Exhaustive Search of Feature Subsets for Support Vector Machine Classification

Jun Kitazono[1] Kenji Nagata[1] Shinichi Nakajima[2] Akira Manda[1] Satoshi Eifuku[3]
Ryoi Tamura[3] Masato Okada[1,4,a)]

Abstract: Feature selection in machine learning is an important process for improving the generalization capability and interpretability of learned models through the selection of a relevant feature subset. In the last two decades, a number of feature selection methods, such as L1 regularization and automatic relevance determination have been intensively developed and used in a wide range of areas. We can select a relevant subset of features, by using these feature selection methods. In this study, we apply an exhaustive search, instead of these methods, to the neural data recorded in the area of brain involved in face recognition. We evaluate how accurately every subset of recorded neurons can discriminate faces, by using SVM classifiers and cross validation. We show that there are a number of highly accurate neuron subsets. All of these results demonstrate that we should not select only one feature subset but exhaustively evaluate every feature subset.

Keywords: feature selection, support vector machine, cross validation, exhaustive search, generalization capability, inferior temporal cortex, face recognition

1. Introduction

Supervised learning algorithms learn from training samples consisting of pairs of inputs and outputs, and construct an appropriate input-output function that reflects the input-output relation in the training samples. The constructed function is expected to output correctly based on the input in the training samples, and even for novel inputs that are not contained in the training samples. How accurately the function can predict the output for a novel input is referred to as the generalization capability. To construct an input-output function with a high generalization capability is a shared goal among supervised learning algorithms. Feature selection [1], [2], [3] is an important technique for improving the generalization capability. Feature selection is a technique for selecting which features (elements of inputs) are relevant for making an accurate output prediction. In the last two decades, a number of feature selection methods, such as L1 regularization [4] and automatic relevance determination (ARD) [5], [6], [7], [8], [9] have been intensively developed and used in a wide range of areas. We can select a relevant subset of features, by using these feature selection techniques.

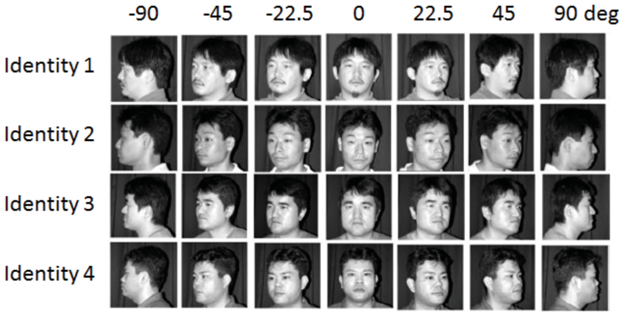

Fig. 1 Face images [10]

In this study, we apply an exhaustive search, instead of these conventional techniques, to the neural data recorded in the inferior temporal (IT) cortex of a macaque monkey. The IT cortex is a terminal of visual information processing in our brain, and is associated with object recognition, such as face identification. We use the data from the activities of 23 neurons recorded using an electrode, when a monkey is discriminating the identities of face images (**Fig. 1**, [10]). This image dataset consists of face images of four different identities viewed from seven different angles (totally, $4 \times 7 = 28$ images). We train the support vector machine [11], [12], [13] to discriminate identities regardless of the viewing angles using this data. We evaluate how accurately every subset of neurons can distinguish between identities, by performing a cross validation [14]. As a result, we show that there are multiple subsets of neurons that can perfectly distinguish between identities.

2. Method

In this section, we describe the methods used in this study. First, we describe the support vector machine [11], [12], [13]. Second, we describe the cross validation. Finally, we illustrate the exhaustive search for feature subsets.

[1] Graduate School of Frontier Sciences, The University of Tokyo
[2] Nikon Corporation
[3] Graduate School of Medicine and Pharmaceutical Sciences, University of Toyama
[4] RIKEN Brain Science Institute
[a)] okada@k.u-tokyo.ac.jp

2.1 Support Vector Machine

The support Vector Machine classifier, simply called the SVM, is a state-of-the-art model for classification that has a high generalization capability [11], [12], [13]. An SVM learns the relationship between the input data and their classes from training samples, and predicts the class of novel data.

Let us consider the following training data set.

$$\left\{(\boldsymbol{x}_i, t_i) | \boldsymbol{x}_i \in \mathbb{R}^D, t_i \in \{+1, -1\}\right\}_{i=1}^{N}, \tag{1}$$

where $\boldsymbol{x}_i$ is a D-dimensional feature vector, t_i is a class label of $\boldsymbol{x}_i$, and N is the number of samples. An SVM can find a hyperplane in the feature vector space that separates the samples with $t_i = 1$ from those with $t_i = -1$ using this data set. The obtained hyperplane is referred to as a decision boundary. Novel samples are then classified based on which side of the boundary they fall on. The decision boundary is expressed as a linear equation as

$$y(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b = 0, \tag{2}$$

where $\boldsymbol{w}$ is a weight vector. The SVM finds $\boldsymbol{w}$ and b that satisfy $y(\boldsymbol{x}_i) > 0$ for $t_i = 1$ and $y(\boldsymbol{x}_i) < 0$ for $t_i = -1$, that is, $ty(\boldsymbol{x}) > 0$ for all samples.

First, let us consider a case where the samples are linearly separable, that is, $\boldsymbol{w}$ and b exist such that all the samples satisfy $ty(\boldsymbol{x}) > 0$ [11]. The optimization problem to find a $\boldsymbol{w}$ and b that maximizes the margin is formulated as a quadratic programming problem as

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2, \tag{3}$$

subject to

$$t_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) - 1 > 0 \; (i = 1, \ldots, N). \tag{4}$$

This quadratic problem can be solved using a Lagrange multiplier.

Next, we consider a case where the samples are not completely linearly separable, as frequently occurs in real-life data [12], [13]. In this case, SVMs tolerate a restricted number of misclassifications by introducing slack variables $\xi_i \geq 0, (i = 1, \ldots, N)$. The slack variables represent penalties for misclassifications, in which $\xi_i = 0$ is for the correctly classified samples that are outside the margin and $\xi_i = |t_i - y(\boldsymbol{x}_i)|$ is for the other samples. The resulting optimization problem is as follows.

$$\min_{\boldsymbol{w},b,\xi} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}\xi_i, \tag{5}$$

subject to

$$t_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) - 1 > 1 - \xi_i \; (i = 1, \ldots, N), \tag{6}$$

$$\xi_i > 0 \; (i = 1, \ldots, N), \tag{7}$$

where C is a regularization constant. The regularization constant represents the trade-off between the margin maximization and misclassification. This quadratic problem can also be solved using a Lagrange multiplier. This method is called the soft-margin method.

2.2 Cross Validation

Cross validation (CV) is a technique for estimating how the capability of a learning model (classifier, regressor, etc.) is generalized for new data that are not used in the training [14]. A data set is divided into two parts in the CV. One part is used for the training of the model, and the other part is for validating the model's capability. This training and validating operation is iterated using different partitioning. The CV is effective when the number of the available data is limited. We explain the K-fold cross validation for a SVM below.

Let us consider the same data set as in subsection 2.1. First, we segment the data set into K equal size parts $C_1, \ldots, C_K$. For each $k = 1, \ldots, K$, we train the SVM using the data other than the k-th part C_k. We denote the decision boundary as $y_{\backslash k}(\boldsymbol{x})$. Then, we predict the class labels of the data in C_k using this $y_{\backslash k}(\boldsymbol{x})$ boundary, and compare them against the true class labels t. We iterate this operation for every $k = 1, \ldots, K$, and calculate the following cross validation error (CVE):

$$\mathrm{CVE} = \frac{1}{N}\sum_{k=1}^{K}\sum_{i\in C_k} L(t_i, y_{\backslash k}(\boldsymbol{x}_i)), \tag{8}$$

$$L(t, y(\boldsymbol{x})) = \begin{cases} 0 \; (ty(\boldsymbol{x}) > 0) \\ 1 \; (ty(\boldsymbol{x}) < 0) \end{cases}. \tag{9}$$

$L(t, y(\boldsymbol{x}))$ indicates whether the prediction of the class label of each sample is correct or not. When the prediction is correct, $L(t, y(\boldsymbol{x})) = 0$, and when the prediction is incorrect, $L(t, y(\boldsymbol{x})) = 1$. CVE represents the ratio of the number of incorrectly predicted data to the total number of data. A small CVE indicates that the generalization capability of the SVM is high. In this study, we used an N-fold CV, where N is the number of data. The N-fold CV is called a leave-one-out CV (LOOCV).

2.3 Calculate CVE for All Subsets

We selected subset A of D features and set $\boldsymbol{x}_{i_A} := (x_{i_d})_{d\in A} \in \mathbb{R}^{|A|}$. We then applied the LOOCV to the data set $\{(\boldsymbol{x}_{i_S}, t_i)\}_{i=1}^{N}$ and calculated the CVE. We carry out this process for all the $(2^D - 1)$ subsets.

3. Analyses of Feature Subsets with CVE = 0

As shown in section 4, there are multiple feature subsets with CVE = 0. We analyze the structures of these subsets. First, we show all the subsets with CVE = 0. We then visualize the weights of the decision boundaries $\boldsymbol{w}$ of these subsets by using the principal component analysis (PCA) and show that these subsets are clustered into several groups.

4. Apply Exhaustive Search to Neural Data

We apply the exhaustive search described above to neural data in this study.

4.1 Data and Settings of Simulations

The data contains the activities of 23 neurons in the anterior inferior temporal (AIT) cortex measured by conducting a single-unit recording, when the monkey was performing a sequential delayed matching-to-sample task requiring the identification of a

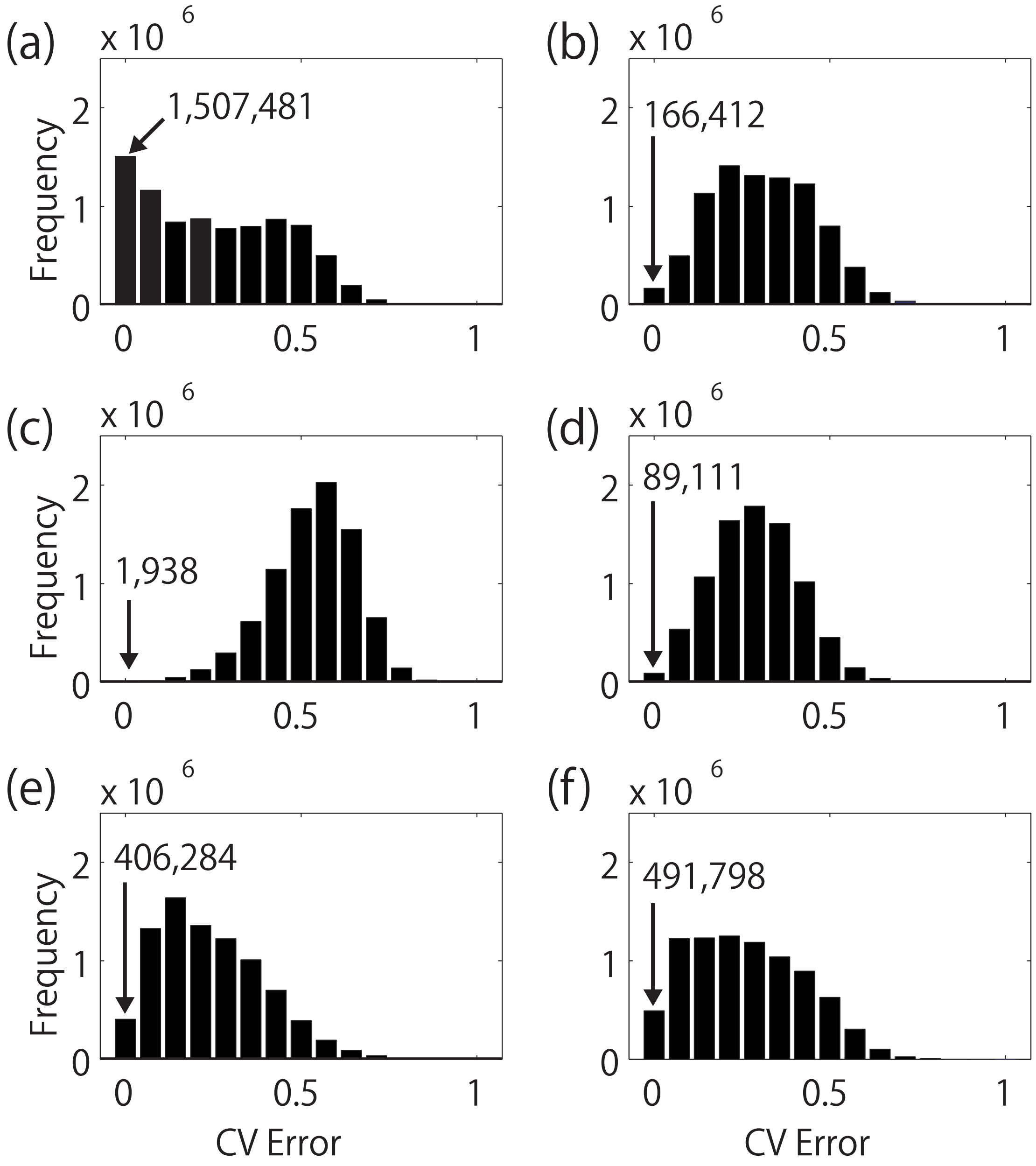

Fig. 2 CVE histograms. Each panel corresponds to (a) Identity 1 vs. 2, (b) Identity 1 vs. 3, (c) Identity 1 vs. 4, (d) Identity 2 vs. 3, (e) Identity 2 vs. 4, and (f) Identity 3 vs. 4. The inset numbers indicate the number of feature subsets with CVE = 0.

face (I-DMS task). Briefly, in the I-DMS task, a sample face image was presented to the subject and then a test face image was presented after a short delay period. The subject was required to answer whether or not the identity in test face image matches that of the sample face image. The presented image dataset (Fig. 1) consisted of the face images of four different identities viewed from seven different angles (totally, $4 \times 7 = 28$ images). The sample face was presented from a frontal view, and the test face was presented from one of seven angles. You can find the details of the experimental procedure in appendix A.1 and [10]. In this study, we used a mean firing rate for each neuron during a period from 64–496 ms after the onset of each test face image.

We used the exhaustive search on the data described above. We used the firing rates of the neurons and the identities of the faces as the inputs to and outputs for the SVM, respectively. We set $C = 5$ throughout this study. We trained the SVM to discriminate between the identities regardless of view angles. For example, we trained the SVM to distinguish seven images of identity 1 from those of identity 2 ($N = 7\times2 = 14$) using the firing rates of the 23 neurons ($D = 23$). We evaluated all the subsets of neurons using CVE.

5. Results

In this section, we present the simulation results. First, we

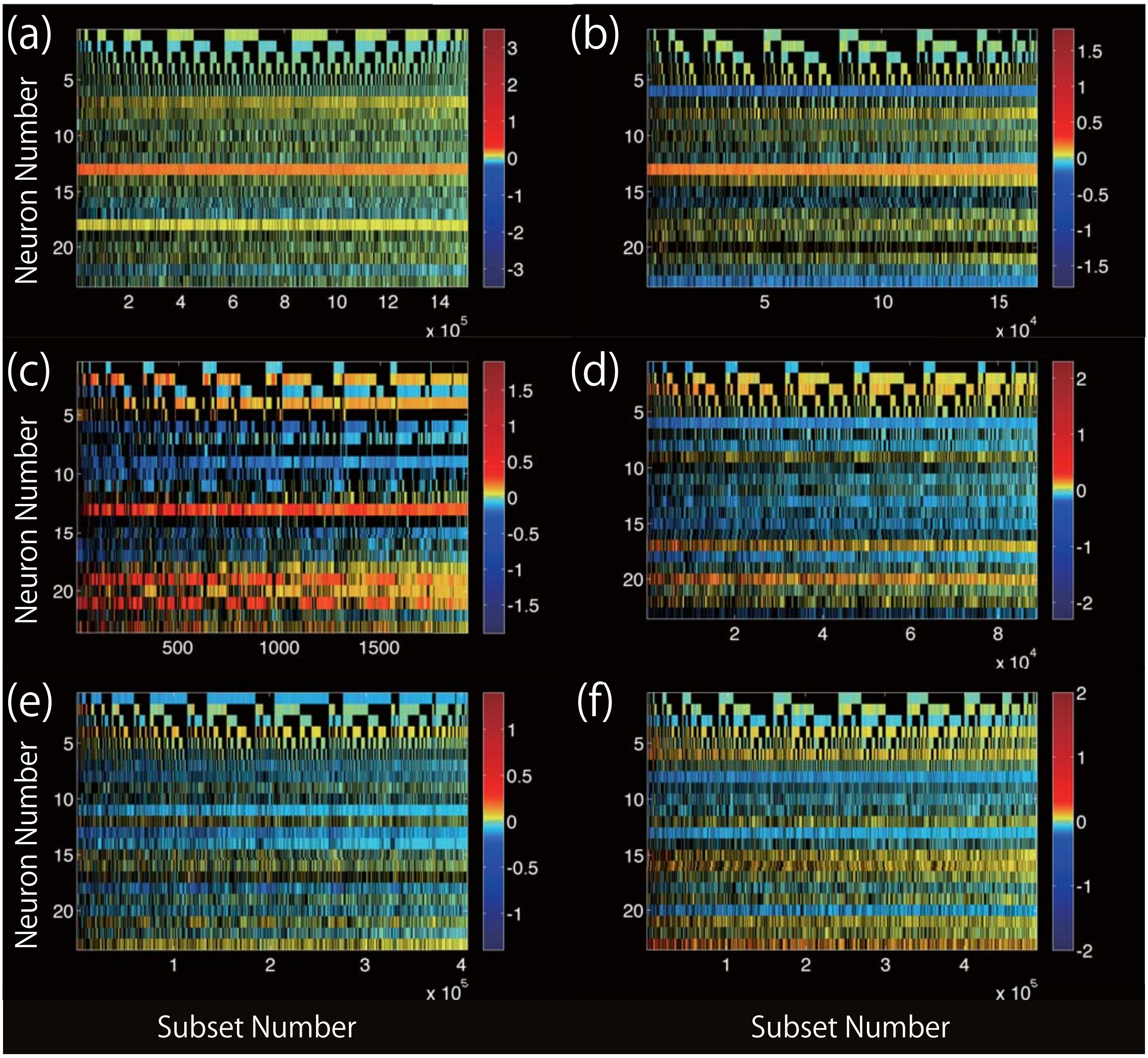

Fig. 3 Feature subsets with CVE = 0. The horizontal and vertical axes represent the number of subsets and neurons, respectively. The black cells indicate that the feature is wihin the subset. (a) Identity 1 vs. 2. (b) Identity 1 vs. 3. (c) Identity 1 vs. 4. (d) Identity 2 vs. 3 (e) Identity 2 vs. 4. (f) Identity 3 vs. 4.

present the results from the exhaustive search. Next, we focus on the feature subsets with CVE = 0.

5.1 Results of Exhaustive Search

Figure 2 shows the CVE histograms. Each panel corresponds to (a) Identity 1 vs. 2, (b) Identity 1 vs. 3, (c) Identity 1 vs. 4, (d) Identity 2 vs. 3, (e) Identity 2 vs. 4, and (f) Identity 3 vs. 4. The inset numbers indicate the number of neuron subsets with CVE = 0. We can see that these histograms are substantially different from each other. For example, on one hand there are about one and a half million subsets with CVE = 0 in Fig. 2(a), while on the other there are about two thousand subsets with CVE = 0 in Fig. 2(c).

5.2 Analyses of Neuron Subsets with CVE = 0

We focus on the Neuron subsets with CVE = 0 and analyze their structures in this section. We present the weight vectors of the decision boundaries when using the subsets with CVE = 0 in **Fig. 3**. These decision boundaries are calculated using all the N samples. Rows of this matrix correspond to neurons, and columns to subsets. The color of cells indicates a value of weight vector. Red indicates positive value and blue indicates negative value. Black indicates that a value equals to zero, that is, the neuron is not contained in the subsets. This figure shows that the subsets seem to be clustered into several groups. We visualize the decision boundaries $\boldsymbol{w}$ of the subsets using the principal component analysis (PCA) in **Fig. 4** to confirm this. We can confirm that the subsets are clustered into several groups. As shown above, the neuron subsets with CVE = 0 are clustered into several groups.

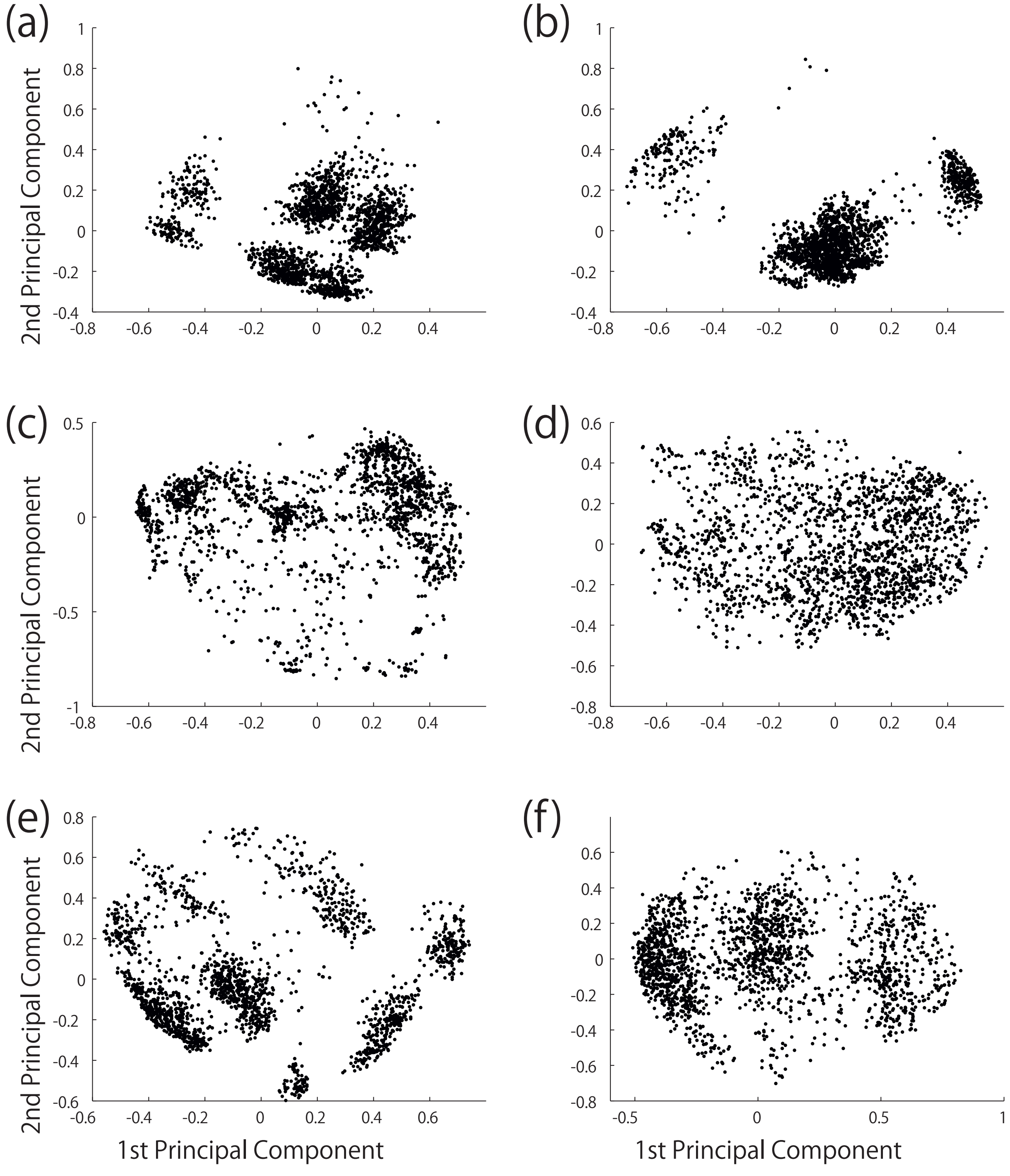

Fig. 4 Visualization of decision boundaries $\boldsymbol{w}$ of feature subsets with CVE = 0 using PCA. (a) Identity 1 vs. 2. (b) Identity 1 vs. 3. (c) Identity 1 vs. 4. (d) Identity 2 vs. 3 (e) Identity 2 vs. 4. (f) Identity 3 vs. 4. We ploted 1,938 randomly selected points for visibility in each panel.

6. Summary

We analyzed neural data to show the importance of an exhaustive search of the feature subsets in this study. We showed that there are a number of subsets of neurons with a low CVE. We also showed that these subsets are clustered into several groups. These results might suggest that multiple subsets of neurons may be used to build a robust classification of faces. Another suggestion might be that these subsets may have different criteria for classification. Thus, by performing the exhaustive search, we may discover knowledge that is hidden within the data.

In this study, we analyzed 23 neurons, i.e., the number of features is 23. Since the computational complexity of the exhaustive search grows exponentially with the number of features, the exhaustive search is often not impractical. To develop a computationally efficient method is a future challenge.

Acknowledgments This work was partially supported by a Grant-in-Aid for JSPS Fellows [10J06155(J.K.)], a Grant-in-Aid for Young Scientists (B) [No. 22700230(K.N.)], a Grant-in-Aid for Scientific Research (C) [Nos. 23500465(S.E.), 23500480(R.T.)], a Grant-in-Aid for Scientific Research on Innovative Areas "Face perception and recognition: Multidisciplinary approaching to understanding face processing mechanism (Project No. 4002)" [Nos. 20119006(S.E.), 60227296(R.T.), 23119708(M.O.)], and a Grant-in-Aid for Scientific Research (A) [No. 20240020(M.O.)].

References

[1] Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., Vapnik, V. N.: Feature Selection for SVMs, *Advances in neural information processin systems*, Vol. 13 (2001).

[2] Guyon, I. and Elisseeff, A.: An Introduction to Variable and Feature Selection, *J. Mach. Learn. Res.*, Vol. 3, 1157 (2003).

[3] Liu, H., Yu, L.: Toward Integrating Feature Selection Algorithms for Classification and Clustering, *IEEE Trans. Knowl. Data Eng.*, Vol. 17, 491 (2005).

[4] Tibshirani, R.: Regression shrinkage and selection via the lasso, *J. Royal Stat. Soc. B*, Vol. 58, 267 (1996).

[5] Mackay, D. J. C.: Bayesian methods for backprop networks, *Models of Neural Networks, III*, Springer (1994).

[6] Neal, R. M.: *Bayesian Learning for Neural Networks*, Springer. Lecture Notes in Statistics 118 (1996).

[7] Bishop C. M.: Bayesian PCA, In M. S. Kearns, S. A. Solla, and D. A. Cohn (Eds.), *Advances in Neural Information Processing Systems*, Vol. 11, pp. 382-388. MIT Press (1999).

[8] Tipping, M. E.: The Relevance Vector Machine, In S. A. Solla, T. K. Leen, and K.-R. Muller (Eds.), *Advances in Neural Information Processing Systems*, Vol. 12, pp. 652-658. MIT Press (2000).

[9] Yamashita O, Sato MA, Yoshioka T, Tong F, and Kamitani Y: Sparse estimation automatically selects voxels relevant for the decoding of fMRI activity patterns, *Neuroimage*, Vol. 42, Issue 4, pp. 1414-1429 (2008).

[10] Eifuku, S., De Souza, W. C., Tamura, R., Nishijo, H., and Ono, T.: Neuronal Correlates of Face Identification in the Monkey Anterior Temporal Cortical Areas, *J Neurophysiol.*, Vol. 91, 358 (2004).

[11] Vapnik, V. N.: *Estimation of dependences based on empirical data*, Springer (1982).

[12] Bennett, K. P.: Robust linear programming discrimination of two linearly separable sets, *Optimization Methods and Software*, Vol. 1, 23 (1992).

[13] Cortes, C. and Vapnik, V. N.: Support vector networks, *Machine Learning*, Vol. 20, 273 (1995).

[14] Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection, *Proc. 14th Int. Joint Conf. Artif.*, Vol. 2, 1137 (1995).

Appendix

A.1 Detail of the Data

In this appendix, we give an account for the data used in this study. For more technical information, see Eifuku et al., 2004.

A.1.1 Behavioral task

The monkey was trained to perform a sequential delayed matching-to-sample task, which requires the identification of familiar individuals by face (I-DMS task). In the I-DMS task, a sample (480 ms) image was presented after fixation, and then test (match or nonmatch 480 ms) images were presented after a period of interstimulus delay (992 ms). The stimulus set consisted of 28 faces (7 facial views × 4 facial identities). All visual stimuli were presented within the receptive field (RF) center of each recorded neuron that was mapped in advance of the experiment (see Recording of neural activity). In the I-DMS task, the images of the sample faces were always from the frontal view (0°), whereas the test stimuli were from one of seven images of the faces viewed from one of seven different angles (profiles from left to right:-90, -45, -22.5, 0, 22.5, 45, and 90°). The monkey was required to identify the same person who had been shown in the sample; if the test stimulus was a match, the monkey was trained to push a lever within 800 ms after the onset of a match. Some intervening (nonmatch) stimuli were presented until a match finally appeared (range: 0 to 3 intervening stimuli).

A.1.2 Recording of neural activity

First, we retrained the monkey to perform the I-DMS task. After the monkeys learned the I-DMS task at a performance level of more than 95% correct, we began recording the neuronal activity. We first isolated a single neuronal activity from the anterior inferior temporal gyrus. In advance of the experiment, the size and location of the excitatory RF region were mapped by using a mouse-controlled stimulus during a visual-fixation task. For this purpose, seven types of stimuli were used: a 2° diameter spot, a 10 × 10° random-dot field, and 10 × 10○ facial stimuli shown from five different angles (-90, -45, 0, 45, and 90°). The RF center was drawn on a tracing made on a monitor that duplicated the stimulus seen by the monkey. We then proceeded to record the neuronal activity during the performance of the I-DMS task.