Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

# Discriminative Vanishing Component Analysis

**Chenping Hou**[1], **Feiping Nie**[2], **Dacheng Tao**[3]
[1]College of Science, National University of Defense Technology
[2]Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University
[3]Center for Quantum Computation and Intelligent Systems and the Faculty of Engineering and Information Technology, University of Technology, Sydney
hcpnudt@hotmail.com, feipingnie@gmail.com, dacheng.tao@uts.edu.au

## Abstract

Vanishing Component Analysis (VCA) is a recently proposed prominent work in machine learning. It narrows the gap between tools and computational algebra: the vanishing ideal and its applications to classification problem. In this paper, we will analyze VCA in the kernel view, which is also another important research direction in machine learning. Under a very weak assumption, we provide a different point of view to VCA and make the kernel trick on VCA become possible. We demonstrate that the projection matrix derived by VCA is located in the same space as that of Kernel Principal Component Analysis (KPCA) with a polynomial kernel. Two groups of projections can express each other by linear transformation. Furthermore, we prove that KPCA and VCA have identical discriminative power, provided that the ratio trace criteria is employed as the measurement. We also show that the kernel formulated by the inner products of VCA's projections can be expressed by the KPCA's kernel linearly. Based on the analysis above, we proposed a novel Discriminative Vanishing Component Analysis (DVCA) approach. Experimental results are provided for demonstration.

## Introduction

Feature extraction is one of the most important topics in machine learning. Primarily, it addresses the problem of finding the most relevant and informative set of features. It has been commonly recognized that the effectiveness of feature extraction takes great influence on the subsequent procedures, such as classification and clustering (Guyon and Elisseeff 2003). Because of its importance, feature extraction still attracts a lot of research efforts nowadays, although it is a traditional topic and there is much literature already.

In the literature, there are mainly two distinct ways for feature extraction. The first kind of approaches are performed in the original data space. They use the statistical characteristics or similarity measurements of the original data for feature extraction. Typical statistical property based methods include ReliefF (Robnik-Sikonja and Kononenko 2003), Fisher Score (Koller and Sahami 1996) and Laplacian Score (He, Cai, and Niyogi 2006), etc. Similarity-based methods include linear methods, e.g., Principal Component Analysis (PCA) and Linear Discriminative Analysis (LDA) (Bishop 2006), and other nonlinear dimensionality reduction approaches (Van der Maaten, Postma, and den Herik 2009).

The second way to do feature extraction is to transform the original data into another space first. For example, the kernel methods can characterize the original data in a Reproducing Kernel Hilbert Space (RKHS). By kernel trick, they just need the kernel matrix, without knowing the explicit mapping function from the original data space to RHS (Schölkopf and Smola 2001). Due to their effectiveness, the kernel methods have been regarded as one of the most important tools for data mining. Compared with the first kind of approaches, the second type of methods have more flexibility in characterizing original data and have attracted a lot of research interests.

Recently, Livni et al have proposed a novel approach named Vanishing Component Analysis (VCA) (Livni et al. 2013), which belongs to the above-mentioned second category. VCA uses the concept vanishing ideal in computational algebra for data transformation and feature selection, solving the feature extraction problem in a distinctive way. It has sound theoretical foundation and great values in applications. It not only provides a new perspective on the feature extraction problem in machine learning, but also makes contributions to the mathematical area concerning vanishing ideal.

In this paper, we aim to analyze this prominent work from the kernel view. Under a weak assumption, we prove that the projection matrix derived by VCA is located in the same space as that of Kernel Principal Component Analysis (KPCA) (Schölkopf, Smola, and Müller 1998) with polynomial kernel on each category respectively. The explicit linear transformation matrices between two groups of transformed data points are derived. Furthermore, we demonstrate that different mapping results of KPCA and VCA have equal discriminative power if the ratio trace criteria in traditional LDA is adopted for measurement. The linear relations between the kernels of KPCA and VCA are also revealed. Based on the above analysis, we propose a new Discriminative Vanishing Component Analysis (DVCA) approach. The main contributions of this paper include,

(1) We have analyzed VCA in kernel view. The results reveal the relationship between the two types of prominent research, kernel methods and VCA. It also provides a new perspective of VCA and deepens the understanding of the

relations of these approaches.

(2) We have proposed an improved approach, i.e., DVCA. Compared with VCA, it takes the discriminative information among different classes into consideration while VCA does not. Experimental results are provided to show its advantages.

## VCA Revisited

VCA is a novel supervised approach which provides us a new perspective of feature extraction and tightens the relationship between the concept of vanishing ideal in algebra and its application in machine learning. Let us introduce some notation and basic definitions first.

Assume that $\mathbf{X}_i = [\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, \cdots, \mathbf{x}_{n_i}^{(i)}]$ represents the data points in the $i^{\text{th}}$ category and $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_c]$ contains all data points in the original $D$-dimensional space. Here, $n_i$ is the number of points in the class $i$, $c$ is the number of classes and $n$ is the total number of training points, i.e., $n = \sum_{i=1}^{c} n_i$. The set of all polynomials $f(\mathbf{x})$ that satisfied $f(\mathbf{x}) = 0$ for any $\mathbf{x} \in \mathcal{S}$ is known as the vanishing ideal of $\mathcal{S}$, where $\mathcal{S}$ is a data set. It is denoted as $I(\mathcal{S})$. If a set of polynomials can generate $I(\mathcal{S})$ by common operations between them, then this set of polynomials is called a set of generators of $I(\mathcal{S})$. The elements of such a finite set of generators is named as Vanishing Components.

The basic idea of VCA is finding a finite set of generators or vanishing components for the data points in each category. Then, it combines all the vanishing components of each category and uses this combination as the transformation function for all data points. Intuitively, if a function is a vanishing component of the vanishing ideal of $\mathbf{X}_i$, then it will attain zero values on the points belonging to the $i^{\text{th}}$ class, but not necessary to be zero on data points of other categories. By this way, VCA extracts the discriminative information from the data and consequently, it is more convenient to use this kind of representations for the following classification task. In (Livni et al. 2013), the authors provide a quick and effective way to fulfill this goal. It can also be regarded as great progress in the mathematical fields concerning vanishing ideal computation. See more details about VCA there.

## VCA vs KPCA

Since VCA focuses on finding the polynomials with the vanishing property, we directly investigate polynomial kernels. Recalling the results in kernel theory, we know that the polynomial kernel element computed by $(1 + \mathbf{x}^T\mathbf{y})^d$ can be formulated by the inner product of two vectors $\phi(\mathbf{x})$ and $\phi(\mathbf{y})$, where $\phi(\cdot)$ is the mapping function corresponding to this kernel. If we use the polynomial kernel with degree $d$, previous results show that the function $\phi(\cdot)$ can be derived explicitly. It is just the linear combination of all the scaled versions of the monomials with degrees no larger than $d$ (Schölkopf and Smola 2001). Since we only refer to linear transformation, we use the monomials with degrees no larger than $d$ for simplicity in the following deduction, and it takes no effects on the following theoretical results.

KPCA with polynomial kernels can be regarded as performing traditional PCA on the mapping data. The kernel trick is an elegant method to avoid manipulation in high dimensional space explicitly. Using the kernel trick, we just need the kernel and it is not necessary to know the exact mapping function $\phi(\cdot)$. In the following, unless otherwise specified, the kernel that KPCA uses is polynomial.

Based on the results in Proposition 3.1, Theorem 3.2 and the procedures of VCA in (Livni et al. 2013), for the $i^{\text{th}}$ class, VCA aims to find the generators of the null space of $\mathbf{\Phi}_i$, where $\mathbf{\Phi}_i = [\phi(\mathbf{x}_1^{(i)}), \phi(\mathbf{x}_2^{(i)}), \cdots, \phi(\mathbf{x}_{n_i}^{(i)})]$ consists of the mapping results of the data points in the $i^{\text{th}}$ category, by above-mentioned $\phi(\cdot)$. Denote $\mathbf{S}_i$ as the matrix formulated by vectors that span the null space of $\mathbf{\Phi}_i$. $\mathbf{S}_i$ is the $i^{\text{th}}$ group of projecting vectors of VCA, satisfying $\mathbf{S}_i^T\mathbf{\Phi}_i = \mathbf{0}$.

Before analyzing VCA in kernel view, we would like to explain the intuition of our work. The Theorem 3.2 in (Livni et al. 2013) argues that the kernel trick cannot be used to find the vanishing components. The reason is that the authors only use the kernel feature maps of the training points on which the polynomials should vanish. To overcome this problem, we manage to construct vanishing components from the linear combinations of kernel feature maps using *all* training points across classes.

To analyze this problem, we first reveal the intrinsic relationship between KPCA and VCA. Let us discuss the relationship between the linear spaces that they are located in. Denote $\mathbf{\Phi} = [\mathbf{\Phi}_1, \mathbf{\Phi}_2, \cdots, \mathbf{\Phi}_c]$. Commonly, the dimensionality of the mapped data is very high. Thus, it is reasonable to assume that the columns of $\mathbf{\Phi}$ consist of linearly independent vectors. Besides, we can also neglect the projections derived by $\mathbf{S}_i$ if they locate in the null space of all data points, because they project all data points to zero values and eliminating these dimensions with totally zero values makes no difference. For this reason, without loss of generality, we assume that $\text{span}(\mathbf{S}_i) \subset \text{span}(\mathbf{\Phi})$, where $\text{span}(\cdot)$ denotes the space spanned by the columns of the corresponding matrices.

Under these weak assumptions, the space spanned by the projection matrix of VCA is the same as that of the space spanned by all data points.

**Theorem 1.** *Assume that the columns of* $\mathbf{\Phi}$ *are linear independent. Then,* $\text{span}([\mathbf{S}_1, \mathbf{S}_2, \cdots, \mathbf{S}_c]) = \text{span}(\mathbf{\Phi})$.

*Proof.* For any $i \neq j$, we will prove that $\text{span}([\mathbf{S}_i, \mathbf{S}_j]) = \text{span}(\mathbf{\Phi})$.

On one side, we know that $\text{span}(\mathbf{S}_i) \subset \text{span}(\mathbf{\Phi})$ for $i = 1, 2, \cdots, c$. Then, we know that $\text{span}([\mathbf{S}_i, \mathbf{S}_j]) \subset \text{span}(\mathbf{\Phi})$.

On the other side, for any vector $\xi \in \text{span}(\mathbf{\Phi}_i)$, it is not orthogonal to $\text{span}(\mathbf{S}_j)$. Otherwise, $\xi \in \text{span}(\mathbf{\Phi}_j)$. In other words, there is a vector located in both $\text{span}(\mathbf{\Phi}_i)$ and $\text{span}(\mathbf{\Phi}_j)$. It conflicts with the assumption that the columns of $\mathbf{\Phi}$ are linear independent.

Moreover, if $\text{span}([\mathbf{S}_i, \mathbf{S}_j]) \neq \text{span}(\mathbf{\Phi})$, there is a vector $\xi \in \text{span}(\mathbf{\Phi})$ which is orthogonal to $\text{span}([\mathbf{S}_i, \mathbf{S}_j])$. It indicates that $\xi \in \text{span}(\mathbf{\Phi}_i)$ and $\xi$ is orthogonal to $\text{span}(\mathbf{S}_j)$. It conflicts with the above statement. □

We will show that the projections derived by KPCA and VCA are related by a linear transformation. Based on the

above theorem, $[\mathbf{S}_1, \mathbf{S}_2, \cdots, \mathbf{S}_c]$ can be formulated as

$$[\mathbf{S}_1, \mathbf{S}_2, \cdots, \mathbf{S}_c] = \mathbf{\Phi}\mathbf{G}, \tag{1}$$

where $\mathbf{G}$ is a matrix formulated by all coefficients. It has full row rank since $\text{rank}([\mathbf{S}_1, \mathbf{S}_2, \cdots, \mathbf{S}_c]) = \text{rank}(\mathbf{\Phi})$.

Assume that the SVD decomposition of $\mathbf{\Phi}$ is $\mathbf{\Phi} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ and $\mathbf{B} = \mathbf{G}^T\mathbf{V}\mathbf{\Sigma}$, where we only pick up the non-zero singular values and the corresponding singular vectors. It is named as thin SVD in the following parts. Denote the projection results derived by KPCA and VCA as $\mathbf{P}$ and $\bar{\mathbf{P}}$ respectively. Then, $\mathbf{P} = \mathbf{U}^T\mathbf{\Phi}$ and $\bar{\mathbf{P}} = [\mathbf{S}_1, \mathbf{S}_2, \cdots, \mathbf{S}_c]^T\mathbf{\Phi}$. Their relationship is shown in Theorem 2.

**Theorem 2.** *$\bar{\mathbf{P}} = \mathbf{B}\mathbf{P}$ and $\mathbf{P} = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\bar{\mathbf{P}}$.*

*Proof.* Note that

$$\begin{aligned}\bar{\mathbf{P}} &= [\mathbf{S}_1, \mathbf{S}_2, \cdots, \mathbf{S}_c]^T\mathbf{\Phi} = (\mathbf{\Phi}\mathbf{G})^T\mathbf{\Phi} \\ &= (\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{G})^T\mathbf{\Phi} = \mathbf{G}^T\mathbf{V}\mathbf{\Sigma}\mathbf{U}^T\mathbf{\Phi} = \mathbf{B}\mathbf{P}.\end{aligned} \tag{2}$$

Recall the definition $\mathbf{B} = \mathbf{G}^T\mathbf{V}\mathbf{\Sigma}$ and the fact that $\mathbf{G}$ is a matrix with full row rank, we know that $\mathbf{B}$ is a matrix with full column rank. In other words, $\mathbf{B}^T\mathbf{B}$ is an invertible matrix. Then,

$$\begin{aligned}&\bar{\mathbf{P}} = \mathbf{B}\mathbf{P} \Rightarrow \mathbf{B}^T\bar{\mathbf{P}} = \mathbf{B}^T\mathbf{B}\mathbf{P} \\ &\Rightarrow \mathbf{P} = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\bar{\mathbf{P}}.\end{aligned} \tag{3}$$

□

From Theorem 1 we know that these two projections share the same linear space, therefore, it is not surprising that they can express each other by linear transformations. Intuitively, they should have similar discriminative power. The following results show that the discriminative power of the two projections is equal, provided that the ratio trace criteria in LDA is employed for measurement. The reason why we use ratio trace criteria is that it is one of the most popular metrics in supervised learning. Before going into the details, we provide some lemmas.

**Lemma 1.** *For any matrix $\mathbf{A}$, if $\mathbf{U}^T\mathbf{U} = \mathbf{I}$, we have the equation $(\mathbf{U}\mathbf{A}\mathbf{U}^T)^+ = \mathbf{U}\mathbf{A}^+\mathbf{U}^T$, where $(\cdot)^+$ denotes the Moore-Penrose inverse of a matrix.*

*Proof.* Assume that the thin SVD decomposition of $\mathbf{A}$ is $\mathbf{A} = \mathbf{U}_A\mathbf{\Sigma}_A\mathbf{V}_A^T$. Then,

$$\begin{aligned}&(\mathbf{U}\mathbf{A}\mathbf{U}^T)^+ = (\mathbf{U}\mathbf{U}_\mathbf{A}\mathbf{\Sigma}_\mathbf{A}\mathbf{V}_\mathbf{A}^\mathbf{T}\mathbf{U}^T)^+ \\ =&(\mathbf{U}\mathbf{U}_A\mathbf{\Sigma}_A(\mathbf{U}\mathbf{V}_A)^T)^+ = \mathbf{U}\mathbf{V}_A\mathbf{\Sigma}_A^{-1}(\mathbf{U}\mathbf{U}_A)^T \\ =&\mathbf{U}(\mathbf{V}_A\mathbf{\Sigma}_A^{-1}\mathbf{U}_A^T)\mathbf{U}^T = \mathbf{U}\mathbf{A}^+\mathbf{U}^T.\end{aligned} \tag{4}$$

□

**Lemma 2.** *Assume that $\mathbf{A}$ is an invertible matrix and $B$ is a matrix with full column rank, we have the equation $(\mathbf{B}\mathbf{A}\mathbf{B}^T)^+ = \mathbf{B}\mathbf{A}^+\mathbf{B}^T$.*

*Proof.* Assume the thin SVD decomposition of $\mathbf{B}$ is $\mathbf{B} = \mathbf{U}_B\mathbf{\Sigma}_B\mathbf{V}_B^T$. Then,

$$\begin{aligned}&(\mathbf{B}\mathbf{A}\mathbf{B}^T)^+ = (\mathbf{U}_B\mathbf{\Sigma}_B\mathbf{V}_B^T\mathbf{A}\mathbf{V}_B\mathbf{\Sigma}_B\mathbf{U}_B^T)^+ \\ =&\mathbf{U}_B(\mathbf{\Sigma}_B\mathbf{V}_B^T\mathbf{A}\mathbf{V}_B\mathbf{\Sigma}_B)^+\mathbf{U}_B^T \\ =&\mathbf{U}_B(\mathbf{\Sigma}_B\mathbf{V}_B^T\mathbf{A}\mathbf{V}_B\mathbf{\Sigma}_B)^{-1}\mathbf{U}_B^T \\ =&\mathbf{U}_B\mathbf{\Sigma}_B^{-1}\mathbf{V}_B^T\mathbf{A}^{-1}\mathbf{V}_B\mathbf{V}_B\mathbf{\Sigma}_B^{-1}\mathbf{U}_B^T = (\mathbf{B}^T)^+\mathbf{A}^{-1}\mathbf{B}^+.\end{aligned} \tag{5}$$

In the above deduction, the second equation holds based on Lemma 1. The third equation holds since $\mathbf{B}$ is a matrix with full column rank. □

**Lemma 3.** *If $\mathbf{B}$ is a matrix with full column rank, then $\mathbf{B}^T(\mathbf{B}^T)^+ = \mathbf{B}^+\mathbf{B} = \mathbf{I}$*

*Proof.* Assume that the thin SVD decomposition of $\mathbf{B}$ is $\mathbf{B} = \mathbf{U}_B\mathbf{\Sigma}_B\mathbf{V}_B^T$. Then,

$$\begin{aligned}&\mathbf{B}^T(\mathbf{B}^T)^+ = \mathbf{V}_B\mathbf{\Sigma}_B\mathbf{U}_B^T\mathbf{U}_B\mathbf{\Sigma}_B^{-1}\mathbf{V}_B^T = \mathbf{V}_B\mathbf{V}_B^T = \mathbf{I}, \\ &\mathbf{B}^+\mathbf{B} = \mathbf{V}_B\mathbf{\Sigma}_B^{-1}\mathbf{U}_B^T\mathbf{U}_B\mathbf{\Sigma}_B\mathbf{V}_B^T = \mathbf{V}_B\mathbf{V}_B^T = \mathbf{I}.\end{aligned} \tag{6}$$

□

**Lemma 4.** *Assume that $\mathbf{A}$ is an invertible matrix and $B$ is a matrix with full column rank. For any matrix $\mathbf{C}$, we have $\text{Tr}((\mathbf{B}\mathbf{A}\mathbf{B}^T)^+\mathbf{B}\mathbf{C}\mathbf{B}^T) = \text{Tr}(\mathbf{A}^{-1}\mathbf{C})$, where $Tr(\cdot)$ represents the trace of a matrix.*

*Proof.* Recalling Lemma2 and Lemma3, we have

$$\begin{aligned}&\text{Tr}((\mathbf{B}\mathbf{A}\mathbf{B}^T)^+\mathbf{B}\mathbf{C}\mathbf{B}^T) = \text{Tr}((\mathbf{B}^T)^+\mathbf{A}^{-1}\mathbf{B}^+\mathbf{B}\mathbf{C}\mathbf{B}^T) \\ =&\text{Tr}(\mathbf{A}^{-1}\mathbf{B}^+\mathbf{B}\mathbf{C}\mathbf{B}^T(\mathbf{B}^T)^+) = \text{Tr}(\mathbf{A}^{-1}\mathbf{C}).\end{aligned} \tag{7}$$

□

Based on the above four lemmas, we show our main results on the relationship between KPCA and VCA in terms of the discriminative power.

**Theorem 3.** *Recall that the projections derived by KPCA and VCA are $\mathbf{P}$ and $\bar{\mathbf{P}}$ respectively. If we adopt the same ratio trace criteria as in traditional LDA for measurement, then $\mathbf{P}$ and $\bar{\mathbf{P}}$ have the same discriminative power. In other words, the following equation holds.*

$$\text{Tr}((\bar{\mathbf{P}}\mathbf{L}_w\bar{\mathbf{P}}^T)^+(\bar{\mathbf{P}}\mathbf{L}_b\bar{\mathbf{P}}^T)) = \text{Tr}((\mathbf{P}\mathbf{L}_w\mathbf{P}^T)^{-1}(\mathbf{P}\mathbf{L}_b\mathbf{P}^T)), \tag{8}$$

*where $\mathbf{L}_b$ and $\mathbf{L}_w$ are the matrices concerning the between- and within- scatters and they are defined as follows.*

$$\begin{aligned}\mathbf{L}_w &= \mathbf{I} - \mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T. \\ \mathbf{L}_b &= \mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T.\end{aligned}$$

*where $\mathbf{H} = \{0,1\}^{n\times C}$ is an indicator matrix, i.e., $H_{ij} = 1$ if $\mathbf{x}_i$ belongs to the $j$th class and $H_{ij} = 0$ otherwise.*

*Proof.* Recalling the results in Theorem 2, we know that $\bar{\mathbf{P}} = \mathbf{B}\mathbf{P}$ and $\mathbf{B}$ is a matrix with full column rank. Based on Lemma 4, we have

$$\begin{aligned}&\text{Tr}((\bar{\mathbf{P}}\mathbf{L}_w\bar{\mathbf{P}}^T)^+(\bar{\mathbf{P}}\mathbf{L}_b\bar{\mathbf{P}}^T)) \\ =&\text{Tr}((\mathbf{B}\mathbf{P}\mathbf{L}_w\mathbf{P}^T\mathbf{B}^T)^+(\mathbf{B}\mathbf{P}\mathbf{L}_b\mathbf{P}^T\mathbf{B}^T)) \\ =&\text{Tr}((\mathbf{P}\mathbf{L}_w\mathbf{P}^T)^{-1}(\mathbf{P}\mathbf{L}_b\mathbf{P}^T)).\end{aligned} \tag{9}$$

The second equation holds since $\mathbf{P}\mathbf{L}_w\mathbf{P}^T$ is an invertible matrix as in traditional LDA.

Finally, we would like to reveal the relationship between KPCA and VCA. Interestingly, the kernel formulated by the inner products of VCA's projections can be expressed by that of KPCA linearly. □

**Theorem 4.** *Note that the projections derived by KPCA and VCA are* $\mathbf{P}$ *and* $\bar{\mathbf{P}}$ *respectively. The corresponding kernels are* $\mathbf{K} = \mathbf{P}^T\mathbf{P}$ *and* $\bar{\mathbf{K}} = \bar{\mathbf{P}}^T\bar{\mathbf{P}}$ *respectively. Then,* $\bar{\mathbf{K}} = \mathbf{KQ} = \mathbf{RK}$*, where* $\mathbf{Q} = \mathbf{G}\mathbf{G}^T\mathbf{\Phi}^T\mathbf{\Phi}$ *and* $\mathbf{R} = \mathbf{\Phi}^T\mathbf{\Phi}\mathbf{G}\mathbf{G}^T$.

*Proof.* Note that $\mathbf{\Phi} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ and $\mathbf{P} = \mathbf{U}^T\mathbf{\Phi}$, we have

$$\begin{aligned}\mathbf{K} &= \mathbf{P}^T\mathbf{P} = \mathbf{\Phi}^T\mathbf{U}\mathbf{U}^T\mathbf{\Phi} = \mathbf{V}\mathbf{\Sigma}\mathbf{U}^T\mathbf{U}\mathbf{U}^T\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \\ &= \mathbf{V}\mathbf{\Sigma}\mathbf{U}^T\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \mathbf{\Phi}^T\mathbf{\Phi}.\end{aligned} \tag{10}$$

Since $[\mathbf{S}_1, \mathbf{S}_2, \cdots, \mathbf{S}_c] = \mathbf{\Phi}\mathbf{G}$,

$$\begin{aligned}\bar{\mathbf{K}} &= \bar{\mathbf{P}}^T\bar{\mathbf{P}} = ([\mathbf{S}_1, \mathbf{S}_2, \cdots, \mathbf{S}_c]^T\mathbf{\Phi})^T[\mathbf{S}_1, \mathbf{S}_2, \cdots, \mathbf{S}_c]\mathbf{\Phi} \\ &= ((\mathbf{\Phi}\mathbf{G})^T\mathbf{\Phi})^T(\mathbf{\Phi}\mathbf{G})^T\mathbf{\Phi} = \mathbf{\Phi}^T\mathbf{\Phi}\mathbf{G}\mathbf{G}^T\mathbf{\Phi}^T\mathbf{\Phi} \\ &= \mathbf{KQ} = \mathbf{RK}.\end{aligned}$$

□

In summary, we have provided some theoretical results which can reveal the intrinsic relationship between KPCA and VCA from different aspects.

### Why kernel can help

We would like to explain why kernel can help. In VCA, the authors have proved the following result.

**Theorem 5.** *(Livni et al. 2013) Let* $\mathbf{K}_i$ *be the reproducing kernel and function* $f \in$ *span*$(\mathbf{K}_i(\cdot, \mathbf{x}_j^{(i)}))$ *such that* $f$ *vanishes on all* $\mathbf{x}_j^{(i)}$ *for* $j = 1, 2, \cdots, n_i$*. Then* $f$ *is the zero function.*

In this theorem, it is assumed that the function should have the form $f \in \text{span}(\mathbf{K}_i(\cdot, \mathbf{x}_j^{(i)}))$. It only concerns the points in the $i$-th class. Different from the setting of this theorem, we assume that the function should satisfy $f \in \text{span}(\mathbf{K}(\cdot, \mathbf{x}_j^{(i)}))$, which is formulated on all training points. In other words, Livin et al only use the kernel feature maps of the training points on which the polynomials should vanish, while we manage to construct vanishing components from the linear combinations of kernel feature maps using all training points across classes. It is more common in real applications as in most kernel learning algorithm (Schölkopf and Smola 2001). Thus, *our results do not conflict with Theorem 5 proved by Livin et al., but provide a different point of view to VCA and make the kernel trick on VCA become possible*.

Interestingly, if we constrain the function as the form concerning only the $i^{\text{th}}$ kernel $\mathbf{K}_i$, it is not surprising that the function $f$ with vanishing property is the zero function since it is located in the null space of $\mathbf{K}_i$.

## Discriminative Vanishing Component Analysis

In the section, we first reformulate VCA in the kernel format based on the above analysis. Then, by adding a discriminative objective function, we present our Discriminative Vanishing Component Analysis (DVCA) approach.

### Kernel Formulation of VCA

Based on the above explanations and the results in Theorem 1, we can reformulate VCA by finding $\{\mathbf{S}_i\}$ that satisfies:

$$\mathbf{S}_i^T\mathbf{\Phi}_i = \mathbf{0}, \;\; \mathbf{S}_i^T\mathbf{S}_i = \mathbf{I}, \;\; \mathbf{S}_i = \mathbf{\Phi}\mathbf{G}_i, \tag{11}$$

where $\mathbf{G}_i$ is formulated by the corresponding columns of $\mathbf{G}$ in formulating $\mathbf{S}_i$ as shown in Eq. (1).

Recall that Hilbert's basis theorem (Cox, Little, and O'Shea 2007) guarantees the existence of vanishing component. Considering the results in Theorem 1, we can approximate the above-mentioned problem in Eq. (11) as follows.

$$\begin{aligned}&\min_{\mathbf{S}_1, \mathbf{S}_2, \cdots, \mathbf{S}_c} \sum_{i=1}^{c}\sum_{j=1}^{n_i} \|\mathbf{S}_i^T\phi(\mathbf{x}_j^{(i)})\|_2^2 \\ &\text{s.t.} \quad \mathbf{S}_i = \mathbf{\Phi}\mathbf{G}_i, \;\; \mathbf{S}_i^T\mathbf{S}_i = \mathbf{I}, \;\; i = 1, 2, \cdots, c.\end{aligned} \tag{12}$$

where $\|\cdot\|_2$ denotes the 2-norm of a vector. Commonly, the objective function can attain zero value since the dimension of $\phi(\mathbf{x}_j^{(i)})$ is often much larger than the number of points belonging to the $i^{\text{th}}$ category.

This problem can be separated into $c$ sub-problems. For the $i^{\text{th}}$ problem, the objective function can be represented by kernel $\mathbf{K} = \mathbf{\Phi}^T\mathbf{\Phi}$ as defined in Theorem 4.

$$\begin{aligned}&\sum_{j=1}^{n_i} \|\mathbf{S}_i^T\phi(\mathbf{x}_j^{(i)})\|_2^2 = \sum_{j=1}^{n_i} \|\mathbf{G}_i^T\mathbf{\Phi}^T\phi(\mathbf{x}_j^{(i)})\|_2^2 \\ &= \text{Tr}(\mathbf{G}_i^T\mathbf{\Phi}^T\mathbf{\Phi}_i\mathbf{\Phi}_i^T\mathbf{\Phi}\mathbf{G}_i) = \text{Tr}(\mathbf{G}_i^T\mathbf{K}_i\mathbf{K}_i^T\mathbf{G}_i),\end{aligned} \tag{13}$$

where $\mathbf{K}_i$ is part of $\mathbf{K}$ and it is formulated by inner products between data points in $\mathbf{\Phi}$ and $\mathbf{\Phi}_i$, i.e., $\mathbf{K}_i = \mathbf{\Phi}^T\mathbf{\Phi}_i$.

Correspondingly, the constraints become

$$\mathbf{S}_i^T\mathbf{S}_i = \mathbf{I} \Rightarrow \mathbf{G}_i^T\mathbf{\Phi}^T\mathbf{\Phi}\mathbf{G}_i = \mathbf{I} \Rightarrow \mathbf{G}_i^T\mathbf{K}\mathbf{G}_i = \mathbf{I}. \tag{14}$$

### Discriminative Vanishing Component Analysis

In essence, VCA tries to find $\mathbf{S}_i$ that satisfy $\mathbf{S}_i^T\mathbf{\Phi}_i = 0$ . In other words, it computes the projection matrix $\mathbf{S}_i$ which can map the data points in the $i^{\text{th}}$ class to zero values. It does not consider the projections for points in other categories. Apparently, mapping data points in the $i^{\text{th}}$ category close to zeros and simultaneously mapping the points in other categories as far away from zeros as possible will benefit the subsequent classification.

In the literature, there are many approaches considering the dissimilarities between samples from different categories, such as Linear Discriminative Analysis (LDA) and its kernel extension (Mika et al. 1999), and Biased Discriminative Analysis (BDA) (Zhou and Huang 2001). In this paper, we adopt the idea from BDA. By adding the requirement that the mapping results of data points in different categories should be as far away from each other as possible, we propose our Discriminative Vanishing Component Analysis (DVCA) approach.

Mathematically, for the $i^{\text{th}}$ category, the discriminative requirement has the following objective function:

$$\max \sum_{j \neq i}\sum_{t=1}^{n_j} \left\| \mathbf{S}_i^T \left( \phi(\mathbf{x}_t^{(j)}) - \frac{1}{n_i}\sum_{s=1}^{n_i} \phi(\mathbf{x}_s^{(i)}) \right) \right\|^2. \tag{15}$$

It indicates that the mapping results of data from other categories should be far away from the center of $\{\phi(\mathbf{x}_s^{(i)})\}_{s=1}^{n_i}$.

Denote $\mathbf{e}_i = [1, 1, \cdots, 1]_{n_i}^T$ as a size $n_i$ column vector with all elements being 1. Note that $\mathbf{S}_i = \mathbf{\Phi}\mathbf{G}_i$, the above function can be reformulated as follow

$$\sum_{j\neq i}\sum_{t=1}^{n_j} \mathrm{Tr}(\mathbf{G}_i^T\mathbf{\Phi}^T(\phi(\mathbf{x}_t^{(j)}) - \frac{1}{n_i}\mathbf{\Phi}_i\mathbf{e}_i) (\phi(\mathbf{x}_t^{(j)}) - \frac{1}{n_i}\mathbf{\Phi}_i\mathbf{e}_i)^T\mathbf{\Phi}\mathbf{G}_i). \tag{16}$$

Denote $\tilde{\mathbf{\Phi}}_i$ as the matrix formulated by the points which do not belong to the $i^{\text{th}}$ category. Denote $\tilde{n}_i$ as the number of columns of $\tilde{\mathbf{\Phi}}_i$. Denote $\mathbf{f}_i = [1, 1, \cdots, 1]_{\tilde{n}_i}^T$ as a column vector with size $\tilde{n}_i$ and all 1 elements. Then, the above objective function can be reformulated as follows.

$$\begin{aligned}&\mathrm{Tr}\left(\mathbf{G}_i^T\mathbf{\Phi}^T\tilde{\mathbf{\Phi}}_i(\tilde{\mathbf{\Phi}}_i)^T\mathbf{\Phi}\mathbf{G}_i\right)\\ &+(\tilde{n}_i/n_i^2)\mathrm{Tr}\left(\mathbf{G}_i^T\mathbf{\Phi}^T\mathbf{\Phi}_i\mathbf{e}_i\mathbf{e}_i^T\mathbf{\Phi}_i^T\mathbf{\Phi}\mathbf{G}_i\right)\\ &-(2/n_i)\mathrm{Tr}\left(\mathbf{G}_i^T\mathbf{\Phi}^T\mathbf{\Phi}_i\mathbf{e}_i\mathbf{f}_i^T\tilde{\mathbf{\Phi}}_i^T\mathbf{\Phi}\mathbf{G}_i\right).\end{aligned} \tag{17}$$

Likewise, denote $\tilde{\mathbf{K}}_i$ as part of $\mathbf{K}$, which is formulated by inner products between data points in $\mathbf{\Phi}$ and $\tilde{\mathbf{\Phi}}_i$, i.e., $\tilde{\mathbf{K}}_i = \mathbf{\Phi}^T\tilde{\mathbf{\Phi}}_i$. The above formulation in Eq. (17) becomes

$$\mathrm{Tr}\Big(\mathbf{G}_i^T\Big(\tilde{\mathbf{K}}_i(\tilde{\mathbf{K}}_i)^T + (\tilde{n}_i/n_i^2)\mathbf{K}_i\mathbf{e}_i\mathbf{e}_i^T\mathbf{K}_i^T - (2/n_i)\mathbf{K}_i\mathbf{e}_i\mathbf{f}_i^T(\tilde{\mathbf{K}}_i)^T\Big)\mathbf{G}_i\Big). \tag{18}$$

By combining the objective functions in Eq. (13) and Eq. (18), and using the constraints in Eq. (14), we have the following formulation of DVCA.

$$\begin{aligned}&\min_{\mathbf{G}_1,\cdots,\mathbf{G}_c}\sum_{i=1}^{c}\mathrm{Tr}(\mathbf{G}_i^T\mathbf{K}_i\mathbf{K}_i^T\mathbf{G}_i) - \lambda\mathrm{Tr}\Big(\mathbf{G}_i^T\Big(\tilde{\mathbf{K}}_i(\tilde{\mathbf{K}}_i)^T\mathbf{G}_i\\ &+ (\tilde{n}_i/n_i^2)\mathbf{K}_i\mathbf{e}_i\mathbf{e}_i^T\mathbf{K}_i^T - (2/n_i)\mathbf{K}_i\mathbf{e}_i\mathbf{f}_i^T(\tilde{\mathbf{K}}_i)^T\Big)\mathbf{G}_i\Big)\\ &\text{s.t. } \mathbf{G}_i^T\mathbf{K}\mathbf{G}_i = \mathbf{I}.\end{aligned} \tag{19}$$

where $\lambda$ is a parameter to balance two factors, i.e., the vanishing property and the discriminative requirement.

## Solution

The optimization problem of DCA in Eq. (19) can be divided into $c$ sub-problems. Thus, in the following derivation, we only focus on the $i^{\text{th}}$ sub-problem.

Denote the eigen-decomposition of $\mathbf{K}$ as $\mathbf{K} = \mathbf{\Gamma}\mathbf{\Lambda}\mathbf{\Gamma}^T$, where $\mathbf{\Lambda}$ consists of all the non-zero eigen-values. Denote $\tilde{\mathbf{\Gamma}}$ as the matrix formulated by vectors that span the null space of $\mathbf{K}$. $\mathbf{G}_i$ can be expressed as $\mathbf{G}_i = \mathbf{\Gamma}\mathbf{W}_i + \tilde{\mathbf{\Gamma}}\mathbf{F}_i$.

For the sake of convenience, denote

$$\mathbf{M} = \tilde{\mathbf{K}}_i(\tilde{\mathbf{K}}_i)^T + (\tilde{n}_i/n_i^2)\mathbf{K}_i\mathbf{e}_i\mathbf{e}_i^T\mathbf{K}_i^T - (2/n_i)\mathbf{K}_i\mathbf{e}_i\mathbf{f}_i^T(\tilde{\mathbf{K}}_i)^T.$$

Then, the objective function and constraints in Eq. (19) become:

$$\begin{aligned}&\mathrm{Tr}(\mathbf{G}_i^T(\mathbf{K}_i\mathbf{K}_i^T + \lambda\mathbf{M})\mathbf{G}_i)\\ =&\mathrm{Tr}((\mathbf{\Gamma}\mathbf{W}_i + \tilde{\mathbf{\Gamma}}\mathbf{F}_i)^T(\mathbf{K}_i\mathbf{K}_i^T + \lambda\mathbf{M}))(\mathbf{\Gamma}\mathbf{W}_i + \tilde{\mathbf{\Gamma}}\mathbf{F}_i)\\ =&\mathrm{Tr}\left(\mathbf{W}_i^T\mathbf{\Gamma}^T(\mathbf{K}_i\mathbf{K}_i^T + \lambda\mathbf{M})\mathbf{\Gamma}\mathbf{W}_i\right).\end{aligned} \tag{20}$$

$$\begin{aligned}&\mathbf{G}_i^T\mathbf{K}\mathbf{G}_i = \mathbf{I} \Rightarrow \ (\mathbf{\Gamma}\mathbf{W}_i + \tilde{\mathbf{\Gamma}}\mathbf{F}_i)^T\mathbf{K}(\mathbf{\Gamma}\mathbf{W}_i + \tilde{\mathbf{\Gamma}}\mathbf{F}_i) = \mathbf{I}\\ &\Rightarrow \ \mathbf{W}_i^T\mathbf{\Gamma}^T\mathbf{K}\mathbf{\Gamma}\mathbf{W}_i = \mathbf{I} \Rightarrow \ \mathbf{W}_i^T\mathbf{\Lambda}\mathbf{W}_i = \mathbf{I}.\end{aligned} \tag{21}$$

The above two equations hold since $\tilde{\mathbf{\Gamma}}$ is in the null spaces of $\mathbf{K}$, and $\tilde{\mathbf{K}}_i$ and $\mathbf{K}_i$ are parts of $\mathbf{K}$.

Furthermore, $\mathbf{\Lambda}$ is invertible as $\mathbf{\Lambda}$ only consists of the non-zero eigen-values. Denote $\bar{\mathbf{W}}_i = \mathbf{\Lambda}^{1/2}\mathbf{W}_i$. By combining the objective function in Eq. (20) and the constraint in Eq. (21), DVCA can be reformulated as

$$\begin{aligned}&\min_{\bar{\mathbf{W}}_i}\mathrm{Tr}\left(\bar{\mathbf{W}}_i^T\mathbf{\Lambda}^{-1/2}\mathbf{\Gamma}^T(\mathbf{K}_i\mathbf{K}_i^T + \lambda\mathbf{M})\mathbf{\Gamma}\mathbf{\Lambda}^{-1/2}\bar{\mathbf{W}}_i\right)\\ &\text{s.t. } \bar{\mathbf{W}}_i^T\bar{\mathbf{W}}_i = \mathbf{I}.\end{aligned} \tag{22}$$

The optimal solution to the above problem can be derived by the eigen-decomposition strategy. After deriving the optimal $\bar{\mathbf{W}}_i$, we compute $\mathbf{W}_i = \mathbf{\Lambda}^{-1/2}\bar{\mathbf{W}}_i$ and $\mathbf{G}_i = \mathbf{\Gamma}\mathbf{W}_i$. We drop $\tilde{\mathbf{\Gamma}}\mathbf{F}_i$ since it locates in the null space of $\mathbf{K}$ and it contributes no discriminative information.

Another problem of the above solution is to determine how many eigen-vectors should be selected to form $\bar{\mathbf{W}}_i$. Recalling the intuition of VCA, in our algorithm, we only pick up the eigen-vectors which corresponds to the negative and zero eigen-values.

Finally, as in VCA, the $i^{\text{th}}$ group of projections of training points and testing points can also be computed by kernel trick. In other words, the $i^{\text{th}}$ projection of any training point $\mathbf{x}_j^{(t)}$ can be calculated by

$$\mathbf{S}_i^T\phi(\mathbf{x}_j^{(t)}) = \mathbf{G}_i^T\mathbf{\Phi}^T\phi(\mathbf{x}_j^{(t)}) = \mathbf{G}_i^T K_j^{(t)}, \tag{23}$$

where $K_j^{(t)}$ is the column of $\mathbf{K}$ corresponding to $\phi(\mathbf{x}_j^{(t)})$. Similarly, for any testing point $\mathbf{y}$, its $i^{\text{th}}$ projection can also be computed by

$$\mathbf{G}_i^T\mathbf{\Phi}^T\phi(\mathbf{y}) = \mathbf{G}_i^T\mathbf{K}_y, \tag{24}$$

where $\mathbf{K}_y$ is a $n$-dimensional column vector whose elements are calculated by $(1+\mathbf{x}^T\mathbf{y})^d$, where $\mathbf{x}$ represents the training data point. After calculating all the projections, we simply connect these $c$ projections to formulate a new representation of $\mathbf{y}$ as in VCA.

The procedure of DVCA is listed in Algorithm 1.

**Algorithm 1** Discriminative Vanishing Component Analysis (DVCA)

**Run:** Calculate the polynomial kernel matrix $\mathbf{K}$ by $\mathbf{x}_j^{(i)}$.

Calculate the polynomial kernel $\mathbf{K}_y$ between $\{\mathbf{x}_j^{(i)}\}$ and $\mathbf{y}$ if testing point is available.

**for** i=1:c **do**

Compute $\bar{\mathbf{W}}_i$ by solving Eq.(22).

Compute $\mathbf{W}_i = \mathbf{\Lambda}^{-1/2}\bar{\mathbf{W}}_i$ and $\mathbf{G}_i = \mathbf{\Gamma}\mathbf{W}_i$.

Compute the $i^{\text{th}}$ group of projections for training points and testing points (if available) using the kernel trick as shown in Eq. (23) and Eq. (24).

**end for**

Formulate the new representations by all the projections as in VCA.

## Computational Complexity Comparison

In the training process, the most time consumption step of VCA is the decomposition of the matrix $\mathbf{A} \in \mathbb{R}^{n_i \times l}$, where $l$ is the number of candidate functions. It is no larger than $|F|^2 \min\{|F|, s\}$ where $|F| \leq n_i$ and $s$ is the total number of monomials whose degrees are less than $d$. Correspondingly, the most time consumption step of DVCA is the eigen-decomposition of an matrix with scale $n$. Since eigen-decomposition and SVD decomposition has similar computational complexity and different implementations may have different time costs, it is difficult to compare the time efficiency of two methods.

In the testing process, VCA only involves the evaluation of testing points on the generators. Its computational time is linear with respect to the size of generators. Using DVCA, we need to compute the kernel formulated by the testing points and training points. Commonly, it needs more time than VCA, especially when the size of data is large.

# Experimental results

We would like to provide two groups of experimental results for illustration. VCA is a supervised method and the first group contains results for classification. The second is the numerical result concerning computational time. As the emphases of this paper are the kernel view of VCA and the improvement of VCA, we would like to just make comparisons between DVCA, KPCA and VCA. VCA is implemented by the code provided by the authors[1].

There are totally six various data sets collected from different real applications, including DNA data, images, voice data, etc. They are DNA data, ORL data, VOWEL data, VEHICLE data, COIL20 data and ISOLET5 data. All the data are downloaded from open sources[2,3]. The dimensionality ranges from about 10 to 1000 and all the data are scaled as suggested by the providers.

Similar to VCA, we also use KPCA and DVCA as the method of feature selection and use its projections for classification. Linear classifier is conducted by using the LibSvm software (Chang and Lin 2011). The common parameter $d$ is tuned using 5-fold cross validation. Besides, in DVCA, we also determine $\lambda$ by cross validation.

## Classification Accuracy

The original data are randomly split into two parts, training and testing samples. We select fixed number of points for training and the rest are used as testing examples. In VCA, KPCA and DVCA, we use training samples for deriving transformation. Then linear SVM is trained and tested on the transformed data, including training and testing points. With 100 independent runs, the mean classification accuracy with different numbers of training samples are shown in Fig. 1.

As shown in Figure 1, we have the following observations. (1) It is clear that DVCA performs better than VCA in almost all cases. This is mainly due to the fact that we take the dissimilarities between data points of different classes into account by adding the objective function of BDA shown in Eq. (17). (2) With the increase of the number of the training points, the classification accuracy also increases, as is often the case. (3) Although KPCA and VCA have a linear relationship between their projections, their classification results are different. The reason may be that we evaluate their performances by the classification accuracy of linear SVM, while linear SVM does not produce equal results when its inputs have linear relationship.

## Computational Time Comparison

We have tested the algorithm by a naive Matlab implementation on a workstation with 12 processor (3.33G for each) and 47.2GB memory. We separate all the procedure into training and testing sets. The training process includes transformation computation and linear SVM model learning. The testing process includes projecting testing points and classifying testing points by linear SVM. We have selected three representative data sets with largest sample size, dimensionality and class number, and report their time consumptions with different numbers of training samples. The training time and testing time are shown in Table 1.

As seen from the results in Table 1, we have the following observations. (1) The computational time of different methods is dominated by different factors. For example, when the number of classes is large, DVCA consumes more time. (2) The training time of VCA is largely determined by the number of training samples. For instance, on DNA data, when the number of training points is large, the training time of VCA increases drastically. (3) The real testing time consumption of DVCA is comparable to that of VCA, especially when the number of training points is small.

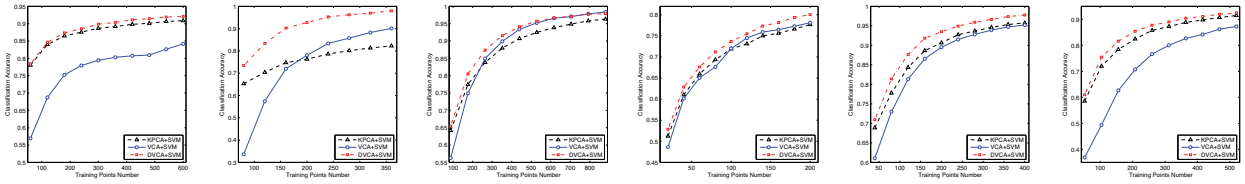
# Discussion and Further Work

The insight into VCA has tightened the relationship between results in algebra and their applications in machine learning. There are also some related works concerning the computation of vanishing ideal for a data set, such as the Approximately Vanishing Ideal (AVI) algorithm (Heldt et al. 2009).

[1] http://www.cs.huji.ac.il/ rlivni73/

[2] http://www.csie.ntu.edu.tw/c̆jlin/libsvmtools/datasets/

[3] http://www.cad.zju.edu.cn/home/dengcai/Data/data.html




(a) (b) (c) (d) (e) (f)

Figure 1: Classification results of different methods on six different data sets with different numbers of training points. (a) DNA (b) ORL (c) VOWEL (d) VEHICLE (e) COIL20 (f) ISOLET5.

Table 1: Computational time of training and testing different methods on DNA, ORL and VEHICLE. For brief, we refer VCA as VCA for feature extraction and Linear SVM for classification. Similarly for DVCA.

| No. | Training time | | Testing time | |
|---|---|---|---|---|
| | VCA | DVCA | VCA | DVCA |
| DNA | | | | |
| 300 | 52.1712 | 0.5531 | 1.0729 | 0.8531 |
| 600 | 61.0889 | 2.2633 | 2.9889 | 2.6550 |
| 900 | 82.6519 | 6.3136 | 3.6290 | 6.1642 |
| 1200 | 219.394 | 13.948 | 4.5547 | 9.3746 |
| 1600 | 115.349 | 19.788 | 4.5775 | 10.942 |
| ORL | | | | |
| 80 | 0.7246 | 0.2731 | 0.1404 | 0.1636 |
| 160 | 0.9686 | 1.7563 | 0.1248 | 0.6469 |
| 240 | 1.1515 | 4.3979 | 0.1088 | 1.0721 |
| 320 | 1.8888 | 9.0895 | 0.0782 | 1.3240 |
| VEHICLE | | | | |
| 40 | 0.0074 | 0.0108 | 0.0082 | 0.0078 |
| 220 | 0.3042 | 0.2534 | 0.0644 | 0.0881 |
| 400 | 2.5052 | 0.5893 | 0.1098 | 0.1261 |
| 580 | 3.4999 | 1.1103 | 0.1108 | 0.1367 |
| 760 | 7.3911 | 1.6921 | 0.0538 | 0.0636 |

Nevertheless, it is limited to the functions with a small number of monomials. Kernel methods has attracted plenty of research interests and the results are fruitful. Our paper aims to deepen the understanding of VCA from the well-known kernel view. It facilitates the understanding and using VCA from both theoretical and practical aspects.

There are still some further works. In theory, we have eliminated the null spaces of $\mathbf{\Phi}$. It is still interesting to investigate its influence. Although the null space of training samples does not take influence on training classifier, it takes effects on testing process. In practice, although we have proposed DVCA to improve VCA, there still exists some other problems to be studied, such as how to reduce the computational complexities and determine the parameter.

## Acknowledgments

This work was supported by NSF of China: No. 61005003, 60975038 and Australian Research Council Projects: DP-140102164, FT-130101457.

## References

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Chang, C.-C., and Lin, C.-J. 2011. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2(3):1–27.

Cox, D. A.; Little, J.; and O'Shea, D. 2007. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra, (Undergraduate Texts in Mathematics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Guyon, I., and Elisseeff, A. 2003. An introduction to variable and feature selection. *Journal of Machine Learning Research* 3:1157–1182.

He, X.; Cai, D.; and Niyogi, P. 2006. Laplacian score for feature selection. In *NIPS 18*. 507–514.

Heldt, D.; Kreuzer, M.; Pokutta, S.; and Poulisse, H. 2009. Approximate computation of zero-dimensional polynomial ideals. *Journal of Symbolic Computation* 44(11):1566–1591.

Koller, D., and Sahami, M. 1996. Toward optimal feature selection. In *ICML*, 284–292.

Livni, R.; Lehavi, D.; Schein, S.; Nachliely, H.; Shalev-shwartz, S.; and Globerson, A. 2013. Vanishing component analysis. In *ICML*, 597–605.

Mika, S.; Ratsch, G.; Weston, J.; Scholkopf, B.; and Mullers, K. R. 1999. Fisher discriminant analysis with kernels. *Neural Networks for Signal Processing IX, Proceedings of the IEEE Signal Processing Society Workshop* 41–48.

Robnik-Sikonja, M., and Kononenko, I. 2003. Theoretical and empirical analysis of relieff and rrelieff. *Machine Learning* 53(1-2):23–69.

Schölkopf, B., and Smola, A. J. 2001. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press.

Schölkopf, B.; Smola, A.; and Müller, K.-R. 1998. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation* 10(5):1299–1319.

Van der Maaten, L.; Postma, E.; and den Herik, H. V. 2009. Dimensionality reduction: a comparative review. Technical Report TiCC-TR 2009-005, Tilburg University.

Zhou, X. S., and Huang, T. S. 2001. Small sample learning during multimedia retrieval using biasmap. In *CVPR*, 11–17.