수정된 커널 주성분 분석 기법의 분류 문제에의 적용

김 병 주[†] · 심 주 용[††] · 황 창 하[††] · 김 일 곤[†††]

요 약

본 논문에서는 학습 자료로부터 비선형 특징추출과 분류를 위한 점진적인 커널 주성분 분석 방법(IKPCA)을 제안한다. 일괄처리 방식의 커널 주성분 분석 방법은 학습 자료의 크기가 클 경우 과도한 계산량이 문제가 된다. 또한 새로 추가 되는 학습 자료가 있을 경우 고유벡터를 계산하기 위해 고유공간 전체를 다시 계산해야 하는 문제점이 있다. IKPCA는 이러한 문제점들을 고유공간 모델의 점진적인 계산과 경험 커널 사상에 의해 해결하였다. IKPCA는 일괄처리방식의 커널 주성분 분석에 비해 기억공간 요구량에 있어 효율적이며 학습 자료의 재학습에 의해 성능을 쉽게 향상 시킬 수 있다. 비선형 자료에 대한 실험을 통해 IKPCA는 일괄처리방식의 커널 주성분 분석 방법에 비해 특징추출과 분류 문제의 성능에 있어 유사한 결과를 나타내었다.

Modified Kernel PCA Applied To Classification Problem

Byung Joo Kim[†] · Joo Yong Sim[††] · Chang Ha Hwang[††] · Il Kon Kim[†††]

ABSTRACT

An incremental kernel principal component analysis (IKPCA) is proposed for the nonlinear feature extraction from the data. The problem of batch kernel principal component analysis (KPCA) is that the computation becomes prohibitive when the data set is large. Another problem is that, in order to update the eigenvectors with another data, the whole eigenspace should be recomputed. IKPCA overcomes these problems by incrementally computing eigenspace model and empirical kernel map. The IKPCA is more efficient in memory requirement than a batch KPCA and can be easily improved by re-learning the data. In our experiments we show that IKPCA is comparable in performance to a batch KPCA for the feature extraction and classification problem on nonlinear data set.

키워드 : 고유 공간 모델(Eigenspace Model), 커널 주성분 분석(Kernel PCA), 경험 커널 사상(Empirical Kernel Map), 비선형 특징 추출 (Nonlinear Feature Extraction)

1. Introduction

Principal Component Analysis (PCA) [1] has proven to be an exceedingly popular technique for dimensionality reduction and it applied many areas such as data compression, image analysis, pattern recognition, regression and time series prediction. PCA traditionally require a batch computation step. The drawback of batch PCA method is that when the data set is large, i.e., the PCA computation becomes prohibitive. Another problem is that, in order to update the subspace of eigenvectors with another data, we have to recompute the whole eigenspace.

To overcome these problems, several methods have been introduced that allow for incremental computation of eigenspace [2, 3]. These methods take the training data sequentially and compute the new set of eigenspace based on the previous space of eigenvectors and the new input training data. Although the eigenspace is computed incrementally, their methods have several limitations, namely they do not consider shift of origin. Hall et al. [4] proposed a incrementally computing eigenspace models that allows shift of origin. They show that fixed mean methods are not proper for classification. Another drawback of PCA is that it only defines a linear projection of the data, the scope of its application is somewhat limited. An approach for nonlinear principal component analysis has been taken by Tipping and Bishop [5]. Various global nonlinear approaches have also been developed such as auto-associative multi-layer perceptrons minimizing the reconstruction error and principal curves [6, 7]. The disadvantage of earlier research is that they require nonlinear optimization techniques. Recently, the kernel trick has also been applied to PCA in terms of the dot product matrix instead of the covariance matrix [8]. This makes it possible to extract non-linear features using kernel functions by solving an eigenvalue problem like PCA. Though KPCA is capable of extracting nonlinear features, KPCA has to store the entire $N \times N$ kernel matrix. It is infeasible for a large number of data N. Recently there are several approach

† 정 회 원 : 영산대학교 컴퓨터 정보공학부 교수
†† 정 회 원 : 대구가톨릭대학교 정보통계학과 교수
††† 정 회 원 : 경북대학교 컴퓨터과학과 교수
논문접수 : 2003년 4월 1일, 심사완료 : 2003년 6월 2일

to make kernel PCA more efficient for large data sets. Rosipal and Girolami [9] take advantage of the simplicity and efficiency of the Expectation Maximization (EM) algorithm for PCA to make KPCA more computationally efficient. However, this algorithm is still a batch method and does not resolve the problem of having to store the entire kernel matrix. Reduced set selection methods proposed by Scholkopf et al. [10] are one way to obtain smaller expansion but they are computationally quite expansive. Smola et al. [11] propose a sparse kernel feature analysis which leads to very sparse expansions require only k kernel functions to be computed for extracting the first k features. But the basic algorithm still needs to store the kernel matrix. Another problem of KPCA is to choose a suitable kernel function for a particular problem [12]. For this problem, Scholkopf et al. [10] propose an empirical kernel map which is more genera kernel.

In this paper, we propose a incremental approach of making PCA nonlinear using empirical kernel map $\Psi_N : \mathbf{R}^d \rightarrow \mathbf{R}^N$ and incrementally eigenspace update method proposed by Hall et al. [4], describing how an IKPCA alleviates the need for storing the kernel matrix and makes KPCA more tractable on large data sets. The empirical kernel map is briefly described in Section 2. The incremental PCA outlined in Section 3. The IKPCA which involves an adaption of the standard formulation of KPCA is described in Section 4. In Section 5 we present the results of the experiments which show the feasibility of our approach. We give the conclusion and the remark in Section 6.

2. Empirical Kernel Map

The standard formulation of PCA is as the eigendecomposition of the covariance matrix of the data. Scholkopf [8] shows that PCA can also be carried out on the dot product matrix.

Let $\{x_N\}$ be a data set with N examples of dimension d which is mapped into feature space $\{\Phi(x_N)\}$. We suppose the mapped data to be centered $\sum_{i=1}^{N} \Phi(x_i) = 0$. The matrix $X = [\Phi(x_1), \Phi(x_2), \cdots, \Phi(x_N)]$ represent the data in a compact way. Standard PCA is based on finding the eigenvalues and orthonormal eigenvectors of the covariance matrix in the feature space.

$$C = \frac{1}{N} \sum_{i=1}^{N} XX^T \tag{1}$$

We are interested in the dot product matrix of size $N \times N$

$$K = \frac{1}{N} X^T X \tag{2}$$

which is called the kernel matrix since $K_{ij} = \frac{1}{N} \Phi(x_i) \cdot \Phi(x_j) = \frac{1}{N} k(x_i, x_j)$.

KPCA is based on the fact that there is a one-to-one correspondence between the non-zero eigenvectors $\{v^k\}$ of C and the non-zero eigenvectors $\{u^k\}$ of K and that they have the same eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_p$ $(p \leq min(d, N))$.

$$v^k = Xu^k / \sqrt{\lambda_k N} \tag{3}$$

$$u^k = X^T v^k / \sqrt{\lambda_k N} \tag{4}$$

where the scaling by $\sqrt{\lambda_k N}$ normalizes the eigenvectors. Thus, the principal eigenvectors of the covariance matrix of the mapped data lie in the span of the Φ-images of the training data.

A direct consequence of this one-to-one correspondence is that one can perform KPCA feature extraction entirely in terms of kernel functions. It requires determining the orthonormal eigenvectors u^k of K and its eigenvalues λ_k, and projecting a point $\Phi(x)$ onto the principal eigenvectors v^k in feature space as defined by equation (3).

$$\Phi(x) \cdot v^k = \left[\sum_{i=1}^{N} u_i^k \{\Phi(x) \cdot \Phi(x_i)\} \right] = \left[\sum_{i=1}^{N} u_i^k k(x, x_i) \right] \tag{5}$$

In KPCA the data set $\{\Phi(x_N)\}$ is high dimensional and can most of the time not even be calculated explicitly. A way around this problem is the idea of an empirical kernel map $\Psi_N : \mathbf{R}^d \rightarrow \mathbf{R}^N$ [10]

$$\begin{aligned} \Psi_N(x) &= [\Phi(x_1) \cdot \Phi(x), \cdots, \Phi(x_N) \cdot \Phi(x)]^T \\ &= [k(x_1, x), \cdots, k(x_N, x)]^T \end{aligned} \tag{6}$$

The empirical map does not map the data into feature space but into a space of size N. This is motivated by the fact that in equation (3) the eigenvectors lie in the span of the mapped data. The empirical kernel map projects each data point onto the subspace spanned by the $\{\Phi(x_N)\}$ and enables to do all calculations in the relevant subspace of F, which we will refer to as the feature space. Since the $\Psi_N(x_N)$ do not form an orthonormal basis in $\mathbf{R}^N$, the dot product in this space is not the ordinary dot product $x \cdot y = \sum_{i=1}^{N} x_i y_i$. But in the case of KPCA we ignore this. The idea is that we have to perform linear PCA on the $\Psi_N(x_N)$ from the empirical kernel map and thus diagonalize its covariance

matrix. Let the $N \times N$ matrix $\Psi = [\Psi_N(x_1), \Psi_N(x_2), \cdots, \Psi_N(x_N)]$, then from equation (6) and definition of the kernel matrix $\Psi = NK$. This means that the covariance matrix of the empirical mapped data is

$$C_\Psi = \frac{1}{N} \Psi\Psi^T = NKK^T = NK^2 \tag{7}$$

So we actually diagonalize NK^2 instead of K as in KPCA. Mika[13] shows that the two matrices have the same eigenvectors $\{u_k\}$. The eigenvalues $\{\lambda_k\}$ of K are related to the eigenvalues $\{k_k\}$ of NK^2 by

$$\lambda_k = \sqrt{\frac{k_k}{N}} \tag{8}$$

and as before one can normalize the eigenvectors $\{v_k\}$ for the covariance matrix C of the data by dividing each u^k by $\sqrt{\lambda_k N}$. Instead of actually diagonalize the covariance matrix C_Ψ, the incremental KPCA is applied directly on the mapped data $\Psi = NK$. It is now relatively easy to adapt the algorithm for KPCA such that it correctly takes into account the centering of the mapped data in an incremental way. This means that we only need to apply the empirical map to one data point at a time and do not need to store the $N \times N$ kernel matrix.

3. Incremental PCA

In this section we briefly outline the method that allows for complete incremental learning using the eigenspace approach. That method uses incremental PCA algorithm and to project every input data immediately onto the subspace. Each input data is then discarded, and its representation consists only of the corresponding principal components stored.

Let $\{x_i\}$ be a data set with N examples of dimension d. We compute the eigensystem by solving the singular value decomposition (SVD) of the covariance matrix C composed as

$$C = \frac{1}{N} \sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})^T \tag{9}$$

where $\bar{x} = \frac{1}{N} \sum_{i=1}^{n} x_i$ is the mean input vector.

The eigenvectors u_i, $i = 1 \cdots N$ corresponding to non-zero eigenvectors of the covariance matrix span a subspace of maximum m dimensions. We can then choose a subset of only k eigenvectors corresponding to the largest eigenvalues to be included in the model. To explain incremental PCA, we assume we have already built a set of eigenvectors $U = [u_j]$, $j = 1 \cdots k$, after having used the input data x_i, $i = 1 \cdots N$. The corresponding eigenvalues are $\lambda = diag(\Lambda)$ and $\bar{x}$ is the mean input vector. Incremental building requires to update these eigenspace to take into account a new input data x_{N+1}.

Here we briefly summarize the method described in [4]. First, we update the mean

$$\bar{x}' = \frac{1}{N+1} (N\bar{x} + x_{N+1}) \tag{10}$$

We then update the set of eigenvectors by adding a new vector and applying a rotational transformation. In order to do this, we first compute the orthogonal residual vector $\hat{h} = (U a_{N+1} + \bar{x}) - x_{N+1}$ and normalize it to obtain $h_{N+1} = \frac{\hat{h}_{N+1}}{|\hat{h}_{N+1}|_2}$ for $|\hat{h}_{N+1}|_2 > 0$ and $h_{N+1} = 0$ otherwise. The new matrix of eigenvectors U' is computed by

$$U' = [U, h_{N+1}] R \tag{11}$$

where $\mathbf{R} \in \mathbf{R}^{(k+1)\times(k+1)}$ is a rotation matrix. R is the solution of the eigenproblem of the following form

$$DR = R\Lambda' \tag{12}$$

We compose $D \in \mathbf{R}^{(k+1)\times(k+1)}$ as

$$D = \frac{N}{N+1} \begin{bmatrix} \Lambda & 0 \\ 0^T & 0 \end{bmatrix} + \frac{N}{(N+1)^2} \begin{bmatrix} aa^T & \gamma a \\ \gamma a^T & \gamma^2 \end{bmatrix} \tag{13}$$

where $\gamma = h_{N+1}^T (x_{N+1} - \bar{x})$ and $a = U^T (x_{N+1} - \bar{x})$. There are other ways to construct matrix D [3, 14]. However, only the method described in [4] allows for the updating of mean.

4. Incremental Kernel Principal Component Analysis (IKPCA)

Although incremental PCA builds the subspace of eigenvectors incrementally, it is limited to linear data. For the case of nonlinear data set, applying feature mapping function method to incremental PCA may be the solution. This is performed by so-called *kernel-trick,* which is an implicit mapping to an infinite dimensional space.

$$K(x, y) = \Phi(x) \cdot \Phi(y) \tag{14}$$

Where K is a given kernel function in input space. When K is semi positive definite, the existence of Φ is proven [15]. But, most of the case the mapping Φ cannot be obtained explicitly, so the vector in the feature space is not observable and only the inner product between vectors can be observed

via kernel function. So in this paper we take empirical kernel map described in Section 2. Tsuda [16] shows that support vector machine with empirical kernel map is identical with the conventional kernel map.

IKPCA Algorithm

Require : $X = (x_1 x_2 \cdots x_n)$: **matrix of training example**
λ : **initial eigenvalue**
U : **initial eigenvector**
$\overline{x}$: **initial mean**

for k = 1 to n : re-learning iteration

$\overline{x}' = \frac{1}{n+1}(n\overline{x} + x_{n+1})$: **update the mean**

$h_{n+1} = (Ua_{n+1} + \overline{x}) - x_{n+1}$: **compute orthogonal residual vector**

$\hat{h}_{n+1} = \frac{h_{n+1}}{|h_{n+1}|_2}$ for $|h_{n+1}|_2 > 0$

$\hat{h}_{n+1} = 0$ *otherwise* : **normalize** $\hat{h}_{n+1}$

$$D = \frac{n}{n+1}\begin{bmatrix} \Lambda & 0 \\ 0^T & 0 \end{bmatrix} + \frac{n}{(n+1)^2}\begin{bmatrix} aa^T & \gamma a \\ \gamma a^T & \gamma^2 \end{bmatrix}$$

where $\gamma = \hat{h}_{n+1}^T(x_{n+1} - \overline{x})$, $a = U^T(x_{n+1} - \overline{x})$
: **construct matrix D**

$DR = R\Lambda'$: **solve eigenproblem**

$U' = [U, \hat{h}_{n+1}]R$: **update eigenvector using the cirterian rule**

end for : end of re-learning iteration

5. Experiment

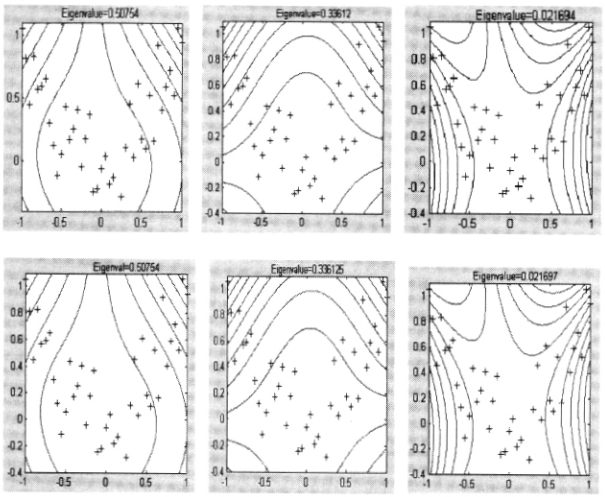
First we shall take a look at simple toy data which will show the validation of IKPCA. It is done by comparing the eigenvector of IKPCA with batch KPCA. Next we will evaluate the usefulness of the nonlinear feature extraction by IKPCA, as a preprocessing step for classification. For this purpose, extracted features by IKPCA will be used for training simple linear classifier. Among many linear classifier, we take linear least squares support vector machine (LS-VM) proposed by Suyken [17]. The reason why we use LS-VM in this paper is that LS-SVM method is computationally attractive and easier to extend than standard support vector machine (SVM). In order to evaluate the IKPCA feature extraction capability, experiments were carried out two class and multiclass data classification problem.

5.1 IKPCA versus Batch KPCA

To evaluate the performance of IKPCA and batch KPCA we take nonlinear data used by Scholkoff [8]. Training data set is generated by

$$y = x^2 + 0.2\varepsilon \ : \ \varepsilon \sim N(0,1),\ x = [-1,1] \qquad (15)$$

In (Figure 1) upper part is batch KPCA and lower is IKCA. From left to right the first 3 eigenvalues are shown in decreasing order. Contour lines indicate constant principal component value. We can see that there is a close similarity between batch KPCA and IKPCA in contour graph and eigenvalues.

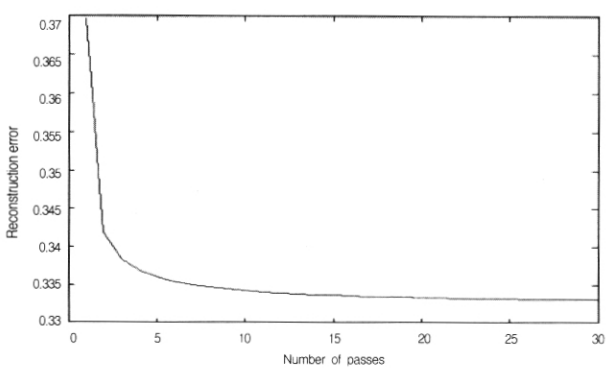


(Figure 1) Two-dimensional toy examples, with data was generated in equation (15) Upper part is batch KPCA and lower is IKPCA

Another factors of performance evaluation are reconstruction error and eigenvector's cosine value similarity between IKPCA and batch way KPCA. Reconstruction error is defined as the squared distance between the Ψ image of x_N and its reconstruction when projected onto the first l principal components.

$$\delta = |\Psi(x_N) - P_l\Psi(x_N)|^2 \qquad (16)$$

(Figure 2) shows the reconstruction error by re-learning in IKPCA. SSE (Sum of Square Error) and MSE (Mean Square Error) value of reconstruction error is 0.36975 and 0.0090. This means the performance of IKPCA is similar to batch KPCA.



(Figure 2) Reconstruction error change by re-learning in IKPCA

Finally, to check the similarity of eigenvectors of both methods we compute the cosine values between each eigenvector of KPCA and the corresponding of IKPCA. B and I are the matrices of eigenvectors of batch KPCA and IKPCA, respectively.

$$B=\begin{bmatrix} 0.53856 & -0.095116 & 0.81604 \\ 0.094315 & 0.49981 & -0.19613 \\ 0.82981 & -0.11002 & -0.5259 \\ 0.11171 & 0.85384 & 0.13795 \end{bmatrix}$$

$$I=\begin{bmatrix} 0.53856 & -0.095114 & 0.8166 \\ 0.094314 & 0.49981 & -0.19364 \\ 0.82981 & -0.11002 & -0.52635 \\ 0.11171 & 0.85384 & 0.13649 \end{bmatrix}$$

<Table 1> shows the $\cos\theta$ and θ values which explain the similarity of eigenvectors of both methods.

〈Table 1〉 Eigenvector's $\cos\theta$ and θ value obtained by IKPCA and batch KPCA

Eigenvector	$\lvert\cos\theta\rvert$	θ
1	1	0
2	1	0
3	1	0

The results of simple toy problem indicate that the proposed method is comparable to the batch way KPCA.

5.2 Two Class Classification

In two class classification problem, we use two data set i.e., classical benchmarking data and real world data.

5.2.1 Two-Spiral Data

Two-spiral classification problem is known to be hard for multilayer perceptrons [17]. Here we test a spiral problem for which two classes have been defined and each class has 60 training data points. A RBF kernel has been taken with $\sigma^2 = 0.1$. Correct classification ratio of linear LS-SVM is 100%. For this particular data experiment, we can see that IKPCA extracts nonlinear features well.

5.2.2 Real World Data

To test the performance of IKPCA for real world data, we use Cleveland heart disease data obtained from the UCI Machine Learning Repository. Data set has 303 patterns and each pattern has 13 attributes. The two classes are highly overlapped. Goal is to distinguish between presence and absence of heart disease in a patient. Like two-spiral data classification problem, same procedure is applied. A RBF kernel has been taken with $\sigma^2 = 0.1$. Correct classification ratio of linear LS-SVM is 100%. For this particular data experiment, we can see that IKPCA extracts nonlinear features well on real world data.

5.3 Multi Class Data

Earlier experiments was carried to classify two class problem. Here we extend experiment to multiclass classification problem. Training data is wine data obtained from the UCI Machine Learning Repository. Wine data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivates. The analysis determined the quantities of 13 constituents found in each of the three types of wines. Detailed attributes are available from web site (http://www.ics.uci.edu/~mlearn/MLSummary.html). Number of instances per class is 59 for class 1, 71 for class 2, and 48 for class 3. In this case we use classifier as multiclass LS-SVM proposed by Suyken [18]. The three classes have been encoded by taking $m = 2$. A RBF kernel has been taken with $\sigma_1^2 = \sigma_2^2 = 0.1$ and $\gamma = 1$. Correct classification ratio of linear multiclass LS-SVM is 100%.

6. Conclusion and Remarks

This paper was devoted to the exposition of a new technique on extracting nonlinear features from incremental data. To develop this technique, we made use of empirical kernel mapping with incremental learning by eigenspace approach. Proposed IKPCA has following advantages.

Firstly, IKPCA has similar feature extracting performance for incremental and nonlinear data comparable to batch KPCA. Secondly, IKPCA is more efficient in memory requirement than batch KPCA. In batch KPCA the $N \times N$ kernel matrix has to be stored, while for IKPCA requirements are $O((k+1)^2)$. Here $k(1 \le k \le N)$ is the number of eigenvectors stored in each eigenspace updating step, which usually takes a number much smaller than N. Thirdly, IKPCA allows for complete incremental learning using the eigenspace approach, whereas batch KPCA recomputes whole decomposition for update the subspace of eigenvectors with another data. Fourthly, IKPCA can easily be improved by relearning the data. Fifthly IKPCA do not require nonlinear optimization techniques. Lastly, experimental results show that extracted features from IKPCA lead to good performance when used as pre-preprocess data for a linear classifier.

Future works is combining IKPCA with other classifier, for example neural network or kernel based learning method so as to compare and improve classification performance.

References

[1] I. T. Jolliffe, "Principal Component Analysis," New York Springer-Verlag, 1986.

[2] S. Chandrasekaran, B. S. Manjunath, Y. F. Wang, J. Winkeler and H. Zhang, "An eigenspace update algorithm for image analysis," Graphical Models and Image Processing, 59(5), pp.321-332, Sep., 1997.

[3] J. Winkeler, B. S. Manjunath and S. Chandrasekaran, "Subset selection for active object recognition," In CVPR, IEEE Computer Society Press, Vol.2, pp.511-516, Jun., 1999.

[4] P. Hall, D. Marshall and R. Martin, "Incremental eigenanalysis for classification," In British Machine Vision Conference, Vol.1, pp.286-295, Sep., 1998.

[5] M. E. Tipping and C. M. Bishop, "Mixtures of probabilistic principal component analysers," Neural Computation 11(2), pp.443-482, 1998.

[6] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," AICHE Journal 37(2), pp.233-243, 1991.

[7] K. I. Diamantaras and S. Y. Kung, "Principal Component Neural Networks : Theory and Applications," New York John Wiley & Sons, Inc., 1996.

[8] B. Scholkopf, A. Smola and K. R. Muller, "Nonlinear component analysis as a kernel eigenvalue problem," Neural Computation 10(5), pp.1299-1319, 1998.

[9] R. Rosipal and M. Girolami, "An Expectation Maximization approach to nonlinear component analysis," Submitted.

[10] B. Scholkopf, S. Mika, C. Burges, P. Knirsch, K. R. Miller, G. Ratsch and A. J. Smola, "Input Space versus Feature Space in Kernel-based Methods," IEEE Transactions on Neural Networks, Vol.10, pp.1000-1017, Sep., 1999.

[11] A. J. Smola O. L. Mangasarian and B. Scholkopf, "Sparse kernel feature analysis," Technical Report 99-03, University of Wisconsin, Data Mining Institute, Madison, 1999.

[12] S. R. Gunn, "Support vector machines for classification and regression," Technical Report ISIS-1-98, Department of Electronics and Computer Science, University of Southampton, 1998.

[13] S. Mika, "Kernel algorithms for nonlinear signal processing in feature spaces," Master's thesis, Technical University of Berlin, November, 1998.

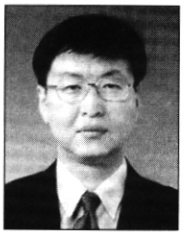
[14] H. Murakami and V. Kumar, "Efficient Calculation of Primary Images from a Set of Images," IEEE Trans. On Pattern Analysis and Machine Intelligence, 4(5), pp.511-515, 1982.

[15] V. N. Vapnik, "Statistical learning theory," John Wiley & Sons, New York, 1998.

[16] K. Tsuda, "Support vector classifier based on asymmetric kernel function," Proc. ESANN, 1999.

[17] J. A. K. Suykens and Vandewalle, "Least squares support vector machine classifiers," Neural Processing Letters, Vol. 9, pp.293-300, 1999.

[18] Accessible at http://www.esat.kuleuven.ac.be/sista/lssvmlab/tutorial.



김 병 주

e-mail : bjkim@ysu.ac.kr

1990년 부산대학교 전자계산학과(이학사)

1992년 부산대학교 대학원 전자계산학 전공(이학석사)

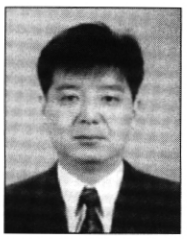
1998년 경북대학교 대학원 전자계산학 전공 박사 수료

1992년~1996년 서라벌 대학 전자계산과 조교수

1996년~2002년 성심외국어대학 정보통신학부 조교수

2003년~현재 영산대학교 컴퓨터 정보공학부 전임강사

관심분야 : 기계학습(machine learning)



심 주 용

e-mail : ds1631@hanmail.net

1983년 서강대학교 화학공학과 졸업

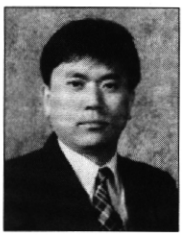
1988년 Northeastern Illinois University 대학원 수학과(이학석사)

1992년 University of Illinois at Urbana-Champaign 대학원 통계학과 박사 수료

1996년 경북대학교 대학원 통계학과(이학박사)

2003년~현재 대구가톨릭대학교 정보통계학과 겸임교수

관심분야 : dynamic programming, machine learning



황 창 하

e-mail : chhwang@cataegu.ac.kr

1982년 경북대학교 사범대학 수학교육과(이학사)

1984년 서울대학교 대학원 계산통계학과(이학석사)

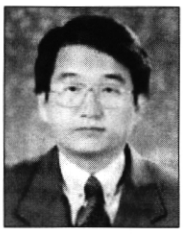
1985년~1987년 한국통신 전임연구원

1991년 미국 미시간대학교 통계학과(이학박사)

1992년~1994년 경성대학교 전산통계학과 조교수

1995년~현재 대구가톨릭대학교 정보통계학과 부교수

관심분야 : 기계학습, 정보과학



김 일 곤

e-mail : ikkim@knu.ac.kr

1991년 서울대학교 전산과학과 박사학위 취득

1997년~1998년 미국 죠지타운대학교 병원 ISIS 연구소 방문연구자

1992년~현재 경북대학교 컴퓨터과학과 교수

관심분야 : 지능형 에이전트 시스템, 분산시스템, 의료정보학