

RESEARCH

Open Access



Improvement on the vanishing component analysis by grouping strategy

Xiaofeng Zhang^{1,2}

Abstract

Vanishing component analysis (VCA) method, as an important method integrating commutative algebra with machine learning, utilizes the polynomial of vanishing component to extract the features of manifold, and solves the classification problem in ideal space dual to kernel space. But there are two problems existing in the VCA method: first, it is difficult to set a threshold of its classification decision function. Second, it is hard to handle with the over-scaled training set and oversized dimension of eigenvector. To address these two problems, this paper improved the VCA method and presented a grouped VCA (GVCA) method by grouping strategy. The classification decision function did not use a predetermined threshold; instead, it solved the values of all polynomials of vanishing component and sorted them, and then used majority voting approach to determine their classes. After that, a strategy of grouping training set was proposed to segment training sets into multiple non-intersecting subsets, which polynomials of vanishing component were later acquired through a VCA method, respectively, and finally combined into an integral set of vanishing component polynomial. What is more important is that it uses the bagging theory in ensemble learning to successfully expound and prove the correctness of the strategy of grouping training sets. It also compares the time complexity for training algorithm with and without grouping training sets, thus demonstrating the effectiveness of the grouping strategy. A series of experiments showed that the GVCA method proposed in the paper has a perfect classification performance with a rapid rate of convergence compared to other statistical learning methods.

Keywords: VCA, Vanishing ideal, Ideal

1 Introduction

Commutative algebra is a discipline of algebra mainly studying commutative ring [1]. It takes the algebraic number theory and algebraic geometry as its study background. In a classic sense, the research object of commutative algebra is the zero point of a polynomial equation set, which correlates with not only the number theory (such as Diophantine equation [2]) but also the manifold pattern (such as the hypersurface defined by polynomial) [3]. While, in a modern sense, the research object of commutative algebra is the topological space with rich structure (structure sheaf) that could be provided by a spectrum of a commutative ring. Commutative algebra integrated with a manifold pattern has been widely applied in machine learning in recent years, as it

can solve some common machine learning problems, such as classification and clustering, from a perspective of a manifold pattern. Vanishing component analysis (VCA) [4] is a method of solving classification problem by applying the theory of commutative algebraic that has emerged in recent years. Therein, the vanishing component refers to a generator (i.e., Grobner basis [5]) for vanishing ideal of polynomial ring space of fitting a manifold pattern, which is in form of a group of polynomials employing feature as their variable. Once acquiring the vanishing component, the natural feature of a data manifold pattern can be captured.

The VCA method uses an ideal space dual to kernel space [6], thus becoming a dual algorithm of a kernel method. However, the VCA method has two problems: (1) A set of polynomials obtained through the VCA method, i.e., vanishing component, should be judged whether it is zero or not, when a test sample is substituted into a classification decision function. But in

Correspondence: coldrain521@gmail.com

¹Department of Computer Science and Engineering of Southeast University, Nanjing, China

²Jiangsu Administration Institute, Nanjing, China

the case of noise, it is difficult to ensure it is strictly zero. Even if a threshold is used, it is impossible to set its value. (2) The VCA method faces a similar difficulty with kernel method, that is, restriction on the scale of training set [7]. This is because, in training algorithm, the number of training samples will influence the number of vanishing component polynomials, the number of monomials contained in vanishing component polynomials, the order of polynomial, etc. [4], thereby largely increasing the amount of calculation. On the other hand, an oversized eigenvector dimension will also lead to an oversized dimension of singular value decomposition (SVD) matrix used in the VCA method, and finally make it harder to solve.

To solve the above two problems of the VCA method, this paper made a theoretical analysis and experimental research to improve the VCA method and presented a grouped VCA (abbreviated as GVCA) method. The GVCA method modified the classification decision function in the VCA method. It did not preset a threshold; instead, it solved the values of all polynomials of vanishing component and sorted them, and then used the majority voting approach to determine their classes. After that, a strategy of grouping training set was proposed on the basis of deduction of ensemble learning theory. In this strategy, the training sets were horizontally or vertically segmented into multiple non-intersecting subsets, which polynomials of vanishing components were later acquired through the VCA method, respectively, and finally combined into an integral set of vanishing component polynomials. By experiment, it is verified that the GVCA method has a perfect classification performance with a rapid rate of convergence.

The main contributions of this paper include that (1) it proposed a classification decision function easier to operate, (2) it put forward a strategy of grouping training sets and utilized the Bagging theory in ensemble learning to successfully expound and prove the correctness of the strategy of grouping training sets, and (3) it deduced the time complexity for training algorithm after grouping, thus demonstrating the effectiveness of such strategy.

The paper is arranged as below: Section 2 introduces the efforts related to commutative algebra, the VCA method, and ensemble learning. Section 3 firstly provides the theoretical basis of commutative algebra, then gives the VCA method and analyzes the problems caused by the threshold of its decision function being hard to set, and over-scaled training sets and oversized dimension of eigenvector. Finally, based on the analysis of problems resulted from setting threshold of a decision function in the VCA method and proposed an improved decision function on the basis of sorted value of vanishing component polynomial, the experiment

showed that such improved decision function was more operational. Furthermore, based on the analysis of problems brought by the over-scaled training set and the oversized dimension of eigenvector, this section raised a strategy which horizontally and vertically grouped the training sets, made analysis on vanishing components respectively, and classified the union of vanishing component in each group. Then, the ensemble learning theory was utilized to verify the correctness and effectiveness of this strategy. By improving these two aspects, a GVCA method was formed. Afterwards, In Section 4, four experiments were conducted with simulation dataset and UCI dataset, which indicates that this method has a perfect classification performance with a rapid rate of convergence. At last, Section 5 presents a conclusion and forecast of the future work.

2 Related work

2.1 Commutative algebra

From the late eighteenth to the mid-nineteenth century, Gauss and Kummer et al. studied the nature of rational integer and the rational integer solution of equation and considered the Elementary Number Theory problems in the quadratic field, cyclotomic field, and their algebraic integer ring [8, 9]. Through abstraction and systematization implemented by Dedekind and Hilbert et al. [10, 11], a new discipline generated to study algebraic number field and its algebraic integer ring, called Algebraic Number Theory [12]. In 1882, the concepts of the Ideal and the Prime Ideal proposed by Dedekind [13] laid a foundation for one-dimensional commutative algebra [14]. At the time later than the number theory, geometry also experienced an algebraization process, and thus the multidimensional commutative algebra started to form its shape [15]. The Ideal Theory [16] proposed by Hilbert et al. at the end of nineteenth century and Noether in the 1920s to the 1930s, and the Valuation Theory [17], the Local Ring Theory [18], and the Dimension Theory [19] established by Krull have furnished classical geometry with brand new algebraic tools and enabled commutative algebra to be an independent discipline.

2.2 VCA method

Livni et al. put forward a vanishing component analysis method with stable values, i.e., the VCA method, to solve the generator (i.e., vanishing component) for vanishing ideal of fitting data manifold pattern. A set of polynomials obtained by this method can be used to represent the structure of manifold pattern. And the method can also be applied in supervised learning field, which has proved to achieve a good precision in experiments [4]. Actually, the emergency of the VCA method

is a result of commutative algebra combined with machine learning, while the key is to apply commutative algebra in solving the generator for vanishing ideal of fitting manifold pattern. As data often contains noises, it is difficult to acquire an analytical solution, so numerical methods are needed to solve an approximate vanishing ideal. The VCA method-related research involves as follows: Buchberger and Möller et al. firstly proposed an algorithm to figure out the vanishing ideal of finite point set, called Buchberger-Möller algorithm [20], which can be regarded as the Euclidean algorithm that solves the maximum common divisor of single variable and the generalization of Gaussian elimination method in a linear system. The obtained Grobner basis has stable values when the coordinate system is measurable [21]. Corless et al. raised a singular value decomposition (SVD) [22] approach for polynomial system and used it to solve the maximum common divisor problem [23], while SVD is the main step for solving the approximate vanishing ideal. Stetter developed the theory proposed by Corless et al. and presented a more general numerical method [24]. Heldt et al. utilized SVD and stable numerical method [25] to solve the approximate vanishing ideal, and these vanishing component polynomials almost composed a border basis [21]. Heldt et al. also worked out the Cohen-Macaulay basis of vanishing ideal [26]. Sauer et al. made use of a strategy of independent coordinate and increased degree of polynomial to calculate the approximate vanishing ideal, thereby acquiring an approximate solution of Buchberger-Möller algorithm [27]. Kiral et al. raised two dualities, namely, the duality between kernel and ideal and the duality between ideal and manifold pattern. Then, these two dualities can be used to design two algorithms: ideal principal component analysis (IPCA) and approximate vanishing ideal component analysis (AVICA), in order to learn the generation features and discriminant features of manifold pattern [28]. Both algorithms can be considered as an extension of kernel primary component analysis (Kernel PCA) algorithm [29]. To sum up, commutative algebra provides an approach to solve the approximate vanishing ideal, and the VCA method uses an ideal space dual to kernel space, thus becoming a dual algorithm of kernel method. This has offered a new idea for solving machine learning problem in kernel space and is of great value in studying kernel method and machine learning.

2.3 Ensemble learning

Ensemble learning refers to a kind of machine learning method which uses a lot of learning devices to study and then integrates each learning outcome under some certain rules, so that the model's stability

and prediction ability can be enhanced. Kleinberg proposed a general stochastic discrimination (SD) method to segment multidimensional space by stochastic process [30]. The SD method is capable to enhance the performance of weak classifier. Based on this, Ho raised a random subspace method (RSM) and constructed a forest with the idea of RSM and the decision tree [31]. Hansen and Salamon proved that introducing ensemble learning into artificial neural network can improve the properties [32]. On the basis of random decision forest (RDF), Breiman employed the Bagging (i.e., bootstrap aggregation) technology, made a theoretical analysis, and provided an error upper-bound [33]. Bagging attempted to achieve similar learning module in small sample set, and then averaged the predicted values. This method uses different learning modules in different datasets to reduce the variance. Schapire put forward a boosting method [34], an iterative technique regulating the weight of observed value based on the last classification. If an observed value has been wrongly classified, it will increase the weight of observed values, and vice versa. Generally, boosting can decrease offset error to build a powerful prediction model. But sometimes, it can also over-fit the training data. Freund and Schapire proposed an Adaboost method [35]. Cho and Kim integrated the results of multiple neural networks that used fuzzy logic, and the experiment showed this method improved the precision of classification [36]. The stacking approach raised by Wolpert is suitable to integrate different types of models and helps to bring offset error and variance down [37].

3 Methodology

3.1 Theoretical basis

Definition 1 (Left Ideal). The nonempty subset $I \subseteq R$ in ring R is called the left ideal in R , if I satisfies the following two conditions [13]:

- (1) The addition operation of I in ring R constitutes a subgroup of additive group of ring R .
- (2) $RI \subseteq R$, i.e., for $\forall a \in R$ and $b \in R$, it satisfies $ab \in R$.

Definition 2 (Right Ideal). The nonempty subset $I \subseteq R$ in ring R is called the right ideal in R , if I satisfies the following two conditions [13]:

- (1) The addition operation of I in ring R constitutes a subgroup of additive group of ring R .
- (2) $IR \subseteq R$, i.e., for $\forall a \in R$ and $b \in R$, it satisfies $ba \in R$.

Definition 3 (Ideal). If I is both left ideal and right ideal, then I is called a bi-ideal, abbreviated as Ideal [13].

Definition 4 (Vanishing Ideal). Given k is a domain, and f_1, f_2, \dots, f_m is a polynomial in the ring $k[x_1, x_2, \dots, x_n]$, then, the set V can be defined as below:

$V(f_1, f_2, \dots, f_m) = \{(u_1, u_2, \dots, u_n) \in k^n : f_i(u_1, u_2, \dots, u_n) = 0\}$. In which, $i = 1, 2, \dots, m$. The set $V(f_1, f_2, \dots, f_m)$ is called an affine variety of the polynomial f_1, f_2, \dots, f_m . Then the set $I(V) = \{f \in k[x_1, x_2, \dots, x_n] : f(u_1, u_2, \dots, u_n) = 0, \forall u \in V\}$ is an ideal of the ring $k[x_1, x_2, \dots, x_n]$, called the vanishing ideal of V [37].

Definition 5 (Grobner basis). Fix a monomial, if it satisfies the following formula, then a finitely generated $G = g_1, \dots, g_k$ will be the Grobner basis of ideal I [5].

$$\langle LT(g_1), \dots, LT(g_k) \rangle = \langle LT(I) \rangle \quad (1)$$

in which, $LT(f)$ represents the leading type of non-zero polynomial f . The coefficient of leading type is called a leading coefficient, denoted as $LC(f)$, and the corresponding term is called a leading term, denoted as $LM(f)$.

Example 1. For a non-zero polynomial $f(x) = a_0x^n + a_1x^{n-1} + \dots + a_m$, its leading type $LT(f) = a_0x^n$, the leading coefficient $LC(f) = a_0$, and the leading term $LM(f) = x^n$. It is easily known that the formula below is valid.

$$LT(f) = LC(f)LM(f) \quad (2)$$

There is an important proposition about ideal and Grobner basis, as below [38]:

Proposition 1. Suppose I is a non-zero ideal in polynomial A , $G = \{g_1, \dots, g_k\}$ is a non-zero ideal in I , then the following statements are equivalent:

- (1) G is the Grobner basis of I .
- (2) $f \in I$, if and only if f is generated by G .
- (3) $f \in I$, if and only if h_1, \dots, h_i exists, making

$$f = \sum_{i=1}^k h_i g_i \quad (3)$$

It is known from Hilbert Basis Theorem and the above ratiocination that the generator of vanishing ideal I in ring R is the Grobner basis [5], namely, vanishing component.

3.2 VCA method and its existing problems

3.2.1 VCA method

In the VCA method, by solving the generator of vanishing ideal (i.e., Grobner basis), it is feasible to obtain the generation features of a manifold pattern, so that an input space can be switched into a feature space. In a feature space, it is easier to judge the class of data. Suppose an input space is $S \subseteq R^n$, then VCA output will be $V = \{f_1(x), \dots, f_k(x)\}$, in which $f_i(x)$ is a polynomial, and it satisfies $\forall x \in S, f \in I(S), f(x) = 0$. That is a vanishing ideal. When V is a finite set to get a group of generators of vanishing ideal of S . These generators are vanishing components, which compose the generation feature of manifold pattern S . Therefore, the VCA method is an application of Buchberger-Möller¹ in nature. What it works out is the Grobner basis of vanishing ideal in polynomial ring of fitting manifold pattern.

The VCA method firstly initializes three sets, i.e., the candidate polynomial set $C_1 = \{f_1, \dots, f_n\}$, in which, $f_i(x) = x_i$, non-vanishing component polynomial set $F = \{f(\cdot) = 1/\sqrt{m}\}$ and vanishing component polynomial set $V = \emptyset$. Next, the algorithm *FindRangeNull()* used to solve zero space is applied to solve the new non-vanishing component polynomial set F_1 and vanishing component polynomial set V_1 , and later combined with original sets F and V , thus composing the current non-vanishing component polynomial set F and vanishing component polynomial set V . At the same time, a new candidate polynomial set C_t is figured out. In case of $C_t = \emptyset$, it is time to finish and output the final non-vanishing component polynomial set F and vanishing component polynomial set V . But if $C_t \neq \emptyset$, it is necessary to conduct iterative computations on the above steps until the termination condition is satisfied at the end; that is to say, all vanishing component polynomials are worked out, thereby forming an integral non-vanishing component polynomial set (i.e., the Grobner basis of vanishing ideal). In the VCA method, after the Grobner basis of vanishing ideal is calculated, it is possible to implement classification via the classification decision function. Assuming $\{p_1^l(x), \dots, p_{n_l}^l(x)\}$ is the generator of vanishing ideal of class l , then for an example of any class l , it satisfies $|p_j^l(x)| = 0$. However, for examples of other classes, there are some polynomials not equaling to zero at least.

3.2.2 Problem of setting threshold for classification decision function

(1) Setting threshold for classification decision function: As mentioned above, the classification decision function

in the VCA method chooses $|p_j^l(x)|$ as the feature of example x . If each class belongs to different algebraic sets, then the data can be linearly classified in such feature space. Nevertheless, its classification decision function formula

$$|p_j^l(x)| = 0 \quad (4)$$

has some problems. Regarding data without noise, Eq. (2) is theoretically valid (considering the factors like calculation error, it is actually non-valid). While for data with noise, this equation approximates to zero, that's

$$|p_j^l(x)| \approx 0 \quad (5)$$

Hence, both above situations need to consider setting threshold. If let ε be threshold, when $|p_j^l(x)| < \varepsilon$, Eq. (5) will be valid. The problem is what value should this threshold ε supposed to be. Next, the paper will verify it through experiment.

(2) A test carried out according to the problem led by threshold setting: This paper built the following two groups of polynomials to generate analog data set for testing:

$$x_1^2 + 0.01x_2^2 + x_3^2 = 1 \quad (6)$$

$$x_1^2 + x_3^2 = 1.4 \quad (7)$$

The experimental data is generated using sampling approaches. By adding noise or not, the sampling methods are classified into noiseless sampling and noisy sampling. The noiseless sampling method is as follows: For formula (6), generate a stochastic number x_1 between $[-1,1]$ and a stochastic number x_2 between $[-10,10]$. If both of them can satisfy $1 - x_1^2 - 0.01x_2^2 \geq 0$, the above sampling is a successful one, and can be used to calculate the candidate value of x_3 . But if neither of them can satisfy the above condition, it needs to return and re-sample. The algorithm will finish until it gets enough amount of sampling. For formula (7), generate a stochastic number x_1 between $[-1.1832, 1.1832]$, use the same approach to calculate x_3 , and generate a stochastic number between $[-1,1]$ and evaluate it as x_2 . The algorithm will finish until it gets enough amount of sampling. The noisy sampling method is similar to noiseless sampling method, in which the only difference is the Gaussian noise $\mu = 0$, $\sigma = 0.02, \dots, 0.10$ added in the process of generating dataset.

3.2.2.1 Experiment without noise The following experiment aims to verify the impact of the threshold ε , used to judge whether the vanishing component polynomial value is zero or not, on the classification

performance. This experiment applies a noiseless sampling method, and two types of experiment data are sampled from above formulas (6) and (7), without noise. The iterative times are ten. Both training examples and tested examples are 200. The experiment is conducted with a fixed training set and testing set method. See the experimental results in Table 1.

Table 1 demonstrates (1) the value of threshold ε , used to judge whether the polynomial value is zero or not, should be small in case of no noise. Because it can be seen from experimental results when ε is small, the values of *Precision*, *Recall*, and *F1* are all very good, but should not be smaller than machine precision; otherwise, the results cannot be judged. (2) When the threshold ε gradually increases, the performance gradually decreases and tends to change monotonously. This shows the VCA method is sensitive to the change of threshold in case of no noise, and there is a rule.

3.2.2.2 (b) Experiment with noise The following experiment also aims to verify the impact of the threshold ε , used to judge whether the vanishing component polynomial value is zero or not, on the classification performance, but there is a difference that this experiment adds the Gaussian noise. All experimental data are sampled from above formulas (6) and (7), added with the Gaussian noise $\mu = 0$, $\sigma = 0.1$. Both training examples and tested examples are 200. The experiment is conducted under fixed training set and testing set method. See the experimental results in Table 2.

Table 2 demonstrates (1) after adding noise, for the same threshold ε , the experimental performance is lower somewhat than the above experiment without noise. This shows the noise can affect performance. (2) After adding noise, on a whole, the smaller the threshold ε is, the better the performance will be. But under the effect of noise, this result has some exceptional circumstances. For example, when $\varepsilon = 10^{-1}$, there is a better result than others. Besides, when $\varepsilon = 10^0$, the results are not the worst one. This suggests the setting of threshold ε is relevant to the features of noise in case

Table 1 The impact of the threshold ε of classification decision function in the VCA method on the classification performance (without noise)

ε	10^{-8}	10^{-7}	10^{-6}	10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{-1}	10^0
P	1.0	1.0	1.0	1.0	1.0	0.85	0.83	0.83	0.75
R	1.0	1.0	1.0	1.0	1.0	0.73	0.71	0.74	0.69
F1	1.0	1.0	1.0	1.0	1.0	0.78	0.77	0.78	0.72

Table 2 The impact of the threshold ε of classification decision function in the VCA method on the classification performance (with noise)

ε	10^{-8}	10^{-7}	10^{-6}	10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{-1}	10^0
P	0.88	0.86	0.75	0.75	0.75	0.74	0.78	0.77	0.72
R	0.84	0.81	0.51	0.51	0.60	0.65	0.75	0.76	0.70
F1	0.85	0.83	0.60	0.60	0.66	0.71	0.76	0.76	0.71

that noise exists. On the premise of having no idea of noise distribution in advance, it is impossible to set a reasonable value for ε . This directly makes the classification decision function in the VCA method hard to operate.

3.2.3 Problems resulted from over-scaled training set and oversized eigenvector dimension

The influence produced by over-scaled training set and oversized eigenvector dimension: On the one hand, it can be known from the VCA method that the over-scaled training set, i.e., too many training examples, will result in too many vanishing component polynomials, too many monomials contained in the vanishing component polynomial, and too high order of polynomials. Among them, the worst impact is produced by too high order of polynomials. On the other hand, the dimension of eigenvector has a great impact on the computing time of an algorithm, which is because the eigenvector dimension is directly embodied in the number of variables in an algorithm. When there are too many variables, both the number of candidate polynomial and its maximum number of times will increase rapidly, and the SVD algorithm used to solve zero space will correspondingly quickly become more complex. Both situations will bring about a too long or even intolerable computing time in the VCA method and consequently let the VCA method be less practical.

An experiment against the problem caused by an over-scaled training set: The following experiment aims to directly verify the influence of the order of vanishing component polynomial on the performance and, accordingly, indirectly verify the influence of over-scaled training set on the performance. Two types of experimental

data are sampled from the above two groups of polynomials, i.e., formulas (6) and (7).

Both the training examples and testing examples are 200. All these 400 examples form an experimental dataset. Having considered that the training sets usually have noise in real situation, the Gaussian noise is set with $\mu = 0$, $\sigma = 0.1$. Afterwards, the order of vanishing component polynomial is set as the integer between [2, 12]. Then VCA method is used to solve, and the experimental results are as shown in Table 3. Table 3 indicates that the experimental performance increases with added order of polynomial in the beginning (with the number of times from 2 to 7), but when it reaches to some certain value (with the number of times from 7 to 12), the performance will slowly increase or no longer increase. The reason lies on that a too high order of polynomial may lead to over-fitting phenomena, so that the change of performance goes to a plateau. This proves that too high order of vanishing component polynomial exactly greatly affects the classification performance. Moreover, it can explain over-scaled training set not only increases computation difficulty but also affects the experimental performance.

3.3 GVCA method

According to the abovementioned two problems of the VCA method, i.e., a problem caused by setting threshold ε of its classification decision function and a problem resulted from over-scaled training set and oversized eigenvector dimension, this paper proposed a grouping-based VCA (i.e., grouped VCA, abbreviated as GVCA) method to solve the problems. The GVCA method improves the classification decision function in the original VCA method and raises a strategy of grouping training set.

3.3.1 Classification decision function in the GVCA method

The classification decision function in the GVCA method does not preset a threshold ε ; instead, it solves the values of all vanishing component polynomials and sorts them (in an order from large to small according to their absolute values). Later, the top ranked $N\% = (10\%, 20\%, 30\%, \dots)$ ones of all polynomials are selected and judged the class through a majority voting approach. The classification decision function of the GVCA method is as shown in Algorithm 1.

Table 3 The influence of the order of vanishing component polynomial in the VCA method on the classification performance

The order of polynomial	2	3	4	5	6	7	8	9	10	11	12
P	0.75	0.63	0.70	0.82	0.84	0.87	0.86	0.88	0.88	0.88	0.88
R	0.50	0.60	0.60	0.74	0.80	0.83	0.83	0.85	0.85	0.85	0.85
F1	0.60	0.62	0.64	0.78	0.82	0.85	0.85	0.86	0.86	0.86	0.86

Algorithm 1: GVCA algorithm with enhanced decision function	
Input:	
1:	Generators of n classes by VCA: $\{f_{n1}, f_{n2}, \dots, f_{nm}\}$ and their labels;
2:	TestData ;
Output:	
3:	TestData 's label;
4:	Initialize;
5:	For $i = 1, 2, \dots, n$ Do
6:	For $j = 1, 2, \dots, m$ Do
7:	Compute polynomial $Value_{ij} = f_{ij}(\text{TestData})$;
8:	End For
9:	End For
10:	Top Rank $\{Value_{ij}\}$ and f_{ij} ;
11:	Vote to get the label of TestData by major f_{ij} 's labels;

This classification decision function has two inputs. One is the vanishing component polynomial set $\{f_{n1}, \dots, f_{nm}\}$ of each class of data produced by training steps via the VCA method, in which n represents the number of class and m represents the number of polynomial corresponding to each class. The other one is the test set Test Data, in which the output is the labels of test set. The main steps are as follows: 5–9: construct double circles with class number n and testing example number m , respectively, then substitute the testing example Test Data into the vanishing component polynomial set $\{f_{n1}, \dots, f_{nm}\}$ of each class, and later figure out the absolute $|Value_{ij}|$ of polynomial value. 10–11: take the top ranked $N\%$ results of the calculated absolute value $|Value_{ij}|$ of vanishing component polynomial and choose the class of maximum amount as the class label of testing example.

3.3.2 Training set grouping strategy in the GVCA method

- (1) Grouping the training set: this paper proposed the GVCA method to address two situations of over-scaled training set and too high dimension of eigenvector. Regarding the problem of over-scaled training set, it is a practicable way to group the examples, i.e., to horizontally segment the entire training set into several training subsets (for instance, take 10/20/30/40/50 examples as a group), then acquire the features (i.e., vanishing component polynomial) by the original VCA method respectively and combine the vanishing component polynomial in multiple grouped training sets into the vanishing component polynomial in an integral training set. This approach is called the horizontal grouping method.

The GVCA method based on horizontally grouped training set is as shown in Algorithm 2. Meanwhile, regarding the problem of too high eigenvector dimension of training set, a feasible approach is to group

the features, i.e., to randomly group a feature set into several non-intersecting subsets, and project the originally integral training set into those feature sets, thereby composing several new grouped training sets. After that, the original VCA method is used to obtain vanishing component polynomial respectively, and finally, the vanishing component polynomial in multiple grouped training sets is combined into the vanishing component polynomial in an integral training set. This approach is called the vertical grouping method. The GVCA method based on vertically grouping training sets is as shown in Algorithm 3. Furthermore, to handle both problems of over-scaled training set and too high eigenvector dimension, the horizontal and vertical grouping approaches can be applied simultaneously.

Algorithm 2: GVCA algorithm with grouped training data by row segment	
Input:	
1:	training data TrainData ;
2:	group size GroupSize ;
Output:	
3:	TrainData 's generator set G .
4:	Initialize
5:	Segment TrainData by GroupSize to get some groups;
6:	Get group's number GroupNumber ;
7:	For $i = 1, 2, \dots, \text{GroupNumber}$ Do
8:	Do original VCA in grouped training data;
9:	Get generators sets $\{G_1, G_2, \dots, G_{\text{GroupNumber}}\}$;
10:	End For
11:	Combine all generators from $\{G_1, G_2, \dots, G_{\text{GroupNumber}}\}$ to get TrainData 's generator set G ;

Algorithm 3: GVCA algorithm with grouped training data by column segment	
Input:	
1:	training data TrainData ;
2:	feature set FeatureSet ;
3:	sub feature sets SubFeatureSets ;
Output:	
4:	TrainData 's generator set G .
5:	Initialize;
6:	Segment FeatureSet to SubFeatureSets ;
7:	Project TrainData on SubFeatureSets to get some Groups;
8:	Get Group's number GroupNumber ;
9:	For $i = 1, \dots, \text{GroupNumber}$ Do
10:	Do original VCA in grouped training data;
11:	Get generators sets $\{G_1, G_2, \dots, G_{\text{GroupNumber}}\}$;
12:	End For
13:	Combine all generators from $\{G_1, G_2, \dots, G_{\text{GroupNumber}}\}$ to get TrainData 's generator set G ;

- (2) Prove the correctness of the strategy of grouping training set: the following paragraphs expound and prove the correctness of the strategy of grouping training set in the GVCA method. There are vertically and horizontally grouping cases, stated respectively below.

3.3.2.1 (a) Horizontal grouping In nature, the VCA method lies on doing feature mapping $x \rightarrow |p_j^l(x)|$ on dataset, i.e., mapping the dataset for x into a dataset for $t = (|p_1^l(x)|, \dots, |p_{n_l}^l(x)|)$, in which, $\{p_1^l(x), \dots, p_{n_l}^l(x)\}$ is the generator for vanishing ideal of class l . Horizontally grouping the training set is equivalent to Bagging integrated learning implemented after bootstrap sampling on t . Due to the correspondence between Grobner basis and dataset x , the results of classifying t can be directly mapped back to x , thus guaranteeing its correctness.

3.3.2.2 (b) Vertical grouping Vertical grouping is equivalent to attribute Bagging done to t , which means to repeatedly sample the eigenvector set to obtain the feature subsets. Either group is allowed to cross each other, in other words, different feature subsets are allowed to contain several same features. In accordance with the theory raised in [39], its correctness can be ensured as well.

- (3) Analysis on time complexity of grouped training set: Assume the size of training set is M before being grouped, and the dimension of eigenvector is N . From the original VCA method, the following conclusions [4] can be obtained:
- (a) The training under the VCA method will finish after iteration for $t \leq M + 1$ times at the most.
 - (b) The highest order of polynomial in both non-vanishing component polynomial set F and vanishing component polynomial set V are M order at the most.
 - (c) $|F| \leq M$, and $|V| \leq |F|^2 \times \min\{|F|, N\}$.
 - (d) The time complexity of all vanishing component polynomial in V is computed to be $O(|F|^2 + |F| \times |V|)$.

Before grouping the training set, usually $M \gg N$; therefore, the time complexity of all vanishing component polynomial in V is computed to be:

$$\begin{aligned} O(|F|^2 + |F| \times |V|) &\leq O(M^2 + M \times |V|) \\ &\leq O(M^2 + M \times |F|^2 \times \min\{|F|, N\}) \\ &\leq O(M^2 + M \times M^2 \times \min\{M, N\}) \\ &\leq O(M^2 + M \times M^2 \times N) \\ &= O(M^2 + M^3 \times N) \end{aligned} \quad (8)$$

According to formula (8), it can be known that the time complexity of non-grouped training set is:

$$O(M^3 \times N) \quad (9)$$

Now, the training set is segmented into k groups that do not intersect, and the example number of each group is m , so it is easily known that $M = km$. Besides, the eigenvector dimension is till N , and after grouping, usually $m \leq N$. Therefore, the time complexity of all vanishing component polynomial in each subgroup V_i , $i = 1, \dots, k$ is computed to be:

$$\begin{aligned} O(|F_i|^2 + |F_i| \times |V_i|) &\leq O(m^2 + m \times |V_i|) \\ &\leq O(m^2 + m \times |F_i|^2 \times \min\{|F_i|, N\}) \\ &\leq O(m^2 + m \times m^2 \times \min\{m, N\}) \\ &\leq O(m^2 + m \times m^2 \times m) = O(m^4) \end{aligned} \quad (10)$$

According to formula (10), it can be known that the time complexity of grouped training set is:

$$O(k \times m^4) \quad (11)$$

In addition, due to $M = km$, the total time complexity of grouped training set is

$$O(k \times m^4) = O(k \times (M/k)^4) = O(M^4/k^3) \quad (12)$$

Then, according to formulas (9) and (12), the ratio of time complexity before and after grouping training set is

$$M^4/k^3 \times 1/(M^3 \times N) = M/(k^3 \times N) \quad (13)$$

It is known from formula (13) that, when the grouping number k is large enough, the ratio of time complexity before and after grouping training set will be small enough. However, as the value of k can be M at the most (in real situation, it should be a suitable value approximating to M), so the minimal value of such ratio will be:

$$M/(k^3 \times N) \geq M/(M^3 \times N) = 1/(M^2 \times N) \quad (14)$$

The formula (14) is the lower bound of such ratio.

Example 1. Suppose the size of training set before being grouped is $M = 1000$, the eigenvector dimension is $N = 100$, the training set is segmented into $k = 20$ groups, and the size of training set in each group is $m = 50$. It can be seen from formula (13) that the time complexity of grouped training set has been reduced to the following ratio of that before grouping.

$$\begin{aligned} M/(k^3 \times N) &= 1000/(20^3 \times 100) = 10/20^3 \\ &= 1/800 \end{aligned} \quad (15)$$

In conclusion, the strategy of grouping training set can significantly lower the time complexity of the VCA method.

Table 4 UCI dataset used in the experiment

Dataset	Example no.	Eigenvector	Classification no.
Wine	178	13	3
Transfusion	748	5	2
Connectionist Bench	528	10	10
Breast Cancer Wisconsin	699	10	2
Indian Liver Patient	583	10	2
Mammographic Masses	961	6	2
Iris	150	4	3

4 Results and discussion

This paper had totally designed and completed four groups of experiments. At first, by experiment, it found out a proportion of sorted vanishing component polynomial suited to the classification decision function in the GVCA method. Next, by experiment, it found out a suitable size of grouped training set (including horizontally and vertically). At last, on the basis of the above two experiments, the classification and convergence performance of the GVCA method were tested on simulation dataset and UCI dataset, respectively.

4.1 Experimental settings

The data used in simulation dataset were sampled from the abovementioned formulas (6) and (7), with all examples added with Gaussian noise at $\mu = 0$, $\sigma = 0.02$ ($\sigma = 0.06$, $\sigma = 0.1$). In order to generate balanced vanishing component polynomial, the number of two classes of examples was set to be equal.

The UCI standard dataset is a dataset for machine learning proposed by the University of California, Irvine. It is a frequently used standard testing dataset. Depending on the experimental need, the paper chose two types of data subsets in UCI dataset. One is a dataset with small eigenvector dimension, which includes seven subdatasets such as Wine, Transfusion, Connectionist Bench, Breast Cancer Wisconsin, Indian Liver Patient, Mammographic

Table 5 UCI dataset used in the experiment

Dataset	Example no.	Eigenvector	Classification no.
Onehr	2536	73	2
LSVT	126	309	2
Hill Valley (with noise)	606	100	2
Hill Valley (without noise)	606	100	2
100Plant Species (data Sha 64)	1600	64	100
100Plant Species (data Tex 64)	1600	64	100
100Plant Species (data Mar 64)	1600	64	100

Table 6 The influence of the sorting-purpose polynomial proportion in classification decision function of the GVCA method on the classification performance

N%	10%	20%	30%	40%	50%	60%	70%	80%	90%
P	0.83	0.84	0.86	0.88	0.86	0.86	0.86	0.88	0.87
R	0.76	0.80	0.81	0.84	0.83	0.83	0.83	0.86	0.84
F1	0.79	0.82	0.83	0.85	0.84	0.84	0.84	0.86	0.85

Masses, and Iris. The dataset scale and eigenvector dimension are as shown in Table 4.

And the other one is a dataset with large eigenvector dimension, which includes seven subdatasets such as Onehr, Hill Valley(with noise and without noise), LSVT, and 100Plant Species(data Sha 64, data Tex 64 and data Mar 64). The subset scale and attribute dimension are as shown in Table 5. Their universality lies on their being classification task, full data type, small scaled training set, and convenient to test. Other classification algorithms chosen to compare with the GVCA method involves the decision tree, naive Bayesian classifier, K-neighborhood, SVM (polynomial kernel), and SVM (Gaussian kernel). All of them are commonly used classification algorithm with good performance. Using Weka [40] as the experimental platform, the comparison was made until the performance had been regulated to a good state through a parameter selection method [41] in the experiment. In view of possible default in some certain data, the decision tree J48 algorithm was used together with Laplace smoothing method. The K-neighborhood employed IBk algorithm and set the neighborhood parameter to be 7. The SVM used SMO algorithm, and considering the computational complexity, the exponent parameter of polynomial kernel was set to be 2, and the gamma parameter of Gaussian kernel was set as 0.01.

4.2 An experiment studying the influence of the proportion of vanishing component polynomial of classification decision function in the GVCA method on the classification performance

Likewise, the paper used the following experiment to verify the rationality of classification decision function in

Table 7 The influence of the horizontal grouping of simulation dataset in the GVCA method on the classification performance

N	σ				
	0.02	0.04	0.06	0.08	0.10
10	0.67	0.73	0.76	0.69	0.67
20	0.84	0.87	0.87	0.88	0.89
30	0.84	0.85	0.84	0.87	0.86
40	0.88	0.87	0.82	0.85	0.84
50	0.86	0.84	0.82	0.83	0.83

Table 8 The influence of the horizontal grouping of UCI dataset in the GVCA method on classification performance

Group size	Wine			Transfusion			Connectionist Bench			Breast Cancer Wisconsin		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
10	0.90	0.86	0.88	0.72	0.60	0.66	0.84	0.76	0.79	0.86	0.94	0.90
20	0.94	0.90	0.92	0.91	0.51	0.65	0.75	0.72	0.73	0.85	0.93	0.89
30	0.94	0.90	0.92	0.74	0.60	0.66	0.90	0.87	0.89	0.96	0.97	0.96
40	0.94	0.92	0.93	0.62	0.58	0.60	0.89	0.85	0.87	0.91	0.96	0.94
50	0.95	0.93	0.94	0.77	0.63	0.69	0.84	0.81	0.82	0.88	0.94	0.91
Group size	Indian Liver Patient			Mammographic Masses			Iris			AVG		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
10	0.58	0.58	0.58	0.81	0.81	0.81	0.74	0.63	0.68	0.78	0.74	0.76
20	0.56	0.57	0.56	0.82	0.81	0.81	0.96	0.95	0.95	0.83	0.77	0.79
30	0.58	0.54	0.56	0.81	0.79	0.80	0.94	0.93	0.93	0.84	0.80	0.82
40	0.58	0.58	0.58	0.80	0.80	0.80	0.94	0.93	0.93	0.81	0.80	0.81
50	0.56	0.55	0.56	0.80	0.77	0.78	0.84	0.67	0.74	0.81	0.77	0.78

the proposed GVCA method. The experiment aims to examine the influence of sorting-purpose polynomial proportion on the performance. In the experiment, the results of polynomial were sorted in an order from small to large according to their absolute values, and then the class with the maximum numbers in top ranked $N\%$ of the total number was determined as the class label. The experimental results are as shown in Table 6. This indicates when $N\%$ increases to a certain quantity, specifically in this table, when $N\% = 20\%$ until $N\% = 70\%$, the experimental performance remains stable. When $N\% = 80\%$, the performance turns to be better yet with limited superiority. So, there is a stable interval from 20 to 70% in the number of vanishing component polynomial required to sort in classification decision function of the GVCA method. During this range, there is no large change in performance. For this reason, the proportion of sorted vanishing component polynomial adopted in this paper is $N\% = 20\%$.

4.3 Experiments studying the influence of the size of grouped training set in the GVCA method on the classification performance

A. An experiment conducted on simulation dataset

An experiment was conducted on simulation dataset to study the relation between the size of grouped training set and the classification performance. Firstly, this paper used a simulation dataset and sampled all experimental data from the above formulas (6) and (7), with all examples added with Gaussian noise at $\mu = 0$, $\sigma = 0.02$, $\sigma = 0.04$, ..., $\sigma = 0.10$. In order to generate balanced vanishing component polynomial, the number of two classes of examples was set to be equal. Due to small eigenvector dimension in simulation dataset, only horizontal grouping experiment was implemented. The size of horizontally grouped training set was set as 10/20/30/40/50. The experimental results are as shown in Table 7, in which, N represents the size of grouped dataset, and the experimental performance is represented by F1 value corresponding to different σ . It can be known from Table 7 that, though the values

Table 9 The influence of the vertical grouping of UCI dataset in the GVCA method on classification performance

Group number	Wine			Transfusion			Connectionist Bench			Breast Cancer Wisconsin		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
2	0.90	0.88	0.89	0.92	0.73	0.81	0.76	0.82	0.79	0.91	0.90	0.90
3	0.89	0.92	0.90	0.87	0.79	0.83	0.86	0.85	0.85	0.89	0.87	0.88
4	0.86	0.90	0.88	0.76	0.80	0.78	0.87	0.89	0.88	0.91	0.85	0.88
Group number	Indian Liver Patient			Mammographic Masses			Iris			AVG		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
2	0.86	0.87	0.86	0.87	0.87	0.87	0.92	0.90	0.91	0.83	0.77	0.79
3	0.85	0.86	0.85	0.83	0.80	0.81	0.90	0.93	0.91	0.84	0.80	0.82
4	0.83	0.87	0.85	0.81	0.80	0.81	0.88	0.91	0.89	0.86	0.80	0.83

Table 10 Comparison of the classification performance on simulation dataset by the GVCA method and other classification algorithms

Algorithm	Precision	Recall	F1
DT	0.90	0.88	0.89
NB	0.87	0.84	0.85
KNN	0.83	0.76	0.79
POLYK	0.78	0.65	0.71
RBFK	0.76	0.72	0.74
GVCA	0.94	0.92	0.93

of standard deviation σ are varied, the optimum experimental performance appears under a moderate scale of grouped dataset (that is $N = 20$ to $N = 40$, in this table). The reason of this phenomenon is the undersized or oversized scale of grouped dataset will result in under-fitting or over-fitting.

B. An experiment conducted on UCI dataset

Below, UCI standard dataset was chosen to study the influence of the size of grouped training set in the GVCA method on the classification performance.

Horizontal grouping means to group the dataset example. The paper tested seven subdatasets contained in UCI standard dataset, such as Wine, Transfusion, Connectionist Bench, Breast Cancer Wisconsin, Indian Liver Patient, Mammographic Masses, and Iris. The number of grouped dataset was 10/20/30/40/50, respectively. The evaluation criterion was set as Precision, Recall, and F1. After several tests, the performance of different training sets was averaged, see results in Table 8. This set of experimental results shows a total trend: when the

grouped scale in the GVCA method is placed in a middle position, that is, taking 30 or 40 examples to compose one subgroup, the obtained performance will be the best. And the performance may slightly decrease with a lower or higher value than 30 and 40. The reason for this is when the grouping scale is moderate, neither under-fitting nor over-fitting will happen. Therefore, the GVCA method can acquire an optimum performance in the vertical grouping test.

Vertical grouping means to group the eigenvector dimension. After several tests, the performance of different training sets was averaged, see results in Table 9.

4.4 Experiments comparing the performance of the GVCA method and other classification algorithms

4.4.1 An experiment conducted on simulation dataset

This paper utilized simulation dataset to compare the GVCA method with other classification algorithms. The experimental data were still sampled from the above formulas (6) and (7), with all examples added with Gaussian noise at $\mu=0, \sigma=0.02 (\sigma=0.06, \sigma=0.1)$. The training set used 10, 20, 30, 40, 50, 100, 150, and 200 examples, respectively. The machine learning methods for contrast are decision tree, naive Bayesian classifier, K-neighborhood, SVM (polynomial kernel), and SVM (Gaussian kernel). The evaluation criterion was set as Precision, Recall, and F1. After several tests, the results of different training set sizes were averaged, see results in Table 10, in which DT represents naive Bayesian classifier, KNN represents K-neighborhood, POLYK represents SVM (polynomial kernel), and RBFK represents

Table 11 Comparison of the classification performance on UCI dataset by the GVCA method and other classification methods

Algorithm	Wine			Transfusion			Connectionist Bench			Breast Cancer Wisconsin		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
DT	0.94	0.90	0.92	0.84	0.82	0.82	0.75	0.73	0.74	0.97	0.97	0.97
NB	0.90	0.86	0.88	0.81	0.83	0.80	0.66	0.64	0.65	0.98	0.98	0.98
KNN	0.99	0.99	0.99	0.77	0.81	0.78	0.82	0.81	0.81	0.96	0.97	0.96
POLYK	0.94	0.92	0.93	0.79	0.82	0.75	0.97	0.97	0.97	0.98	0.98	0.98
RBFK	0.94	0.90	0.92	0.67	0.82	0.73	0.66	0.64	0.65	0.99	0.99	0.99
GVCA	0.94	0.90	0.92	0.77	0.63	0.69	0.90	0.87	0.89	0.97	0.96	0.96
Algorithm	Indian Liver patient			Mammographic Masses			Iris			AVG		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
DT	0.56	0.75	0.64	0.93	0.93	0.93	0.81	0.73	0.77	0.86	0.84	0.85
NB	0.84	0.50	0.51	0.88	0.86	0.86	0.76	0.64	0.69	0.83	0.81	0.81
KNN	0.86	0.83	0.80	0.80	0.79	0.79	0.73	0.78	0.75	0.86	0.88	0.87
POLYK	0.79	0.54	0.57	0.74	0.82	0.77	0.76	0.82	0.79	0.88	0.89	0.87
RBFK	0.56	0.75	0.64	0.64	0.64	0.64	0.80	0.64	0.71	0.81	0.82	0.81
GVCA	0.84	0.50	0.63	0.70	0.71	0.70	0.84	0.87	0.85	0.87	0.82	0.85

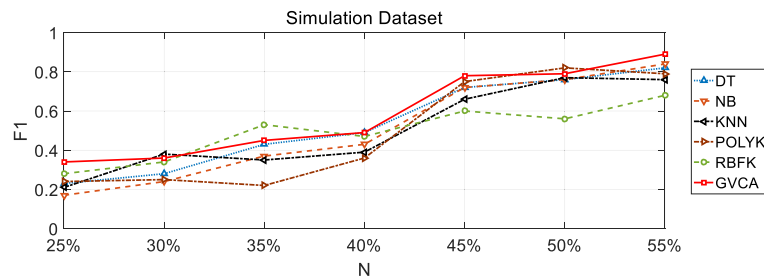


Fig. 1 Comparison of the convergence rate of simulation dataset by the GVCA method and other classification algorithms

SVM (Gaussian kernel). It can be discovered from Table 10 that the average performance of the GVCA method is higher than other five machine learning methods.

4.4.2 An experiment conducted on UCI dataset

This paper tested seven subdatasets contained in UCI standard dataset, such as Wine, Transfusion, Connectionist Bench, Breast Cancer Wisconsin, Indian Liver Patient, Mammographic Masses, and Iris. The evaluation criterion was set as Precision, Recall, and F1. After several tests, the performance of different training sets were averaged, see results in Table 11. These results indicate that compared to some classification algorithms, the proposed the GVCA method can achieve perfect average performance, which fully displays its strong stability and perfect performance.

4.5 Experiments comparing the convergence rate by the GVCA method and other classification algorithms

The following experiments aim to test the convergence performance by the GVCA method. For simulation dataset and UCI dataset, the 25, 30, 35, 40, 45, 50, and 55% of total example number were adopted as the training set, and the balanced ones were taken as testing set and compared with common machine learning methods. The evaluation indexes are F1 values corresponding to different training set scales and different classification algorithms.

4.5.1 An experiment conducted on simulation dataset

After several tests, the performance of different training sets was averaged, see results in Fig. 1.

4.5.2 An experiment conducted on UCI dataset

The paper tested the convergence rate of seven subdatasets contained in UCI standard dataset, such as Wine, Transfusion, Connectionist Bench, Breast Cancer Wisconsin, Indian Liver Patient, Mammographic Masses, and Iris. The experimental results are as shown in Fig. 2, which indicates the GVCA method can still obtain a good performance with a small training set scale (less than 50%) compared to other methods. This is because the Grobner basis acquired by the GVCA method on a moderate-scale grouped training set could well characterize the inner structure of a manifold pattern. Thus, the GVCA method can get a rapid rate of convergence and quickly achieve favorable learning performance.

5 Conclusions

This paper analyzed the characteristics and existing problems of the VCA method, and then improved it from both aspects to form a GVCA method. (1) The classification decision function is based on the sorting of values that vanishing components take from tested data, while the non-vanishing component makes its decision depending on the number of value being zero taken from tested data. This avoids the problem of not easily set threshold of classification decision function for the VCA method and enhances feasibility in real application.

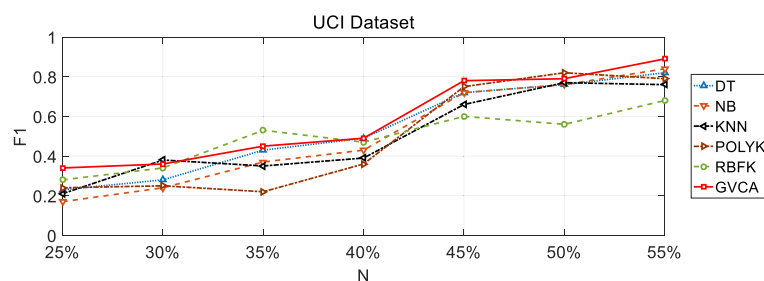


Fig. 2 Comparison of the convergence rate of UCI dataset by the GVCA method and other classification algorithms

(2) A strategy of grouping training set was proposed, which segmented training set into several non-intersecting subsets, solved the vanishing polynomial on subset, and combined them into the vanishing component polynomial set of an integral training set, to apply to solve the problem of large-scaled training set. (3) The integrated learning theory was utilized to prove the correctness of the strategy of grouping training set. The analysis of time complexity before and after grouping demonstrates it can effectively reduce computational time. A series of experimental results show that the GVCA method obtains a perfect experimental performance and quicker convergence compared to other classification algorithms.

In future, the dual relation between ideal space and kernel space can be further used to switch the computation of kernel space into that of ideal space, thereby realizing the application of commutative algebra to efficiently solve the machine learning problem.

Abbreviation

AVICA: Approximate vanishing ideal component analysis; DT: Decision tree; GVCA: Grouped vanishing component analysis; IPCA: Ideal principal component analysis; Kernel PCA: Kernel primary component analysis; KNN: K-nearest neighbor; NB: Naive Bayesian; PCA: Principal component analysis; POLYK: Support vector machine with polynomial kernel; RBFK: Support vector machine with Gaussian kernel; SD: Stochastic discrimination; SVD: Singular value decomposition; SVM: Support vector machine; UCI: University of California Irvine; VCA: Vanishing component analysis

Author's contributions

XZ is the writer of this paper. He proposed the main idea, deduced the performance of the GVCA method, completed the simulation, and analyzed the result. The author read and approved the final manuscript.

Competing interests

The author declares that he has no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 7 March 2018 Accepted: 18 April 2018

Published online: 08 May 2018

References

1. D Eisenbud, Commutative algebra: With a view toward algebraic geometry. Springer Science & Business Media **150** (2013)
2. M. Waldschmidt, "Diophantine Approximation and Diophantine Equations," 2011.
3. JB Tenenbaum, V De Silva, JC Langford, A global geometric framework for nonlinear dimensionality reduction. *Science* **90**(5500), 2319–2323 (2000)
4. R Livni, D Lehari, S Schein, H Nachlieli, S Shalev-Shwartz, A Globerson, in *Proceedings of The 30th International Conference on Machine Learning*. Vanishing component analysis (2013), pp. 597–605
5. D Lazard, in *Computer algebra*. Gröbner bases, Gaussian elimination and resolution of systems of algebraic equations (Springer, 1983), pp. 146–156
6. N Cristianini, J Shawe-Taylor, *An introduction to support vector machines and other kernel-based learning methods* (Cambridge University Press, 2000)
7. CM Bishop et al., *Pattern recognition and machine learning*, vol 4 (Springer, New York, 2006), p. no. 4
8. CF Gauss, *Disquisitiones arithmeticae*, vol 157 (Yale University Press, 1966)
9. EE Kummer, *De numeris complexis, qui radicibus unitatis et numeris integris realibus constant* (Gratulationsschrift der Univ. Breslau in Jubelfeier der Univ., Königsberg, 1844), pp. 185–212
10. R Dedekind, *Theory of Algebraic Integers* (Cambridge University Press, 1996)
11. R Courant, D Hilbert, *Methods of mathematical physics*, vol 1 (CUP Archive, 1966)
12. S Lang, *Algebraic number theory*, vol 110 (Springer Science & Business Media, 2013)
13. B Stenström, *Rings of quotients: An introduction to methods of ring theory*, vol 217 (Springer Science & Business Media, 2012)
14. G Dedekind, "Belt press apparatus with heat shield," Jun. 22 1982, uS Patent 4,336,096.
15. MF Atiyah, IG Macdonald, *Introduction to commutative algebra*, vol 2 (Addison-Wesley Reading, 1969)
16. GE Noether, *On a theorem of pitman* (The Annals of Mathematical Statistics, 1955), pp. 64–68
17. N Bourbaki, *Algebra I: chapters 1–3* (Springer Science & Business Media, 1998)
18. CA Weibel, *An introduction to homological algebra* (Cambridge university press, 1995), p. no. 38
19. O Endler, W Krull, *Valuation Theory* (Springer, 1972)
20. HM Möller, B Buchberger, *The construction of multivariate polynomials with preassigned zeros* (Springer, 1982)
21. D Heldt, M Kreuzer, S Pokutta, H Poulisse, Approximate computation of zero-dimensional polynomial ideals. *J. Symb. Comput.* **44**(11), 1566–1591 (2009)
22. GH Golub, CF Van Loan, *Matrix computations*, vol 3 (JHU Press, 2012)
23. RM Corless, PM Gianni, BM Trager, SM Watt, in *Proceedings of the 1995 international symposium on Symbolic and algebraic computation*. The singular value decomposition for polynomial systems (ACM, 1995), pp. 195–207
24. HJ Stetter, *Numerical polynomial algebra* (SIAM, 2004)
25. NJ Higham, *Accuracy and stability of numerical algorithms* (SIAM, 2002)
26. AM Garsia, Combinatorial methods in the theory of Cohen-Macaulay rings. *Adv. Math.* **38**(3), 229–266 (1980)
27. T Sauer, Approximate varieties, approximate ideals and dimension reduction. *Numerical Algorithms* **45**(1–4), 295–313 (2007)
28. F J. Király, M. Kreuzer, and L. Theran, "Dual-to-kernel learning with ideals," arXiv preprint arXiv:1402.0099, 2014.18.
29. S Mika, B. Schölkopf, A. J. Smola, K.-R. Müller, M. Scholz, and G. Rätsch, "Kernel pca and de-noising in feature spaces." in NIPS, vol. 4, no. 5. Citeseer, 1998, p. 7.
30. E Kleinberg, *Stochastic discrimination* (Annals of Mathematics and Artificial Intelligence, 1990)
31. TK Ho, The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Mach. Intell.* **20**(8), 832–844 (1998)
32. LK Hansen, P Salamon, Neural network ensembles. *IEEE Trans. Pattern Anal. Mach. Intell.* **12**(10), 993–1001 (1990)
33. L Breiman, *Bagging predictors* (Machine Learning, 1996)
34. RE Schapire, "A brief introduction to boosting," 1999.
35. RE Schapire, Y Freund, P Bartlett, WS Lee, Boosting the margin: A new explanation for the effectiveness of voting methods. *Ann. Stat.* **26**(5), 1651–1686 (1998)
36. S Cho, JH Kim, Multiple network fusion using fuzzy logic. *IEEE Trans. Neural Netw.* **6**(2), 497–501 (1995)
37. DH Wolpert, Stacked generalization. *Neural Networks* **5**(2), 241–259 (1992)
38. WVD Hodge, W Hodge, D Pedoe, *Methods of algebraic geometry*, vol 2 (Cambridge University Press, 1994)
39. WW Adams, P Lounstaunau, *An introduction to Grobner bases* (American Mathematical Soc., 1994)
40. R Bryll, R Gutierrezosuna, F Quek, Attribute bagging: Improving accuracy of classifier ensembles by using random feature subsets. *Pattern Recogn.* **36**(6), 1291–1302 (2003)
41. M Hall, E Frank, G Holmes, B Pfahringer, P Reutemann, IH Witten, The Weka data mining software: An update. *ACM SIGKDD explorations newsletter* **11**(1), 10–18 (2009)