of reference points contains the point that satisfies Theorem 1 or 2. In such as case, sampling a larger number of reference points may improve the coverage of FVDM.

D. Experiments on real-world data

The data used in this experiment is Astroparticle, which was used in [18]. In this experiment, we have a total of 4000 samples (2000 for each class), and every data point has 4 attributes. Principal Component Analysis [10] was first applied to the data to preprocess them. Then, the data were normalized so that they lie between the interval [0 1] on each dimension. After that, FVDM was applied and the results are presented in Table III.

TABLE III
THE RESULT OF FVDM ON REAL-WORLD DATA

class	+	-
No. of instances	2000	2000
No. of CHVs	168	192
coverage	85.5%	96.35%
run time(seconds)	29.4753	

From Table III, it can be found that FVDM achieved satisfactory performance on both classes. However, its performance on the two classes are quite different. This observation suggests that FVDM might be sensitive to the distribution of data, which could be an interesting issue deserving further investigation.

V. CONCLUSION AND DISCUSSION

In this paper, an algorithm named FVDM was proposed to approximately find the convex hull vertices of a data set. Different from the existing methods, FVDM only takes the similarity information between data points as input, without relying on the attribute values. This characteristic will be particularly useful for scenarios where little information about the space in which the data lie is available, e.g., the kernel space. Experimental results on two synthetic datasets and one real-world dataset demonstrate the effectiveness of FVDM. In the future, potential drawbacks of FVDM, e.g., sensitivity to data distributions, will be analyzed and addressed. Real-world applications of FVDM will also be further explored.

VI. ACKNOWLEDGMENT

This work was supported in part by the 973 Program of China under Grant 2011CB707006, the National Natural Science Foundation of China under Grants 61175065 and 61329302, the Program for New Century Excellent Talents in University under Grant NCET-12-0512, the Science and Technological Fund of Anhui Province for Outstanding Youth under Grant 1108085J16, and the European Union Seventh Framework Programme under Grant 247619.

VII. APPENDIX

In this appendix, we prove that RBF kernel and polynomial kernel (p ≥ 2), map all data points into the feature space in which all data points become vertices of the convex hull.

If we use RBF kernel function $\mathrm{K}(\mathbf{x},\mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2/(2\delta^2)}$, since K(**x, x**)=1, the kernel function will remap the data points into the kernel space *K*, and all data points will be remapped to the surface of a unit hypersphere, so all data points become the convex hull vertices in *K*. In polynomial kernel situation, we can prove following theorem:

Theorem 3: If use polynomial kernel function $\mathrm{K}(\mathbf{x},\mathbf{y}) = (\mathbf{x}\cdot\mathbf{y}+1)^p$, the kernel function will remap all data points into the convex hull vertices in the kernel space when $p \geq 2$ and $p \in N$.

Proof:

We prove the theorem by contradiction. Suppose the conclusion is not true, i.e., there is a point $\boldsymbol{x}$ in the feature space which would be remapped to $\phi(\boldsymbol{x})$ in the kernel space and $\phi(\boldsymbol{x})$ is not the convex hull vertices in the kernel space.

Suppose $\phi(\boldsymbol{x}_i)(\mathrm{i} = 1,2,\cdots \mathrm{m})$ is all the convex hull vertices in the kernel space.

$\therefore\ \phi(\boldsymbol{x}) = \sum_{i=1}^{m}\alpha_i\phi(\boldsymbol{x}_i)$, here $\alpha_i \geq 0$ and $\sum_{i=1}^{m}\alpha_i = 1$

$\therefore \phi(\boldsymbol{x})\cdot\phi(\boldsymbol{y}) = \sum_{i=1}^{m}\alpha_i\phi(\boldsymbol{x}_i)\cdot\phi(\boldsymbol{y})$ holds for any $\boldsymbol{y}\in R^n$

$\because \phi(\boldsymbol{x})\cdot\phi(\boldsymbol{y}) = K(\boldsymbol{x},\boldsymbol{y})$ for any $\boldsymbol{x},\boldsymbol{y}\in R^n$

$\therefore \mathrm{K}(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{m}\alpha_i\mathrm{K}(\boldsymbol{x}_i,\boldsymbol{y})$

That is

$(\boldsymbol{x}\cdot\boldsymbol{y}+1)^p = \sum_{i=1}^{m}\alpha_i(\boldsymbol{x}_i\cdot\boldsymbol{y}+1)^p$ for any $\boldsymbol{y}\in R^n$

So, for first-order, we have

$$C_p^1(\boldsymbol{x}\cdot\boldsymbol{y}) = \sum_{i=1}^{m}\alpha_i C_p^1(\boldsymbol{x}_i\cdot\boldsymbol{y}) \Longrightarrow \boldsymbol{x}\boldsymbol{y} = \sum_{i=1}^{m}\alpha_i(\boldsymbol{x}_i\cdot\boldsymbol{y}) \quad (3)$$

And for second-order

$$C_p^2(\boldsymbol{x}\cdot\boldsymbol{y})^2 = \sum_{i=1}^{m}\alpha_i C_p^2(\boldsymbol{x}_i\cdot\boldsymbol{y})^2 \Longrightarrow (\boldsymbol{x}\cdot\boldsymbol{y})^2 = \alpha_i(\boldsymbol{x}_i\cdot\boldsymbol{y})^2$$

If we set $\boldsymbol{y} = [b \quad 0 \quad \cdots \quad 0]^{\mathrm{T}}\ (b \neq 0)$. Then

$$(\boldsymbol{x}^1 b)^2 = \sum_{i=1}^{m}\alpha_i(\boldsymbol{x}_i^1 b)^2 \Longrightarrow (\boldsymbol{x}^1)^2 = \sum_{i=1}^{m}\alpha_i(\boldsymbol{x}_i^1)^2$$

$\mathbf{g}^i$ represents the *i*-th component of $\mathbf{g}$. Here, $\boldsymbol{x}^1$ is the first component of $\boldsymbol{x}$.

Similarly, we can get

$$(\boldsymbol{x}^j)^2 = \sum_{i=1}^{m}\alpha_i(\boldsymbol{x}_i^j)^2 \text{ for j=1,2,…n} \quad (4)$$

$$\begin{aligned}\therefore \sum_{i=1}^{m}\alpha_i(\boldsymbol{x}_i-\boldsymbol{x})^2 &= \sum_{i=1}^{m}\alpha_i(\boldsymbol{x})^2 - 2\sum_{i=1}^{m}\alpha_i\boldsymbol{x}_i\cdot\boldsymbol{x} + \sum_{i=1}^{m}\alpha_i(\boldsymbol{x}_i)^2\\ &= (\boldsymbol{x})^2 - 2(\boldsymbol{x})^2 + \sum_{i=1}^{m}\alpha_i\sum_{j=1}^{n}\left(\boldsymbol{x}_i^j\right)^2\\ &= -(\boldsymbol{x})^2 + \sum_{j=1}^{n}\sum_{i=1}^{m}\alpha_i\left(\boldsymbol{x}_i^j\right)^2\\ &= -(\boldsymbol{x})^2 + \sum_{j=1}^{n}(\boldsymbol{x}^j)^2\\ &= -(\boldsymbol{x})^2 + (\boldsymbol{x})^2 = 0\end{aligned}$$

$\therefore\ \alpha_i = 0\ or\ \boldsymbol{x} = \boldsymbol{x}_i$ for any $\mathrm{i} = 1,2,\cdots \mathrm{m}$

So, $\phi(\boldsymbol{x})$ is the convex hull vertices, which contradicts with the assumption.

The theorem is thus proved.

REFERENCES

[1] Bayer, Valentina. "Survey of algorithms for the convex hull problem." preprint (1999).

[2] Khosravani, Hamid R., Antonio E. Ruano, and Pedro M. Ferreira. "A simple algorithm for convex hull determination in high dimensions." Intelligent Signal Processing (WISP), 2013 IEEE 8th International Symposium on. IEEE, 2013.

[3] Barber, C. Bradford, David P. Dobkin, and Hannu Huhdanpaa. "The quickhull algorithm for convex hulls." ACM Transactions on Mathematical Software (TOMS) 22.4 (1996): 469-483.

[4] Graham, Ronald L. "An efficient algorith for determining the convex hull of a finite planar set." Information processing letters 1.4 (1972): 132-133.

[5] Bennett, Kristin P., and Erin J. Bredensteiner. "Duality and geometry in SVM classifiers." ICML. 2000.

[6] Haasdonk, Bernard. "Feature space interpretation of SVMs with indefinite kernels." Pattern Analysis and Machine Intelligence, IEEE Transactions on 27.4 (2005): 482-492.

[7] Peng, Xinjun, and Yifei Wang. "Geometric algorithms to large margin classifier based on affine hulls." Neural Networks and Learning Systems, IEEE Transactions on 23.2 (2012): 236-246.

[8] Chand, Donald R., and Sham S. Kapur. "An algorithm for convex polytopes." Journal of the ACM (JACM) 17.1 (1970): 78-86.

[9] Jarvis, Ray A. "On the identification of the convex hull of a finite set of points in the plane." Information Processing Letters 2.1 (1973): 18-21.

[10] Jolliffe, Ian. "Principal component analysis." John Wiley & Sons, Ltd, 2005.

[11] Preparata, Franco P., and Se June Hong. "Convex hulls of finite sets of points in two and three dimensions." Communications of the ACM 20.2 (1977): 87-93.

[12] Chan, Timothy M. "Optimal output-sensitive convex hull algorithms in two and three dimensions." Discrete & Computational Geometry 16.4 (1996): 361-368.

[13] Brönnimann, Herve, et al. "Space-efficient planar convex hull algorithms." Proc. Latin American Theoretical Informatics. 2002.

[14] Wang, Di, et al. "Online Support Vector Machine Based on Convex Hull Vertices Selection." (2013): 1-1.

[15] Bordes, Antoine, and Léon Bottou. "The Huller: a simple and efficient online SVM." Machine Learning: ECML 2005. Springer Berlin Heidelberg, 2005. 505-512.

[16] Mavroforakis, Michael E., and Sergios Theodoridis. "A geometric approach to support vector machine (SVM) classification." Neural Networks, IEEE Transactions on 17.3 (2006): 671-682.

[17] Cooper, Matthew, and Jonathan Foote. "Summarizing video using non-negative similarity matrix factorization." Multimedia Signal Processing, 2002 IEEE Workshop on. IEEE, 2002.

[18] Hsu, Chih-Wei, Chih-Chung Chang, and Chih-Jen Lin. "A practical guide to support vector classification." (2003). Bennett K P, Bredensteiner E J. "Duality and geometry in SVM classifiers." ICML. 2000: 57-64.

[19] Goshtasby, Ardeshir, and George C.Stockman. "Point pattern matching using convex hull edges." Systems, Man and Cybernetics, IEEE ransactions on 5 (1985): 631-637.

[20] Ceulemans, Eva, and Henk AL Kiers. "Selecting among three‐mode principal component models of different types and complexities: A numerical convex hull based method." British Journal of Mathematical and Statistical Psychology 59.1 (2006): 133-150.

[21] Halbwachs N. "Delay analysis in synchronous programs Computer Aided Verification." Springer Berlin Heidelberg, 1993: 333-346.

[22] Chan, Timothy M. "Output-sensitive results on convex hulls, extreme points, and related problems." Discrete & Computational Geometry 16.4 (1996): 369-387.