

# To Obtain Orthogonal Feature Extraction Using Training Data Selection\*

Ye Xu, Shen Furao, Jinxi Zhao  
National Key Laboratory for Novel Software  
Technology, Nanjing University,  
Hankou Road 22, Nanjing, China  
yexu@smail.nju.edu.cn,  
frshen@nju.edu.cn, jxzhao@nju.edu.cn

Osamu Hasegawa  
Imaging Science and Engineering Lab,  
Tokyo Institute of Technology,  
Nagatsuda, Midori-ku, Yokohama, 226-8503,  
Tokyo, Japan  
hasegawa@isl.titech.ac.jp

## ABSTRACT

Feature extraction is an effective tool in data mining and machine learning. Many feature extraction methods have been investigated recently. However, few methods can achieve orthogonal components. Non-orthogonal components distort the metric structure of original data space and contain redundant information. In this paper, we propose a feature extraction method, named as incremental orthogonal basis analysis (IOBA), to cope with the challenging endeavors. First, IOBA learns orthogonal components for original data, not only theoretically but also numerically. Second, an innovative way of training data selection is proposed. This selection scheme helps IOBA pick up numerically orthogonal components from training patterns. Third, by designing a self-adaptive threshold technique, no prior knowledge about the number of components is necessary to use IOBA. Moreover, without solving eigenvalue and eigenvector problems, IOBA not only saves large computing loads, but also avoids ill-conditioned problems. Results of experiments show the efficiency of the proposed IOBA.

## Categories and Subject Descriptors

I.2.6 [Artificial Intelligence]: Learning—Induction; I.5.4 [Pattern Recognition]: Application

## General Terms

Theory, Algorithm, Performance

## 1. INTRODUCTION

Feature extraction is an effective technique to handle the high dimension problem. It has been widely used in many areas.

---

\*This work was supported in part by the China NSF grant (#60573157, #60723003, and #60775046).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CIKM '09, November 2-6, 2009, Hong Kong, China  
Copyright 2009 ACM 978-1-60558-512-3/09/11 ...\$10.00.

Principal Component Analysis (PCA) is the most famous feature extraction algorithm. It attempts to search for directions of original data along which variations are extremal. During recent years, some variances of PCA have been proposed. For instance, Bishop [1] adopts some prior information to obtain the intrinsic dimension and the optimal number of clusters for PCA in the latent variable model. Weng et al. [2] proposes an incremental principal component analysis (CCIPCA) to realize online learning for PCA. Tang et al. [3] uses traditional PCA and nonorthogonal binary feature extraction method to get components. Independent Component Analysis (ICA) is another category of feature extraction algorithms. ICA, along with some ICA-based methods [4] [5], tries to transform original patterns into uncorrelated data in feature space. There are also some other categories of feature extraction methods[6][7].

However, such feature extraction learning models have some shortcomings. (1) Few of those methods can achieve orthogonal components. Even for ICA-based methods, they can not obtain orthogonal components. Cai et al. [8] indicated that non-orthogonality can not preserve the metric structure of original data space and this would cause the degrading of learning performance. Liu et al. [9] dictated that some redundant information would be incorporated into the learned non-orthogonal components. Therefore, orthogonal components are more desired. (2) Nearly all methods need a user to predetermine the number of components. Although for some methods [1], the target dimension estimation can be done. But it deserves noting that such estimation step needs prior knowledge and aggravates the computing load of the algorithms. Therefore, it is necessary to find an approach to automatically generate the proper number of components for feature space. (3) Many of these feature extraction algorithms have to solve eigenvalue and eigenvector problems, which aggravates the computing load to some extent. Moreover, it is very likely for such methods to meet ill-conditioned problem.

In this paper, we introduce a fast method named as incremental orthogonal basis analysis (IOBA) to handle the above problems of typical feature extraction methods. By adjusting the Gram-Schmidt (GS) algorithm [10], we extract an orthogonal representation for the input data, not only theoretically but also numerically. By proposing a data selection scheme, IOBA learns the components efficiently using part of training data rather than all the training patterns. By designing a self-adaptive threshold scheme, IOBA automatically generates the proper number of components

with little extra step and no prior knowledge. Moreover, IOBA obviates solving eigenvalue and eigenvector problems, thus the computation load is low and the algorithm is stable.

## 2. INCREMENTAL ORTHOGONAL BASIS ANALYSIS

As mentioned in section 1, we try to propose a method to cope with the shortcomings of those typical feature extraction methods. Here, we summarize the targets of the proposed IOBA as follows: (1) To learn orthogonal components (base vectors). (2) To automatically determine and adjust the number of components needed without any prior knowledge. (3) To realize a fast and robust incremental procedure.

To fulfill the goals, we propose the IOBA as follows: for training data that come in an online manner, first, we employ the Gram-Schmidt method to incrementally extract components. Then, we design a way to measure the independence between the income data and learned components based on linear dependence theorem, and select those training data that are independent of learned components. And a threshold scheme is proposed to automatically determine the number of components. In the rest of the section, we analyze IOBA algorithm in detail.

### 2.1 Measure the Independence between Input Data and Learned Components

To learn orthogonal components, we need to consider the independence between learned components and input patterns. Suppose we have learned base vectors  $\{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_k\}$ . If an input pattern  $\xi_{k+1}$  is dependent on the learned feature space, the corresponding computed vector  $\mathbf{b}_{k+1}$  is unsuitable to be accepted as a new component. It is because that if a component is transformed from a pattern that has strong dependence on learned components, the computation error would arise and affect the orthogonality among components [11]. In this case, if we accept  $\mathbf{b}_{k+1}$  as a new component, the numerical orthogonality between  $\{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_k\}$  and  $\mathbf{b}_{k+1}$  can't be guaranteed. Therefore, it is necessary to consider the independence between learned components and input patterns.

But simply considering the independence from theoretical aspect is not enough. Take the famous Hilbert Matrix for example: the  $n \times n$  Hilbert matrix  $H(n)$  is a positive definite symmetric matrix. Thereby, theoretically speaking, the dimension of the space spanned by its column vectors should be  $n$ . However, in *Matlab* 6.1, we validate that there are only about 18 numerically independent base vectors in the space spanned by  $H(100)$ 's 100 column vectors. The example dictates that it may cause large bias if we merely take theoretical independence into consideration. Here, based on the linear dependence theorem [11], we propose a technique to measure the numerical independence of between input patterns and learned components.

The linear dependence theorem indicates that the independence between a space  $S$  and a vector  $\mathbf{x}$  can be measured by the projection distance of  $\mathbf{x}$  upon the basis space spanned by the basis vectors of  $S$ .

Now we apply this theorem to address our problem. We presume that the already learned basis vectors (components) are  $\mathbf{b}_1, \mathbf{b}_2, \dots, \text{and } \mathbf{b}_k$ . The basis vector  $\mathbf{b}_{k+1}$  is learned from the input pattern  $\xi_{k+1}$ . We attempt to measure the linear

dependence between the input pattern  $\xi_{k+1}$  and the space  $span\{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_k\}$  according to their distance.

$$\|\xi_{k+1} - span\{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_k\}\|_2 \quad (1)$$

In addition, the distance between  $\xi_{k+1}$  and  $span\{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_k\}$  can be rewritten as the following expression.

$$\min_{a_1, a_2, \dots, a_k} \|\xi_{k+1} - (a_1 \mathbf{b}_1 + a_2 \mathbf{b}_2 + \dots + a_k \mathbf{b}_k)\|_2 \quad (2)$$

Therein,  $\alpha = (a_1, a_2, \dots, a_k)^\top$  is a coordinate vector. Here, we define matrix  $\mathbf{B}_k = (\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_k) \in \mathbb{R}^{d \times k}$ , then (2) is written in a matrix representation.

$$\min_{\alpha} \|\mathbf{B}_k \alpha - \xi_{k+1}\|_2 \quad (3)$$

The column vectors in  $\mathbf{B}_k$  are orthogonal. Thereby, we can construct an orthogonal matrix  $\mathbf{Q}$  in the following form:  $\mathbf{Q} = [\mathbf{B}_k, \mathbf{Q}_1]$ . Therein,  $\mathbf{Q} \in \mathbb{R}^{d \times d}$  and  $\mathbf{Q}_1 \in \mathbb{R}^{d \times (d-k)}$ .

Since orthogonalization transformation never changes the 2-norm of a vector, (2) is equal to the follow expression:

$$\|\mathbf{B}_k \alpha - \xi_{k+1}\|_2 = \left\| \begin{pmatrix} \alpha \\ \mathbf{0} \end{pmatrix} - \begin{pmatrix} \mathbf{B}_k^\top \xi_{k+1} \\ \mathbf{Q}_1^\top \xi_{k+1} \end{pmatrix} \right\|_2 \quad (4)$$

We let  $\alpha = \mathbf{B}_k^\top \xi_{k+1}$ . Thereby, the value of expression (2) is equal to  $\|\mathbf{Q}_1^\top \xi_{k+1}\|_2$ . The theorem below will denote the relationship between  $\|\mathbf{Q}_1^\top \xi_{k+1}\|_2$  and  $r_{k+1, k+1}$  which is obtained in the process of learning components. (See the detailed algorithm in section 2.3.)

**THEOREM 1.** *The 2-norm of  $\mathbf{Q}_1^\top \xi_{k+1}$  shown in (4) is equal to  $r_{k+1, k+1}$  achieved in the process of learning components, i.e.,  $r_{k+1, k+1} = \|\mathbf{Q}_1^\top \xi_{k+1}\|_2$ .*

**PROOF.** The proof is quite straightforward and we omit it here for brevity.  $\square$

Based on the theorem portrayed above, we can measure the numerical independence between the already learned basis spanned by components  $span\{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_k\}$  and input data  $\xi_{k+1}$  by  $r_{k+1, k+1}$ .

### 2.2 Data Selection Based on Threshold Policy

To obtain orthogonal components, we need to select orthogonal components into basis  $\mathbf{M}$  according to the measurement discussed in section 2.1. Those computed base vectors should be ruled out of basis  $\mathbf{M}$  if they are learned from input data that are dependent on  $\mathbf{M}$ . Otherwise, the numerical orthogonality of learned components can't be guaranteed. Therefore, we adopt the value of  $r_{k+1, k+1}$  as a criterion to select training data. If  $r_{k+1, k+1}$  is large, i.e., the independence between  $\mathbf{b}_{k+1}$  and learned components is strong, we accept  $\mathbf{b}_{k+1}$  into basis  $\mathbf{M}$ . Otherwise, we reject computed component  $\mathbf{b}_{k+1}$ . Through the designed training data selection scheme, only those data that are independent of learned components can be selected to learn components.

The number of components is a crucial factor worth considering in feature extraction task. If it is too small, the learned components will have a large distinction from the space spanned by the original data. On the contrary, a too large target dimension will render it difficult in learning components and waste storage capacity. Many methods need a user to predetermine the target dimension for feature space. Although for PCA and some PCA based methods, the target dimension can be determined based on the accumulation

ratio whose lower bound comes from the tolerable approximation error, this lower bound has to be predetermined by users as parameters. Therefore, traditional methods may learn an improper dimension for feature space.

In IOBA, we address the issue by a self-adaptive threshold policy: if  $r_{k+1,k+1} \geq T$ , we accept potential component  $\mathbf{b}_{k+1}$  into basis  $M$ . In this case, the number of learned components is automatically determined based on the independence. It means that all the new accepted components are transformed from patterns independent of learned components; we learn the proper number of components. And the target dimension is completely data-dependent.

$d + 1$  vectors of  $d$  dimension must be linearly dependent. For that reason, the number of vectors in the basis is no more than  $d$ . Thereby, if the compression ratio, i.e., the dimension of basis  $\dim(\mathbf{B})$  dividing the dimension of input pattern  $d$ , is small currently, it is imperative to accept potential basis vectors and vice versa.

As described above, the proposed threshold  $T$  should satisfy the following two constraints. i) The orthogonality of components should be ensured. ii) The difficulty of accepting components increases with the increase of the basis dimension  $\dim(\mathbf{B})$  for feature space. To satisfy the constraints, we propose a self-adaptive threshold policy: if and only if  $\frac{r_{k+1,k+1}}{\max_i r_{ii}} \geq T = \frac{\dim(\mathbf{B})}{d}$ , we accept  $\mathbf{b}_{k+1}$  as a new component.

Because  $\frac{\dim(\mathbf{B})}{d}$  is confined to the interval of  $[0, 1]$ , we use  $\frac{r_{k+1,k+1}}{\max_i r_{ii}}$  instead of  $r_{k+1,k+1}$  for comparison convenience given that  $\frac{r_{k+1,k+1}}{\max_i r_{ii}}$  is definitely between 0 and 1.

Based on the threshold policy, IOBA can select those training data that has little dependence on learned components.

### 2.3 Summary of the Proposed Algorithm

According to the analysis presented above, we give the detailed IOBA in Algorithm 1 as shown below.

---

#### Algorithm 1 Incremental Orthogonal Basis Analysis

---

- 1: Initialize basis  $\mathbf{B} = \emptyset$  and its dimension  $k = \dim(\mathbf{B}) = 0$ .
  - 2: For each input new pattern  $\mathbf{x}_j (j > k)$ :
  - 3: Compute  $r_{i,k+1}$  by  $r_{i,k+1} = \mathbf{x}_j^\top \mathbf{b}_i$ , for  $i = 1 : k$ .
  - 4: Compute  $\mathbf{b}_{k+1}$  by  $\mathbf{b}_{k+1} = \mathbf{x}_j - \sum_{i=1}^k r_{i,k+1} \mathbf{b}_i$ .
  - 5: Compute  $r_{k+1,k+1}$  by  $r_{k+1,k+1} = \|\mathbf{b}_{k+1}\|_2$ .
  - 6: Compute  $\mathbf{b}_{k+1}$  by  $\mathbf{b}_{k+1} = \frac{\hat{\mathbf{b}}_{k+1}}{r_{k+1,k+1}}$ .
  - 7: **if**  $\frac{r_{k+1,k+1}}{\|\hat{\xi}_{k+1}\|} \geq \frac{\dim(\mathbf{B})}{d}$  **then**
  - 8:   Accept  $\mathbf{b}_{k+1}$  as a component and add it into basis  $\mathbf{B}$ .
  - 9:   Update basis dimension  $k \leftarrow k + 1$ .
  - 10: **end if**
  - 11: Goto step 2 to continue the learning process.
- 

The time complexity of IOBA is  $O(Ndk)$ , where  $N$  is the training set size,  $d$  is the dimension of original patterns, and  $k$  is the number of components learned by the algorithm. For comparison with IOBA, we analyze the time complexity of PCA, BPCA (Bayesian PCA) [1], CCIPCA [2], and IICA [5] (an incremental version of ICA) as following: the time complexity of PCA is  $O(d^3 + Nd^2 + kd^2)$ ; BPCA is  $O(Nd^3k)$ ; CCIPCA is  $O(Ndk)$ ; IICA is  $O(Ndkh)$ . Here,  $h$  is the iteration time in the stage of IICA.

The whole procedure of IOBA needs not solve the eigenvalue and eigenvector problems. Therefore, the proposed method can enjoy a low computing load and high stability.

## 3. EXPERIMENT

In the following section, we use some datasets from the UCI Machine Learning Repository [12] to test the proposed algorithm. Such datasets include Hill, Ionosphere, Musk, Optical Digits, Sonar, and Waveform.

### 3.1 Component Quality

In this section, we try to test the quality of the components generated using the proposed method. Here, we take this policy: the more patterns in the training set that can be represented linearly by the components, the better the components are. In other words, the degree to which the original data are represented by components reflects their component quality.

If a pattern  $\xi$  can be represented completely using a group of components  $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_k$ , the vector  $\xi$  is in the space  $\text{span}\{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_k\}$  that is spanned by the components. It means that the distance between vector  $\xi$  and the space  $\text{span}\{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_k\}$  is zero. It is proper to evaluate the component quality by defining a cost function  $\mathcal{E}(\mathbf{B})$  as follows:  $\mathcal{E}(\mathbf{B}) = \frac{1}{N} \sum_{j=1}^N \|\xi_j - \text{span}\{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_k\}\|_2$ . The value of the function  $\mathcal{E}$  denotes the average distance from every pattern in a dataset to the space spanned by components.

We perform IOBA under the each database sequentially to learn the components. Then we compute the cost function of learned components and present the result in Table 1. In addition, PCA, BPCA, CCIPCA, and IICA are executed to make a comparison with the proposed IOBA. IOBA and BPCA automatically learn the number of components, as listed in Table 3. But for PCA, CCIPCA, and IICA, users must predetermine the target dimension as a parameter. The target dimension can affect the linear representation quality to some extent. Therefore, to guarantee fair comparison, for PCA, CCIPCA, and IICA, we set them up with the same value in IOBA.

Table 1 shows that the cost function of components learned by the IOBA is averagely smaller than that of components learned by any other algorithm. The components learned by IOBA are orthogonal among each other. Therefore, the components obtained by IOBA contain the least redundant information and soundly reflect the structure of original data space under the same condition. Hence it is no wonder that IOBA can achieve the best component quality.

### 3.2 Recognition performance

Next, we apply the recognition rate as a benchmark to compare different feature extraction methods. For each dataset, a group of components for transformation matrix is learned by Algorithm 1. Then we adopt the learned components to transform original data into low dimension feature space data. In the transformed feature space, 1-nearest neighbor (1-NN) rule is used to classify the test patterns. PCA, BPCA, CCIPCA, and IICA are also used as comparison.

The result is listed in Table 2. The best recognition rate is highlighted in bold typeface. To demonstrate that IOBA can achieve a better recognition performance, we also list the recognition rate that is obtained merely by NNC in Table 2. According to the analysis presented in section 3.1, for IOBA and BPCA, the target dimension is generated automatically.

However, in PCA, CCIPCA, and IICA, users need to predetermine it as a parameter. Here, we tune these parameters for PCA, CCIPCA, and IICA by ten times 10-fold cross-validation of training set to achieve an optimal recognition performance for the respective parameter setting. Furthermore, we show the target dimension and the compression ratio, i.e., the target dimension divided by the dimension of input vectors for PCA, BPCA, CCIPCA, and IICA in Table 3.

**Table 1:  $\mathcal{E}(\mathbf{B})$ : The best and near best performance are highlighted in bold typeface.**

Dataset	IOBA	PCA	BPCA	CCIPCA	IICA
Hill	<b>0.060</b>	<b>0.051</b>	<b>0.054</b>	<b>0.052</b>	0.710
Ionosphere	<b>0.265</b>	<b>0.270</b>	0.450	1.280	0.825
Musk	<b>0.153</b>	<b>0.166</b>	0.460	0.985	2.710
Optical Digits	0.233	<b>0.151</b>	0.182	<b>0.175</b>	0.625
Sonar	<b>0.146</b>	<b>0.142</b>	0.122	0.775	0.559
Waveform	<b>0.295</b>	0.486	0.720	<b>0.285</b>	0.759
Average	<b>0.192</b>	<b>0.211</b>	0.331	0.592	1.031

**Table 2: Recognition rate of the experiment: the best recognition rate is highlighted in bold typeface.**

Dataset	IOBA	PCA	BPCA	CCIPCA	IICA	NNC
Hill	56.3%	<b>57.9%</b>	<b>57.6%</b>	<b>57.6%</b>	55.4%	56.2%
Ionosphere	<b>88.0%</b>	87.1%	87.4%	86.6%	<b>87.9%</b>	86.1%
Musk	<b>97.5%</b>	<b>97.7%</b>	<b>97.9%</b>	97.4%	96.2%	<b>97.5%</b>
Optical Digits	97.6%	<b>98.1%</b>	97.9%	97.9%	<b>98.1%</b>	98.0%
Sonar	<b>86.1%</b>	82.4%	80.0%	76.4%	78.1%	81.8%
Waveform	<b>79.8%</b>	78.3%	78.3%	76.3%	<b>79.5%</b>	76.1%
Average	<b>84.1%</b>	83.6%	83.2%	82.0%	82.5%	82.6%

**Table 3: Target Dimension ( $Dim$ ) and Compression Ratio ( $CR$ )**

Dataset	IOBA	PCA	BPCA	CCIPCA	IICA
	$Dim/CR$	$Dim/CR$	$Dim/CR$	$Dim/CR$	$Dim/CR$
Hill	9/ <b>0.09</b>	24/0.24	19/0.19	15/0.15	55/0.55
Ionosphere	22/ <b>0.65</b>	30/0.88	26/0.76	28/0.82	29/0.85
Musk	79/0.48	72/0.43	62/ <b>0.37</b>	90/0.54	139/0.84
Opt	26/ <b>0.41</b>	40/0.63	48/0.75	60/0.94	32/0.5
Sonar	16/ <b>0.27</b>	29/0.48	19/0.32	40/0.67	22/0.37
Waveform	10/ <b>0.48</b>	14/0.67	15/0.71	16/0.76	12/0.57
Average $CR$	<b>0.40</b>	0.56	0.52	0.65	0.61

From Table 2, we know that for Ionosphere, Musk, Sonar and Waveform, IOBA has the best or near the best recognition performance. The average recognition rate of IOBA is higher than that for any other classifier for all databases. For the Hill and Optical Digits datasets, IOBA works a little worse than some methods. But Table 3 shows that, the compression ratio in IOBA under the two database is 0.09 and 0.41, which is much smaller than that in other methods. That means, from the aspect of compression ratio, IOBA is better than other algorithms under the two databases. Therefore, we can conclude that IOBA could reach better recognition rate and compression ratio.

## 4. CONCLUSION

As described in this paper, we propose an incremental algorithm for feature extraction tasks. The proposed method learns orthogonal components, not only theoretically but also numerically. Based on a self-adaptive threshold criterion, the target dimension can be generated automatically and adjusted adaptively during learning. Users no longer need to predetermine the target dimension. The incremental iteration method without solving eigenvalues and eigenvectors guarantees a low computing load and robustness.

In the experiments described herein, we compare IOBA with several typical feature extraction methods. Results show that the quality of components generated by IOBA is the best among these algorithms. With respect to the recognition rate and compression ratio, IOBA performance is good. Although another method might sometimes achieve slightly better recognition rate, IOBA fulfills the best compromise among component quality, classification power, and storage efficiency. When considering time complexity, the performance of IOBA is good.

## 5. REFERENCES

- [1] C. Bishop. Bayesian PCA. In *the Advances in Neural Information Processing System*, pages 382–388, 1999.
- [2] J. Weng, Y. Zhang, and W. S. Hwang. Candid covariance-free incremental principal component analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(8):1034–1040, 2003.
- [3] F. Tang, R. Crabb, and H. Tao. Representing images using nonorthogonal haar-like bases. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(12):2120–2134, 2007.
- [4] J. Chien and B. C. Chen. A new independent component analysis for speech recognition and separation. *IEEE Trans. Audio, Speech, and Language processing*, 14(4):1245–1254, 2006.
- [5] I. Dagher and R. Nachar. Face recognition using ipca-ica algorithm. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(6):996–1000, 2006.
- [6] X. He, D. Cai, and J. Han. Learning a maximum margin subspace for image retrieval. *IEEE Trans. Knowledge and Data Engineering*, 20(2):189–201, 2008.
- [7] C. J. Veenman and D. M. J. Tax. Less: A model-based classifier for sparse subspaces. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(9):1496–1500, 2005.
- [8] D. Cai and X. He. Orthogonal locality preserving indexing. In *the ACM SIGIR Conf. Information Retrieval*, pages 3–10, 2005.
- [9] H. Liu and L. Yu. Toward integrating feature selection algorithms for classification and clustering. *IEEE Trans. Knowledge and Data Engineering*, 17(4):491–502, 2005.
- [10] G. H. Golub and C. F. V. Loan. *Matrix Computations, 3rd Edition*. Johns Hopkins University Press, 1996.
- [11] X. He. Numerical dependence theorem and its application (in chinese). *Numerical Mathematics, A Journal of Chinese Universities*, 1(1):11–19, 1979.
- [12] C. L. Blake and C. J. Merz. Uci repository of machine learning databases. *Irvine, CA: University of California Department of Information*, 1996.