Learning Active Fusion of Multiple Experts' Decisions: An Attention-Based Approach

Journal Article

Author(s):
Mirian, Maryam S.; Ahmadabadi, Majid N.; Araabi, Babak N.; Siegwart, Roland

Publication date:
2011-02

Permanent link:
https://doi.org/10.3929/ethz-a-010000225

Rights / license:
In Copyright - Non-Commercial Use Permitted

Originally published in:
Neural Computation 23(2), https://doi.org/10.1162/NECO_a_00079

Learning Active Fusion of Multiple Experts′ Decisions: An Attention-Based Approach

Maryam S. Mirian
mmirian@ut.ac.ir
Control and Intelligent Processing Center of Excellence, Department of Electrical and Computer Eng., University of Tehran, Tehran, Iran

Majid Nili Ahmadabadi
mnili@ut.ac.ir
Babak N. Araabi
araabi@ut.ac.ir
Control and Intelligent Processing Center of Excellence, University of Tehran, Tehran, Iran, and School of Cognitive Sciences, Institute for Research in Fundamental Sciences, Tehran, Iran

Roland R. Siegwart
rsiegwart@ethz.ch
ASL, ETHZ, Switzerland, Autonomous Systems Lab, Institute of Robotics and Intelligent Systems, ETH Zurich, 8092 Zurich, Switzerland

In this letter, we propose a learning system, active decision fusion learning (ADFL), for active fusion of decisions. Each decision maker, referred to as a local decision maker, provides its suggestion in the form of a probability distribution over all possible decisions. The goal of the system is to learn the active sequential selection of the local decision makers in order to consult with and thus learn the final decision based on the consultations. These two learning tasks are formulated as learning a single sequential decision-making problem in the form of a Markov decision process (MDP), and a continuous reinforcement learning method is employed to solve it. The states of this MDP are decisions of the attended local decision makers, and the actions are either attending to a local decision maker or declaring final decisions. The learning system is punished for each consultation and wrong final decision and rewarded for correct final decisions. This results in minimizing the consultation and decision-making costs through learning a sequential consultation policy where the most informative local decision makers are consulted and the least informative, misleading, and redundant ones are left unattended. An important property of this policy is that it acts locally. This means that the system handles any nonuniformity in the local decision maker′s

expertise over the state space. This property has been exploited in the design of local experts. ADFL is tested on a set of classification tasks, where it outperforms two well-known classification methods, Adaboost and bagging, as well as three benchmark fusion algorithms: OWA, Borda count, and majority voting. In addition, the effect of local experts design strategy on the performance of ADFL is studied, and some guidelines for the design of local experts are provided. Moreover, evaluating ADFL in some special cases proves that it is able to derive the maximum benefit from the informative local decision makers and to minimize attending to redundant ones.

1 Introduction

Decision fusion—asking a set of experts' response to a query and making decisions accordingly—has been a hot research topic. Decision fusion is a challenging problem, as each expert's knowledge and expertise are in general incomplete and nonuniform over the problem domain. Not only is each expert's suggestion imperfect over the problem domain, it also can be misleading in response to a number of queries—those that are not posed in an expert's area of expertise. To reflect these facts, we refer to an expert or a decision maker as a local decision expert (LDE). Consultation with LDEs is not cost free. By "cost," we mean whatever resources should be allocated for getting the LDEs' suggestions. For example, in a medical diagnostic problem, consultation with another physician may require some additional medical tests as different physicians may look at a single problem from different perspectives. This will at least cost the patient money and time, disregarding the test's side effects, which are important in most cases. Therefore, in this letter, we propose a learning system attentive decision fusion learning (ADFL), that learns both whom to consult with and the final decision based on the consultations.

In the proposed method, it is assumed that suggestions, which we also refer to as outputs or decisions, of each LDE is a probability distribution over all of the possible final decisions. This natural assumption is not restrictive, because it necessitates homogeneity in neither the structure nor the inputs of the experts. Therefore, LDEs can have different structures and inputs.

The decision fusion problem is cast in an episodic sequential decision-making framework, and the costs of consultation and decision making are modeled in the form of a reward function. Then a continuous reinforcement learning (RL) method is employed to learn the optimal solution of the problem. The learning results in a single policy for sequential consultation with LDEs and making the final decision accordingly. The policy is employed to select either the next LDE to be consulted or a final decision to be made based on the consultations already made in the current episode.

One of the major properties of learned policy is being nonuniform over the problem domain. It means that ADFL acts locally, not globally as in common fusion methods, through learning to attend to the most informative LDEs indirectly for each portion of the problem domain. Another important attribute of ADFL is to learn the best final decision based on the consulted LDEs′ suggestions. These two properties resolve the challenges of decision fusion: knowledge incompleteness and nonuniformity of the LDEs′ expertise over the problem domain and the cost reduction of consultation and decision making.

The rest of this letter is organized as follows. First, the related studies are reviewed. Then the problem is defined formally. In section 3, the proposed approach is explained, followed by an explanation on how this approach is applied to classification tasks. The case studies on the selected UCI data sets are mentioned and the results reported and analyzed. In section 6, the effect of LDEs′ properties on the performance of ADFL is studied, and some guidelines for the design of LDEs are provided. Moreover, some major properties of ADFL are verified through testing its performance in some special cases. The conclusions and the next steps of this work are discussed in the last section.

2 Related Studies

There are two classes of related studies and methods to the focus of this research. The first covers the studies associated with active and adaptive selection of inputs by a decision maker—sensors and features—to cut the costs of information processing and improve decision quality. These studies mainly offer top-down attention control methods, or what some researchers in the robotics domain refer to as active perception. The second category contains research on fusion of inputs. In this research, the inputs are suggestions of LDEs, and therefore we review the research on combining the decisions of multiple decision makers.

Most of the existing attention control mechanisms are hand-designed based on heuristics or biologically inspired algorithms. However, it is preferable for attention control and decision-making policy to be learned together at the same time since the optimal attention strategy is a function of the decision-making policy and vice versa. Nevertheless, there are very limited learning approaches to concurrent learning of attention and decision policy (for example, Paletta & Pinz, 2000; Borji, Ahmadabadi, Araabi, & Hamidi, 2010; Paletta, Fritz, & Seifert, 2005; Minut & Mahadevan, 2001). Attention strategy in these studies is learned in the sensory space. In the method introduced by Mirian, Firouzi, Ahmadabadi, and Araabi (2009), ADFL also works in the decision space, and the attention strategy is being learned in this alternative space. In fact, we adopt the idea of augmented action from Shariatpanahi and Nili Ahmadabadi (2008) to formulate the problems of attention and decision learning as a single decision-making problem in the decision space.

The main idea behind active and budgeted learning techniques (Danziger, Zeng, Wang, Brachmann, & Lathrop, 2007; Lizotte, Madani, & Greiner, 2003) is to choose the most informative data from a set of training samples to reduce the cost of learning and decision making. ADFL also tries to reduce the costs, but it does so through attentive sequential selection of consultations in the recall mode.

A variety of strategies for decision making are based on combining the decisions of multiple decision makers. Ensemble-based methods such as mixture of experts (Jordan & Jacobs, 1994), stacked generalization (Wolpert, 1992), Adaboost (Schapire, 2003), and Bagging (Polikar, 2007) are examples. In addition, there are plenty of standard decision fusion methods: classifier fusion (Zhu, 2003; Woods, Kegelmeyer, & Bowyer, 1997; Verikas, Lipnickas, Malmqvist, Bacauskiene, & Gelzinis, 1999), majority voting (Polikar, 2006), Borda count (Polikar, 2006) and OWA (Filev & Yager, 1998). Nevertheless, the main superiority of ADFL over these methods is its unsupervised learning of the attentive sequential selection of decision makers to consult with and formation of locally optimal decision policies over the decision space. This superiority is experimentally verified in this letter.

3 Proposed Approach

The idea proposed here is to learn how to sequentially fuse the individual decision makers' decision to minimize a specific cost function. In other words, the learner agent learns to sequentially select those more helpful LDEs in every state, combine their opinions, and learn to reach a final decision. Here, the cost function is a combination of the reward that the learner gets in return for its final decision minus the costs of asking LDEs about their decisions. We have named this the attentive decision fusion learning (ADFL) agent.

3.1 Problem Statement.

We assume that the ADFL agent has access to l LDEs: $e_1, e_2, e_3, \ldots, e_l$.[1] Each e_i looks at a segment of the entire input feature space ($\boldsymbol{F}_i$), an n-dimensional space represented by $\boldsymbol{F}$ ($\boldsymbol{F}_i \subseteq \boldsymbol{F}$). Here, $\boldsymbol{F}_i$s are overlapping subsets of the $\boldsymbol{F}$ space and $\boldsymbol{F} = \bigcup_{i=1}^{l} \boldsymbol{F}_i$. Moreover, all LDEs have the same set of decisions expressed by $D = \{d_1, d_2, \ldots, d_c\}$, where c is the total number of possible decisions. The decision (output) of the ith LDE when given $\boldsymbol{f}_i \in \boldsymbol{F}_i$ as input (by looking at point $\boldsymbol{f} \in \boldsymbol{F}$) is its degree of support for all possible decisions:

$$\begin{aligned}\boldsymbol{e}_i(\boldsymbol{f}_i) = \lfloor &Pr(d_{e_i} = d_1 | \boldsymbol{f}_i) Pr(d_{e_i} = d_2 | \boldsymbol{f}_i) Pr(d_{e_i} = d_3 | \boldsymbol{f}_i) \ldots \\ &Pr(d_{e_i} = d_c | \boldsymbol{f}_i) \rfloor. \end{aligned} \tag{3.1}$$

[1] All vectors are in bold characters. Sets are denoted by capital letters.

This can be simplified as

$$[Pr(d_k|\boldsymbol{f}_i)], \quad \boldsymbol{k} = 1, 2, 3, \ldots, c.$$

Note that the ADFL agent cannot decide based on the features and should make its decision based on consultation with LDEs. In other words, the ADFL agent is an attentive decision fuser. Therefore, the state ($\boldsymbol{s}$), action ($\boldsymbol{a}$), and decision policy (π) of the ADFL agent for every observation—or, equivalently, query—$\boldsymbol{f} \in \boldsymbol{F}$ are defined as

$$\begin{cases} \boldsymbol{s} = [s_1 s_2 \ldots s_l],\, s_i = \begin{cases} \boldsymbol{e}_i(\boldsymbol{f}_i), i \in \text{selected experts so far} \\ 0_{1\times c} \qquad \text{otherwise} \end{cases} \\ \qquad\qquad a \in A,\, A = T \cup D \\ \qquad\qquad \pi = Pr(a|\boldsymbol{s}), \end{cases} \tag{3.2}$$

where $T = \{t_1, t_2, \ldots, t_l\}$ and t_i is consultation with the ith LDE. The state of ADFL is composed of the decisions of consulted LDEs. This definition is similar to the description of decision profile in Kuncheva, Bezdek, and Duin (2001).

As Figure 1 and equation 3.2 show, an ADFL agent at each state decides between two sets of possible actions: consult another LDE (*Consultation actions* $\in T$) or declare a final decision (*Decision actions* $\in D$). In Figure 1, consultation with an LDE is modeled by closing a switch. The ADFL agent updates its state ($\boldsymbol{s}$) after every consultation. Sequential consultation with LDEs continues until the ADFL agent decides to stop consulting and makes a final decision. The ADFL agent's goal is to find an optimal policy (π^*) under which a specific utility function is maximized:

$$\pi^* = \text{argmax}_\pi\, Q^\pi(\boldsymbol{s}, a) \quad \forall \boldsymbol{s}. \tag{3.3}$$

The utility function Q (called Q-value as well) and the optimization method, which is done over every state, vary according to the information about the problem at hand. However, the utility function, in its general form, is the expected reinforcement that the ADFL agent receives for its decision in state $\boldsymbol{s}$. In other words, by maximizing its expected reward, the ADFL agent learns to reach a reasonable trade-off between the quality of its final decision and the cost of consulting with LDEs. This is achieved by considering predefined costs for consulting each LDE, as well as the benefits of making a reasonable final decision. Learning the optimal policy is modeled in the next section.

3.2 Attentive Decision Fusion Learning: Formulation. Since in general we do not know the best policy (the optimal sequence of consultations) and the exact area of expertise of the experts, we opt for learning the value of decisions and finding the optimal policy (π^*) at each state

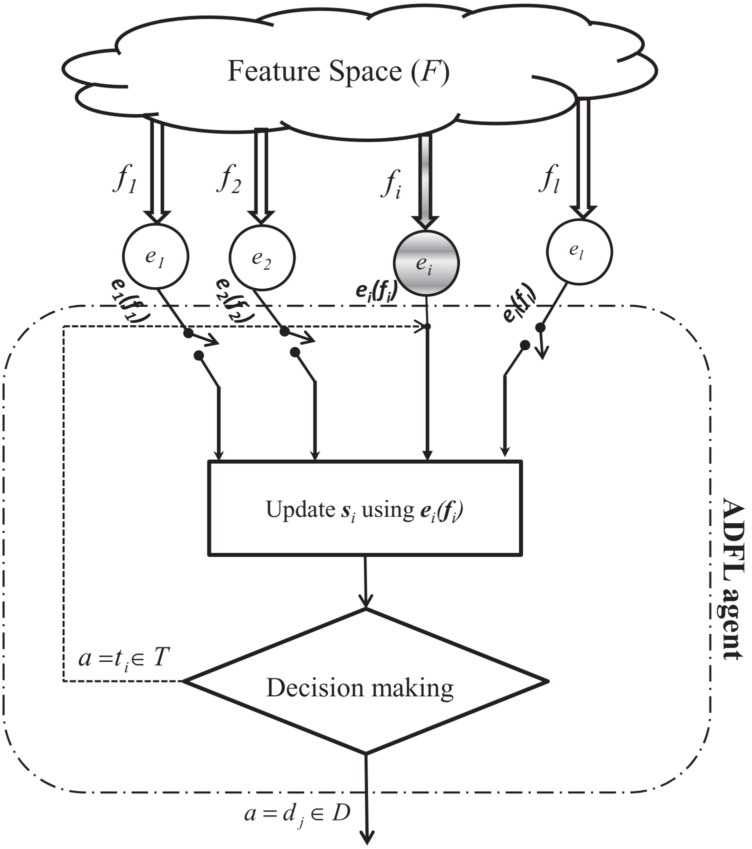

Figure 1: A cycle of decision making and state updating in ADFL.

(see equations 3.2 and 3.3). Thus, we have defined a flexible mechanism to set rewards and punishments for each decision. Reinforcement learning (RL) (Sutton & Barto, 1999) was chosen as the optimization method since it is one of the natural candidates for learning efficient sequential decision making. As we will show, the state of the ADFL problem—or of the ADFL agent—is continuous while the actions are discrete. This continuous space is called the decision space because it is composed of the LDEs' decisions (see the definition of state (***s***) in equation 3.2). There are different variants of RL methods for handling the continuous state: fuzzy Q-learning (Berenji, 1996), variants of the tile coding method (Whiteson, Taylor, & Stone, 2007), and the Bayesian Q-learning approach (Firouzi, Ahmadabadi, & Araabi, 2008). Among the possible variants, we have used Bayesian Q-learning approach (Firouzi et al., 2008; Firouzi et al., 2009) because of its uncertainty handling

Table 1: Key Elements of Assumed MDP to Formulate ADFL.

State (S)	$s_{init} = 0_{l \times c}$ = the initial state of ADFL agent before consulting any LDE (*Null*). $\boldsymbol{s} = [\boldsymbol{s}_1 \boldsymbol{s}_2 \ldots \boldsymbol{s}_l]$, $\boldsymbol{s}_i = \begin{cases} \boldsymbol{e}_i(\boldsymbol{f}_i), i \in \text{selected experts so far} \\ 0_{1\times c} \qquad \text{otherwise} \end{cases}$ $\boldsymbol{e}_i(\boldsymbol{f}_i) = [Pr(d_{e_i} = d_1 \mid \boldsymbol{f}_i) Pr(d_{e_i} = d_2 \mid \boldsymbol{f}_i) Pr(d_{e_i} = d_3 \mid \boldsymbol{f}_i) \ldots Pr(d_{e_i} = d_c \mid \boldsymbol{f}_i)]$ l = the number of LDEs $c = \lvert \boldsymbol{D} \rvert$ = the size of the decision actions
Actions (A)	$\boldsymbol{A} = \boldsymbol{T} \cup \boldsymbol{D}$

Transition function (*Tran*)	*if* $a = t_i \in T \Rightarrow \boldsymbol{s}_i \leftarrow \boldsymbol{e}_i(\boldsymbol{f}_i)$
	if $a = d_j \in D \Rightarrow \boldsymbol{s} \leftarrow$ Terminal state

Reward function (r)	r = High positive, *if* a = *Correct Decision* $\in D$ r = High negative, *if* a = *Wrong Decision* $\in D$ r = (Small negative) × (Number of already consulted experts), *if* $a \in T$

and flexibility in generating state prototypes. Nevertheless, our approach is theoretically general and independent of the employed learning core. Here, we first cast the ADFL problem in an episodic Markov decision process (MDP) form and then use the RL method to solve it.

The corresponding MDP is defined by a 4-tuple (*S, A, Tran, r*) in which S is the ADFL agent's set of states and A is the set of its actions, *Tran* is the state transition function, and r is the reward received. More details are given in Table 1.

As shown in Table 1, the state and action sets of the ADFL problem are exactly those of ADFL agent (see equation 3.2). The initial state is *Null*, which means that no consultation has been made. The transition function either concatenates the opinion of an LDE to the state (when the ADFL agent consults with that LDE) or transfers the state to the terminal state if a final decision is made. Since the ADFL agent attempts to maximize its expected reward, the rewards and punishments are defined such that the expected reward maximization results in making a correct final decision with the fewest possible consultations. It is done by setting a large reward or punishment for correct or wrong final decisions. In addition, the punishment for consultation—or, equivalently, the cost of consultation—is linearly increased with the number of already consulted LDEs. The slope of this is set to be small in order not to penalize the ADFL agent too much for consultations, which forces it to make premature final decisions. Because as the number of consultations can be easily extracted from the state, it does not violate the Markov property of the ADFL problem.

Similar to ordinary Q-learning (Watkins & Dayan, 1992), in ADFL the value of each state-action under policy $\pi^*(Q^{\pi^*}(\boldsymbol{s}, a))$ is learned while following a Q-value-based soft policy, like Boltzman or $\in$-greedy (Sutton & Barto, 1999). Since π^* is a greedy policy, the soft policy is shifted gradually

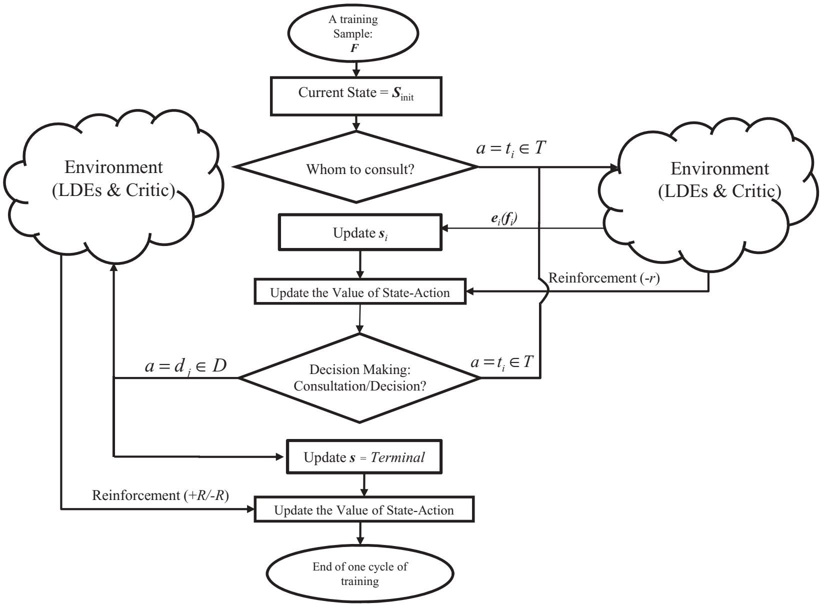
A training Sample: F
Current State = S_init
Whom to consult?
a = t_i ∈ T
Environment (LDEs & Critic)
Environment (LDEs & Critic)
e_i(f_i)
Update s_i
Reinforcement (-r)
Update the Value of State-Action
a = d_j ∈ D
Decision Making: Consultation/Decision?
a = t_i ∈ T
Update s = Terminal
Reinforcement (+R/-R)
Update the Value of State-Action
End of one cycle of training

Figure 2: A learning cycle of the ADFL agent.

toward the greedy one as learning the Q-values proceeds. Ultimately the Q-values are learned or converged in some learning cycles.

Each learning cycle starts with posing the ADFL agent a query. It is done by giving the agent a training sample (see Figure 2). Actually, the query is to ask the *ADFL* agent's opinion about a stimulus—a point in the feature space ($\boldsymbol{f} \in \boldsymbol{F}$). Since the agent cannot decide based on the features and should consult LDEs, it initializes its state to *Null* and decides who to consult first.[2] This decision is made according to a softmax policy on the possible consultation actions in the *Null* state: $a = \text{softmax}(Q(Null, t)\ \forall t \in T)$. After the consultation, the ADFL receives a punishment associated with the consultation and updates its state (see Table 1) along with its Q-value using the employed learning rule—Bayesian Q-learning here.

After the first consultation, the agent decides between two possible options using a softmax policy: either consulting an LDE or making a final decision: $a = \text{softmax}(Q(\boldsymbol{s}, b) \forall b \in T \cup D)$ based on the acquired information. If it decides to perform another consultation, it gets a punishment

[2] Each action (either consultation action or final action) has a Q-value, and this value shows how much that action is beneficial for the agent. The probability of selecting each action is proportional to its Q-value, that is, the probability of selecting more beneficial actions is higher.

again and updates its state as well as its corresponding Q-value. This process is repeated for every consultation. In case the ADFL agent makes a final decision, it receives the corresponding reward or punishment in return and updates its corresponding Q-value after setting its next state to the terminal state (see Figure 2 and Table 1). The learning cycle ends here.

The learning cycle is repeated over the training samples multiple times until a stop criterion is met. The stop criterion is met when the error on the evaluation samples is increased for a reasonable number of successive learning cycles. Thereafter, the ADFL agent is tested on the test data. The test cycle is the same as the learning cycle except no knowledge updating is involved.

4 Application in Classification Tasks

Up to this point, ADFL has been introduced in the most general form. In this section, we explain the realization of the proposed approach on a well-known type of decision-making task. A classification task is considered due to a set of reasons: various appropriate tasks for active decision fusion learning (like medical diagnosis) are kinds of classification; LDEs here can be simply replaced by local classifiers; and we can benchmark our approach in comparison with well-established classification methods in addition to well-known decision fusion approaches.

In order to evaluate ADFL, we need a test system composed of a set of readily existing LDEs on a benchmark problem. Because such a test system does not exist, we have selected some medical diagnoses data sets and a few others from the UCI data sets (Blake & Merz, 1998) and manually designed LDEs on them. The key design issue of LDEs is their locality attribute, which is aimed at generating diversity in LDEs′ areas of expertise. This can be achieved through making LDEs different in their input feature space or output decision space, or both. In this research, we have chosen generating diversity in the input space.[3]

There is a wide spectrum of different strategies (Ebrahimpour, Kabir, Esteky, & Yousefi, 2008) on localization of the input space: from fully random to fully hand-designed partitioning of the space. In this letter, we have selected a less random method balanced random subspace making (BRSM), and pre-knowledge subspace making (PKSM), which acts based on both preknowledge (prior knowledge available about the features) and heuristics, which we discussed later.

BRSM is illustrated in Figure 3a. In BRSM, by binning the features and selecting from the bins, we find feature-diverse but performance-wise nearly similar LDEs. PKSM (see Figure 3b), is based on two heuristics. The first one

[3] Utilizing the output space facilitates the development of hierarchical decision making; thus, it can be helpful in problems with a high-dimensional decision space.

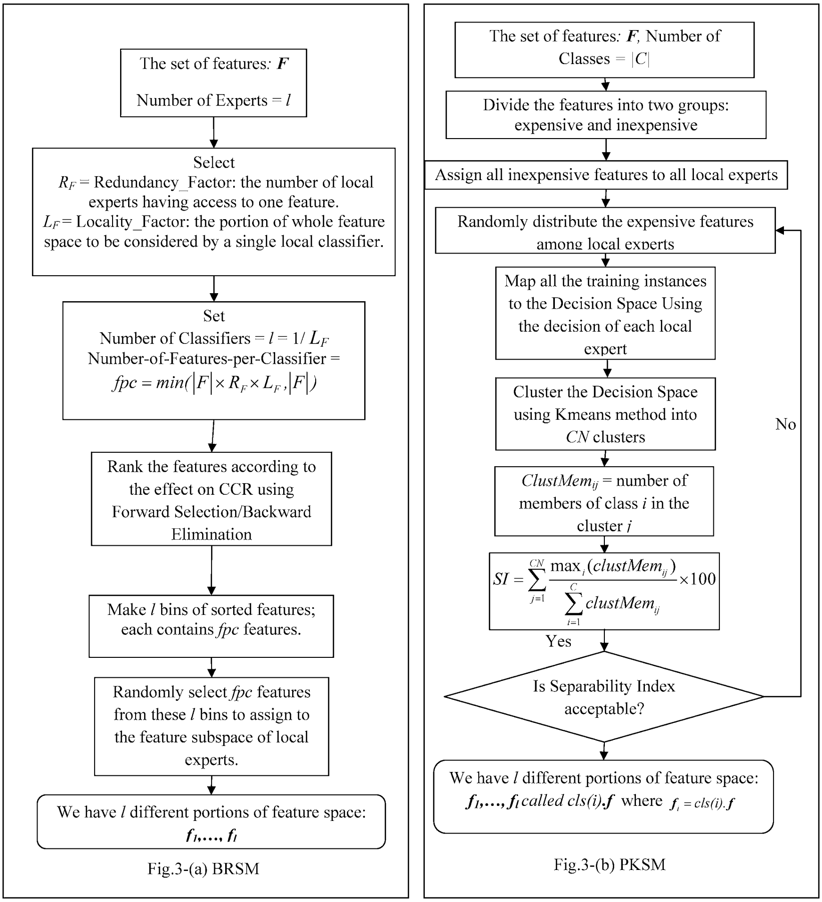
The set of features: F
Number of Experts = l
Select
R_F = Redundancy_Factor: the number of local experts having access to one feature.
L_F = Locality_Factor: the portion of whole feature space to be considered by a single local classifier.
Set
Number of Classifiers = l = 1/ L_F
Number-of-Features-per-Classifier =
$fpc = min(|F| \times R_F \times L_F, |F|)$
Rank the features according to the effect on CCR using Forward Selection/Backward Elimination
Make l bins of sorted features; each contains fpc features.
Randomly select fpc features from these l bins to assign to the feature subspace of local experts.
We have l different portions of feature space: $f_1, \dots, f_l$
Fig.3-(a) BRSM
The set of features: F, Number of Classes = |C|
Divide the features into two groups: expensive and inexpensive
Assign all inexpensive features to all local experts
Randomly distribute the expensive features among local experts
Map all the training instances to the Decision Space Using the decision of each local expert
Cluster the Decision Space using Kmeans method into CN clusters
$ClustMem_{ij}$ = number of members of class i in the cluster j
$SI = \sum_{j=1}^{CN} \frac{\max_i (clustMem_{ij})}{\sum_{i=1}^{C} clustMem_{ij}} \times 100$
Yes
No
Is Separability Index acceptable?
We have l different portions of feature space: $f_1, \dots, f_l$ called cls(i).f where $f_i = cls(i).f$
Fig.3-(b) PKSM

Figure 3: (a) BRSM. (b) PKSM.

is that a design with expensive features (like MRI data) given to a limited number of LDEs is preferred to a design that distributes all features among LDEs while disregarding the cost of generating those features. Providing all LDEs with not-very-expensive-to-observe features is recommended in the second heuristic. Therefore, PKSM is established for the manual design of LDEs such that 1) LDEs' decisions (see equation 3.1) have a reasonable separability index (it will be discussed in section 6.1) and all LDEs have access to low-cost features while just a limited number of them are provided with expensive features.

To realize the proposed approach for classification, three consecutive phases are defined. First, different feature subspaces are generated with BRSM or PKSM using the training data. Then one local classifier (LDE)

Table 2: The Selected Data Sets.

Data Sets	Number of Features	Number of Output Classes	Number of Instances
Heart (statlog)	13	2	270
Hepatitis	19	2	155
Liver Disorder (bupa)	6	2	345
Pima Indian Diabetes	8	2	768
Ionosphere	34	2	351
Sonar	60	2	208
Glass	9	6	214
Vehicle	18	4	846
Waveform	40	3	500
Satimage	36	6	6435
Dermatology	34	6	366

is assigned to each subspace and trained. The classification method for LDEs (such as k-NN, naive Bayes, support vector machines (SVM), and MLP artificial neural networks) is chosen considering the properties of data set (see Kotsiantis, Zaharakis, & Pintelas, 2006). Here, we have opted for three general methods—k-NN, naive Bayes, and SVM—and compared the results. In the last phase, the ADFL agent learns (through employed Bayesian Q-learning) to maximize the expected reward through sequential selection of the most appropriate LDEs for consultation and declaring the classification decision.

5 Case Studies: Evaluating the Approach on the Selected UCI Data Sets

We have benchmarked the performance of an implementation of our method over 11 sample data sets from UCI machine learning repository (see Table 2) against several well-known approaches that have some similarities to our approach from different aspects.

Approaches selected for benchmarking are (1) a holistic k-NN in the feature space; (2) Bagging (Polikar, 2007), (3) Adaboost (Schapire, 2003); (4) a holistic k-NN in the decision space;[4] and (5) fusion-based methods in three decision levels: only labels (majority voting), rank of the labels (Borda count), and continuous outputs like a posteriori probabilities OWA with gradient descent learning of the optimal weights; Filev & Yager, 1998). The first three methods work in the feature space, while the rest, along with ADFL, work in the decision space.

The results of Bagging with k-NN base learners and those of Adaboost with k-NN and SVM base learners are adopted from Garcia-Pedrajas (2009)

[4] This method works based on the pool of original training instances that are represented in the decision space of designed local experts plus their corresponding class labels.

and Garcia-Pedrajas and Ortiz-Boyer (2009); the rest of methods are our experiments. In Garcia-Pedrajas (2009) and Garcia-Pedrajas and Ortiz-Boyer (2009), 50% of each data set is used for training and the rest for the test. We followed the same data partitioning policy. Nevertheless, for ADFL, the situation is harder since 8% of the training data is used for validation to find the appropriate learning stop point.

Each experiment is performed over five replicates of randomly generated training and test data, and the results are averaged. The reported results are the average correct classification rates (CCR) on the test data along with the statistical variances. For ADFL, CCR, and statistical variances on the training data, along with the consultation ratio on the test, data are reported as well. Consultation ratio is defined as

$$\text{Consultation ratio} = \frac{\text{Number of consulted LDEs}}{\text{Total number of LDEs}}. \tag{5.1}$$

To have a fair comparison with the ensemble-based methods reported in Garcia-Pedrajas (2009) and Garcia-Pedrajas and Ortiz-Boyer (2009), we have also used SVM classifiers with gaussian kernels for LDEs. SVM learning algorithm in multiclass problems is performed using functions from the LIBSVM library (Chih-Jen Lin, n.d.). The same parameters for SVM are also used: C (bound on the Lagrangian multipliers) is 10 and lambda (conditioning parameter for the QP method) is 0.1. To employ Bayesian Q-learning, some initial parameter settings are made: the learning rate is decreased from 0.4 to 0.1 in an exponential manner. The discount factor is 1 (because the last action of every episode has the highest importance) and the temperature used for softmax action selection is exponentially changed from 0.8 to 0.01 (in order to gradually move toward full greedy mode). The cost of consultation is −1, the reward of correct decision making is +10, and the punishment associated with wrong decision making is −10. Table 3 shows the results of ADFL+BRSM (ADFL with BRSM-based LDEs) in addition to those of the benchmark methods. For every data set, the best result gained in each class of methods (ensemble-based and decision-space methods) is shown in boldface type. The same is done for the best result of ADFL over the three types of LDEs; k-NN, NB, and SVM. The name of a data set is marked with ☒ ☑ when the best result of ADFL is significantly lower (higher) than that of its strongest competitor. An empty box is used when there is no significant difference between ADFL's best and the strongest benchmark method's performances. Table 3 demonstrates that ADFL+BRSM defeated all of its fusion-based competitors in all of the data sets but not the ensemble-based methods. This issue is analyzed in section 6.1. Nevertheless, as the consultation ratio indicates, the ADFL agent after training genuinely consults with more knowledgeable LDEs in every state and LDEs that are recognized as unhelpful are left unattended. In this, ADFL is superior to the other methods, which need to consider the decisions of all LDEs together or use a single

Table 3: Comparing ADFL+BRSM and the Benchmark Methods.

		Feature Space Methods				Decision Space Methods						
			Ensemble									
			AdaBoost				Classification	Fusion Methods			ADFL+BRSM	
Dataset	Design Desc.	Classification k-NN	k-NN (Garcia-Pedrajas & Ortiz-Boyer, 2009)	SVM (Garcia-Pedrajas, 2009)	Bagging (k-NN) (Garcia-Pedrajas, 2009)	Base Learner	KNN	Majority Voting	Borda Count	OWA	Test CCR	Consultation Ratio
Heart T	$R_F = 1\ L_F = 0.2$	75.30	74	80.22		k-NN	75.53	66.4	62.6	66.6	74.51	3.9/5
☒	$l = 5\ fpc = 2$	(0.013)					(0.008)	(0.01)	(0.09)	(0.02)	(0.001)	
					82.89	NB	73.11	66.1	61.7	64.1	**75.90**	3.4/5
							(0.002)	(0.016)	(0.019)	(0.05)	(0.016)	
						SVM	75.55	**67.40**	59.4	64.4	74.63	2.1/5
							(0.003)	(0.06)	(0.01)	(0.08)	(0.0006)	
Hepatitis	$R_F = 2\ L_F = 0.2$	79.8	74	**76.79**		k-NN	78.6	71.7	58.4	59.0	80.14	4/5
□	$l = 5\ fpc = 9$	(0.02)					(0.05)	(0.04)	(0.03)	(0.01)	(0.001)	
					82.18	NB	77.6	70.7	57.1	58.1	78.2	3.7/5
							(0.05)	(0.01)	(0.02)	(0.03)	(0.004)	
						SVM	79.52	**76.10**	68.47	63.52	**82.23**	2.6/5
							(0.001)	(0.01)	(0.003)	(0.02)	(0.006)	
Liver Disorder	$R_F = 2\ L_F = 0.2$	65.40	57.5			k-NN	63.5	56.1	57.6	65.1	66.9	3.3/5
(Bupa) R	$l = 5\ fpc = 3$	(0.09)					(0.001)	(0.04)	(0.02)	(0.08)	(0.007)	
☑				**67.77**	60.93	NB	59.9	57.67	55.4	62.54	64.20	3.1/5
							(0.005)	(0.02)	(0.09)	(0.015)	(0.007)	
						SVM	62.85	62.85	60.1	**68.9**	**71.42**	2.7/5
							(0.003)	(0.03)	(0.02)	(0.03)	(0.001)	

Table 3: *(Continued)*

		Feature Space Methods				Decision Space Methods						
			Ensemble									
			AdaBoost				Classification	Fusion Methods			ADFL+BRSM	
Dataset	Design Desc.	Classification k-NN	k-NN (Garcia-Pedrajas & Ortiz-Boyer, 2009)	SVM (Garcia-Pedrajas, 2009)	Bagging (k-NN) (Garcia-Pedrajas, 2009)	Base Learner	KNN	Majority Voting	Borda Count	OWA	Test CCR	Consultation Ratio
Pima Indian Diabetes ☒	$L_F = 0.25\ R_F = 2$ $l = 4\ fpc = 4$	74.17 (0.08)	68		74.27	k-NN	70.1 (0.007)	66.0 (0.02)	68.1 (0.05)	62.7 (0.03)	75.36 (0.001)	2.7/4
				76.20		NB	72.6 (0.003)	69.1 (0.04)	**69.2** (0.01)	60.2 (0.01)	**75.92** (0.05)	2.5/4
						SVM	70.1 (0.013)	63.1 (0.05)	66.4 (0.03)	59.9 (0.07)	72.14 (0.001)	2.1/4
Ionosphere ☑	$L_F = 0.25\ RF = 2$ $l = 4\ fpc = 17$	81.92 (0.015)	84	83.71		k-NN	81.6 (0.007)	**79.3** (0.02)	73.1 (0.01)	71.4 (0.04)	84.05 (0.008)	2.9/4
					86.95	NB	80.2 (0.008)	78.1 (0.03)	74.1 (0.06)	70.6 (0.08)	82.37 (0.06)	2.2/4
						SVM	76.11 (0.004)	66.11 (0)	69.20 (0)	63.88 (0.01)	**88.90** (0.001)	2.6/4
Sonar ☑	$RF = 2\ LF = 0.25$ $l = 4\ fpc = 30$	73.60 (0.013)	78	73.85		k-NN	77.1 (0.04)	**68.2** (0.04)	59.3 (0.09)	66.3 (0.01)	**82.42** (0.08)	2.5/4
					80.19	NB	75.6 (0.003)	61.3 (0.02)	57.8 (0.03)	64.1 (0.04)	80.53 (0.004)	2.3/4
						SVM	65.7 (0.001)	65.45 (0)	64.2 (0.03)	60.30 (0.01)	72.70 (0.004)	2.1/4
Glass □	$RF = 10\ LF = 0.2$ $l = 5\ fpc = 9$	64.21 (0.03)				k-NN	59.4 (0.007)	52.3 (0.02)	55.4 (0.04)	59.0 (0.01)	64.54 (0.09)	3.5/5
			64.5	60.47	63.93	NB	**60.8** (0.03)	53.8 (0.01)	56.9 (0.03)	**59.3** (0.05)	**65.32** (0.002)	2.7/5
						SVM	56.3 (0.007)	54.5 (0.01)	51.3 (0.002)	52.0 (0.04)	64.06 (0.003)	2.6/5

Table 3: *(Continued)*

		Feature Space Methods				Decision Space Methods						
			Ensemble									
			AdaBoost				Classification	Fusion Methods			ADFL+BRSM	
Dataset	Design Desc.	Classification k-NN	k-NN (Garcia-Pedrajas & Ortiz-Boyer, 2009)	SVM (Garcia-Pedrajas, 2009)	Bagging (k-NN) (Garcia-Pedrajas, 2009)	Base Learner	KNN	Majority Voting	Borda Count	OWA	Test CCR	Consultation Ratio
Vehicle ☒	$RF = 2$ $LF = 0.2$ $l = 5$ $fpc = 9$	68.44 (0.02)	67.5			k-NN	67.8 (0.001)	61.40 (0.01)	**62.24** (0.02)	62.10 (0.003)	**75.20** (0.007)	3.6/5
				77.19	68.68	NB	64.53 (0.005)	59.30 (0.003)	60.12 (0.01)	60.50 (0.04)	73.81 (0.0002)	3/5
						SVM	61.62 (0.0002)	55.58 (0.01)	59.65 (0.005)	59.25 (0.01)	65.11 (0.006)	2.3/4
Waveform ☒	$RF = 2$ $LF = 0.2$ $l = 5$ $fpc = 10$	81.46 (0.05)	68		83.61	k-NN	78.37 (0.0007)	73.52 (0)	54.90 (0.007)	69.60 (0.017)	76.47 (0.001)	3.1/5
				85.83		NB	80.40 (0.0004)	75.00 (0.004)	50.98 (0)	**77.45** (0.07)	**82.84** (0.003)	3.1/5
						SVM	74.07 (0.005)	70.98 (0.001)	71.51 (0.001)	64.70 (0.064)	78.92 (0.0004)	2.6/5
Satimage ☑	$RF = 2$ $LF = 0.2$ $l = 5$ $fpc = 18$	85.55 (0.07)	87.5	86.29		k-NN	91.04 (0.002)	70.50 (0.004)	71.35 (0.05)	**71.42** (0.01)	**93.23** (0.001)	2.7/5
					89.82	NB	90.30 (0.009)	68.30 (0.002)	70.56 (0.01)	69.30 (0.007)	90.34 (0.04)	2.6/5
						SVM	78.82 (0.03)	68.50 (0.03)	70.10 (0.004)	71.4 (0.003)	86.67 (0.005)	2.1/5
Dermatology □	$RF = 2$ $LF = 0.25$ $l = 4$ $fpc = 13$	91.03 (0.009)	92		95.41	k-NN	89.5 (0.05)	87.5 (0)	87.5 (0)	86.25 (0.003)	90.00 (0.0012)	2.1/4
				97.05		NB	95.00 (0.001)	82.5 (0.001)	85.0 (0.01)	81.25 (0.01)	**97.55** (0.0013)	2.1/4
						SVM	90.10 (0.0012)	85.82 (0.001)	**90.8** (0.03)	90.0 (0.01)	91.25 (0.0003)	2/4

consultation policy over all states. In addition, a comparison ADFL's CCR (shown in Table 3) and the average CCR of LDEs in a BRSM-based design (shown in Table 6) shows that the CCR of ADFL has been boosted compared with those of the BRSM-based LDEs. However, this boosting has not been sufficient to defeat the ensemble-based methods. In Table 4 it is evident that when LDEs were redesigned using PKSM, ADFL's CCR was meaningfully improved and ADFL defeated the benchmark methods in all but the Heart and Waveform data sets. We elaborate on the results more in section 6.1. Another important characteristic of ADFL is its low statistical variance (stated with the numbers in parentheses below the mean CCRs in Tables 3 and 4). It is a sign of robustness to splitting the data into train and test partitions.

We performed the sign test (Friedman, 1940), Wilcoxon test (Wilcoxon, 1945) and *t*-test (Zimmerman, 1997) to check if the superiority of ADFL+PKSM over the benchmark methods is statistically significant (see Table 5). The numbers of ADFL+PKSM's win, draw, and loss against the methods are reported as well. As the results show, ADFL+PKSM (with the best LDEs) works better than the best ensemble-based and fusion-based methods over the data sets with 90% and 95% confidence respectively (see the two last columns in Table 5).

6 Discussion

This section begins with a discussion on the performance of ADFL with PKSM-based and BRSM-based LDEs. Then the basic properties of ADFL are verified through testing its ability to cope with the existence of duplicated and systematically incorrect LDEs. Finally, a brief discussion on ADFL's time complexity is given.

6.1 ADFL and LDEs' Design. ADFL with BRSM-based LDEs did not outperform ensemble-based methods over majority of the data sets. To find the reason, we have studied the distribution of the training samples, produced by mostly attended BRSM-based LDEs, in the decision space. It has been observed that the training instances are scattered irregularly in the decision space composed of BRSM-based LDEs. In other words, the samples of each class have not formed distinct granules in the decision space. This means that the ADFL method has not been able to form sufficient decision boundaries in the decision space, or observation that directed us to define a desired property for the placement of training samples in the decision space. This property is the clusterability of the training samples. Then we defined the Separability Index (*SI*) to measure the clusterability and developed the PKSM to maximize it. *SI* is defined as

$$SI = \sum_{j=1}^{CN} \frac{\max_i(clustMem_{ij})}{\sum_{i=1}^{C} clustMem_{ij}} \times 100, \tag{6.1}$$

Table 4: Comparing ADFL+PKSM and the Benchmark Methods.

			Decision Space Methods					
			Classification	Fusion Methods			ADFL+PKSM	
Data Set	Feature Space Methods	Base Learner	k-NN	Majority Voting	Borda Count	OWA	Test CCR	Consultation Ratio
Heart ☒	Refer to **Table 3**.	k-NN	71.10 (0.01)	66.1 (0.03)	61.1 (0.03)	63.1 (0.01)	77.9 (0.002)	3.4/5
		NB	79.7 (0.04)	72.4 (0.06)	**76.5** (0.05)	69.1 (0.07)	**81.92** (0.008)	3.5/5
		SVM	70.15 (0.09)	66.10 (0.05)	58.1 (0.05)	63.5 (0.007)	76.12 (0.001)	2.3/5
Hepatitis ☑		k-NN	77.3 (0.01)	71.3 (0.02)	57.1 (0.02)	61.0 (0.03)	81.35 (0.007)	3.4/5
		NB	77.1 (0.02)	**72.1** (0.06)	58.3 (0.02)	63.3 (0.07)	79.5 (0.003)	3.1/5
		SVM	80.0 (0.02)	71.3 (0.05)	66.6 (0.02)	67.1 (0.02)	**86.20** (0.003)	3.4/5
Liver Disorder (Bupa) ☑		k-NN	60.2 (0.04)	61.1 (0.03)	59.4 (0.02)	61.0 (0.2)	71.34 (0.002)	3.9/5
		NB	60.1 (0.004)	59.16 (0.01)	55.9 (0.01)	63.6 (0.03)	65.3 (0.001)	2.2/5
		SVM	63.15 (0.09)	65.5 (0.013)	64.6 (0.09)	**69.0** (0.001)	**72.62** (0.001)	2.7/5
Pima Indian Diabetes □		k-NN	71.15 (0.09)	69.15 (0.001)	68.7 (0.09)	62.9 (0.001)	76.10 (0.01)	2.9/4
		NB	72.3 (0.07)	75.3 (0.03)	**76.6** (0.06)	66.3 (0.03)	**76.34** (0.001)	1.9/4
		SVM	70.5 (0.01)	63.9 (0.001)	65.2 (0.04)	61.1 (0.06)	73.25 (0.001)	2.0/4

Table 4: *(Continued)*

		Decision Space Methods						
			Classification	Fusion Methods			ADFL+PKSM	
Data Set	Feature Space Methods	Base Learner	k-NN	Majority Voting	Borda Count	OWA	Test CCR	Consultation Ratio
Ionosphere ☑		k-NN	83.88 (0.001)	**81.16** (0.001)	80.5 (0.06)	63.88 (0.04)	**88.9** (0.006)	2.2/4
		NB	82.7 (0.005)	78.9 (0.04)	75.4 (0.08)	71.9 (0.07)	83.47 (0.05)	2.1/4
		SVM	79.38 (0.007)	68.15 (0.005)	70.40 (0.09)	64.7 (0.07)	88.10 (0.004)	2.5/4
Sonar ☑		k-NN	78.5 (0.06)	69.5 (0.02)	61.8 (0.06)	67.2 (0.08)	83.66 (0.0)	2.4/4
		NB	80.9 (0.001)	75.00 (0.01)	**75.2** (0.03)	54.5 (0.01)	**84.09** (0.009)	2.4/4
		SVM	65.1 (0.08)	67.17 (0.004)	65.3 (0.06)	61.42 (0.05)	73.81 (0.009)	2.2/4
Glass ☑		k-NN	60.2 (0.001)	54.6 (0.01)	59.6 (0.001)	**60.19** (0.05)	68.15 (0.001)	2.6/5
		NB	60.34 (0.03)	60.1 (0.01)	58.1 (0.03)	57.1 (0.013)	**72.11** (0.0014)	2.3/5
		SVM	57.1 (0.09)	55.5 (0.09)	52.1 (0.004)	52.9 (0.001)	66.15 (0.009)	2.0/5
Vehicle ☑		k-NN	68.60 (0)	**64.35** (0.06)	54.04 (0.001)	62.95 (0.02)	**78.40** (0.003)	2.9/5
		NB	66.27 (0.015)	60.12 (0.001)	61.6 (0.09)	61.8 (0.005)	74.93 (0.009)	3.1/5
		SVM	62.7 (0.08)	55.9 (0.02)	60.15 (0.07)	58.18 (0.001)	66.29 (0.001)	2.4/5

Table 4: *(Continued)*

		Decision Space Methods						
			Classification	Fusion Methods			ADFL+PKSM	
Data Set	Feature Space Methods	Base Learner	k-NN	Majority Voting	Borda Count	OWA	Test CCR	Consultation Ratio
Waveform		k-NN	75.4	**75.82**	57.1	69.12	76.9	2.4/5
☒			(0.09)	(0.001)	(0.009)	(0.02)	(0.0003)	
		NB	81.33	75.00	65.39	75.88	**82.9**	2.1/5
			(0.01)	(0.04)	(0.002)	(0.06)	(0.001)	
		SVM	74.03	71.3	72.81	65.9	78.02	2.2/5
			(0.09)	(0.06)	(0.09)	(0.07)	(0.0001)	
Satimage		k-NN	90.16	**74.18**	59.30	63.95	**95.02**	3.4/5
☒			(0.002)	(0.002)	(0)	(0.01)	(0.0001)	
		NB	90.02	69.50	72.4	71.50	91.5	2.2/5
			(0.001)	(0.04)	(0.03)	(0.04)	(0.005)	
		SVM	79.30	68.00	71.6	73.5	87.9	2.0/5
			(0.006)	(0.07)	(0.07)	(0.06)	(0.006)	
Dermatology		k-NN	90.9	88.9	89.1	89.6	91.60	2.2/4
□			(0.08)	(0.09)	(0.01)	(0.06)	(0.0012)	
		NB	95.5	75.00	82.5	85.5	**97.25**	3.1/4
			(0)	(0)	(0.012)	(0)	(0.0003)	
		SVM	93.20	86.7	**93.9**	91.2	92.78	2.1/4
		(85.6)	(0.007)	(0.06)	(0.05)	(0.04)	(0.0003)	

Table 5: Sign (*ps*), Wilcoxon (*pw*), and *t*-Test(*pt*) Results of ADFL+PKSM (with Best Base learners) versus Benchmark Methods.

	Bagging +k-NN	Adaboost +SVM	Adaboost +k-NN	Ensemble (Bests)	Fusion (Best)
Win/draw/loss	9/0/2	8/2/1	11/0/0	7/2/2	10/0/1
PKSM+ADFL	$ps = 0.0654$	$ps = 0.0117$	$ps = 0.0009$	$ps = 0.0654$	$ps = 0.0117$
(Best)	$pw = 0.0048$	$pw = 0.0097$	$pw = 0.0009$	$pw = 0.0322$	$pw = 0.0019$
	$pt = 0.0068$	$pt = 0.0123$	$pt = 0$	$pt = 0.0406$	$pt = 0.0009$

where $clustMem_{ij}$ is the number of members of class i in the cluster j and CN is the number of clusters.

In Table 6, BRSM and PKSM methods are compared in terms of the LDEs' CCR and the separability index, in addition to the CCR of the ADFL. The statistical variance of the separability index is reported in parentheses. More details about the PKSM method, including LDEs' feature sets ($\boldsymbol{F}_i, i = 1, 2, \ldots, c$), are given in Table 7.

As can be inferred from Table 6, switching from BRSM to PKSM-based LDEs improves the performance of fusion-based methods as well except in the Hepatitis and Waveform data sets. However, their improvement is not sufficient to surpass the ensemble-based methods, where as ADFL can defeat them. Another observation is that ADFL surpasses OWA, a learning fusion method.

Table 6 shows an increase in both LDEs' average CCR and *SI* results in improved ADFL performance in all but in two cases: Waveform and Dermatology, with SVM and NB LDEs respectively, where the reductions are less than 1%. An increase in the average CCR of LDEs results in higher *SI* in most cases, while the inverse is not always true. It means that improvement in *SI* does not necessarily require considerable enhancement of LDEs' CCR. In fact, in some cases, an increase in *SI* by switching from BRSM to PKSM has resulted in higher CCR for ADFL, while the average and maximum CCRs of LDEs are reduced. The cases are the Vehicle, Glass, Dermatology, and Ionosphere data sets with k-NN, SVM, SVM, and NB LDEs, respectively. To investigate the problem further, we generated 50 sets of LDEs for the Liver and Hepatitis data sets, respectively, using the BRSM method. We found that the correlation of *SI* and the average CCR of LDEs is a small, negative value. All in all, it can be inferred that *SI* is a proper preevaluation mechanism to find out whether the designed LDEs are potentially suitable for the ADFL algorithm even if their CCRs are not high. Note that the design of LDEs with a high CCR is difficult in practice; increasing *SI* is much simpler.

Heart and Waveform are the only data sets where replacing BRSM-based LDEs with PKSM-based ones did not help ADFL to defeat ensemble-based benchmarks. Nevertheless, it assisted ADFL in improving its best

Table 6: Comparing the Design of LDEs in BRSM Versus PKSM and Their Effect on ADFL.

Data Set	Base Learners' Algorithm	Local CCRs of LDEs (Average)		*SI*		ADFL's CCR		OWA's CCR	
		BRSM	PKSM	BRSM	PKSM	BRSM	PKSM	BRSM	PKSM
Heart	k-NN	66,67,65,66,66 [66]	65,66,62,70,68 [66.2]	69.5 (4)	71.2 (5)	**74.51** (0.001)	**77.9** (0.002)	66.6 (0.02)	63.1 (0.01)
	NB	60,64,58,61,61 [60.8]	70,70,68,69,69 [69.2]	70 (12)	75.4 (4)	**75.90** (0.016)	**81.92** (0.008)	64.1 (0.05)	69.1 (0.07)
	SVM	61,61,62,58,59 [60.2]	61,61,63,66,62 [62.6]	67.3 (6)	68.3 (6)	**74.63** (0.0006)	**76.12** (0.001)	64.4 (0.08)	63.5 (0.007)
Hepatitis	k-NN	54,53,56,51,53 [53.4]	70,71,72,69,66 [69.6]	73.4 (4)	83.1 (5)	**80.14** (0.001)	**81.35** (0.007)	59.0 (0.01)	61.0 (0.03)
	NB	53,52,59,53,56 [54.6]	74,76,73,72,72 [73.4]	77.4 (3)	79.5 (9)	**78.2** (0.004)	**79.5** (0.003)	58.1 (0.03)	63.3 (0.07)
	SVM	58,57,57,60,58 [58]	72,74,70,73,73 [72.4]	79.2 (7)	87.2 (2)	**82.23** (0.006)	**86.20** (0.003)	63.52 (0.02)	67.1 (0.02)
Liver Disorder (bupa)	k-NN	57,57,55,55,57 [56.2]	65,68,62,63,63 [64.2]	66.3 (4)	69.7 (3)	**66.9** (0.007)	**71.34** (0.002)	65.1 (0.08)	61.0 (0.2)
	NB	46,51,47,55,49 [49.6]	62,60,63,60,63 [61.6]	60.2 (4)	63.2 (4)	**64.20** (0.007)	**65.3** (0.001)	62.54 (0.015)	63.6 (0.03)
	SVM	58,61,55,49,53 [55.2]	64,57,63,60,64 [61.6]	52.6 (5)	54.1 (2)	**71.42** (0.001)	**72.62** (0.001)	68.9 (0.03)	69.0 (0.001)
Pima Indian Diabetes	k-NN	67,60,62,62 [62.7]	69,64,67,59 [64.7]	69.2 (4)	72.1 (6)	**75.36** (0.001)	**76.10** (0.01)	62.7 (0.03)	62.9 (0.001)
	NB	63,62,60,66 [62.7]	65,65,60,68 [64.5]	72.3 (3)	76.3 (1)	**75.92** (0.05)	**76.34** (0.001)	60.2 (0.01)	66.3 (0.03)
	SVM	61,59,60,59 [59.7]	63,67,66,65 [65.2]	68.3 (5)	70.3 (5)	**72.14** (0.001)	**73.25** (0.001)	59.9 (0.07)	61.1 (0.06)
Ionosphere	k-NN	76,77,73,82 [77]	82,80,80,78 [80]	85.3 (4)	89.2 (7)	**84.05** (0.008)	**88.9** (0.006)	71.4 (0.04)	63.88 (0.04)
	NB	83,80,78,81 [80.5]	81,80,81,79 [80.25]	81.3 (2)	83.1 (7)	**82.37** (0.06)	**83.47** (0.05)	70.6 (0.08)	71.9 (0.07)
	SVM	78,83,86,82 [82.2]	77,78,80,83 [79.5]	70.1 (2)	73.7 (3)	**88.90** (0.001)	**88.10** (0.004)	63.88 (0.01)	64.7 (0.07)

Table 6: *(Continued)*

Data Set	Base Learners' Algorithm	Local CCRs of LDEs (Average)		*SI*		ADFL's CCR		OWA's CCR	
		BRSM	PKSM	BRSM	PKSM	BRSM	PKSM	BRSM	PKSM
Sonar	k-NN	63,60,63,68	71,71,70,71	83.5	89.2	**82.42**	**83.66**	66.3	67.2
		[63.5]	[70.7]	(4)	(7)	(0.08)	(0.09)	(0.01)	(0.08)
	NB	67,71,64,70	69,70,73,74	84.3	94.7	**80.53**	**84.09**	64.1	54.5
		[68]	[71.5]	(7)	(5)	(0.004)	(0.009)	(0.04)	(0.01)
	SVM	58,65,63,61	65,65,64,65	82.6	87.5	**72.70**	**73.81**	60.30	61.42
		[61.7]	[64.7]	(9)	(3)	(0.004)	(0.009)	(0.01)	(0.05)
Glass	k-NN	58,65,55,63,61	61,62,55,61,67	75.4	79.4	**64.54**	**68.15**	59.0	60.19
		[60.4]	[61.2]	(7)	(10)	(0.09)	(0.001)	(0.01)	(0.05)
	NB	60,60,57,60,61	63,59,60,69,52	70.3	84.4	**65.32**	**72.11**	59.3	57.1
		[59.6]	[60.6]	(4)	(9)	(0.002)	(0.0014)	(0.05)	(0.013)
	SVM	69,62,64,54,58	60,63,55,63,57	70.3	74.1	**64.06**	**66.15**	52.0	52.9
		[61.4]	[59.6]	(6)	(5)	(0.003)	(0.009)	(0.04)	(0.001)
Vehicle	k-NN	59,62,60,63,63	59,57,60,56,61	74.4	96.8	**75.20**	**78.40**	62.10	62.95
		[61.4]	[58.6]	(9)	(3)	(0.007)	(0.003)	(0.003)	(0.02)
	NB	53,57,52,50,59	61,72,68,53,65	60.1	84.2	**73.81**	**74.93**	60.50	61.8
		[54.2]	[63.8]	(5)	(9)	(0.0002)	(0.009)	(0.04)	(0.005)
	SVM	51,50,54,53,52	53,57,59,61,57	61.2	80.1	**65.11**	**66.29**	59.25	58.18
		[52]	[57.4]	(7)	(9)	(0.006)	(0.001)	(0.01)	(0.001)
Waveform	k-NN	73,72,69,65,65	73,70,66,62,77	70.8	76.4	**76.47**	**76.9**	69.60	69.12
		[68.8]	[69.6]	(5)	(4)	(0.001)	(0.0003)	(0.017)	(0.02)
	NB	78,67,73,65,77	72,75,77,75,71	70.2	77.9	**82.84**	**82.9**	77.45	75.88
		[72]	[74]	(4)	(8)	(0.003)	(0.001)	(0.07)	(0.06)
	SVM	80,76,76,66,79	74,76,82,73,79	65.4	74.1	**78.92**	**78.02**	64.70	65.9
		[75.4]	[76.8]	(4)	(7)	(0.0004)	(0.0001)	(0.064)	(0.07)
Satimage	k-NN	79,79,88,84,90	91,88,85,88,89	75.5	94.5	**93.23**	**95.02**	71.42	63.95
		[84]	[88.2]	(7)	(4)	(0.001)	(0.0001)	(0.01)	(0.01)
	NB	84,85,79,83,82	80,86,82,85,87	79.1	81.3	**90.34**	**91.5**	69.30	71.50
		[82.6]	[84]	(4)	(5)	(0.04)	(0.005)	(0.007)	(0.04)
	SVM	87,86,87,83,83	81,83,86,77,88	80.4	90.0	**86.67**	**87.9**	71.4	73.5
		[85.2]	[83]	(8)	(10)	(0.005)	(0.006)	(0.003)	(0.06)
Dermatology	k-NN	89,89,82,84	95,91,91,84	80.9	85.3	**90.00**	**91.60**	86.25	89.6
		[86]	[90.2]	(6)	(5)	(0.0012)	(0.0012)	(0.003)	(0.06)
	NB	88,86,89,89	91,89,88,88	86.3	94.0	**97.55**	**97.25**	81.25	85.5
		[88]	[89]	(6)	(5)	(0.0013)	(0.0003)	(0.01)	(0)
	SVM	81,84,76,89	81,82,85,79	80.1	82.4	**91.25**	**92.78**	90.0	91.2
		[82.5]	[81.7]	(5)	(5)	(0.0003)	(0.0003)	(0.01)	(0.04)

Table 7: Feature Splits for the Best *SI* and the Corresponding Base Learner Algorithm.

Data Set	Best *SI* by PKSM (var)	Base Learner	Feature Sets of LDEs
Heart	75.4 (4)	NB	$\boldsymbol{F}_1 = [1, 2, 3]$ $\boldsymbol{F}_2 = [1, 2, 3, 5]$ $\boldsymbol{F}_3 = [1, 2, 4, 6, 3, 8]$ $\boldsymbol{F}_4 = [1, 2, 6, 7, 5, 3]$ $\boldsymbol{F}_5 = [1, 2, 12, 7, 6, 13]$
Hepatitis	87.2 (2)	SVM	$\boldsymbol{F}_1 = [1, 2, 4, 5, 6, 7, 10, 11, 12, 13, 19]$ $\boldsymbol{F}_2 = [1, 2, 3, 4, 5, 6, 7, 10, 11, 12, 13, 8, 9]$ $\boldsymbol{F}_3 = [1, 2, 14, 15, 16, 17, 18]$ $\boldsymbol{F}_4 = [1, 2, 4, 5, 6, 7, 10, 11, 12, 13, 14, 15, 17, 18]$ $\boldsymbol{F}_5 = [1, 2, 5, 18, 19]$
Liver Disorder (bupa)	69.7 (3)	k-NN	$\boldsymbol{F}_1 = [6, 2, 5]$ $\boldsymbol{F}_2 = [6, 1, 5]$ $\boldsymbol{F}_3 = [6, 2, 3]$ $\boldsymbol{F}_4 = [6, 2, 3, 4]$ $\boldsymbol{F}_5 = [6, 3, 4, 5]$
Pima Indian Diabetes	76.3 (1)	NB	$\boldsymbol{F}_1 = [1, 3, 4, 6, 7, 8]$ $\boldsymbol{F}_2 = [1, 2, 3, 5, 7, 8]$ $\boldsymbol{F}_3 = [1, 3, 4, 5, 6, 8]$ $\boldsymbol{F}_4 = [1, 2, 5, 6, 7, 8]$
Ionosphere	89.2 (7)	k-NN	$\boldsymbol{F}_1 = [6,15,8,3,14,17,12,1,11,2,9,5,13, 30]$ $\boldsymbol{F}_2 = [7,6,11,21,8,10,13, 14,28, 18, 20]$ $\boldsymbol{F}_3 = [25,16,12,13,20,9,15,19,34, 18,23,10,17]$ $\boldsymbol{F}_4 = [18,19,26,25,23,32,31,20,28,29,27]$
Sonar	94.7 (5)	NB	$\boldsymbol{F}_1 = [17,26,10,11, 19,2, 39,14,7, 6,4, 15,34, 5,22,1,3, 25,32, 30,12]$ $\boldsymbol{F}_2 = [8,11,9, 50,18, 49,44, 26,22,17, 19,51,39,15,30, 10, 12, 25]$ $\boldsymbol{F}_3 = [21, 16, 50, 22, 19, 56, 55, 46, 38, 52, 15,18, 60,51, 59, 44, 58, 25, 17,47]$ $\boldsymbol{F}_4 = [57,54,23,60,59,37,48, 46, 36,50,35, 52,51,21, 49,44,45,53,33$
Glass	84.4 (9)	NB	$\boldsymbol{F}_1 = [9, 2, 6]$ $\boldsymbol{F}_2 = [5,2, 3, 4]$ $\boldsymbol{F}_3 = [2, 3, 9, 8]$ $\boldsymbol{F}_4 = [6, 2, 4]$ $\boldsymbol{F}_5 = [2, 3, 6, 7, 4]$
Vehicle	96.8 (3)	k-NN	$\boldsymbol{F}_1 = [9,11, 15, 4, 16, 7, 12, 18, 6, 2]$ $\boldsymbol{F}_2 = [17, 18, 5, 12, 11, 10, 8, 9, 4, 13, 3]$ $\boldsymbol{F}_3 = [6, 8, 14, 9, 12, 13, 7, 5, 18, 2, 4]$ $\boldsymbol{F}_4 = [2, 7, 15,12, 17, 4, 1, 3, 8, 14, 10, 18, 6, 9]$ $\boldsymbol{F}_5 = [12,17, 10, 7, 1, 3,13, 8, 2, 9, 4,11, 18, 16]$
Waveform	77.9 (8)	NB	$\boldsymbol{F}_1 = [14, 2, 22, 13, 6, 17, 5, 24, 26, 25,18, 36, 8,15, 34, 7, 1,28,16, 35, 30, 19, 3, 32, 27]$ $\boldsymbol{F}_2 = [8,17,25,32,33,24,16,10,3,35, 28,31, 7, 19, 11, 2, 22, 9, 13, 12, 1, 34 15, 36]$ $\boldsymbol{F}_3 = [15, 28,17, 7, 29, 3, 36, 6,26,24,10, 18,12,35,13, 4, 25,33,16,34, 2, 11]$

Table 7: *(Continued)*

			$\boldsymbol{F}_4$ = [22,13,7,10,30,6,32,31,21,29, 24, 35,9, 5, 1,11,14,36,8,3]
			$\boldsymbol{F}_5$ = [28,25,20, 2, 8,15, 23,3,10, 4,30,7, 21,17,18, 22, 34, 33,13, 29,14,24, 32,12]
Satimage	94.5 (4)	k-NN	$\boldsymbol{F}_1$ = [28,9,19,30,1,20,26,29, 8,13,32,15,31, 5, 14, 6,3, 4,27, 2,12,23,17, 18, 16]
			$\boldsymbol{F}_2$ = [17, 18,31,28, 3,13, 1,21, 2,35,26,34,14, 5,30,23,11,25,16, 8,10]
			$\boldsymbol{F}_3$ = [12,9,4,24,3,16,22,13, 6, 14, 5, 28,35,18,29,32,21,11,36,31, 7]
			$\boldsymbol{F}_4$ = [18,2,17,11, 36, 8, 13, 22, 7,34, 9,32,30, 5,25,20,1,33,31,4,26]
			$\boldsymbol{F}_5$ = [22,18,12, 4, 10,6,26,35, 33,27, 24, 32,14, 25,31, 3,21,5, 9,16,34,19, 17,30]
Dermatology	94.0 (5)	NB	$\boldsymbol{F}_1$ = [23, 7,11,1,34,5,10,21,17, 2, 3]
			$\boldsymbol{F}_2$ = [30,15,11,9,26,24,10,5,18,7,3]
			$\boldsymbol{F}_3$ = [32,20,13,30,11,23,31,19,4,24,14,18]
			$\boldsymbol{F}_4$ = [4,27,33, 6, 13, 26, 28,32,24, 25,14]

performance over LDEs from 75.9 to 81.92 in Heart and kept its performance in Waveform unchanged.

It is clear that *SI* is maximized in two abnormal circumstances: where all LDEs are perfect (CR = 100), or they are totally and systematically wrong (CCR = 0). We study the second situation in the next section by adding one systematically wrong decision maker to our set of LDEs. It is obvious that this decision maker reduces the average CCR of LDEs.

6.2 Duplicated and Systematically Wrong LDEs. An important property of ADFL, which is not usually a concern in classification and ensemble-based methods but is highly desired in information fusion systems, is its robustness to some sorts of problematic design of LDEs. To verify this property, we tested ADFL's robustness against adding some duplicated and systematically wrong LDEs to the set of existing LDEs. Such LDEs cause difficulty for common decision fusion methods. However, our experiments revealed that ADFL can automatically detect and manage consultation with duplicated LDEs and makes the best use of systematically incorrect ones as well.

6.2.1 Duplicated Decision Makers. For each data set, the design with the best performance is selected (from Table 4). Then one of the PKSM-based LDEs is duplicated at a time, and ADFL is executed again. The results are averaged over the LDEs (see Table 8). Each duplicated LDE adds $c - 1$ dimensions to the decision space, which are actually redundant. The results show that the fusion methods and k-NN in the decision space experience

Table 8: Duplication Test Scenario.

Data Set (LDEs' Base Learner)	Description	Average CCR of LDEs	k-NN on Decision Space	Majority Voting	Borda Count	OWA	ADFL	Consultation Ratio
Heart (NB)	Original	70	79.7 (0.04)	72.4 (0.06)	76.5 (0.05)	69.1 (0.07)	**81.92** (0.008)	3.5/5
	With duplicated LDE	70	75.1 (0.02)	70.9 (0.04)	72.3 (0.09)	66.3 (0.01)	82.4 (0.005)	3.3/6
Hepatitis (SVM)	Original	71.2	80.0 (0.02)	71.3 (0.05)	66.6 (0.02)	67.1 (0.02)	**86.20** (0.003)	3.4/5
	With duplicated LDE	71.2	78.2 (0.01)	70.45 (0.05)	63.81 (0.007)	65.5 (0.01)	**85.34** (0.004)	3.3/6
Liver (k-NN)	Original	64	60.2 (0.04)	61.1 (0.03)	59.4 (0.02)	61.0 (0.2)	**71.34** (0.002)	3.9/5
	With duplicated LDE	64	54.6 (0.07)	56.7 (0.04)	54.3 (0.03)	55.4 (0.04)	71.5 (0.004)	**4/6**
Pima Indian Diabetes (NB)	Original	64.5	72.3 (0.07)	75.3 (0.03)	76.6 (0.06)	66.3 (0.03)	**76.34** (0.001)	1.9/4
	With duplicated LDE	64.5	68.1 (0.07)	69.1 (0.05)	71.2 (0.01)	60.1 (0.07)	**75.8** (0.05)	2.1/5
Ionosphere (k-NN)	Original	79.2	83.88 (0.001)	81.16 (0.001)	80.5 (0.06)	63.88 (0.04)	**88.9** (0.006)	2.2/4
	With duplicated LDE	79.2	82.7 (0.004)	80.4 (0.05)	80.1 (0.09)	64.2 (0.07)	87.3 (0.005)	2.5/5
Sonar (NB)	Original	70	80.9 (0.001)	75.00 (0.01)	75.2 (0.03)	54.5 (0.01)	**84.09** (0.009)	2.4/4
	With duplicated LDE	70	78.8 (0.05)	72.2 (0.04)	74.7 (0.06)	53.3 (0.09)	84.3 (0.005)	2.1/5

Table 8: *(Continued)*

Data Set (LDEs' Base Learner)	Description	Average CCR of LDEs	k-NN on Decision Space	Majority Voting	Borda Count	OWA	ADFL	Consultation Ratio
Glass (NB)	Original	61	60.34 (0.03)	60.1 (0.01)	58.1 (0.03)	57.1 (0.013)	**72.11** (0.0014)	2.3/5
	With duplicated LDE	61	59.9 (0.05)	56.9 (0.04)	54.7 (0.08)	52.8 (0.09)	71.8 (0.004)	2.5/6
Vehicle (k-NN)	Original	61.5	68.60 (0)	64.35 (0.06)	54.04 (0.001)	62.95 (0.02)	**78.40** (0.003)	**2.9/5**
	With duplicated LDE	61.5	66.72 (0.005)	60.7 (0.03)	50.20 (0.04)	60.13 (0.05)	**77.91** (0.001)	3.1/6
Waveform (NB)	Original	74.5	81.33 (0.001)	75.00 (0.04)	65.39 (0.002)	75.88 (0.06)	**82.9** (0.001)	2.1/5
	With duplicated LDE	74.5	80.59 (0.009)	76.05 (0.08)	65.45 (0.07)	71.92 (0.03)	**82.50** (0.009)	2.4/6
Satimage (k-NN)	Original	88	90.16 (0.002)	74.18 (0.002)	59.30 (0)	63.95 (0.01)	**95.02** (0.0001)	3.4/5
	With duplicated LDE	88	90.5 (0.005)	73.9 (0.008)	59.1 (0.06)	64.2 (0.05)	94.6 (0.004)	3.7/6
Dermatology (NB)	Original	87	95.5 (0)	75.00 (0)	82.5 (0.012)	85.5 (0)	**97.25** (0.0003)	3.1/4
	With duplicated LDE	87	95.0 (0.03)	73.2 (0)	81.3 (0.04)	84.8 (0.04)	97.6 (0.007)	2.9/5

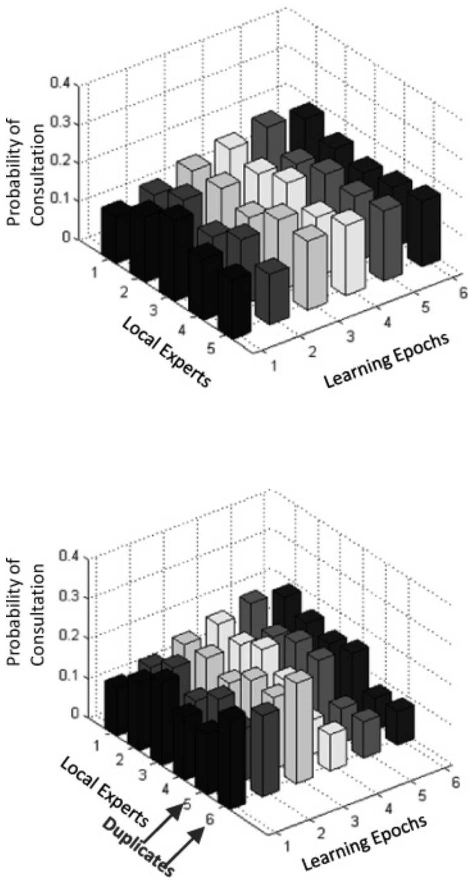

Figure 4: The pattern of consultations in the duplicated LDE scenario for the Hepatitis data set. (Top) There is no duplicated LDE. (Bottom) LDE 5 is duplicated as 6. At the end of learning, it can be simply observed that both are equally probable for consultations.

a performance drop in data sets where we have on average weaker LDEs (Liver Disorder, Pima Indian Diabetes, Glass, and Vehicle). However, as expected, in data sets where the classifiers are on average stronger (Dermatology, Satimage, and Ionosphere), the drop is on average smaller. In contrast, ADFL was robust against this redundancy in all of the cases. In addition, Figure 4 illustrates that ADFL learns not to consult with both an LDE and its copy.

6.2.2 Systematically Wrong Decision Makers. In the binary problems, an LDE that assigns class 1 with probability 1 to data originally belonging to class 2 and vice versa was added. This is actually an output-inverted classifier. Table 9 shows the results of this scenario. For the data sets with multiple classes, an LDE was added that announces class 1, 2,. . . and N for data belonging to class N,. . . , 2, and 1, respectively. See Table 10 for the results.

As the results demonstrate, ADFL has discovered the situations and benefited from them by learning to consult with the systematically wrong LDEs only (see Figure 5). It has also reached the perfect recognition rate with consulting such LDEs. k-NN in the decision space has reached 100% recognition rate as well, as the existence of systematically wrong LDEs results in placing the training samples of each class on a separate hyperplane in the decision space. It means $SI = 100$. However, as Tables 9 and 10 show, the fusion methods have experienced big drops in their performance.

6.3 Time Complexity. ADFL employs a set of local experts and learns attentive active fusion of their decisions. Therefore, the time complexity of ADFL in the training mode is composed of two parts: that of training the local experts and the time complexity of attentive decision fusion learning. The time complexity in the first part depends on the type of base learners (e.g. ANN, naıve Bayes, k-NN) and is independent of ADFL itself. The second part is the time complexity of the employed continuous reinforcement learning method, which is a matter of choice. In the recall mode, the time complexity of attentive decision fusion is negligible against the time complexity of local experts.

7 Conclusions and Future Work

In this letter, we proposed the ADFL approach for learning attentive decision fusion in the decision space. In this approach, the problem of learning which decision maker to consult with and learning the final decision are modeled as a single sequential decision-making problem. The coupled problem was cast in a Markov decision process with continuous state and discrete actions, and then a continuous RL method was employed to solve it. ADFL was tested on a set of classification tasks where it defeated two well-known classification methods, Adaboost and Bagging, with 90% confidence, in addition to benchmark fusion algorithms—OWA, Borda count, and majority voting—with 95% confidence.

In addition to satisfactory CCR, ADFL has other distinct characteristics. From the fusion perspective, it learns the proper sequence of consultation with the decision makers for each part of the decision space. This means that instead of learning a single consultation strategy over the entire decision space, it learns attentive local consultation policies. This characteristic is highly desired, especially when decision making in subspaces is more accurate than in the original space (Harandi, Nili Ahmadabadi, & Araabi,

Table 9: Output-Inverted Test by Using PKSM-Based LDEs in the Binary Data Sets.

Data Set (LDEs' Base learner)	Description (Average CCR of LDEs)	Average CCR of LDEs[a]	k-NN on Decision Space	Majority Voting	Borda Count	OWA	ADFL	Consultation Ratio
Heart (NB)	Original	70	79.7 (0.04)	72.4 (0.06)	76.5 (0.05)	69.1 (0.07)	**81.92** (0.008)	3.5/5
	With output-inverted LDE	58.3	100 (0)	64.3 (0.08)	70.2 (0.01)	61.3 (0.08)	**100** (0)	1.3/6
Hepatitis (SVM)	Original	71.2	80.0 (0.02)	71.3 (0.05)	66.6 (0.02)	67.1 (0.02)	**86.20** (0.003)	3.4/5
	With output-inverted LDE	59.3	100 (0)	59.4 (0.05)	54.3 (0.04)	60.9 (0.04)	**100** (0)	1.6/6
Liver Disorder (k-NN)	Original	64	60.2 (0.04)	61.1 (0.03)	59.4 (0.02)	61.0 (0.2)	**71.34** (0.002)	3.9/5
	With output-inverted LDE	53.3	100 (0)	52.9 (0.05)	50.1 (0.03)	55.4 (0.09)	**100** (0)	1.2/6
Pima Indian Diabetes (NB)	Original	64.5	72.3 (0.07)	75.3 (0.03)	76.6 (0.06)	66.3 (0.03)	**76.34** (0.001)	1.9/4
	With output-inverted LDE	51.6	100 (0)	68.9 (0.08)	70.25 (0.04)	59.5 (0.01)	**100** (0)	1.1/5
Ionosphere (k-NN)	Original	79.2	83.88 (0.001)	81.16 (0.001)	80.5 (0.06)	63.88 (0.04)	**88.9** (0.006)	2.2/4
	With output-inverted LDE	63.3	100 (0)	73.5 (0.04)	69.3 (0.04)	53.8 (0.05)	**100** (0)	1.2/5
Sonar (NB)	Original	70	80.9 (0.001)	75.00 (0.01)	75.2 (0.03)	54.5 (0.01)	**84.09** (0.009)	2.4/4
	With output-inverted LDE	56	100 (0)	63.9 (0.04)	69.1 (0.09)	50.1 (0.04)	**100** (0)	1.3/5

[a] By adding a systematically incorrect classifier with an actual CCR of zero, the average local CCR is clearly reduced.

Table 10: Systematically Incorrect Test by Using PKSM-Based LDEs in Multiple-Class Data Sets.

Data Set (LDEs' Base Learner)	Description	Average CCR of LDEs	k-NN on Decision Space	Majority Voting	Borda Count	OWA	ADFL	Consultation Ratio
Glass (NB)	Original	61	60.34 (0.03)	60.1 (0.01)	58.1 (0.03)	57.1 (0.013)	**72.11** (0.0014)	2.3/5
	With systematically wrong LDE	50.8	100 (0)	52.9 (0.04)	50.4 (0.06)	50.8 (0.01)	**100** (0)	1.5/6
Vehicle (k-NN)	Original	61.5	68.60 (0)	64.35 (0.06)	54.04 (0.001)	62.95 (0.02)	**78.40** (0.003)	2.9/5
	With systematically wrong LDE	51.2	100 (0)	58.4 (0.03)	47.9 (0.01)	55.5 (0.06)	**100** (0)	1.4/6
Waveform (NB)	Original	74.5	81.33 (0.001)	75.00 (0.04)	65.39 (0.002)	75.88 (0.06)	**82.9** (0.001)	2.1/5
	With systematically wrong LDE	62	100 (0)	62.9 (0.05)	54.5 (0.06)	69.4 (0.09)	**100** (0)	1.3/6
Satimage (k-NN)	Original	89.3	90.16 (0.0002)	74.18 (0.002)	59.30 (0)	63.95 (0.01)	**95.02** (0.0001)	3.4/5
	With systematically wrong LDE	74.4	100 (0)	63.80 (0.05)	48.72 (0.001)	52.78 (0.05)	**100** (0)	1.5/6
Dermatology (NB)	Original	87	95.5 (0)	75.00 (0)	82.5 (0.012)	85.5 (0)	**97.25** (0.0003)	3.1/4
	With systematically wrong LDE	69.5	100 (0)	62.03 (0.03)	73.7 (0.002)	65.9 (0.06)	**100** (0)	1.2/5

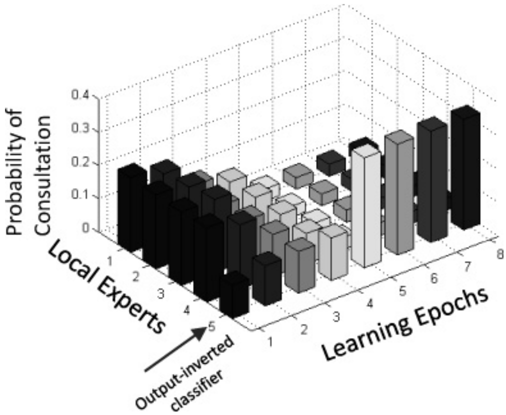

Figure 5: The pattern of consultations in a binary systematically wrong scenario for the Pima Indian Diabetes data set. At the end of learning, the output inverted classifier is the most attended one among all the LDEs.

2009). An important point about this locality attribute is that it is learned, not hand-designed.

From the attention perspective, ADFL tries to minimize its consultation cost through finding more informative decision makers in a case-by-case manner. This property is highly desired in a number of real-world applications such as medical diagnostic tasks and any other high-cost consultations. In addition, it creates a degree of freedom for the design of decision makers. It means that we do not have to build decision makers that work well all over the data set. Instead, since ADFL automatically finds local attentive consultation polices, decision makers that are solely expert on subsets of data are good as be used by ADFL. Developing such experts is much easier than making holistically good ones.

Furthermore, the property of selecting more informative decision makers and rejecting redundant and less informative ones was evaluated by duplication of decision makers and adding systematically wrong decision makers to the system. The results proved that ADFL makes the maximum possible benefit from the most informative decision maker—a systematically wrong one here—and does not blindly consult with the others. In addition, it learns not to consult with both a decision maker and its copy. This characteristic is highly desired from the fusion perspective as well; it provides an automatic method to implicitly rank the decision makers for each part of the decision space from the viewpoint of being informative. This property is rare in the common decision fusion methods.

From the application perspective, ADFL can be considered as a learning decision support system (LDSS) that sequentially suggests whom to consult with and helps in making the final decision based on the consultation results. Such LDSSs are helpful in domains like medical diagnostic and e-consultant systems. We examined the first application in this letter.

ADFL can be used to construct classification systems as well. For the classification tasks, the LDEs are usually not given and should be manually designed, as done by BRSM and PKSM. Our current research is focused on defining the desired properties of the local classifiers from an ADFL perspective and proposing efficient and automatic LDE design methods accordingly.

In this letter, we introduced k-NN on the decision space as well. The results showed that k-NN is a promising method for decision making in the decision space provided that a granular distribution of data can be formed in that space. This formation can be acquired by the proper design of LDEs. Nevertheless, k-NN is a hard classifier—it considers only a fixed number of neighbors—while ADFL generates soft decision boundaries through an attentive method. These characteristics justify higher performance of ADFL over k-NN in the decision space.

Acknowledgments

This research is supported by the University of Tehran and has also been realized in close collaboration with the BACS project supported by EC contract number FP6-IST-02′140, Action line: Cognitive Systems. We thank G. N. Pedrajas for sharing his data and valuable information.

References

Berenji, H. R. (1996). Fuzzy Q-learning for generalization of reinforcement learning. In *Proceedings of the Fifth IEEE International Conference on Fuzzy Systems, 1996.* Piscataway, NJ: IEEE.

Blake, C. L., & Merz, C. J. (1998). *UCI repository of machine learning databases, 1998.* Irvine: University of California at Irvine.

Borji, A., Ahmadabadi, M. N., Araabi, B. N., & Hamidi, M. (2010). Online learning of task-driven object-based visual attention control. *Image and Vision Computing, 28*, 1130–1145.

Chih-Jen Lin, C. C. (N.d.). LIBSVM. *LIBSVM: Library for support vector machines*. Available online at http://www.csie.ntu.edu.tw/∼cjlin/libsvm/.

Danziger, S. A., Zeng, J., Wang, Y., Brachmann, R. K., & Lathrop, R. H. (2007). Choosing where to look next in a mutation sequence space. *Bioinformatics, 23*(13), i104–i114.

Ebrahimpour, R., Kabir, E., Esteky, H., & Yousefi, M. R. (2008). View-independent face recognition with mixture of experts. *Neurocomputing, 71*(4–6), 1103–1107.

Filev, D., & Yager, R. R. (1998). On the issue of obtaining OWA operator weights. *Fuzzy Sets and Systems, 94*(2), 157–169.

Firouzi, H., Ahmadabadi, M. N., & Araabi, B. N. (2008). A probabilistic reinforcement-based approach to conceptualization. *International Journal of Intelligent Technology, 3*, 48–55.

Firouzi, H., Nili Ahmadabadi, M., N. Araabi, B., Amizadeh, S., Mirian, M. S., & Roland Siegwart. (2009). Interactive learning in continuous multimodal space: A Bayesian approach to action based soft partitioning. Manuscript submitted for publication.

Friedman, M. (1940). A comparison of alternative tests of significance for the problem of *m* rankings. *Annals of Mathematical Statistics, 11*(1), 86–92.

Garcia-Pedrajas, N. (2009). Constructing ensembles of classifiers by means of weighted instance selection. *IEEE Transactions on Neural Networks, 20*(2), 258–277.

Garcia-Pedrajas, N., & Ortiz-Boyer, D. (2009). Boosting K-nearest neighbor classifier by means of input space projection. *Expert Systems with Applications, 36*(7), 10570–10582.

Harandi, M. T., Nili Ahmadabadi, M., & Araabi, B. N. (2009). Optimal local basis: A reinforcement learning approach for face recognition. *International Journal of Computer Vision, 81*(2), 191–204.

Jordan, M. I., & Jacobs, R. A. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation, 6*(2), 181–214.

Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2006). Supervised machine learning: A review of classification techniques. *Artificial Intelligence Review, 26*(3), 159–190.

Kuncheva, L. I., Bezdek, J. C., & Duin, R. P. (2001). Decision templates for multiple classifier fusion: An experimental comparison. *Pattern Recognition, 34*(2), 299–314.

Lizotte, D. J., Madani, O., & Greiner, R. (2003). Budgeted learning of na ve-bayes classifiers. In *Proceedings of the 19th Conference in Uncertainty*. San Francisco: Morgan Kaufmann.

Minut, S., & Mahadevan, S. (2001). A reinforcement learning model of selective visual attention. In *Proceedings of the Fifth International Conference on Autonomous Agents*. Norwood, MA: Kluwer.

Mirian, M. S., Firouzi, H., Ahmadabadi, M. N., & Araabi, B. N. (2009). Concurrent learning of task and attention control in the decision space. In *Proceedings of IEEE/ASME Advanced Intelligent Mechatronics* (pp. 1353–1358). Piscataway, NJ: IEEE.

Paletta, L., Fritz, G., & Seifert, C. (2005). Q-learning of sequential attention for visual object recognition from informative local descriptors. In *Proceedings of the 22nd International Conference on Machine Learning*. New York: ACM.

Paletta, L., & Pinz, A. (2000). Active object recognition by view integration and reinforcement learning. *Robotics and Autonomous Systems, 31*(1), 71–86.

Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine, 6*(3), 21–45.

Polikar, R. (2007). Bootstrap-inspired techniques in computational intelligence. *IEEE Signal Processing Magazine, 24*(4), 59–72.

Schapire, R. E. (2003). The boosting approach to machine learning: An overview. *Lecture Notes in Statistics.* (pp. 149–172). New York: Springer-Verlag.

Shariatpanahi, F., & Nili Ahmadabadi, M. N. (2008). Biologically inspired framework for learning and abstract representation of attention control. In L. Paletta,

J. K. Tsotsos, & E. Rome (Eds.), *Attention in Cognitive Systems. Theories and Systems from an Interdisciplinary Viewpoint* (pp. 324–324). New York: Springer.

Sutton, R. S., & Barto, A. G. (1999). Reinforcement learning. *Journal of Cognitive Neuroscience, 11*(1), 126–134.

Verikas, A., Lipnickas, A., Malmqvist, K., Bacauskiene, M., & Gelzinis, A. (1999). Soft combination of neural classifiers: A comparative study. *Pattern Recognition Letters, 20*(4), 429–444.

Watkins, C. J., & Dayan, P. (1992). Q-learning. *Machine Learning, 8*(3), 279–292.

Whiteson, S., Taylor, M. E., & Stone, P. (2007). Adaptive tile coding for value function approximation (Tech. Rep. AI-TR-07-339). Austin: University of Texas.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin, 1*(6), 80–83.

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks, 5*(2), 241–259.

Woods, K., Kegelmeyer, W. P., & Bowyer, K. (1997). Combination of multiple classifiers using local accuracy estimates. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19*(4), 405–410.

Zhu, Y. (2003). *Multisensor decision and estimation fusion*. New York: Springer.

Zimmerman, D. W. (1997). A note on interpretation of the paired-samples t test. *Journal of Educational and Behavioral Statistics, 22*(3), 349–360.

Received January 18, 2010; accepted August 2, 2010.