Simultaneous Pattern Classification and Multidomain Association Using Self-Structuring Kernel Memory Networks

Tetsuya Hoya, *Senior Member, IEEE*, and Yoshikazu Washizawa, *Member, IEEE*

***Abstract*—In this paper, a novel exemplar-based constructive approach using kernels is proposed for simultaneous pattern classification and multidomain pattern association tasks. The kernel networks are constructed on a modular basis by a simple one-shot self-structuring algorithm motivated from the traditional Hebbian principle and then, they act as the flexible memory capable of generalization for the respective classes. In the self-structuring kernel memory (SSKM), any arduous and iterative network parameter tuning is not involved for establishing the weight connections during the construction, unlike conventional approaches, and thereby, it is considered that the networks do not inherently suffer from the associated numerical instability. Then, the approach is extended for multidomain pattern association, in which a particular domain input cannot only activate some kernel units (KUs) but also the kernels in other domain(s) via the cross-domain connection(s) in between. Thereby, the SSKM can be regarded as a simultaneous pattern classifier and associator. In the simulation study for pattern classification, it is justified that an SSKM consisting of distinct kernel networks can yield relatively compact-sized pattern classifiers, while preserving a reasonably high generalization capability, in comparison with the approach using support vector machines (SVMs).**

***Index Terms*—Constructive approach, kernel method, pattern classification, self-structuring neural networks.**

I. Introduction

IN THE area of pattern classification, artificial neural networks (ANNs) have played a significant role. One of the widely used ANNs, namely, multilayered perceptron neural networks (MLP-NNs), which were pioneered in the early 1960s and proposed as a natural extension of perceptrons [1], [2], have been used for various pattern classification problems (for the general issue of pattern classification tasks, see, e.g., [3] and [4]). In MLP-NNs, sigmoidal functions are used for representing the nonlinearity, and the network parameters, such as the weight vectors between the input and hidden and those between hidden and output layers, are usually adjusted by the backpropagation (BP) algorithm [5]–[7]. However, it is now well recognized that, in practice, learning of the MLP-NN parameters by BP-type algorithms quite often suffers from becoming stuck in local minima and requiring long period of learning, both of which are good reason for detracting their utility in online processing. This account also holds for training ordinary radial basis function neural networks (RBF-NNs) [8]–[11] or a family of self-organizing feature maps (SOFMs) [12]–[14], since tuning the network parameters resorts to a gradient–descent type optimization algorithm, which normally requires iterative and long training to obtain an input–output mapping.

In the early 1990s, the effectiveness of kernel discriminant analysis [15] was rediscovered by Specht, which led him to define the notion of probabilistic neural networks (PNNs) [16], [17]. Subsequently, Nadaraya–Watson kernel regression [18], [19] was reformulated as generalized regression neural networks (GRNNs) [20]. In the ANN context, both PNNs and GRNNs have layered structures as in MLP-NNs and are categorized into a family of RBF-NNs, in which the hidden neurons are represented by Gaussian response functions (or, Gaussian kernels) and connected via the weights to the output nodes with a linear operation (i.e., normalized sum). While the roots of PNNs and GRNNs differ from each other, the only difference between these networks (in the strict sense) is, in practice, confined to their implementation; for PNNs, the weights between the RBFs and the output node(s) (which are given identical to the target values for both the PNNs and GRNNs) are normally fixed to binary (0/1) values, whereas GRNNs generally do not hold such restriction in the weight setting (for this issue, see also [21]). Thus, unlike MLP-NNs, SOFMs, or ordinary RBF-NNs, it is essentially *not* necessary for PNNs and GRNNs to tune a number of network parameters in order to obtain a good convergence rate for achieving a reasonable generalization performance, or to worry about any numerical instability, such as local minima, or long and iterative training of the network parameters. By exploiting the property of GRNNs/PNNs, simple and quick incremental learning is possible, due to their inherent memory-based architecture,[1] whereby the network growing/shrinking is straightforwardly performed [23], [24]. Moreover, it is reported in [25] that a PNN constructed by a simple incremental training scheme even exhibits a capability to accommodate new classes, while maintaining a reasonably high generalization capability.

In a similar context, a number of constructive approaches using Gaussian kernels have been proposed and applied to pattern classification tasks in the last decade [26]–[29], though all these approaches seem to still require a rather mathematically

Manuscript received August 22, 2005; revised August 4, 2006 and November 6, 2006; accepted November 12, 2006.

T. Hoya is with the Department of Mathematics, College of Science and Technology, Nihon University, Tokyo 101-8308, Japan, and also with the Laboratory for Advanced Brain Signal Processing, Brain Science Institute, RIKEN, Saitama 351-0198, Japan (e-mail: hoya@brain.riken.jp).

Y. Washizawa is with the Laboratory for Advanced Brain Signal Processing, Brain Science Institute, RIKEN, Saitama 351-0198, Japan.

Digital Object Identifier 10.1109/TNN.2006.889940

[1] In general, the original RBF-NN scheme has already exhibited a similar property; in [22], it is stated that a reasonable initial performance still can be obtained merely by setting the centers to a subset of the examples.

1045-9227/$25.00 © 2007 IEEE

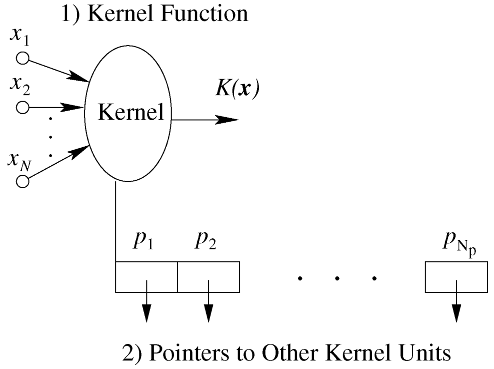

Fig. 1. KU (represented in the simplest form).

complex parameter approximation procedure for the improved accuracy (i.e., optimization of centroid vectors and/or the radii for RBFs). On the other hand, in the machine learning context, support vector machines (SVMs) have recently attracted great interest and can be regarded as one of the modern and "state-of-the-art" methods for pattern classification [30][31][32], and later some approaches for the extension to multiclass problems have been proposed [33], [34]. Nevertheless, the approaches using SVMs also essentially involve a rather arduous optimization procedure, as in the aforementioned constructive approaches, and the implementation can even become prohibitive for real world problems [31].

In other disciplines than ANNs, it is interesting to note that a number of models similar to RBF-NNs such as generalized context model (GCM) [35], the extended model, i.e., attention learning covering map (ALCOVE) [36] and Gaussian mixture model (GMM) (for the detail, see, e.g., [32]) have been proposed independently; the former two (i.e., GCM and ALCOVE) were proposed within the psychological context, while the latter has been described within a general statistical learning context.

Then, the lines of research works described previously indicate that the concept of "kernels," the term of which can be backdated to kernel discriminant analysis [15], is a key for general pattern classification problems. Moreover, it has been considered (see, e.g., [22] and [29]) that the functionality of a kernel such as an RBF represents that of a local receptive field as in the neurophysiological works in the late 1950s by Mountcastle [37] and Hubel and Wiesel [38].

II. Self-Structuring Kernel Memory (SSKM)

By exploiting the concept of "kernels," the SSKM [i.e., originally termed self-organizing kernel memory (SOKM)] was proposed as a new form of ANNs and given as a basis for modeling various cognitive/psychological functionalities [24], [39]–[41].

Kernel memory is composed of a set of kernel units (KUs), representing the most elementary constituents of memory, and their mutual connections [viz. *link weights* (LWs)]. As in Fig. 1, a KU used in this paper consists of the following two elements [41]: 1) the kernel function $K(\boldsymbol{x})$, given the input data $\boldsymbol{x} = [x_1, x_2, \dots, x_N]$ and 2) multiple addressing pointers to other KUs $p_i (i = 1, 2, \dots, N_p)$, which are used for establishing connections with other KUs (i.e., LWs). Then, a pattern classifier consisting of multiple distinct kernel networks (constructed within the SSKM principle for general M-class problems to be described later) can be formed as illustrated in Fig. 2. Thus, each kernel network is responsible for a particular class.

A. Kernel Function

Here, a kernel function is defined as a certain distance metric between the two vectors $\boldsymbol{x}$ and $\boldsymbol{t}$

$$K(\boldsymbol{x}) = K(\boldsymbol{x}, \boldsymbol{t}) = D(\boldsymbol{x}, \boldsymbol{t}) \tag{1}$$

where $\boldsymbol{t}$ is called the *template vector* of the KU, with the same dimension as $\boldsymbol{x}$ (i.e., $\boldsymbol{t} = [t_1, t_2, \dots, t_N]$) and the function $D(\cdot)$ yields a certain metric between $\boldsymbol{x}$ and $\boldsymbol{t}$. Thus, a variant of kernel functions, as defined by (1), can be considered, such as the inner product, Euclidean distance, Epanechnikov quadratic, etc. (for a concise summary and relevant issues of kernel functions, see, e.g., [32]). Of particular interest here is Gaussian response function (or an RBF), i.e.,

$$K(\boldsymbol{x}) = K(\boldsymbol{x}, \boldsymbol{c}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{c}\|_2^2}{\sigma^2}\right) \tag{2}$$

where $\|\cdot\|_2$ denotes $L2$-norm and where $\boldsymbol{c}^2$ and σ are called the centroid vector and radius, respectively, since not only the function embraces the similarity measurement of two vectors but also the output is strictly bounded within the range from 0 to 1, which evidently describes that a Gaussian response function itself performs a local pattern matching (see also [22]). Although, within the kernel memory context, it is still possible to consider a *mixture of kernel representations* rather than a single, we hereafter refer to the term "kernel" as Gaussian kernel, without loss of generality.

Then, if there are no lateral connections (i.e., represented by the LWs in between) within the kernel memory, with applying the topological equivalence property to the distinct kernel memory networks [24] (i.e., a collection of RBFs for a particular class can be represented by a subnetwork within a PNN), and if the decision unit is replaced by the output node with a sum operator, the structure is eventually reduced to a PNN [16], [17].

Note that, within the kernel memory context, the output of the kernel function $K(\boldsymbol{x})$ (or the activation/excitation) is not always necessarily transferred directly to the other KUs via the LWs; in the family of RBF-NNs, each output is always calculated as the total sum of the weight value *times* the activation of the hidden (i.e., the RBF) units, which in general yields the final output. Instead, where appropriate, the links between the kernel K_i and others are established by using the aforementioned addressing pointers $p_{i,1}, p_{i,2}, \dots, p_{i,N}$, each of which specifies the absolute locations of adjacent kernels within the memory space, and the LW w_{ij}[3] between the kernel K_i and $K_j (j = 1, 2, \dots, N_p, i \neq j)$ will be assigned, the value of which represents the *strength* of the connection in between. As stated previously, since the actual data is stored within the template (or centroid) vector $\boldsymbol{c}_i$,

[2] Instead of the template vector $\boldsymbol{t}$, we hereafter use the notation of centroid vector $\boldsymbol{c}$ for convenience.

[3] Within the general scheme of kernel memory [41], not only unidirectional but also bidirectional LW connections are considered. In such a case, we may assign different values for w_{ij} and w_{ji}. However, for simplicity, we assume $w_{ij} = w_{ji}$ throughout this paper.

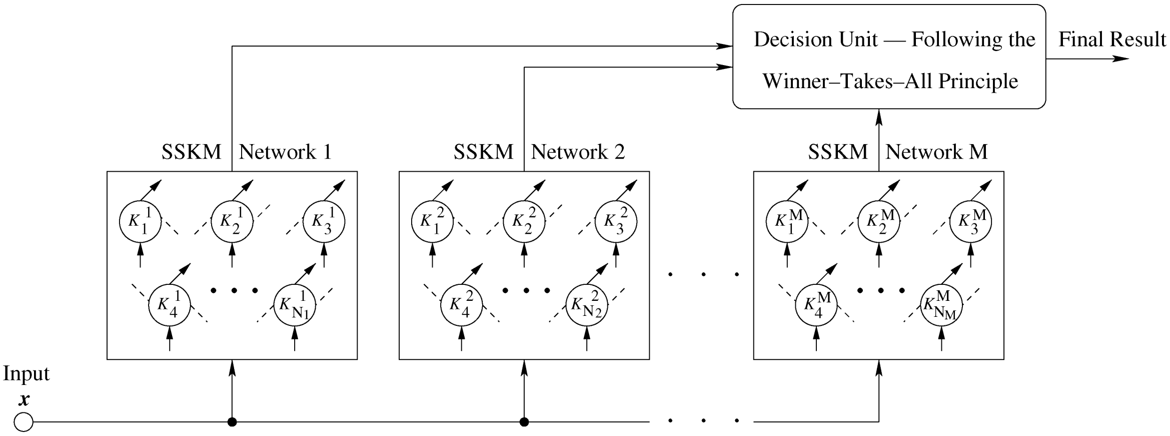
Decision Unit — Following the
Winner–Takes–All Principle
Final Result
SSKM Network 1
SSKM Network 2
SSKM Network M
K_1^1 K_2^1 K_3^1 K_4^1 K_{N1}^1
K_1^2 K_2^2 K_3^2 K_4^2 K_{N2}^2
K_1^M K_2^M K_3^M K_4^M K_{NM}^M
Input
$\boldsymbol{x}$

Fig. 2. Pattern classifier based upon distinct SSKM networks (for general M-class problems)—each kernel network is responsible for a particular class.

the change in the value of the LWs does not affect the data stored within the template vector at all.

Moreover, unlike traditional layered-type neural networks, there is essentially no structural constraint, e.g., any sparse or lateral connections are allowed, while modeling the "dense" structures similar to correlation matrix memory or two-dimensional (2-D)-map retina-like models such as by SOFMs is also possible, by effectively setting the addressing pointers of a KU (for a thorough discussion, see [41]). Nevertheless, within the SSKM context, both the number of neurons (kernels) and the weight connections are dynamically varied during the learning (or construction) phase without introducing any structural constraints in this paper, as will be described in Section II-B. Then, as justified in the simulation study to be given later, it is also noted that, due to the presence of lateral connections, a reasonable classification performance can be preserved in the case of noisy data.

In another respect, it is said that SSKM lies between the symbolic connectionist models and ANNs, while each node (kernel) can exhibit generalization capability to a certain extent. In contrast, the self-structuring memory does not inherently involve the aforementioned numerically oriented problems, i.e., long and iterative training or the associated numerical instability, unlike conventional ANN models.

B. Construction of Distinct SSKM Networks

In this paper, we will exploit a simplified version of the LW update scheme proposed in [40] and [41] in order for reduction in the degree of freedom in the parameter setting, as well as for the analytical tractability in the behavior of the kernel networks within the SSKM.

In [42, p. 62], Hebb postulated that "*When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased.*" Eventually, within the SSKM context, the following two conjectures can be drawn for simultaneous pattern classification and association.

Conjecture 1: When a pair of kernels K_i and $K_j (i \neq j)$ in SSKM are excited repeatedly,[4] a new LW w_{ij} between K_i and K_j is formed.

Conjecture 2: When a kernel K_i is excited and one of the LWs is connected to the kernel K_j, the excitation of K_i is transferred to K_j via the LW w_{ij}. Then, if K_j is not excited, the activation of K_j is recalculated by the relation

$$K_j(\boldsymbol{x}) = w_{ij} K_i(\boldsymbol{x}) (i \neq j). \tag{3}$$

In this paper, Hebb's original postulate of the increase in terms of the amount of connections among cells is not considered for keeping the structural as well as analytical simplicity, though the implementation may lead to a more flexible data representation and thereby complex and interesting behaviors of the neural memory can be observed. Moreover, to build robust pattern classifiers, we here implicitly exploit the "supervisedness," i.e., the *a priori* knowledge that each pattern vector in the training data set *always* comes with the corresponding class label. Thereby, instead of a single and large scheme of kernel memory [40], [41], a multiple number of distinct SSKM networks responsible for the respective classes will be constructed. (In other words, the LW connections across the SSKM networks are not allowed in this paper.) This strategy not only can greatly alleviate the computational load (i.e., the search for excited kernels during the construction) but also avoids the risk of generating connections among the KUs with different class labels and possible resultant misclassifications, in comparison with the original single and large approach in [40] and [41]. Then, the simplified algorithm to construct the nth SSKM network $(n = 1, 2, \ldots, M)$ is given as follows.

Constructing an SSKM Network

Step 1) Initially, there is only a single kernel $(i = 1)$ in the nth network $(n = 1, 2, \ldots, M)$, with the template vector identical to the first input vector presented ($k = 1$; i.e., the first pattern that falls in class n in the training data set), namely, $\boldsymbol{c}_1^n = \boldsymbol{x}_{n,1}$.

[4] In this paper, the repetitive excitations are not considered for simplicity.

Step 2) For $k = 2$ to {number of input data to be presented (i.e., all the training patterns that fall in class n)}, do the following.

Step 2.1) Calculate all the activations of the kernels K_i^n $(\forall i)$ in the nth SSKM network by the input data $\boldsymbol{x}_{n,k}$, [as given by (2)]. Then, if

$$K_i^n(\boldsymbol{x}_{n,k}) \geq \theta_K \tag{4}$$

where θ_K is given a certain threshold, the kernel K_i^n is excited. For all the nonexcited kernels by the direct input $\boldsymbol{x}_{n,k}$, check if the excitation of kernels via the LWs $\boldsymbol{w}_i^n$ occurs, by following the principle in **Conjecture 2**. Then, mark all the excited kernels.

Step 2.2) If there is no kernel excited by the input vector $\boldsymbol{x}_{n,k}$, add a new kernel into the network (i.e., $i \leftarrow i+1$), with setting its template vector $\boldsymbol{c}_i^n = \boldsymbol{x}_{n,k}$. Otherwise, if there are no LWs between the pairs of the excited kernels, establish new LW connections among all the excited kernels.

Then, the previous algorithm is repetitively applied to construct a total of M distinct SSKM networks for an M-class pattern classification task.

Note that, although the manner in the addition of the KUs in a single kernel network exactly follows the principle of resource allocation network (RAN) [26], the difference between RAN and SSKM still exists, since, within the SSKM, lateral connections among the Gaussian KUs can be established, where appropriate [i.e., in Step 2.2), due to **Conjectures 1** and **2**], in order to consolidate the pattern space spanned during the construction phase. To describe this, Fig. 3 illustrates the pattern space formed by a pair of Gaussian KUs K_i and K_j[5] connected via the LW w_{ij} in between. As in the figure (and under the assumption that $w_{ij} = 1$, for convenience), when a pattern presented is located at point B in the pattern space, both the KUs K_i and K_j will be activated simultaneously, since the point is within the intersection spanned by these two Gaussian KUs. (Thus, if there is such simultaneous activation, a new LW w_{ij} between these two KUs will be formed, if it does not exist yet, by following **Conjecture 1**.) Then, consider the case where the LW between K_i and K_j is already formed and another pattern which locates at point A in the space (i.e., outside the region covered by K_j) is presented. In conventional kernel-based networks (i.e., such as GRNNs/PNNs), while only the local space spanned by K_i is considered, the concatenated region covered by both K_i and K_j represents a local pattern space within SSKM. This implies that the simultaneous activation of adjacent kernels (i.e., connected via the LW in between) and, thereby, the concatenated region can represent a more precise structure of the local pattern space (i.e., in shape wise), in comparison with that spanned by a single kernel with a larger radius. (In this manner, the extension to more than two simultaneous activations is straightforward.)

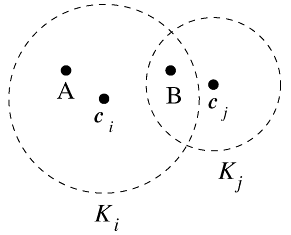

Fig. 3. Illustrative example of the pattern space spanned by a pair of Gaussian KUs K_i and K_j connected via the LW in between.

C. Testing Phase

For the testing, we also require to configure the mixture of kernel networks so as to yield the final outputs (i.e., the classification results). Then, as in Fig. 2, we here employ an ordinary "winner-takes-all" scheme for yielding the classification results. In summary, the testing procedure by means of the distinct SSKM networks is given as follows.

Summary of Testing the SSKM

Step 1)

- Present the input data $\boldsymbol{x}$ to all the SSKM networks (i.e., $n = 1, 2, \ldots, M$) and compute the activations of all the KUs by (2) within all the networks.
- Check also the activations via the LWs of K_i^n (if established during the construction phase), $\boldsymbol{w}_i^n = [w_{i,1}^n, w_{i,2}^n, \ldots, w_{i,N_{i,p}}^n]$, by following the principle in the aforementioned **Conjecture 2** [i.e., using (2)–(4)].
- Mark all the excited kernels.

Step 2)

- Obtain the maximally activated kernel $K_{\max}$ among all the marked kernels within the SSKM, i.e.,

$$K_{\max} = \max\left(K_j^n(\boldsymbol{x})\right) (\forall j, n) \tag{5}$$

where the index j denotes the jth KU of the marked kernels in Step 1).

- Then, the final classification result by the SSKM is obtained by simply referring to the value n of $K_{\max}$.

III. Extension of SSKM to Multidomain Pattern Association

Now, provided that a multiple number of pattern classifiers responsible for the respective domain data are independently constructed by using the algorithm described in Section II, we extend the idea of the LWs between the KUs to that of cross-domain LWs.

Conjecture 3: If the KUs $K_i^n, K_j^n, K_k^n, \ldots$, in the nth distinct network of SSKM $1, 2, \ldots, N_d$ (N_d: number of domains) are excited simultaneously (and repeatedly[6]), new cross-domain LWs among such KUs are formed.

[5] Note that, for convenience, the superscript n representing a particular class is omitted within this paragraph for denoting both KUs and LWs.

[6] In this paper, the repetitive excitations are again not considered for simplicity.

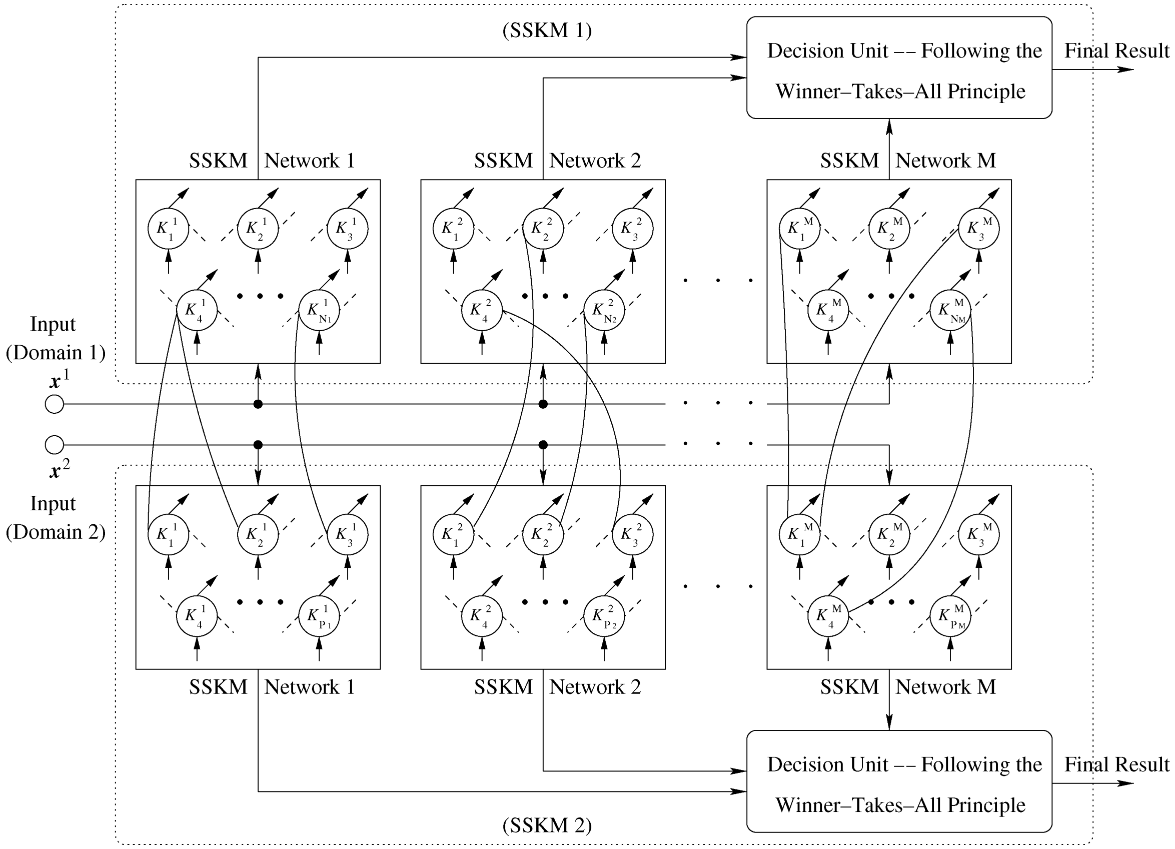

Fig. 4. Simultaneous dual-domain (i.e., $N_d = 2$) pattern associator and classifier based upon distinct SSKM networks and an illustrative example of the cross-domain LWs.

Then, via such cross-domain LWs, associative pattern classification can also be performed; to simplify the story, consider the situation where only two SSKM-based pattern classifiers as illustrated in Fig. 4 ($N_d = 2$), e.g., one for a particular auditory domain data (i.e., SSKM 1) and the other for visual (i.e., SSKM 2), are constructed, as well as the cross-domain LWs between some of the KUs among these classifiers are established (e.g., the LW between K_4^1 in SSKM 1 and K_1^1/K_2^1 in SSKM 2,..., besides the LWs between the KUs within the distinct networks), within the SSKM principle.[7] Then, a particular set of auditory pattern data (i.e., given by $\boldsymbol{x}^1$) can activate not only some of the KUs within the kernel network(s) of one domain but also those in other domain(s), i.e., visual; the activation of some KUs in the visual part occurs *without the presentation of any visual input data* to SSKM 2

$$K_j^n(\text{in SSKM } q) = v_{ij} K_i^n(\boldsymbol{x}^p)(\text{in SSKM } p)(\forall i, j) \quad (6)$$

where $n(= 1, 2, \ldots, M)$ denotes the network ID of SSKM 1/2, v_{ij} is the cross-domain LW value, $p = 1(2)$, and $q = 2(1)$. Thereby, with this configuration, "association" of pattern data, or in a more cognitive sense of memory association, can be modeled. [However, note that, as given by (6), establishment of the cross-domain LWs is restricted between the networks in SSKM p and q with the same network IDs in this paper.]

Related to the aforementioned cross-domain pattern association, one of the active areas of research has been data/sensory fusion by exploiting a mixture of experts (see, e.g., [43]–[46]). However, the objective here is rather different from these previous works; since their works are mostly targeted at the improvement in classification accuracy of a particular object by (somehow) combining the outputs obtained from the classifiers, which are responsible for the respective data domains and independently constructed using (normally) conventional layered-type neural networks, generally no lateral (or cross)-connections between the pattern classifiers are considered. Hence, the associative data processing such as i) the activation from K_4^1 in SSKM 1 with the input $\boldsymbol{x}^1$ and ii) the subsequent activation from K_1^1 and/or K_2^1 in SSKM 2 via the cross-domain LWs in between (i.e., without giving the input $\boldsymbol{x}^2$), as in Fig. 4, is not generally considered in conventional data/sensory fusion or other ANN approaches. Moreover, since establishment of such cross-domain links and subsequent activation transfer via such links does not always occur but rather is dependent upon the input data presented, the system comprising the SSKM can, e.g., flexibly reconfigure its task-planning mechanism to cope with the incessantly varying surrounding environment using multidomain data.

IV. Simulation Study

In the simulation study, three data sets extracted from three public handwritten/spoken digit databases, i.e., OptDigits [47],

[7] Generalizing the notion of cross-domain LWs to the case where $N_d > 2$ is straightforward.

TABLE I
THREE DATA SETS USED FOR THE PATTERN CLASSIFICATION TASKS

Dataset	Total Num. of Patterns in the Training Set	Total Num. of Patterns in the Testing Set	Length of Each Pattern Vector	Num. of Classes
OptDigits	1200	400	64	10
PenDigits	1200	400	16	10
SFS	540	360	256	10

TABLE II
MINIMUM AND MAXIMUM DISTANCES BETWEEN ALL THE PAIRS OF PATTERNS IN THE TRAINING DATA SET COMPUTED FOR THE THREE DATA SETS

Dataset	Minimum Distance	Maximum Distance
OptDigits	1.0	9.3
PenDigits	0.1	5.7
SFS	2.4	11.4

TABLE III
SUMMARY OF THE PATTERN CLASSIFICATION RESULTS—WITH THE PARAMETERS CHOSEN AFTER PERFORMING FIVEFOLD CV

	kNN				
Dataset	1NN	2NN	3NN	4NN	5NN
OptDigits	96.5%	95.8%	96.0%	95.3%	95.8%
PenDigits	97.0%	96.8%	97.5%	97.3%	96.8%
SFS	96.9%	94.7%	97.2%	96.6%	97.5%

	SVMs		
Dataset	C. Rates	Num. SVs	Params.
OptDigits	96.3%	710	$\sigma = 7.9$, $c = 100$
PenDigits	96.3%	286	$\sigma = 4.8$, $c = 100$
SFS	98.3%	846	$\sigma = 9.9$, $c = 10$

	Distinct SSKM Networks		
Dataset	C. Rates	Num. KUs	Params.
OptDigits	96.3%	782	$\sigma = 3.8$, $\theta_K = 0.7$
PenDigits	96.8%	697	$\sigma = 1.0$, $\theta_K = 0.8$
SFS	97.5%	519	$\sigma = 5.4$, $\theta_K = 0.7$

PenDigits [48], and speech filing system (SFS) [49], were used; for the former two, the data set contains a subset of the pattern data extracted from the corresponding databases (under the same names) obtained from the University of California at Irvine (UCI) Machine Learning Repository. The original databases for both the OptDigits and PenDigits contain some thousands of handwritten digits ready for performing optical/pen-based recognition tasks. For the third (SFS), the data set was composed by the feature data obtained using the original SFS database for spoken digit voice recognition tasks. For the pattern classification tasks, the feature extraction was based upon a combined linear predicting coding (LPC) and mel-cepstral analysis, which is a commonly used scheme for speech coding (see, e.g., [50] and [51]). Then, a more detailed description of the three data sets used is given in Table I.

For the (regular) pattern classification tasks, the three methods—k-nearest neighbors (kNNs, $k = 1, 2, \ldots, 5$), SVMs, and the method based upon the distinct SSKM networks—were used. Then, a performance comparison was made between the three methods.

For the SVMs, a simple technique for multiclass classification tasks was introduced: a total of ten distinct SVMs, each of which acts as a binary pattern classifier (i.e., to judge whether the input data given falls into one particular class or not), were constructed, and then, similar to the SSKM approach, the ordinary "winner-takes-all" scheme was applied to generate the final classification results

$$\{\text{Final result by SVMs}\} = \arg\left(\max\left(y_i(\boldsymbol{x})\right)\right) \tag{7}$$

where $y_i(\boldsymbol{x})$ $(i = 1, 2, \ldots, 10)$ is the output obtained from the ith SVM, when a pattern vector $\boldsymbol{x}$ was presented.

For the SSKM, simultaneous activations of only the nearest neighbors (i.e., excluding the subsequent activations by "neighbors of neighbors" due to the activation transfer via the LWs) were considered for the purpose of analytical simplicity in the simulation study.

A. Choice of the Radii for Gaussian Response Function

For both the SVMs and SSKM, the Gaussian response function given in (2) was chosen as kernel functions. Then, selection of the radii σ gives a significant impact upon the performance. Although the choice is normally made by trial and error, it has been empirically shown that a unique setting for σ chosen within the following range can, as a rule of thumb, still yield a reasonable generalization performance [11], [40], [41]:

$$d_{\min} < \sigma < d_{\max} \tag{8}$$

where the values $d_{\min}$ and $d_{\max}$ correspond to the minimum and maximum distance between all the pairs of patterns in the training data set, respectively. For the three data sets, both the minimum and maximum distances are computed as shown in Table II.

B. Classification Results

The results and the performance comparisons for the regular pattern classification tasks are then summarized in Table III. Note that, for both the SVMs and distinct SSKM networks, all the classification rates shown were computed based upon the fivefold cross-validation (CV) data sets (see [3] and [52]); the original training set was first divided into five subsets (i.e., one for validation and the rest for training the classifiers), and the parameters which yielded the highest score and then minimum number of nodes [i.e., the number of support vectors (SVs) for SVMs, whereas that of KUs for SSKM] during the

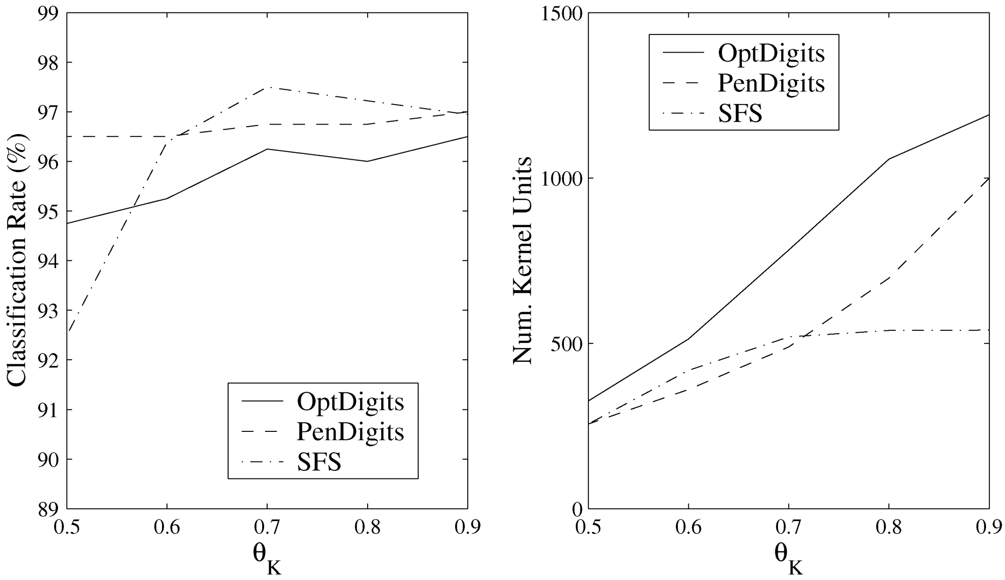

Fig. 5. Comparison of the classification rates (left) and total number of KUs generated within the distinct SSKM networks during the construction phase (right), with varying the value θ_K.

CV were used to construct a pattern classifier for testing (i.e., using the third data set). During performing the CV, the value of the radius σ was varied within the range given by (8) for both the SVMs and SSKM. For the SVMs, another parameter, i.e., $c(= 0.1, 1, 10, 100)$: upper bound for the weight coefficients, was also varied, whereas $\theta_K(= 0.6, 0.7, 0.8, 0.9)$ as given in (4) for the SSKM during the CV (then, the degree of freedom was eventually two for both the SVMs and SSKM). For all the simulations in this paper, the value of the LWs for the SSKM was fixed as $w_{ij}/v_{ij} = 1.0$ [i.e., in (3) and (6)] and unaltered. Then, the classification results given in Table III were obtained using the testing data set with the networks constructed with the aforementioned two parameters (i.e., σ and c for the SVMs, whereas σ and θ_K for the SSKM) chosen during performing the CV.

C. Choice of the Activation Threshold Factor θ_K

We then investigated the performance of SSKM by varying the parameter θ_K with fixing the other, i.e., σ, to the values found best by performing the fivefold CV for the respective data sets (as in Table III; $\sigma = 3.8$, 1.0, and 5.4 for the OptDigits, PenDigits, and SFS data sets, respectively. In addition, note that the simulation results shown hereafter are those using the full version of the training sets). Fig. 5 compares both the classification rates (left) and number of KUs generated (right), when varying only the value θ_K. As in the figure, it is observed that, while the number of KUs generated can be greatly varied by the factor θ_K, the overall classification rates still remained satisfactory, without any catastrophic degradation for all the cases. In particular, it is notable that, while a total of 697 KUs were required for the PenDigits data set with the setting of $\theta_K = 0.8$ (as in Table III), a similar performance (96.5%) can also be obtained by the setting $\theta_K = 0.5$ with a much smaller number of the kernels (i.e. 256 in this case, whereas 697 as in Table III). This observation indicates that, unlike SVMs, the number of KUs can be controlled reasonably by the factor θ_K within the proposed SSKM approach, with introducing no serious performance degradation.

D. Discussion

Note that, as shown in Table III, the classification rates obtained using the SSKM approach were virtually identical to those by the SVMs for all the three data sets, while the number of SVs/KUs varied greatly with the choice of the parameters during the simulation. Nevertheless, for the SVMs, the performance was obtained at the expense of rather complex mathematical operations, i.e., 1) multiple presentation of the whole training data (i.e., requiring $M(= 10)$ times of the presentation by the SVMs, whereas just once for the SSKM approach) and 2) iterative operations for relatively large matrices due to the utility of such as quadratic programming, during the construction (training) phase, for the sake of the convergence rate/improved accuracy in the classification rate.

In this view, it is generally considered that, as a rule of thumb, the training time required for SVMs is much more than that of a one-pass algorithm such as SSKM, while the testing time can be less[8]; i.e., if Gaussian kernels are chosen for the SVMs, the structure of each SVM is equivalent to that of an ordinary RBF-NN, requiring only forwarding the activation of the hidden to the output layer, while a further step is still required in the case of SSKM in order to check the existence of/compute the activation transfer via the lateral connections among the KUs. Hence, it is considered that the testing time for SSKM is largely dependent upon the number of LWs among KUs in each distinct network generated during the construction.

In comparison with RAN, it is said that the training time for SSKM is much less than that for RAN; in RAN, every time a

[8] For kNNs, only the computational resources required for the testing are considered; in kNNs, since all the training patterns must be held during the testing phase and since the testing is achieved by simply finding the k-nearest pattern vectors among the training vectors, the testing time is exactly dependent upon the size of the training data set and, thus, normally longer than all the other algorithms used in the paper. For the detail, see, e.g., [4].

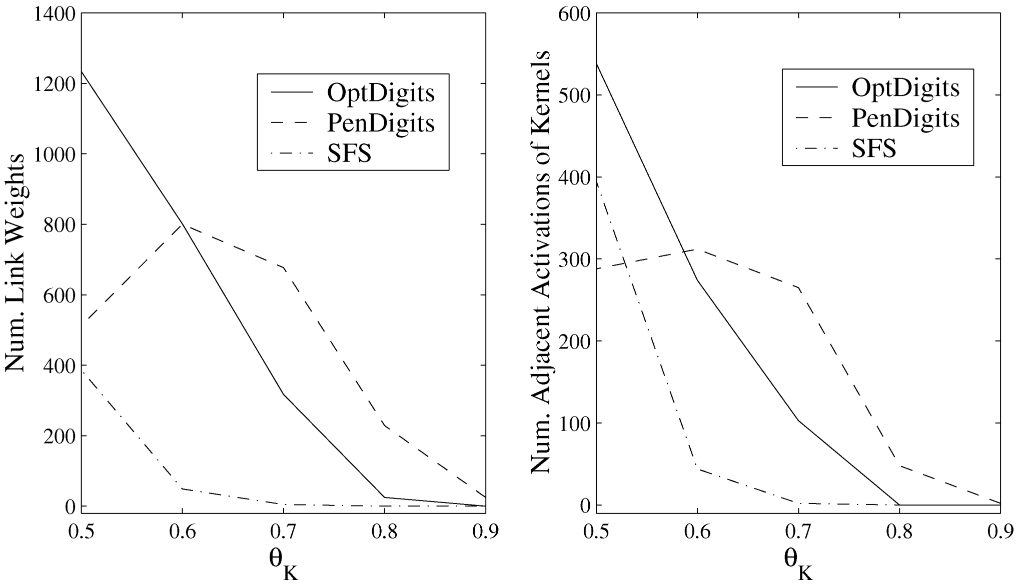

Fig. 6. Variation in the total number of LWs generated during the construction phase (left) and that of adjacent activations of KUs occurred during the testing phase (right), due to the factor θ_K.

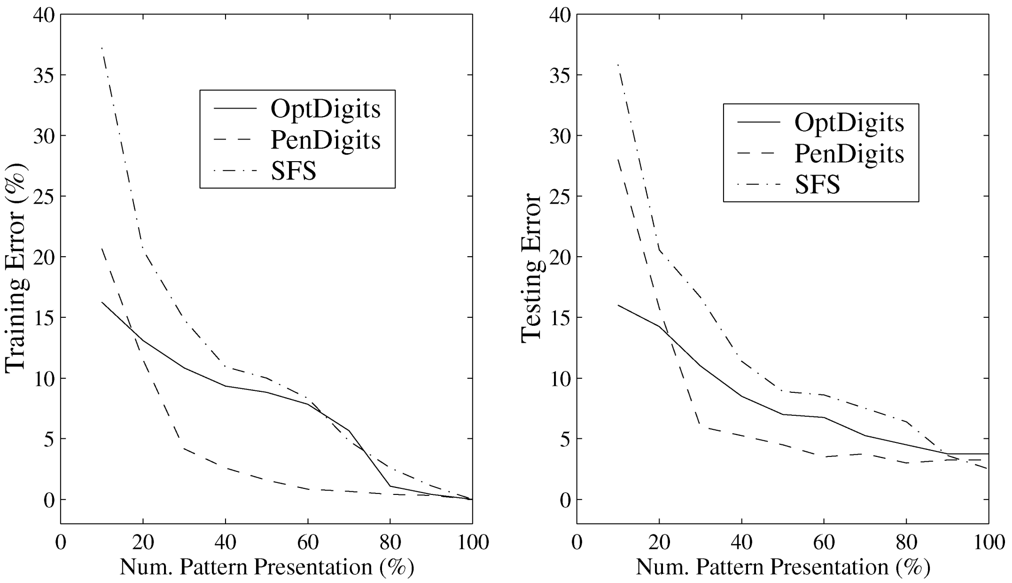

Fig. 7. Variation in the classification errors using training/testing data sets during the construction phase.

new unit is about to be added to the network, the computation of minimum distance between the incoming training pattern and all the existing units in the network is required, while, within SSKM, such computation [i.e., corresponding to the judgement if the KU is activated by (4)] is required only for the KUs within a particular SSKM network with the same class ID as a new training pattern is given. Moreover, unlike RAN, each SSKM network responsible for the corresponding class is constructed on a modular basis and distinct from others. In other words, this manner of construction is equivalent to that of the subnetworks within a PNN, and, hence, accommodation of new classes [25] can be naturally achieved, the capability of which is also desirable in general pattern classification problems.

E. Variation in the Number of LWs

Next, as observed in Fig. 6, the total numbers of LWs generated within all the distinct SSKM networks during the construction phase and occurrences of adjacent kernel activations during the testing phase were relatively higher for smaller values of θ_K. In contrast, during the construction phase, the number of LWs was steadily growing, besides the number of KUs, as shown in Fig. 8, while the training/testing errors were almost consistently decreasing in Fig. 7. However, it is also observed that the tendency of steady growing in the number of LWs is not eminent for the SFS but for the OptDigits and PenDigits data sets.

These results then indicate that the number of patterns is insufficient to consolidate well the pattern space, by examining the longer vector length (256) compared to that of the other two (i.e., 64 for OptDigits and 16 for PenDigits) as in Table I.

F. Robustness to Varying the Presentation Order of Training Patterns

Since the construction of SSKM networks is based upon a one-pass algorithm, we investigated the impact upon the clas-

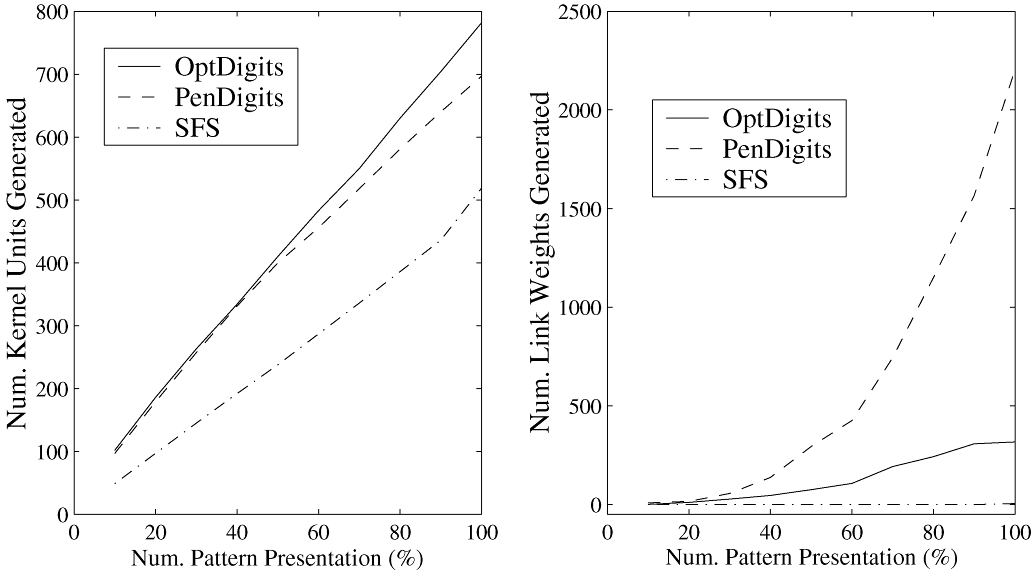

Fig. 8. Variation in the total number of KUs (left) and LWs during the construction phase (right).

TABLE IV
CLASSIFICATION RESULTS OF SSKM—USING THE SUBSETS OF TRAINING DATA WITH VARYING THE PATTERN PRESENTATION ORDER

	Testing Error Rates				
Dataset	Subset1	Subset2	Subset3	Subset4	Subset5
OptDigits	16.7%	15.0%	13.5%	13.5%	16.5%
PenDigits	17.2%	15.7%	18.7%	19.2%	17.7%
SFS	23.9%	25.6%	23.3%	25.8%	25.3%

	Num. Kernel Units Generated				
Dataset	Subset1	Subset2	Subset3	Subset4	Subset5
OptDigits	103	104	97	98	101
PenDigits	82	76	81	78	75
SFS	55	53	57	53	53

TABLE V
VARIATION IN THE TESTING ERRORS WHERE THE NUMBER OF NOISY PATTERNS WAS INCREASED FROM 0 TO 80(40)

	Testing Error Rates				
Dataset	0	20(10)	40(20)	60(30)	80(40)
OptDigits(SVM)	16.5%	21.3%	31.3%	34.5%	41.3%
OptDigits(RAN)	19.2%	22.2%	25.2%	27.5%	29.2%
OptDigits(SSKMa)	16.7%	22.0%	37.7%	44.5%	49.0%
OptDigits(SSKMb)	16.7%	20.2%	30.2%	34.2%	39.7%
PenDigits(SVM)	22.5%	25.8%	25.5%	23.5%	25.8%
PenDigits(RAN)	20.7%	23.7%	24.0%	34.2%	40.5%
PenDigits(SSKMa)	18.5%	21.5%	25.5%	43.5%	44.2%
PenDigits(SSKMb)	17.2%	20.2%	21.2%	32.5%	38.0%
SFS(SVM)	15.3%	24.2%	28.9%	28.9%	33.6%
SFS(RAN)	18.1%	21.4%	23.9%	24.7%	27.2%
SFS(SSKMa)	24.2%	37.2%	44.2%	43.9%	50.3%
SFS(SSKMb)	23.9%	37.2%	44.2%	43.9%	49.7%

sification performance where the presentation order of training patterns was varied. For this, we used five different subsets of the original training data with randomly sorted orders. Then, the total number in each subset was set only to one-sixth of the original (i.e., 200 for the Opt/PenDigits, whereas 90 for the SFS), while the number of patterns in the testing set was unaltered. This was to simulate a rather hard situation where only an incomplete set of training patterns is available. For the SSKM, while the radii values σ were chosen the same as those found by performing the fivefold CV (in Table III), the setting $\theta_K = 0.5$ was used for all three data sets (also for the simulation study of the noisy training data to be described in Section IV-G). Table IV shows both the testing error rates and number of KUs generated during the construction.

As shown in Table IV, since the error rates as well as the number of KUs generated remained almost intact, it can be empirically confirmed that the impact upon the overall performance of the SSKM due to varying the presentation order of the training pattern vectors is negligible.

G. Robustness to Noisy Training Patterns

Next, we investigated the robustness to noisy training patterns. To examine this, the total number of training patterns was again reduced to one-sixth of the full version, and some of the patterns in the training data sets were replaced by noisy patterns. Then, the original testing data sets were used for testing. Tables V and VI compare the testing error rates and numbers of SVs/RBFs(for RAN)/KUs and LWs obtained by the SVMs, RAN, and SSKM, respectively, where the number of noisy patterns was increased from 0 to 80(40 for the SFS) with a step size of 20 (10 for the SFS). In order to see how the lateral connections are effective within the SSKM, we also compared the results of the SSKM with (SSKMb)/without LWs (SSKMa) (i.e., by simply setting $w_{ij} = 0, \forall i, j$). For RAN [26], the degree of

TABLE VI
VARIATION IN THE NUMBER OF SVs/RBFs (FOR RAN)/KUs LWs, WHERE THE NUMBER OF NOISY PATTERNS WAS INCREASED FROM 0 TO 80(40)

	Num. SVs/RBFs/KUs(LWs)				
Dataset	0	20(10)	40(20)	60(30)	80(40)
OptDigits(SVM)	287	374	480	605	644
OptDigits(RAN)	171	177	183	192	195
OptDigits(SSKMa)	103(0)	112(0)	123(0)	133(0)	139(0)
OptDigits(SSKMb)	103(33)	112(28)	123(32)	133(29)	139(21)
PenDigits(SVM)	128	225	321	407	464
PenDigits(RAN)	119	142	170	176	186
PenDigits(SSKMa)	82(0)	90(0)	104(0)	109(0)	124(0)
PenDigits(SSKMb)	82(37)	90(33)	104(33)	109(15)	124(14)
SFS(SVM)	344	364	384	411	446
SFS(RAN)	90	90	90	90	90
SFS(SSKMa)	55(0)	58(0)	61(0)	68(0)	74(0)
SFS(SSKMb)	55(5)	58(3)	61(1)	68(1)	74(2)

freedom is generally higher than SVM and SSKM; there are six parameters to fix during the construction phase (while only two or three for both SVM and SSKM). Then, we chose almost the same values as those employed in [26], i.e., $\alpha = 0.05$, $\epsilon = 0.2$, $\delta_{\min} = 0.07$, $\delta_{\max} = 0.7$, $\kappa = 0.87$, and $\tau = 2$.

As shown in Table VI, the numbers of SVs are much higher than those corresponding to the other three algorithms for all the cases, while the error rates by SVMs are relatively lower than the others. Similarly, the lower error rates were also achieved by RAN for both the OptDigits and SFS cases, with relatively small numbers of RBFs. However, this can be expected due to optimization of other parameters during the construction (i.e., adjusting the center positions and radii values for the RBFs, as well as the weight connections between the hidden and output layers by applying a gradient–descent method). Besides this, although the most memory-expensive factor for all the four algorithms (i.e., RAN, SSKMa, SSKMb, and SVMs) is in practice considered to be the support/centroid vectors, other multiple-valued parameters are also required to be held for both SVMs and RAN; i.e., unlike SSKM, both SVMs and RAN eventually yield a fully connected RBF network (i.e., if Gaussian kernels are used). Then, the independent radii values for RAN and weight values between all the hidden and output units for both SVMs and RAN must be preserved during both the construction and testing phase. In contrast, for the SSKM in this paper, only the LW values (i.e., for the partial/sparse lateral connections among the KUs) are essentially required to be stored, besides the centroid (or template) vectors.

Then, by comparing the results of SSKMa with SSKMb as in Tables V and VI, it can be empirically confirmed that the presence of LWs somewhat contributes to preservation of the reasonable classification performance in noisy situations, while keeping a relatively small number of KUs as in Table VI. This is particularly true for the cases of OptDigits and PenDigits, as the number of noisy patterns is increased from 40 to 80.

H. Multidomain Pattern Association Tasks

The associative pattern classification task in this paper was designed to imitate the situation where a specific voice sound (auditory) input to a particular area of memory excites not only the area responsible for the auditory modality but (in parallel) the visual counterpart (i.e., dual-domain pattern association). As aforementioned, this is then somewhat relevant to the issue of modeling associations between different cognitive modalities or in a more general context of memory association.

To simulate the memory association, we used the SFS data set to represent the auditory part of the memory, while the PenDigits was used for the visual counterpart. The composite network structure as in Fig. 4 was formed using the training set of both the SFS (for SSKM 1) and a subset of PenDigits (for SSKM 2; with the same number of training patterns as the SFS, i.e., 540). During the training, not only the LWs between the corresponding units in a single distinct SSKM network but also those across SSKM networks (i.e., cross-domain LWs) were thus established (with the aforementioned constraint that only the KUs in SSKM 1 and 2 with the same network IDs were allowed to be connected, where appropriate). Then, in the simulation study, we observed how each input pattern in one domain can activate the KUs within the other domain via the cross-domain LWs,[9] using the corresponding testing data set (i.e., the total number of testing patterns was set to 360 for both the SFS and PenDigits).

For imitating such memory association, it is intuitively considered that the presentation order of training patterns can greatly affect the formation of the associative links. In the simulation study, however, we resorted to an artificial data presentation manner, in order to represent a simple but clearly observable manner of establishing pattern associations between the two different modalities within the SSKM. Then, the pattern data were presented alternatively across the two training datasets; viz., the presentation of the first pattern for class $i (i = 1, 2, \ldots, 10)$ in the SFS to SSKM 1, then, the first pattern in the PenDigits for the same class (i.e., digit) as the SFS to SSKM 2, followed by the presentation of the second pattern in the SFS to SSKM 1 and by that in the PenDigits to SSKM 2 . . .

Tables VII and VIII summarize the simulation results of cross-domain pattern association. In Table VII, the behavior was analyzed by varying θ_K from 0.5 to 0.9 for the SSKM of the PenDigits part only (SSKM 2), while the fixed value $\theta_K = 0.7$ was used for the SFS part (SSKM 1), during the construction phase. Similarly, in Table VIII, the behavior was analyzed by varying the θ_K for the SFS part (SSKM 1), whereas the same fixed value $\theta_K = 0.7$ for the PenDigits part (SSKM 2) was used. The upper parts of Tables VII and VIII show the numbers of occurrences where the testing patterns in one domain (i.e., SFS/PenDigits) induced the excitation of KUs in other domain via the cross-domain LWs (i.e., PenDigits/SFS) and where they resulted in correct/incorrect association. In contrast, the lower part in each table shows the total number of i) KUs, ii) LWs generated within a single SSKM network, and iii) associative LWs so generated during the construction phase.

[9] Not to mention, such a network configuration is not possible by the SVMs or RAN, since, as aforementioned, the resulting network structures are equivalent to ordinary fully connected RBF-NNs and cannot be used for performing such pattern association tasks as in this paper.

TABLE VII
SUMMARY OF THE RESULTS FOR THE CROSS-DOMAIN PATTERN ASSOCIATION TASKS—WITH FIXING THE PARAMETERS FOR THE SFS DATA SET

θ_K (for	Num. Correct/Incorrect Pattern Association Occurred	
PenDigits)	SFS → PenDigits	PenDigits → SFS
$\theta_K = 0.5$	27/0	255/7
$\theta_K = 0.6$	27/0	224/2
$\theta_K = 0.7$	27/0	167/1
$\theta_K = 0.8$	27/0	113/0
$\theta_K = 0.9$	18/0	38/0

θ_K (for	SSKM for the PenDigits Part		
PenDigits)	Num. KUs	Num. LWs	Num. Assoc. LWs
$\theta_K = 0.5$	147	173	1112
$\theta_K = 0.6$	202	153	941
$\theta_K = 0.7$	271	103	698
$\theta_K = 0.8$	368	42	551
$\theta_K = 0.9$	488	6	395

TABLE VIII
SUMMARY OF THE RESULTS FOR THE CROSS-DOMAIN PATTERN ASSOCIATION TASKS—WITH FIXING THE PARAMETERS FOR THE PENDIGITS DATA SET

θ_K (for	Num. Correct/Incorrect Pattern Association Occurred	
SFS)	SFS → PenDigits	PenDigits → SFS
$\theta_K = 0.5$	264/11	114/0
$\theta_K = 0.6$	135/2	109/0
$\theta_K = 0.7$	27/0	168/1
$\theta_K = 0.8$	1/0	113/0
$\theta_K = 0.9$	0/0	113/0

θ_K (for	SSKM for the SFS Part		
SFS)	Num. KUs	Num. LWs	Num. Assoc. LWs
$\theta_K = 0.5$	256	386	1035
$\theta_K = 0.6$	418	49	714
$\theta_K = 0.7$	519	5	698
$\theta_K = 0.8$	539	0	691
$\theta_K = 0.9$	540	0	691

As shown in Tables VII and VIII, it is observed that the number of patterns in one domain which induced the activations of the KUs in the other domain was greatly varied by the factor θ_K (i.e., from no occurrence to 275, for a total of 360 testing patterns). In addition, while the number of the LWs remained relatively small, the number of the associative LWs so generated was consistently large for all the cases. This is naturally considered, due to the manner in presenting the input (training) patterns across the two domains (as aforementioned), and, thereby, the lateral connections between SSKM 1 and 2 became much denser than the regular connections (i.e., within each SSKM network). However, despite the large number of the cross-domain connections (i.e., the associative LWs), the number of activations from the KUs in SSKM 2 via the associative LWs (i.e., SFS→PenDigits) was relatively consistently smaller than that in SSKM 1 (i.e., PenDigits→SFS), whereas the number of incorrect pattern associations was relatively larger as θ_K becomes small. This may also indicate that, as observed in the simulation results of the regular pattern classification tasks in Section IV-E, the coverage of the pattern space for SFS is insufficient (i.e., due to the insufficient number of training patterns used) to induce the simultaneous activations for the correct associations, in comparison with that for PenDigits.

V. CONCLUSION

In this paper, a novel kernel-based constructive approach using distinct SSKM networks for simultaneous pattern classification and multidomain association tasks has been proposed. In the simulation study, the following has been confirmed.

1) Since the construction of classifiers is based upon the application of the Hebbian-motivated simple one-pass incremental training scheme to each SSKM network, it requires much less computational complexity than that of SVMs and RAN (as justified in Section IV-C).
2) Unlike SVMs, generation of redundant KUs can be suppressed by adjusting the factor θ_K, while maintaining a reasonable classification rate.
3) As the construction is modular-based (i.e., distinct SSKM networks are constructed one by one, as in Section II-B), it is hence considered that not only ordinary online incremental training but also accommodation of new classes as in PNN [25] is naturally performed.
4) The pattern classifiers based upon distinct SSKM networks can be straightforwardly extended to those capable of processing multidomain data simultaneously, i.e., performing multidomain pattern association tasks, within a single framework of exploiting the concept of LWs, which was not generally considered within the traditional ANN context.

The four aforementioned properties of SSKM are then considered to be the keys for exploring various new as well as versatile application domains. One such application would be to develop a novel connectionist model for simulating the faculty of human language acquisition or other cognitive functionalities that can evolve itself by experiences and simultaneously perform pattern classification/association of multidomain sensory data, which is currently under investigation by the authors.

ACKNOWLEDGMENT

The authors would like to thank all the reviewers and the Associate Editor for their insightful comments that improved the quality of this paper.

REFERENCES

[1] F. Rosenblatt, *Principles of Neurodynamics*. Washington, DC: Spartan, 1962.

[2] B. Widrow, “Generalization and information storage in networks of adaline ‘neurons’,” in *Self-Organizing Systems*, M. C. Yovitz, G. T. Jacobi, and G. D. Goldstein, Eds.. Washington, DC: Spartan, 1962, pp. 435–461.

[3] C. M. Bishop, *Neural Networks for Pattern Recognition*. London, U.K.: Oxford Univ. Press, 1996.

[4] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York: Wiley, 2001.

[5] S. Amari, "Theory of adaptive pattern classifiers," *IEEE Trans. Electron. Comput.*, vol. EC-16, pp. 299–307, 1967.

[6] P. J. Werbos, "Beyond regression: New tools for prediction and analysis in the behavioral sciences," Ph.D dissertation, Appl. Math. Dept., Harvard Univ., Cambridge, MA, 1974.

[7] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.

[8] D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Syst.*, vol. 2, pp. 321–355, 1988.

[9] J. E. Moody and C. J. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Comput.*, vol. 1, pp. 281–294, 1989.

[10] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proc. IEEE*, vol. 78, no. 9, pp. 1481–1497, Sep. 1990.

[11] S. Haykin, *Neural Networks: A Comprehensive Foundation*. New York: Macmillan, 1994.

[12] T. Kohonen, *Self-Organizing Maps*, 2nd ed. Berlin, Germany: Springer-Verlag, 1997.

[13] T. Voegtlin, "Recursive self-organizing maps," *Neural Netw.*, vol. 15, pp. 979–991, 2002.

[14] C. H. Chang, P. Xu, R. Xiao, and T. Srikanthan, "New adaptive color quantization method based on self-organizing maps," *IEEE Trans. Neural Netw.*, vol. 16, no. 1, pp. 237–249, Jan. 2005.

[15] D. J. Hand, *Kernel Discriminant Analysis*. Chichester, U.K.: Wiley, 1984.

[16] D. F. Specht, "Probabilistic neural networks for classification mapping, or associative memory," in *Proc. Int. Conf. Neural Netw.*, 1988, vol. 1, pp. 525–532.

[17] ——, "Probabilistic neural networks," *Neural Netw.*, vol. 3, pp. 109–118, 1990.

[18] E. A. Nadaraya, "On estimating regression," *Theory Probab. Appl.*, vol. 10, pp. 186–190, 1964.

[19] G. S. Watson, "Smooth regression analysis," *Sankhy*, ser. A, vol. 26, pp. 359–372, 1964.

[20] D. F. Specht, "A generalized regression neural network," *IEEE Trans. Neural Netw.*, vol. 2, no. 6, pp. 568–576, Nov. 1991.

[21] P. D. Wasserman, "Advanced methods in neural computing," in *Radial Basis-Function Networks*. New York: Van Nostrand, 1993, ch. 8, pp. 147–176.

[22] T. Poggio and S. Edelman, "A network that learns to recognize three-dimensional objects," *Nature*, vol. 343, no. 18, pp. 263–266, 1990.

[23] T. Hoya and J. A. Chambers, "Heuristic pattern correction scheme using adaptively trained generalized regression neural networks," *IEEE Trans. Neural Netw.*, vol. 12, no. 1, pp. 91–100, Jan. 2001.

[24] T. Hoya, "Notions of intuition and attention modeled by a hierarchically arranged generalized regression neural network," *IEEE Trans. Syst., Man, Cybern. B: Cybern.*, vol. 34, no. 1, pp. 200–209, Feb. 2004.

[25] ——, "On the capability of accommodating new classes within probabilistic neural networks," *IEEE Trans. Neural Netw.*, vol. 14, no. 2, pp. 450–453, Mar. 2003.

[26] J. Platt, "A resource-allocating network for function interpolation," *Neural Comput.*, vol. 3, no. 2, pp. 213–225, 1991.

[27] B. Fritzke, "Supervised learning with growing cell structures," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 1994, vol. 6, pp. 255–262.

[28] J. C. Platt and N. P. Matic, "A constructive RBF network for writer adaptation," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 1997, vol. 9, pp. 765–771.

[29] S. Schaal and C. G. Atkeson, "Constructive incremental learning from only local information," *Neural Comput.*, vol. 10, no. 8, pp. 2047–2084, 1998.

[30] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1996.

[31] M. A. Hearst, "Trends and controversies: support vector machines," *IEEE Intell. Syst.*, vol. 13, no. 4, pp. 18–28, Jul. 1998.

[32] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. New York: Springer-Verlag, 2001.

[33] E. J. Bredensteiner and K. P. Bennett, "Multicategory classification by support vector machines," *Comput. Optim. Appl.*, vol. 12, pp. 53–79, 1999.

[34] J. Weston and C. Watkins, "Support vector machines for multi-class pattern recognition," in *Proc. 7th Eur. Symp. Artif. Neural Netw.*, 1999, pp. 219–224.

[35] R. M. Nosofsky, "Attention, similarity and the identification-categorization relationship," *J. Exp. Psychol. Gen.*, vol. 115, pp. 39–57, 1986.

[36] J. K. Kruschke, "ALCOVE: An exemplar-based connectionist model of category learning," *Psychol. Rev.*, vol. 99, no. 1, pp. 22–44, 1992.

[37] V. B. Mountcastle, "Modality and topographic properties of single neurons of cat's somatic sensory cortex," *J. Neurophysiol.*, vol. 20, pp. 408–434, 1957.

[38] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurons in the cat's striate cortex," *J. Physiol.*, vol. 148, pp. 574–591, 1959.

[39] T. Hoya, "Modeling the notions of intuition and consciousness by hierarchically arranged generalized regression neural networks," in *Proc. Int. Symp. Nonlinear Theory Appl. (NOLTA)*, Zao, Japan, Oct. 2001, pp. 403–406.

[40] ——, "Self-organising associative kernel memory for multi-domain pattern classification," in *Proc. IFAC Workshop Adapt. Learn. Control Signal Process. (ALCOSP)*, Yokohama, Japan, Aug. 2004, pp. 735–740.

[41] ——, *Artificial Mind System—Kernel Memory Approach. Series: Studies in Computational Intelligence (SCI)*. Heidelberg, Germany: Springer-Verlag, Aug. 2005, vol. 1.

[42] D. O. Hebb, *The Organization of Behavior*. New York: Wiley, 1949.

[43] G. J. Wolff, D. G. Stork, K. V. Prasad, and M. Hennecke, "Lipreading by neural networks: Visual preprocessing, learning, and sensory integration," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 1993, vol. 5, pp. 1027–1034.

[44] S. Gutta, J. Huang, I. F. Imam, and H. Wechsler, "Face and hand gesture recognition using hybrid classifiers," in *Proc. Int. Conf. Autom. Face Recognit.*, Killington, VT, 1996, pp. 164–169.

[45] U. Meier, W. Hurst, and P. Duchnowski, "Adaptive bimodal sensor fusion for automatic speechreading," in *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 1996, vol. 2, pp. 833–836.

[46] V. Colla, M. Sgarbi, L. M. Reyneri, and A. M. Sabatini, "A neural approach to a sensor fusion problem," in *Proc. Euro. Symp. Artif. Neural Netw. (ESANN)*, Bruges, Belgium, 1998, pp. 357–362.

[47] E. Alpaydin and C. Kaynak, "UCI machine learning," Univ. California Irvine, Irvine, CA., 1998 [Online]. Available: www.ics.uci.edu/~mlearn

[48] E. Alpaydin and F. Alimoglu, "UCI machine learning," Univ. California Irvine, Irvine, CA., 1998 [Online]. Available: www.ics.uci.edu/~mlearn

[49] M. Huckvale, *Speech Filing System Vs3.0—Computer Tools For Speech Research*. London, U.K.: Univ. College, 1996.

[50] S. Furui, "Cepstral analysis technique for automatic speaker verification," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-29, no. 2, pp. 254–272, Apr. 1981.

[51] J. R. Deller, Jr., J. G. Proakis, and J. H. L. Hansen, *Discrete-Time Processing of Speech Signals*. New York: Macmillan, 1993.

[52] M. Stone, "Cross-validatory choice and assessment of statistical predictors," *J. Roy. Statist. Soc.*, vol. B36, pp. 111–147, 1974.

Tetsuya Hoya (M'01–SM'06) was born in Tokyo, Japan, on September 15, 1969. He received the B.Sc. and M.Sc. degrees in electrical engineering from Meiji University, Japan, in 1992 and 1994, respectively, and the Ph.D. degree from Imperial College of Science, Technology and Medicine, University of London, London, U.K., in 1998.

From April 1994 to September 1994, he was a Research Assistant at the Department of Electronics and Communication, Graduate School, Meiji University. He was then a student at the Department of Electrical and Electronics Engineering, Imperial College of Science, Technology and Medicine, from October 1994 to December 1997. He was a Postdoctoral Research Associate at the Department of Electrical and Electronics Engineering, Imperial College, London, from September 1997 to August 2000. From October 2000 to March 2006, he was a Research Scientist at the Brain Science Institute, RIKEN (The Institute of Physical and Chemical Research), Japan. From April 2003 to March 2007, he was a Visiting Lecturer at Saitama Institute of Technology, Japan. Since April 2007, he has been an Associate Professor at the

Department of Mathematics, College of Science and Technology, Nihon University, Tokyo, Japan. He has published more than 40 technical papers and is the single author of the monograph *Artificial Mind System—Kernel Memory Approach*, which began the new series: *Studies in Computational Intelligence (SCI)* (Springer-Verlag: 2005). His research interest is in a wide spectrum of computational intelligence: artificial intelligence, cognitive neuroscience, combinatoric optimization, computational linguistics, consciousness studies, electroencephalography (EEG), neural networks (connectionism), philosophy, psychology, robotics, and signal processing.

Dr. Hoya was a committee member of the Fourth International Symposium on Independent Component Analysis and Blind Signal Separation (ICA-2003). He is the corecipient of the Best Paper Award at the IEEE International Conference on Very Large Scale Integration Design and Video Technology, Shanhai, China, in May 2005.

Yoshikazu Washizawa (M'06) received the B.E. degree in electrical and computer engineering from the Nagoya Institute of Technology, Nagoya, Japan, in 2002 and the M.E. degree in electrical and electronic engineering from the Tokyo Institute of Technology, Tokyo, Japan, in 2004.

From 2004 to 2005, he was with the Toshiba Corporation. In 2005, he was with the Tokyo Institute of Technology as a Japan Society for the Promotion of Science (JSPS) Research Fellow. Since 2005, he has been a Research Associate at the Laboratory for Advanced Brain Signal Processing, Brain Science Institute, RIKEN, Saitama, Japan. His research interests include pattern recognition, machine learning, signal processing, blind signal separation, and time-frequency representation.

Mr. Washizawa is a member of the Institute of Electronics, Information and Communication Engineers (IEICE) of Japan.