Integration
geoscience
engineering
partnership
geoConvention 2013

A Framework of Sparse Kernel Principal Component Analysis and Its Applications

X. Sun, Lab for Integration of Geology and Geophysics, China University of Petroleum (Beijing)
samzdsun@yahoo.com
S. Z. Sun*, Lab for Integration of Geology and Geophysics, China University of Petroleum (Beijing)
X. Zhou, J. Tian , J. Han, H. Yang ,Tarim Oilfield Co., China National Petroleum Corporation
C. Sun, Dongshenboda Technologies Inc. (Beijing, China)

Summary

In the field of data integration, KPCA is regarded as an excellent technique in capturing and extracting the most principle information for multi-source of data. As a learning mechanism, KPCA usually prefers large training dataset to enhance its generalization. However, the involved computation costs for large datasets could probably become nearly-unaffordable. Currently, it is a major contradiction for KPCA, and is severely impeding its practical applications. For this reason, this paper builds a framework of sparse KPCA by introducing a sparse kernel skill, which could greatly streamline the training dataset while effectively preserving its representative information. With this method, the calculation efficiency for seismic denoising is raised nearly by 7 times than the traditional KPCA, and a much higher fitting rate (98.81%) on fluid identification is also achieved as well, even with much fewer training nodes. At present, results with this method is sufficiently rewarding and encouraging enough to motivate further study. Probably, this method would bring meaningful changes on KPCA's theory and applications in geophysical exploration world.

Introduction

In the past decade, kernel principal component analysis (KPCA) emerged along with the development of support vector machine, where kernel's value is re-emphasized. Compared with PCA, KPCA is regarded to be more sufficient and powerful in tracking and extracting the underlying nonlinear characteristics, which has been fully demonstrated in the seismic attribute optimization (Zhang et al., 2009; Liu et al. 2011).

However, unlike PCA, the Gram matrix in KPCA is much larger, and the nearly-unaffordable computation costs involved in matrix decomposition is severely impeding its further applications. Moreover, intermediate results in feature space (e.g. kernel principal components and eigenvectors)can hardly be transformed back to the original input space for subsequent processing and analysis. Due to these intrinsic defects, KPCA at present mainly rests on attributes optimization in geophysical exploration, while other aspects are seldom touched. Aimed at enlarging its applications, this paper attempts to boost the calculation efficiency by introducing a sparse kernel skill, while multidimensional scaling (MDS) is also proposed to bridge the differences between the feature space and input space. It is no doubt that these improvements will bring profound influences on KPCA's theoryand applications in seismic exploration field.

Framework of traditional KPCA

As like traditional PCA, KPCA should calculate the eigenvectors for its covariance matrix $\mathbf{C}_F$ in feature space F, i.e.

$$\lambda \mathbf{v} = \mathbf{C}_F \mathbf{v}, \quad \text{where } \mathbf{C}_F = \frac{1}{n}\sum_{i=1}^{n} \varphi(\mathbf{x}_i)\varphi(\mathbf{x}_i)^T \ . \tag{1}$$

The φ stands for the mapping function from the input space to the feature space. In fact, we can equivalently accomplish the above decomposition task by employing kernel tricks (without knowing the exact mapping function). Actually, any symmetric function that satisfies the Mercer Condition could be viewed as a kernel function for some feature space (Mercer, 1909; Taylor and Cristianini, 2004). Among these kernel functions, Gauss kernel is one of most frequently engaged one in KPCA, and all the work in this paper is established on this kernel function

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2}{\sigma^2}) \ . \tag{2}$$

According to reproducing kernel theory, the eigenvector $\mathbf{v}_i$ must exist in the hyperspace expanded by the array $\{\varphi(\mathbf{x}_1), \varphi(\mathbf{x}_2), \cdots \varphi(\mathbf{x}_n)\}$, and is formulated as $\mathbf{v}_i = [\varphi(\mathbf{x}_1) \ \varphi(\mathbf{x}_2) \ \cdots \ \varphi(\mathbf{x}_n)]\boldsymbol{\alpha}_i$, where $\boldsymbol{\alpha}_i$ is the related coefficient vector. After a series of formulation (Schölkopf et al., 1997), we will finally arrive at

$$n\lambda\boldsymbol{\alpha}_i = \sum_{j=1}^{n} \boldsymbol{\alpha}_j k(\mathbf{x}_i, \mathbf{x}_j) \ \Leftrightarrow \ n\lambda\boldsymbol{\alpha} = \boldsymbol{K}\boldsymbol{\alpha} \ . \tag{3}$$

Here $\boldsymbol{K}$ represent the Gram matrix formed by $k(\mathbf{x}_i, \mathbf{x}_j)$ and falls into the symmetric positive semi-definite category. By decomposing this matrix, all eigenvectors are then obtained. Actually, kernel components are the projection results on these eigenvectors in feature space. Suppose $\mathbf{Y}$ is the test vector whose mapped vector in feature space is given by $[\varphi(\boldsymbol{y}_1), \varphi(\boldsymbol{y}_2), \cdots \varphi(\boldsymbol{y}_n)]$, then its projected result on $\mathbf{v}_i$ is

$$\mathbf{KP}_i(\mathbf{Y}) = \frac{1}{\sqrt{\lambda_i}} \boldsymbol{\alpha}_i^T [\varphi(\mathbf{x}_1) \ \varphi(\mathbf{x}_2) \ \cdots \ \varphi(\mathbf{x}_n)]^T [\varphi(\boldsymbol{y}_1), \varphi(\boldsymbol{y}_2), \cdots \varphi(\boldsymbol{y}_{n'})] = \frac{1}{\sqrt{\lambda_i}} \boldsymbol{\alpha}_i^T \mathbf{K}_{test}, \tag{4}$$

where **KP** indicates the corresponding kernel component. Moreover, apart from the usage of normalization in equation (4), the eigenvalue λ_i could also indicate the importance of the kernel component. Usually, the first m kernel components could reflect the main information of training dataset, and m is decided by the accumulative contribution rate: $\sum_{i=1}^{m} \lambda_i \Big/ \sum_{i=1}^{d} \lambda_i \geq \gamma$,where γ generally falls into 0.85~0.95.

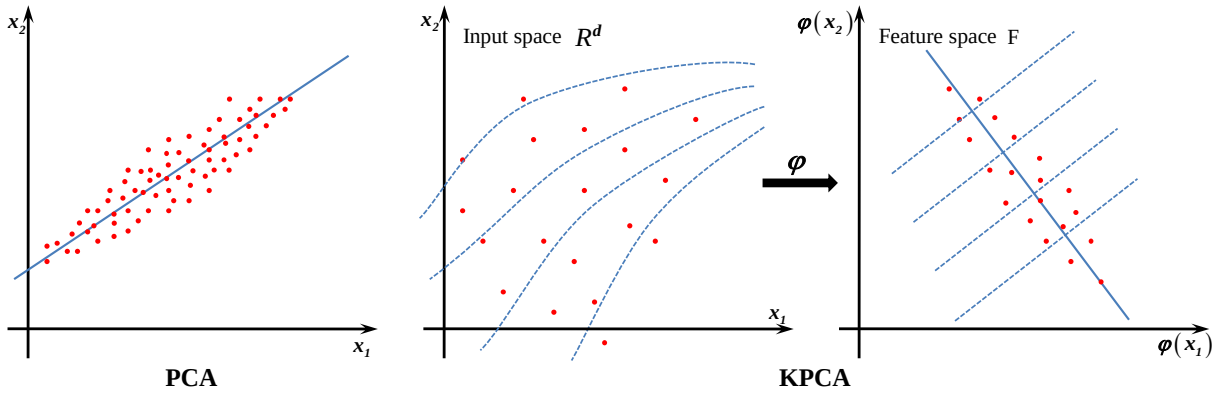

Figure 1: Sketch map of KPCA

Like PCA, KPCA also tends to maximize the variance of transformed results. As Figure 1 shows, KPCA could generally be viewed as an implementation of PCA in feature space, and its unique attraction largely rests on the kernel tactics, which could induce the originally-scattered samples to aggregate along the eigenvector direction. At this step, we can make the following remark.

Remark 1 In spite of the advantages in handling nonlinear problems, the computation efficiency of KPCA could probably slow down to a glacial pace when the training dataset is large. Firstly, dealing with large training dataset demands a significant amount of computing resources to calculate and eigen-decompose the Gram matrix. Besides, every time for a test vector, the kernel function between this vector and each training vector should necessarily be computed. However, as a learning method, KPCA yet prefers large training dataset to enhance its generalization capability. It is a contradiction for KPCA.

Remark 2 One merit of kernel tactic lies in that it enables the implementation of KPCA without knowing the exact mapping function, but this advantage may somewhat cast a new problem that the intermediate results generated in feature space (e.g. kernel principal components and eigenvectors) cannot be transformed back to the original input space for subsequent processing. Probably, their 'pre-images' do not exist in the input space at all, and could only be approached by approximate solutions (Kwok and Tsang, 2004). This may explain the reason why KPCA can hardly been applied on seismic denoising.

Sparse Kernel Skill

The data information that we encounter in real world is often characterized by dynamism, uncertainty and sparsity (Xu et al., 2007). Sparsity implies that the entire training dataset could be represented by a few 'distinctive' ones, which are known as the training nodes. And the sparse kernel skill attempts to extract these distinctive nodes and replace the entire training dataset with these nodes, in an attempt to streamline and optimize KPCA's performance.

Let $\{\varphi(\boldsymbol{x}_i) \in \mathrm{F}, i = 1, \cdots, n\}$ be the entire training dataset of feature space. As we mentioned before, eigenvectors can be treated as a combination of all these training vectors. Assuming that only one training vector $\varphi(\boldsymbol{x}_i)$ is used for reconstruction, the eigen-equation is accordingly transformed into

$$\frac{\varphi(\boldsymbol{x}_i)^T \mathbf{C}_F \varphi(\boldsymbol{x}_i)}{\varphi(\boldsymbol{x}_i)^T \varphi(\boldsymbol{x}_i)} = \lambda_i \;\Leftrightarrow\; \frac{\mathbf{K}_{x_i}{}^T \mathbf{K}_{x_i}}{K_{ii}} = \lambda_i, \tag{5}$$

Here $\mathbf{K}_{x_i} = \left[k(x_1, x_i)\ k(x_2, x_i) \cdots k(x_n, x_i)\right]^T$ is the centralized kernel vector of $\varphi(\boldsymbol{x}_i)$, while λ_i represents the variance obtained by projecting $\varphi(\boldsymbol{x}_i)$ on eigenvectors $\mathbf{v}$. In light of the character of maximizing variance, a larger λ_i should means that $\varphi(\boldsymbol{x}_i)$ is much closer to $\mathbf{v}$. Select the training vector with largest λ as the first training node, then determine the rest nodes on the principle of cosine distance, i.e.

$$\cos\left(\varphi(\boldsymbol{x}_i), \varphi(\boldsymbol{x}_j)\right) = \frac{k(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\sqrt{k(\boldsymbol{x}_i, \boldsymbol{x}_i) k(\boldsymbol{x}_j, \boldsymbol{x}_j)}} \tag{6}$$

Minimum cosine distance demonstrates that the selected node is of the greatest difference with these already-selected ones, which ensures that extracted nodes could basically represent the entire training dataset (Xu, 2009). In most cases, the training nodes only account for 5%~20% of entire training dataset. It is obvious that application of sparse kernel skill would bring significant improvements on the computational efficiency of KPCA.

Application on seismic denoising

In seismic denoising, KPCA ought to show remarkable advantages in preserving the important while suppressing the trivial (such as random noise). Unfortunately, the low computing efficiency and difficulties in estimating the 'pre-image' impede its application value. Therefore, the sparse kernel skill is mainly designed to answer the former challenge, while a multi-dimensional scaling (MDS) for the later.

The MDS is beyond the discussion of this paper, one can refer to Kwok's paper (Kwok and Tsang, 2004) for details.

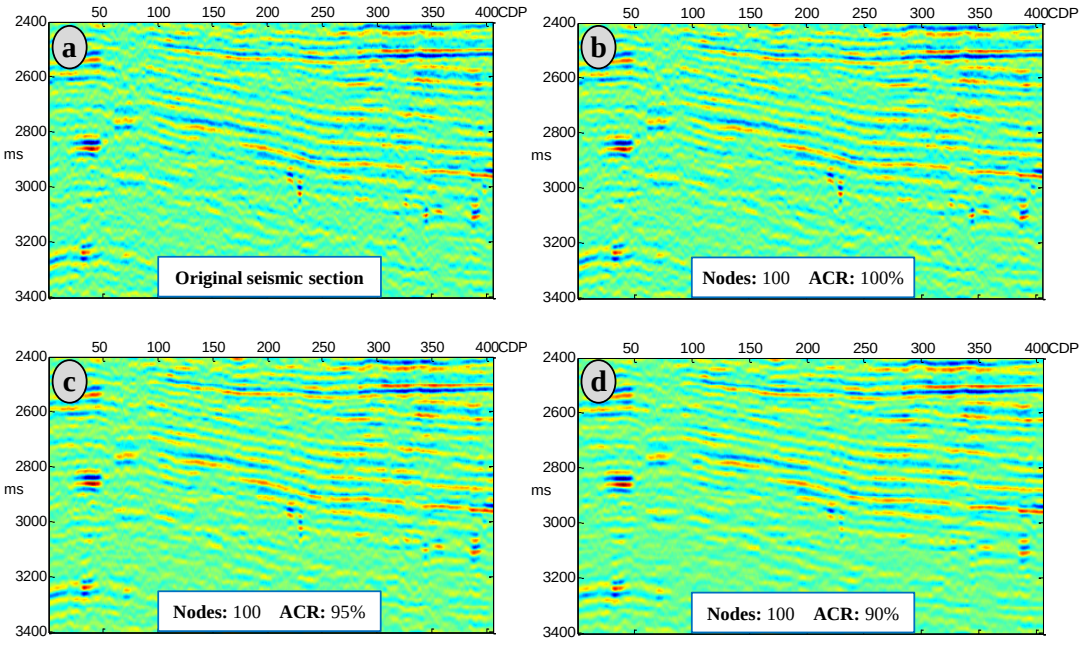

Figure 2: Original seismic section and denoised results with different ACR
(Note: ACR of 100% means no denoising but just reconstruction with extracted nodes)

In practice, we choose an arbitrary line from the well area ZG8, a carbonate reservoir in west China. The original seismic section of this line is given in Figure2 (a), where the favorable carbonate reservoirs are characterized by 'bead-like' reflections on seismic section. Although this is a relative small section of 405 traces with 500 samples, computational speed of ordinary KPCA is quite slow. By employing the sparse kernel skill, we find that 100 nodes are sufficient enough for this case. On extracting these nodes, seismic denoising could be approached through MDS by adjusting the accumulative contribution rate (ACR). It is worth noting that the reconstructed result in Figure 2(b) demonstrates that 100 nodes could almost completely reconstruct the whole section without losing any important information. After several tests and comparisons, an ACR of 95% is determined as the optimum parameter for denoising, whose result is shown in Figure 2(c). More importantly, the calculation efficiency is raised nearly by 7 times.

Application on fluid identification

Fluid identification is essentially a classification task, for which KPCA is more qualified than PCA. As Figure 3 shows, the basic mechanism (Schölkopf et al., 1997; Dejtrakulwong et al., 2012) can be summarized as 3 steps: (1) calculate the kernel components for available training datasets, then form the standard kernel matrix for each fluid class; (2) calculate the kernel component vector for test data;(3) measure the 'distance' between the test data's kernel component vector and each class's kernel matrix, and classify the test data's fluid type by applying a distance sorter.

In this paper, we employ the minimum Mahalanobis distance sorter. Let $\boldsymbol{x}_k$ and ψ_k be the center vector and covariance matrix for the standard kernel matrix of a specific class (suppose the k^{th} class), then Mahalanobis distance for test data $\boldsymbol{y}$ is defined as

$$d_k^2 = (\boldsymbol{y} - \boldsymbol{x}_k)^T \psi_k^{-1} (\boldsymbol{y} - \boldsymbol{x}_k) \tag{7}$$

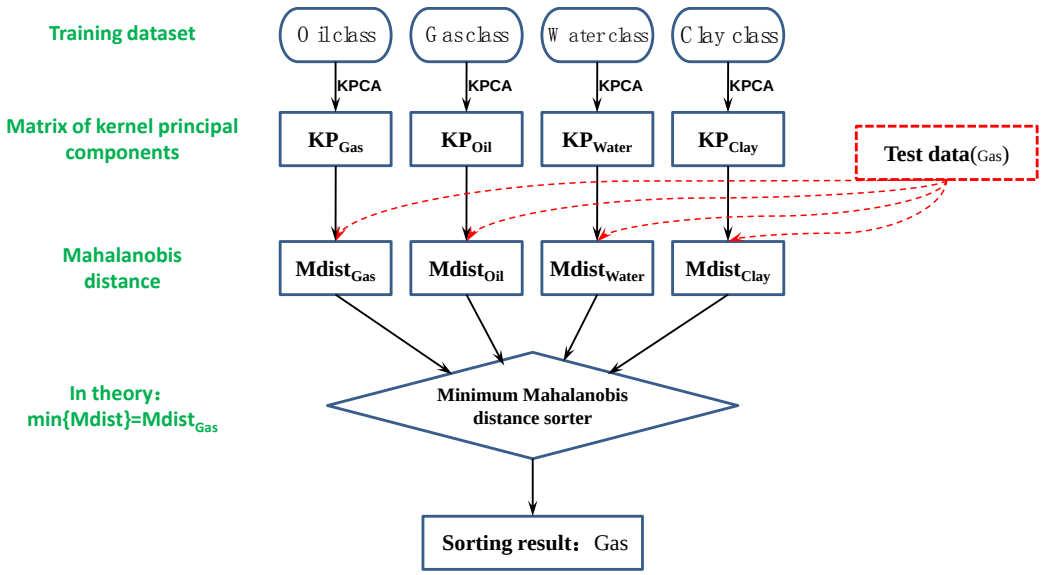

Figure 3: Workflow of fluid identification based on KPCA

This time, the data we used comes from the fluid substitution results on part of a measured porosity log, whose porosity varies between 2.52% and 10.54%. There are 4 fluid classes, and each class owns 129 training vectors, which correspondingly consist of the most common-used attributes in fluid identification (namely, PI, SI, Vp/Vs, Vp, Vs, density, Poisson ratio, LR and MR). And we re-examine these 516 training vector's class based on the 180 training nodes that extracted with sparse kernel skill. The obtained Mahalanobis distance (for KPCA) is shown in Figure 4(b). For contrast, the Mahalanobis distance in PCA is given in Figure 4(a).

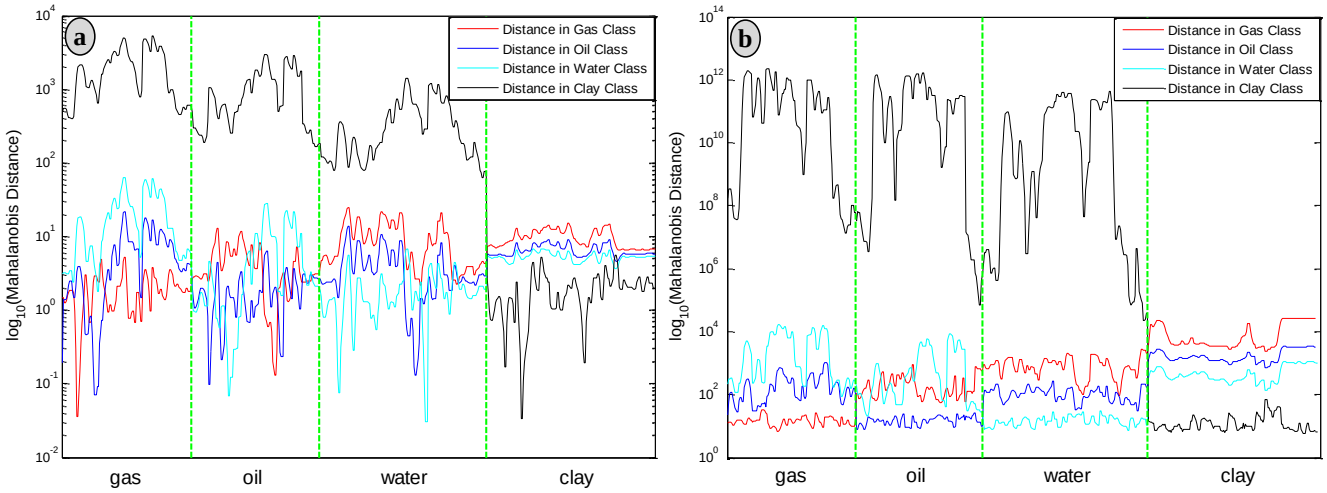

Figure 4: Observed Mahalanobis distance in both PCA (a) and KPCA (b)

Figure 4 (a) shows that Mahalanobis distance is perfectly sorted, i.e. the minimum distance corresponds to its due fluid type, and the overall fitting rate reaches 98.81%, much higher than that derived from PCA (76.48%). These results not only demonstrate that KPCA do have great advantages in identifying different fluid types than PCA, but more importantly reconfirm the feasibility of sparse KPCA in fluid identification.

Conclusions

In this paper, we focus on the construction for a sparse kernel component analysis method, while the two application aspects are mainly designed to justify and support this new method. Due to the sparse

kernel skill, the training dataset could be greatly streamlined and refined, with significant information well-preserved and computing efficiency dramatically speeded up. At present, results with this method is sufficiently rewarding and encouraging enough to motivate further study. It is entirely foreseeable that this method would bring about beneficial changes for overall situation of KPCA, especially in handling the huge dataset of exploration world.

Acknowledgements

The authors would like to thank the financial support from National Basic Research Program of China (Grant No.2011CB201103), the National Science and Technology Major Project (Grant No. 2011ZX05004003) and the National Youth Science Fund Project (41204093). We would also like to thank many of our colleagues at LIGG (Lab for Integration of Geology and Geophysics,CUPB), who have provided helpful inspirations and suggestions.

References

Brito, M.V., 2010, Principal component analysis for stratigraphic imaging improvement and facies predictions: SEG Expanded Abstracts, 2401-2405.

Dejtrakulwong P., T. Mukerji, G. Mavko, 2012, Using kernel principal component analysis to interpret seismic signatures of thin shaly-sand reservoirs: SEG Expanded Abstracts.

Kwok, J., I. Tsang, 2004, the Pre-image problem in kernel method: IEEE Transactions On Neural Networks, **15**(6), 1517-1525.

Liu, L., S. Z. Sun, H. Wang, 2011, 3D Seismic attribute optimization technology and application for dissolution caved carbonate reservoir prediction: SEG Expanded Abstracts, 1968-1972.

Mercer, J., 1909, Functions of positive and negative type and their connection with the theory of integral equations: Philosophical Transactions of the Royal Society of London.

Schölkopf, B., A. Smola, K. Muller, 1997, Kernel principal component analysis: Lecture notes in computer science, **1327**, 583-588.

Schölkopf, B., A. Smola, K. Muller, 1998, Nonlinear component analysis as a kernel eigenvalue problem: Neural Computation, **10**, 1299-1319.

Taylor, J.S., N. Cristianini, 2004, Kernel methods for pattern analysis, Cambridge University Press.

Xu, Y., 2009, A new kernel MSE algorithm for constructing efficient classification procedure: International Journal of Innovative Computing, Information and Control.

Xu, Y., D. Zhang, F. Song, 2007, A method for speeding up feature extraction based on KPCA: Neurocomputing, 1056-1061.

Yin, X., J. Zhou, 2005, Summary of optimum methods of seismic attributes, Oil Geophysical Prospecting, **40**(4), 482-489.

Zhang G., G. Kong, J. Zheng, 2009, Seismic attribute optimization based on kernel principal component analysis: SEG Expanded Abstracts, ID: 1152.