

Classification by Approximation of Normal Distribution for k-neighborhood Data using Steepest Descent Method

Hiromitsu Shiina*, Takayoshi Mihara** & Nobuyuki Kobayashi***

*Faculty of Informatics, Okayama University of Science, Okayama, JAPAN. E-Mail: shiina{at}mis{dot}ous{dot}ac{dot}jp

**Graduate School of Informatics, Okayama University of Science, Okayama, JAPAN. E-Mail: i14im03mt{at}ous{dot}jp

***Faculty of Human Science, Sanyo Gakuen University, Okayama, JAPAN. E-Mail: koba_nob{at}sguc{dot}ac{dot}jp

Abstract—There is a high probability of the same class of data appearing around the position of parameters of the data manifested in feature space where data parameters are obtained. This study proposes a machine learning method that approximates this using a density function of normal distribution. In addition, if small amounts of data of a different class exist where data of the same class is being collected, the impact of those classes will be significant. However, precision can be improved by determining normal density function by approximation from k units of data adjacent to the same class. The proposed technique is designed to speed up learning through semi-optimization using the steepest descent method to optimize the weight α applying k units of data for approximation and the width of the data distribution of the same class to a variance covariance matrix.

Keywords—Classification; Data Occurrence; Neighbors of Learning Data; Machine Learning; Normal Distribution; Parameter Estimation; Steepest Descent Method.

Abbreviations—Gaussian Mixture Model (GMM); Support Vector Machine (SVM).

I. INTRODUCTION

MANY classifications systems that use machine learning have been proposed. For example, for instructor-led learning, decision-tree learning [Quinlan, 1], Support Vector Machine (SVM) [Vapnik & Lerner, 2; Vapnik, 3], the k-NN [Shakhnarovich et al., 4], and the k-means [MacQueen, 5] methods have been proposed. There is also a method to obtain a separation plane to divide the feature space by class as in an SVM. Substantively, this can be considered the distribution of data based on a kernel function. The k-NN method is an instructor-led learning method for estimating classes from k adjacent items in test data. This method considers the classes of data observed in the feature space to be the largest class for data in that vicinity and does not hypothesize the distribution of the data [Duda, 6; Bishop, 7]. An improved method of the k-NN has been proposed by Toyama et al., [8], it is also utilized in other fields [Lee et al., 9]. The k-means method has been improved by some researches [Vij & Kumar, 10; Zhang & Fang, 11]. While it is known to generalize a k-means problem into a Gaussian mixture model [William et al., 12]. Another generalization of the k-means algorithm is the K-SVD algorithm [Aharon et al., 13].

The method for calculating the occurrence probability of the data is utilized while searching for values outside the model, such as for the detection of outliers [Yamanishi et al., 14; Takeuchi & Yamanishi, 15]. However, methods that approximate the distribution using the Gaussian mixture model (GMM) by EM algorithm [Dempster et al., 16] suffer low accuracy because of a high level of approximation error.

Thus, while the approximate values of data for each class may be obtained easily, there is a high probability that approximate values will appear as data of the same class. Consequently, there is a high probability of the same class of data appearing around the position of parameters of data manifested in feature space where the parameters are obtained. The method approximates by recognizing the tendency of data to occur relative to the unimodal normal density function for which the center of the data has been averaged. The occurrence probability of the data of each class is calculated by combining the normal density function approximated for each class of data to cover the entire range of data that can be obtained. When implementing the program, the feature space in which parameters are obtained is segmented, data contained in the cells of each segment are established as the respective representative points, and the probability of belonging to each class is determined for each

cell. Data is considered to belong to the class with the highest probability for each cell to which the data belongs.

In addition, if small amounts of data of different classes exist when data of the same class is collected, the impact of those classes will be significant. However, precision can be improved by determining the normal density function by approximation from k units of data adjacent to the same class. The proposed technique is designed to speed up learning time through semi-optimization using the steepest descent method [Fletcher, 17] to optimize the weight α by applying k units of data used for approximation and the width of data distribution of the same class to a variance-covariance matrix.

Finally, the results of a comparison test between multiple-kernel SVMs and the proposed method are provide as an evaluation to determine a the benchmark for machine learning.

II. MACHINE LEARNING BY CALCULATING OCCURRENCE PROBABILITY FOCUSING ON DATA

The space where data parameters are obtained is segmented into cells and the occurrence probability of each class of data is determined for each cell. Based on the assumption that the same class of data is most likely to occur at the center position of the parameters for data (feature vector), learning data is likely to occur for the cell of the feature vector corresponding to that data, and becomes less likely to occur as the distance from the cell increases. Thus, the reference cell becomes the average position of normal distribution, and cells separated from the center are assigned an occurrence probability approximated from the normal distribution.

2.1. Learning Algorithms by the Occurrence Probability in the Center of Data

The proposed method consists of four steps: (1) probability of unimodal normal density function (2) affiliation probability, (3) classification judgment, and (4) normalization.

Preparation consists of establishing D as the learning data set and $D_s (\subset D)$ as the data set of class label s . The algorithm for the proposed method is as follows. With class label S as the class, the feature vector of the data is $d \in D_s$, and $d = (d_1, d_2, \dots, d_n)$ is the feature vector where n is the number of vectors.

Step 1 (Probability of unimodal normal density function): Data is retrieved for each learning data set D and class labels s ; a normal density function value is added focusing on the cell to which the feature vector d belongs. Under ordinary circumstances, the average μ of the dimension vector n is the variance covariance matrix Σ , and n -dimensional normal density function $p(x)$ is defined as follows.

$$p(x) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu)\right) \quad (1)$$

In this study, because the covariance of variance covariance matrix is zero because the effect of learning data d , the occurrence probabilities of parameter position x and dispersion are determined to be equal. This restriction assumes that the effect of each origin is equal, and is not assumed to be misshapen. For example, Figure 1(a) shows the effect of one data item consisting of two parameters. The numerical values at the bottom of Figure 1 express the occurrence probability.

$$\text{---} \quad \text{---} \quad (2)$$

With the coordinates of the center of the cell C_x , the class label s data set is $D_s (\subset D)$, and the occurrence probability of all cells in the feature space for each class label S is determined by performing the addition expressed by the following equation for all cells. The second normal density function is approximated by following a single data item; the appearance following the addition is shown in Figure 1(b).

$$(3)$$

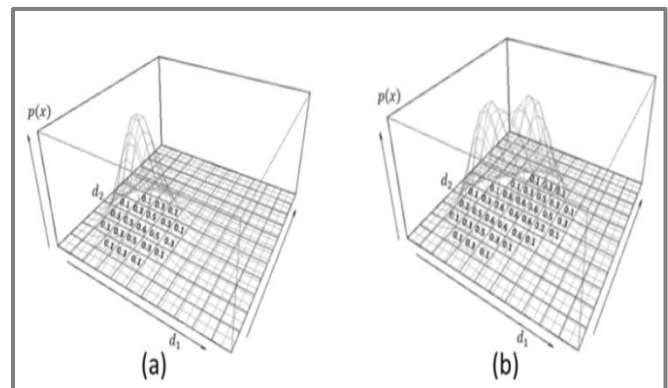


Figure 1: Occurrence Probability of neighborhood data according to Normal Density Function

Step 2 (Affiliation probability): Repeat Step 1 for each learning data d for the entire learning data set D . Figure 2 shows the process of affiliation probability in two classes.

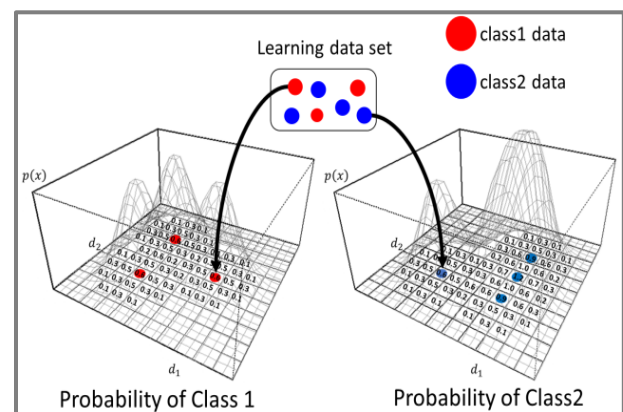


Figure 2: Affiliation Probability in each Class

Step 3 (Classification judgment): Perform normalization such that the total occurrence probability of all cells in the feature space is "1" for each class label s of the learning data.

(4)

Step 4 (Normalization): probability $N_1(s, C_x)$ that data will occur for the entire feature space is determined for each class label s of the learning data by the learning algorithm, thus, an unknown data class $C(d)$ is determined by the class for which the occurrence probability is highest at the representative point C_d for cells belonging to the feature vector of data d .

(5)

For example, if two class determinations of two parameters are color coded, a map similar to that shown in Figure 3 can be created.

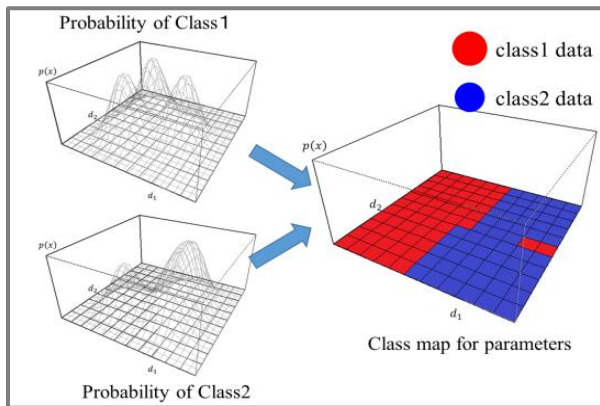


Figure 3: Class Map for Parameter

In addition, if small amounts of data of different classes exist where data of the same class is collected, the impact of those classes will be significant. However, precision can be improved by determining the normal.

2.2. Evaluation using Occurrence Probability in the Center of Data

Evaluation test when the dispersion of normal distribution has been altered. The precision test was conducted using an ordinary data set to study the utility of the proposed method. Three data sets (Pima Indians Diabetes, Heart, and Iris) from the UCI Machine Learning Repository [Lichman, 18] were used for this study. Normalization in the interval $[0, 1]$ was applied the attribute values of each data set, and the attributes were divided into 100 equal segments.

The proposed classifier sets dispersion as a parameter. Figures 4, 5, and 6 show the results of altering dispersion σ for the three data sets. In addition, an SVM line kernel and an RBF kernel were used as a baseline for precision comparison. An LIBSVM RBF kernel and an LIBSVM and LIBLINEAR linear kernel from the Machine Learning and Data Mining Group [Lin, 19; Fan et al., 20] were used as the SVM systems. With the proposed method, the best precision when dispersion was altered was 0.762 (Pima Indians Diabetes), 0.800(Heart), and 0.960(Iris). Precision was equal or somewhat inferior to the SVM. A comparison of the proposed basic method and the SVMs is shown in Table 1.

As can be seen in Table1, the results for the proposed method are equal to or worse than the SVM result; however SVM methods require significant time estimate parameters.

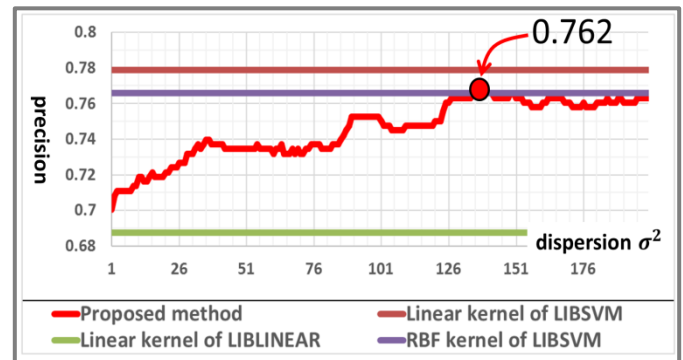


Figure 4: Evaluation of the Pima Indians Diabetes Data Set by Occurrence Probability in the Center of Data

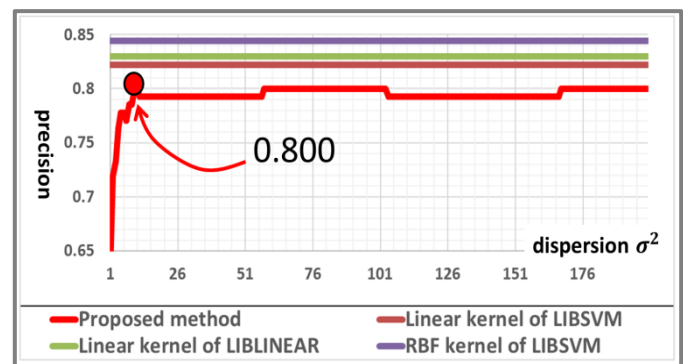


Figure 5: Evaluation of the Heart Data Set by Occurrence Probability in the Center of Data

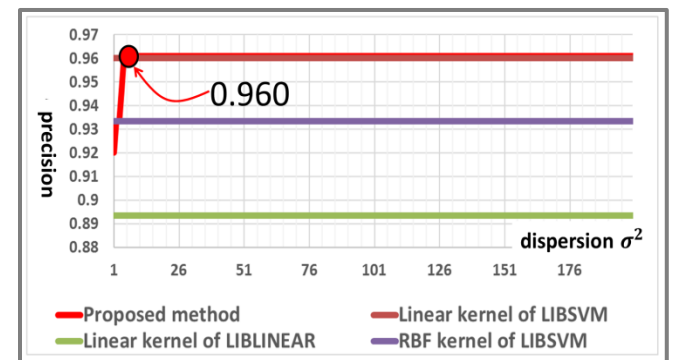


Figure 6: Evaluation of the Iris Data Set by Occurrence Probability in the Center of Data

Table 1: Experimental Results

	(a) Pima Indians Diabetes	(b) Heat	(c) Iris
Linear Kernel of LIBSVM	0.780	0.820	0.960
Linear Kernel of LIBLINEAR	0.682	0.840	0.930
RBF kernel of LIBSVM	0.761	0.830	0.890
Method by the Occurrence Probability in the Center of Data	0.762	0.800	0.960

III. MACHINE LEARNING BY APPROXIMATING NORMAL DENSITY FUNCTION IN k NEIGHBORHOOD

As mentioned previously, if small amounts of data of a different class exists where data of the same class is collected, the impact of that class will be significant. However, precision can be improved by determining the normal density function through approximation from k units of data adjacent to the same class. Here the proposed method also consists of four steps: (1) probability of unimodal normal density function (2) affiliation probability, (3) classification judgment, and (4) normalization.

3.1. Learning Algorithm by Approximating Normal Density Function in k Neighborhood

The occurrence probability for various classes was determined from the sum of the normal density function, focusing on individual data items in the previous section. In this section, we discuss approximation performed with the normal density function using k data items of approximately the same class, rather than a single data item. In the approximation method, mean vector μ , variance covariance matrix Σ , and the estimated amount $\hat{\mu}$, $\hat{\Sigma}$ are determined by the maximum likelihood method which uses the maximum log likelihood function. The average $\hat{\mu}$ and covariance matrix of the parameters of the normal density function from data d are determined by the maximum likelihood method using the following equation.

$$\hat{\mu} = \frac{1}{k} \sum_{i=1}^k d_i \quad (6)$$

$$\hat{\Sigma} = \frac{1}{k} \sum_{i=1}^k (d_i - \hat{\mu})(d_i - \hat{\mu})^T \quad (7)$$

For parameter α to adjust the degree of impact on the surrounding area, the normal density function is approximated from the n data as follows.

$$p(x, \hat{\mu}(d), \hat{\Sigma}(d)) = \frac{1}{\sqrt{(2\pi)^2 |\hat{\Sigma}(d)|}} \exp\left(-\frac{1}{2} (x - \hat{\mu}(d))^T \hat{\Sigma}(d)^{-1} (x - \hat{\mu}(d))\right) \quad (8)$$

Step 1: The average $\hat{\mu}(d)$ and covariance matrix $\hat{\Sigma}(d)$ of parameters of the normal density function from the learning data d and k items of adjacent data $D_{d,k} = \{d_1, d_2, \dots, d_k\}$ for the same class are determined by the maximum likelihood method using the following equation.

$$\hat{\mu}(d) = \frac{1}{k} \sum_{i=1}^k d_i \quad (9)$$

$$\hat{\Sigma}(d) = \frac{1}{k} \sum_{i=1}^k (d_i - \hat{\mu}(d))(d_i - \hat{\mu}(d))^T \quad (10)$$

The effect of one data item from the effects of the neighbor k data with the same class label is not a simple unimodal. It is represented by a normal unimodal distribution with bias. Figure 7 shows a normal distribution because of the influence of the data in case of $k = 3$.

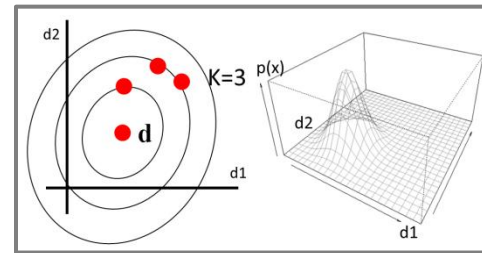


Figure 7: Effect of k Neighborhood Data

The normal density function $p(x, \hat{\mu}(d), \hat{\Sigma}(d))$ from the learning data and the advancement data is expressed as follows.

$$p(x, \hat{\mu}(d), \hat{\Sigma}(d)) = \frac{1}{\sqrt{(2\pi)^2 |\hat{\Sigma}(d)|}} \exp\left(-\frac{1}{2} (x - \hat{\mu}(d))^T \hat{\Sigma}(d)^{-1} (x - \hat{\mu}(d))\right) \quad (11)$$

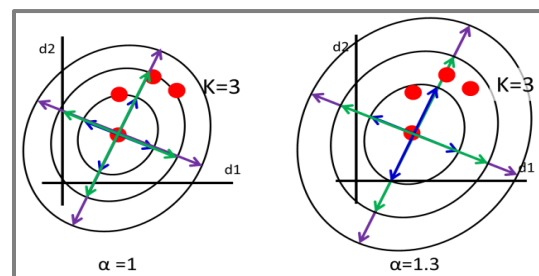


Figure 8: Difference of Effect by Parameter

The effect of α represents the degree of data influences. An example of different α value is shown in Figure 8. With (x_1, x_2) as coordinates of the center of the cell, the occurrence probability $p(x, \hat{\mu}(d), \hat{\Sigma}(d))$ of the feature space cells of each class label s is calculated for all cells. N calculation is expressed by the following equation.

$$N = \sum_{i=1}^N p(x, \hat{\mu}(d), \hat{\Sigma}(d)) \quad (12)$$

Step 2 (Affiliation probability): Repeat Step 1 for each learning data d for the entire learning data set D as in the previous section.

Step3 (Normalize): Conduct normalization such that the sum of the occurrence probability for all cells in the feature space is "1" for each class label s of the learning data as in the previous section.

$$p(x, \hat{\mu}(d), \hat{\Sigma}(d)) = \frac{N}{\sum_{i=1}^N N} \quad (13)$$

Step 4 (Parameter estimation): A parameter search is conducted to determine maximum precision using the steepest descent method for the parameters to the degree of impact on the area surrounding α and the neighborhood number k . An accurate parameter estimation for α is required when determining the initial value of k Step1 of Step 3 results. In the steepest descent method, good accuracy for

parameters α and k can be obtained by repeating the α and k updates and Step1, 2, and 3. Estimation by the steepest descent method has becomes locally minimized, therefore, the steepest descent method was implemented to cause four types to occur randomly.

3.2. Evaluation by Approximating Normal Density Function in k Neighborhood

Figures 9, 10, and 11 show the contour line where the horizontal axis represents neighborhood number k and the vertical axis represents precision α for three data sets (Pima Indians Diabetes, Heart, and Iris). In addition, Figures 9, 10, and 11 shows the convergence value by the steepest descent method. The results of the test were 0.786 (Pima Indians Diabetes), 0.850 (Heart), and 0.987 (Iris). Precision was high in a few areas. In the Heart data set, it relied significantly on the initial values. To obtain optimal parameters, a very large number of precision calculations are required. However, with the steepest descent method, the same accuracy could be obtained in some steps.

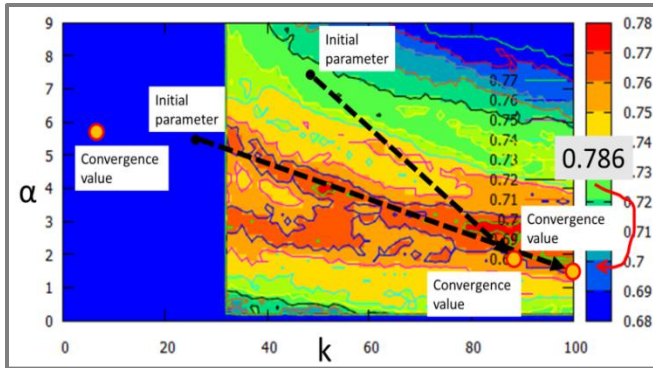


Figure 9: Contour Line Precision (Pima Indians Diabetes)

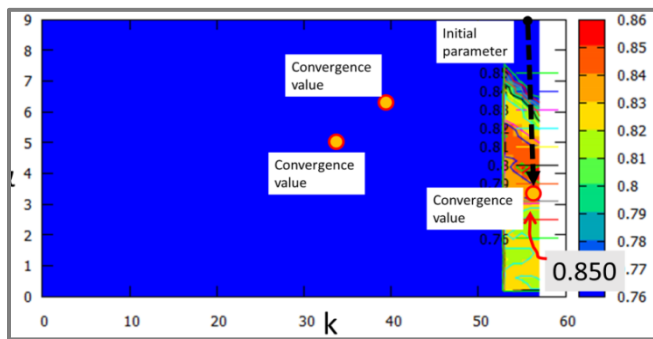


Figure 10: Contour Line Precision (Heart)

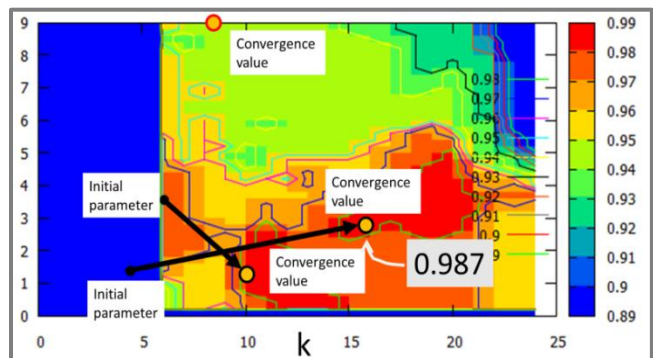


Figure 11: Contour Line Precision (Iris)

Table 2: Comparison of Parameter Estimation Accuracy by Steepest Descent Method

		(a) Pima Indians Diabetes	(b) Heart	(c) Iris
Proposed method by approximating normal density function in k neighborhood	Steepest descent method	0.786	0.850	0.987
	Best parameter	0.797	0.859	0.987
Method by the Occurrence Probability in the Center of Data(Section II)		0.762	0.800	0.960

3.3. Evaluation for Computational Time

A parameter search is conducted to determine maximum precision using the steepest descent method for the parameters to the degree of impact on the area surrounding α and the neighborhood number k . Because the proposed method uses the steepest descent method, the number of repetition of the class judgement step and the precision depend on each initial values of α and k . It is need to the evaluation of the number of repetition until convergence to evaluate computational time. Table 3 shows maximum average and standard deviation of the number of repetition.

Table 3: Comparison of Convergence by Steepest Descent Method

Number of Repetition	(a) Pima Indians Diabetes	(b) Heart	(c) Iris
Average	7.71	1.90	2.28
Standard Deviation	6.269	2.31	3.02
Maximum	29	12	19

IV. CONCLUSION AND FUTURE WORKS

Assuming that data of the same class tends to occur in an area surrounding the feature vector position of learning data, we have proposed to accumulate the entirety of learning data. A precision test was conducted using three data sets. Precision when the mutual impact of data units was considered was not as good as that of an SVM; however, a precision superior to that of the SVM was realized for when the mutual impact of data units was not considered and the proposed method was designed for parameter optimization. The techniques were also designed to speed up calculation time using the steepest descent method.

However, the problem with proposed method is that the feature vector consists of many dimensions, i.e., if many parameters types are considered, the number of cells that represent the feature space increases, thereby requiring greater memory capacity. We plan to improve accuracy for a large number of learning data items, such as product review comments, and reduce calculation volume at the time of judgment. We expect that the can be used for learning time series data.

REFERENCES

- [1] J.R. Quinlan (1986), "Induction of Decision Trees", *Machine Learning*, Vol. 1, No. 1, Pp. 81–107.
- [2] V.N. Vapnik & A. Lerner (1963), "Pattern Recognition using Generalized Portrait Method", *Automation and Remote Control*, Vol. 24, Pp. 774–780.
- [3] V.N. Vapnik (1998), "Statistical Learning Theory", *Wiley-Interscience*, New York.
- [4] G. Shakhnarovich, T. Darrell & P. Indyk (2006), "Nearest-Neighbor Methods in Learning and Vision: Theory and Practice (Neural Information Processing Series)", *The MIT Press*, Cambridge, MA.
- [5] J.B. MacQueen (1967), "Some Methods for Classification and Analysis of Multivariate Observations", *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, Pp. 281–297.
- [6] R.O. Duda, P.E. Hart & D.G. Stork (2000), "Pattern Classification", *Wiley-Interscience*, New York, Pp. 161–214.
- [7] C.M. Bishop (2006), "Pattern Recognition and Machine Learning", *Springer*, New York.
- [8] J. Toyama, M. Kudo & H. Imai (2010), "Probably Correct k-Nearest Neighbor Search in High Dimensions", *Pattern Recognition*, Vol. 43, No. 4, Pp. 1361–1372.
- [9] S.K. Lee, P. Kang & S. Cho (2014), "Probabilistic Local Reconstruction for k-NN Regression and its application to Virtual Metrology in Semiconductor Manufacturing", *Neurocomputing*, Vol. 131, Pp. 427–439.
- [10] R. Vij & S. Kumar (2012), "Improved K-Means Clustering Algorithm for Two Dimensional Data", *CCSEIT '12 Proceedings of the Second International Conference on Computational Science, Engineering and Information Technology*, Pp. 665–670.
- [11] C. Zhang & Z. Fang (2013), "An Improved K-Means Clustering Algorithm", *Journal of Information & Computational Science*, Vol. 10, No. 1, Pp. 193–199.
- [12] H.P. William, S.A. Teukolsky, W.T. Vetterling & B.P. Flannery (2007), "Numerical Recipes 3rd Edition: The Art of Scientific Computing Gaussian Mixture Models and K-Means Clustering", *Cambridge University Press*, New York, Pp. 842–850.
- [13] M. Aharon, M. Elad & A. Bruckstein (2006), "K-SVD: An Algorithm for Designing Over Complete Dictionaries for Sparse Representation", *IEEE Transactions on Signal Processing*, Vol. 54, No. 11, Pp. 4311–4322.
- [14] K. Yamanishi, J. Takeuchi, G. Williams & P. Milne (2004), "On-line Unsupervised Outlier Detection using Finite Mixtures with Discounting Learning Algorithms", *Data Mining and Knowledge Discovery Journal*, Vol. 8, No. 3, Pp. 275–300.
- [15] J. Takeuchi & K. Yamanishi (2006), "A Unifying Framework for Detecting Outliers and Change-points from Time Series", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 18, No. 44, Pp. 482–492.
- [16] A.P. Dempster, N.M. Laird & D.B. Rubin (1977), "Maximum Likelihood Form Incomplete Data via the EM Algorithm", *Journal of the Royal Statistical Society series B*, Vol. 39, No. 1, Pp. 1–38.
- [17] R. Fletcher (1988), "Practical Methods of Optimization", 2nd Edition, *Wiley-Interscience*, New York, Pp. 277–330.
- [18] M. Lichman (2013), "UCI Machine Learning Repository", URL: archive.ics.uci.edu/ml/
- [19] C.J. Lin, "Machine Learning and Data Mining Group", URL: www.csie.ntu.edu.tw/~cjlin/mlgroup/index.shtml
- [20] R.E. Fan, K.W. Chang, C.J. Hsieh, X.R. Wang & C.J. Lin (2008), "LIBLINEAR: A Library for Large Linear Classification", *Journal of Machine Learning Research*, Vol. 9, Pp. 1871–1874.



Hiromitsu Shiina. He received B.E., M.E. and Ph.D. from Toyohashi University of Technology in 1992, 1994 and 1998, respectively. He is currently associate professor of Faculty of Informatics, Okayama University of Science. His research areas included formal language theory, natural language processing and search algorithm.



Takayoshi Mihara. He received B.E from Okayama University of Science in 2014. He is master student in Informatics at Okayama University of Science now. His research areas included statistical algorithm and data mining.



Nobuyuki Kobayashi. He received B.E. and M.E. from Okayama University of Science in 1992 and 1999. He is currently a lecturer of Faculty of Human Sciences, Sanyo Gakuen University. His research areas included natural language processing, search algorithm and e-Learning.