International Journal of Innovative Computing, Information and Control
Volume **7**, Number **1**, January **2011**

pp. 1–09-0848

LEARNING WITH WHICH LOCAL DECISION EXPERT TO CONSULT NEXT CASE STUDY: ARRHYTHMIA DIAGNOSIS

Maryam S. Mirian[1], Majid Nili Ahmadabadi[1,2], Babak N. Araabi[1,2] and Mohammad M. Ajallooeian[1]

[1]Control and Intelligent Processing Centre of Excellence
Department of Electrical and Computer Engineering
University of Tehran
{ mmirian; ajallooeian }@ut.ac.ir

[2] School of Cognitive Science
IPM, Tehran, Iran
{ mnili; arrabi }@ut.ac.ir

Received August 2009; revised January 2010

Abstract. *The proposed approach is based on the classical model of Mixture-of-Experts and tries to concurrently learn two tightly coupled issues. As the main goal, it learns the optimal classification and at the same time, it learns the best sequence of council with previously designed local decision experts to reach the former optimal classification strategy. Local experts are in fact local classifiers who have learned the sub-optimal decision making based on just a portion of the whole feature space. The methodology is that in the first stage we generate different feature spaces by binning the features according to their potential relevance and then randomly selecting from the bins. At the second stage, we train a classifier for each of the resulting feature subsets. Finally, we use a continuous Q-learning variant for learning a combiner for the predictions of these classifiers which is the key contribution of the paper. Actually, the meta-learner in the last stage learns to combine different global models, each induced from a different feature subset. The domain of medical diagnosis (specifically Arrhythmia recognition) by using UCI datasets is opted as the benchmark. The acquired classification rate certifies that the proposed approach is quite comparable with the results have been reported so far. Moreover, this recognition is achieved by as few consultations as possible which is another key different merit for our approach.*
Keywords: Sequential decision making, Continuous Q-learning, Degree of support, Arrhythmia diagnosis, UCI ML repository

1. **Introduction.** Counseling and consulting with a field expert and, sometimes, using a combination of opinions of several professionals is a rather much used way in real-life situations: consulting with a multitude of skilled lawyers for legal cases, reference to a family advisor in familial problems and visiting a physician are some common everyday instances of this strategy. Consultation can enhance the level of confidence to the decision and facilitate looking at a single problem from different perspectives. However, there is a downside that counseling with an expert (or a number of experts) is an expensive and time-consuming procedure. Now, let's transfer the consultation problem into the domain of machine learning. Now, in this context, the stated problem gets even more crucial when the types of information provided by the experts are heterogeneous which requires spending considerable time and resources to prepare and deal with such a training database.

This makes an important consideration in the field of medical diagnosis, where the diagnosis procedure requires numerous sometimes-expensive and even harmful tests. The process of diagnosis starts with visiting a general physician. Then, an expert is introduced for advanced diagnosis based on the observed symptoms and the results of the tests instructed by the already consulted general physician. This process is continued until the diagnosis can undoubtedly be clarified. Here, the patient may be instructed to take tests which are actually unnecessary. Taking this much time might prolong the disease and, sometimes, make the illness chronic.

The stated problem provided us with a reason to re-implement our idea in the field of medical diagnosis. The main idea was previously tested in a completely different robotics test bed [1]. Inspiring from what was stated before about the process of diagnosis, we build an artificial learning decision support system for the diagnosis of diseases or more practically as a supporting system that can provide a human with a secondary opinion in order to make more reliable decisions.

The proposed system learns – by the means of reinforcement learning [2] in a continuous space, using an efficient Bayesian Q-Learning framework [3] – to suggest more knowledgeable experts, in different situations, and the most reasonable sequence of consulting them. This consultation policy varies by situation and is not a static. Apart from the minimization of decision-making cost – which is the main goal of this research-there are also other benefits to the approach proposed in this paper from the perspective of utilizing multiple heterogeneous sources of information for making a more rational decision. Following, a number of key advantages are reviewed from [4]:

- **Statistical Reasons.** A well-known fact is that good performance on training data does not necessarily predict good generalization performance (Here, generalization is defined as the performance of the classifier on data not seen during training). A set of classifiers with similar training performances may very well have different generalization performances. This problem happens particularly if the test dataset used to test the method is not a sufficiently-good representative of the field data which the method is to be used upon. In such cases, fusing the outputs of several different classifiers by the means of averaging may reduce the risk of an unfortunate selection of a poorly-performing classifier. This one of the most important reasons we try to consult with other experts: having several doctors agree on a diagnosis, or several former users agree on the quality of a product, reduces the risk of following the advice of a single doctor (or a single user) whose specific personal opinion may be significantly different than those of others.
- **Large Volumes of Data.** In certain applications, the amount of data to be analyzed can be too large to be effectively handled by a single classifier and training classifiers with a vast amount of data is usually not practical; partitioning the data into smaller subsets, training different classifiers on different partitions of the data, and combining their outputs using an intelligent and adaptive combination rule often proves to be a more efficient approach.
- **Divide and Conquer.** Regardless of the volume of available data, certain problems are just too difficult for a single given classifier to solve. More specifically, the decision boundary that separates data from different classes may be too complex, or lie outside the space of functions that can be generated by the chosen classifier model.
- **Data Fusion.** If we have, on our hand, several sets of data obtained from various sources where the nature of data features are different (heterogeneous features), a single classifier might not be used to learn the information contained in all of the data. In diagnosing a neurological disorder, for example, the neurologist may suggest the patient to take several different tests, such as an MRI scan, an EEG recording, blood

tests, etc. Each test results in data with a different number and type of features, which cannot be used collectively to train a single classifier. In such cases, data from each testing modality can be used to train a different classifier, whose outputs can then be combined. Applications in which data from different sources are combined to make a more informed decision are a part of what is referred to as *data fusion* applications, and ensemble-based approaches have successfully been used for such applications. In [4], two key components of ensemble systems are introduced: first, a strategy is needed to build an ensemble that is as diverse as possible. Some of the more popular ones, such as boosting, AdaBoost, stacked generalization, and mixture of experts are discussed in next section. A second strategy is then needed to combine the outputs of individual classifiers that make up the ensemble in such a way that the correct decisions are amplified, and incorrect ones are cancelled out.

The main idea of this research is to prepare and train a set of local classifiers, and then design a deliberate selection mechanism with the maximum effect on the correct diagnosis rate which minimizes the consultation cost at the same time.

Following from here, the paper contains one review section which is mostly on ensemble-based systems. After the review section, we will explain the framework's structural components, its learning phases and also the evaluation measures. Then, we will introduce the case study and state how we will apply this approach on the arrhythmia dataset and report the results taken so far. At the end, the conclusions and potential future works are discussed.

2. **Review of Related Works.** Integrating different classifiers' decisions to come to a more accurate judgment has been and still is a hot topic of research. According to [5], there exists no unique name for this process: combination of multiple classifiers, classifier fusion, mixture of experts, dynamic classifier selection and classifier ensembles are among the different terms used for this purpose.

There are two basic sorts of data combination: selection and fusion [5]. In this paper, we make use of both selection and fusion: we select some local decision makers to form our local experts and then the final decision is shaped through a learning-based *mixture-of-experts* strategy. Here, it is important to point out the difference between our approach and the classic *mixture-of-experts* strategy where we tried to implicitly learn the gating network while the classical MOE uses a fixed strategy.

Another related perspective is 'Boosting': As Schapire, the pioneer of boosting methods, describes in [6]: "Boosting is a general method for improving the accuracy of any given learning algorithm. It refers to a general and provably effective method of producing a very accurate prediction rule by combining rough and moderately inaccurate rules of thumb." It is worth mentioning that different from traditional boosting algorithms (like AdaBoost [6]) which are done in a supervised manner, our boosting schema is reinforcement-based and the whole story is shaped by a reward signal from a critic. We claim that our proposed structure for decision fusion method implements boosting, as the local correct classification rate (CCR) of each local expert is less than or equal to chance, but by using the learning strategy for fusion, we will improve the performance considerably.

The above reviewed approaches are all related to decision making based on different sources of decisions. Now, from the perspective of controlling the cost while decision-making, the closest research to our work is presented in [7]. There, active learning techniques are proposed for choosing the most informative data which enabled biologists and computer scientists to optimize experimental data choices for rapid discovery of a biological function (p53 cancer rescue mutants). Our method also tries to select the more informative experts rather than the more informative experimental data choices. In other words, we solved the problem not in the feature but rather in the decision space, which is an alternative space proposed below.

3. **Proposed Method: Learning of Classification and Sequence of Consultation: LCSC.** In this section, we will explain our proposed framework in detail. The framework contains three consecutive phases: In the first phase, the input featurespace is partitioned. Each local partition is then assigned to a unique local expert to explore and learn in. In the second phase, local experts are trained on their own specified local parts of the feature space. When the training has converged and the decision of each decision-maker has took its shape, we are ready to start the final phase i.e. training of **L**earner of **C**lassification and **S**equence of **C**onsultation (called LCSC thereafter) through the continuous Bayesian Q-learning [3] in the decision space.

Figure 1 shows a schematic diagram of the whole components and the learning phases. Next, we will describe the learning phases. In every phase, the corresponding involved structural components are described in details. Finally, evaluation measures are described.

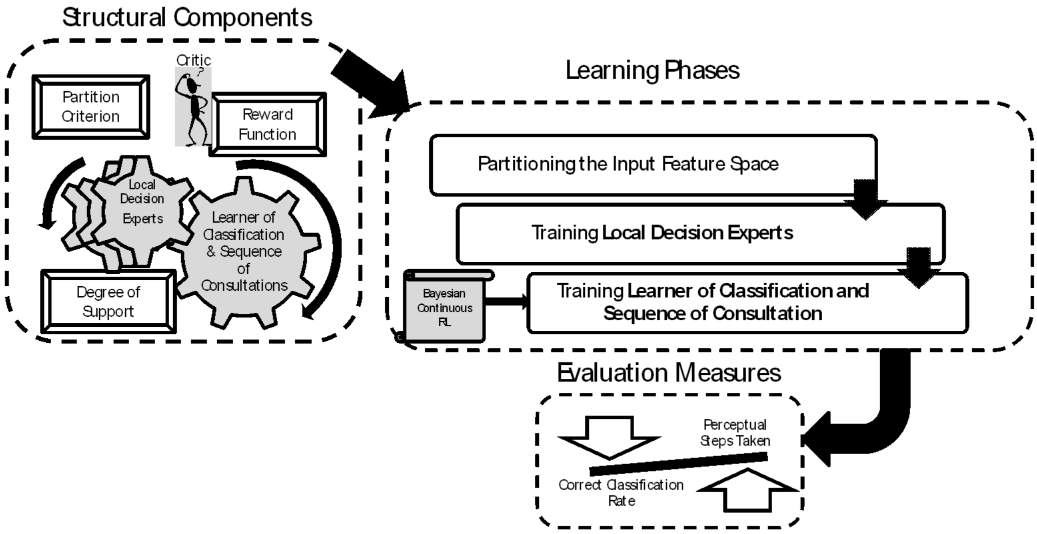

FIGURE 1. A schematic view of the components and phases of LCSC

3.1. **Phase 1: Partitioning the input feature space.** According to the principle of *Divide and Conquer,* there are three main policies for partitioning and designing local classifiers. The locality of classifiers can be achieved through partitioning of a) Input space alone, b) Output space alone, c) Input-output coupled space. For the sake of simplicity, we have chosen the first policy (lone input space partitioning) in this paper.

There exist different strategies [8] from the perspective of available design-time knowledge for finding local feature sub-spaces:

- Random Partitioning (Agnostic): In random partitioning, we randomly partition the input feature space into overlapping subspaces. Then we assign each subspace to a classifier to learn in, so to become our local expert on that part of feature space.
- Partition based on meta-knowledge about the features' semantics: In this strategy, we consider the available meta-knowledge and partition the input feature space so every set of features assigned to an individual classifier has a specific meaning or has a kind of semantic relevance of member features with each other (e.g. being extractable from the same laboratory test, being instructed by same physicians and so on).

The required steps of realizing these two simple strategies are shown in Table 1.

TABLE 1. Partition the input feature space

<table>
<tr><th>Random Partitioning</th><th>Partitioning based on the pre-knowledge</th></tr>
<tr><td colspan="2">The set of features: $F = \{f_1, f_2, \ldots, f_{|F|}\}$, Number of features$= |F|$
The set of output classes: $C = \{c_1, c_2, c_3, \ldots, c_{|C|}\}$, Number of classes $= |C|$
Assume we want to create L classifiers.</td></tr>
<tr><td>1- Rank the features according to effect on CCR using Forward Selection or Backward Elimination[9].
2- Make L bins out of sorted features each with approximately $\frac{|F|}{L}$ features.
3- Randomly select features from these bins. This way, we have designed diverse-in-feature while same-in-strength local experts.</td><td>According to the existing pre-knowledge and the semantic meaning of features, select those features that have a semantic relevance.</td></tr>
<tr><td colspan="2">We have L different portions of feature space: S_{LDE_i}, $i = 1, 2, 3, \ldots, L$</td></tr>
</table>

3.2. **Phase 2: Training local decision experts.** In the second phase, we will train our local classifiers (decision experts) whose corresponding feature sub-spaces have been specified in the previous phase. The diversity of the classifiers can be realized through the input features they look at, output classes that they try to recognize and even their learning algorithms. So, in this step:

- One should opt for a proper classification algorithm according to the different decisive factors such as general accuracy, speed of classification, tolerance to missing values, tolerance to irrelevant attributes, dealing with danger of over-fitting the data, explainability and finally dealing with discrete or continuous attributes [10]. The algorithms can be k-NNs, Bayes' optimal classifiers, SVMs or even artificial neural networks. Here, we select to use k-NN classifiers in order to prevent the dominance of the classification algorithm on our proposed combination method.
- When the classification algorithm is selected, we need to train the classifiers (called LDEs for short) on their specific local part of features to discriminate all $|C|$ classes. The subset of feature space which is observed by i^{th} LDE is shown with s_{LDE_i}.
- By the completion of the training phase, decision will be taken in the form of the *degrees of support* of each classifier for each class. The idea of *degree of support* will be discussed in the next section where degree of support of i^{th} LDE *for all classes* per its state will is shown below:

$$D_{LDE_i} = \left[P\left(Class_j | s = s_{LDE_i}\right)\right]_{j=1}^{|c|} \quad (1)$$

3.3. **Phase 3: Learning of classification and sequence of consultations (LCSC).** The novelty of our approach lies mainly in this phase which we make use of Reinforcement Learning (RL) [11] in a continuous state-space dubbed the decision space[1]. Decision space is defined as:

[1]A comparison between learning in direct perceptual space and also in decision space is proposed in [12].

Decision space is an alternate space formed by combining the degree of support of all classes from the perspective of every classifier (the one which has been already decided to be consulted).

The reason for utilizing RL as the learning core is our need for an optimization method which can learn in concert with costs per every action. It means that we want LCSC to learn how to diagnose the specific class of each patient's disease along with the learning of identification of consultations which are more helpful in reaching such a judgment. Therefore, two types of actions should be considered for LCSC:

- *Information Acquisition actions (IA Actions)* are equivalent to the consultation with an expert. They are performed solely to gather more information when LCSC lacks enough information.
- *Diagnosis Declaration actions (DD Actions)* are equivalent to declaring the final judgment, i.e. the detected class. They are performed when LCSC needs no more information and actually enough experts have already been consulted.

To solve this learning problem, we first make a key assumption: CLCS can be casted in an MDP[2]. Thus, we employ reinforcement learning in a continuous space or in fact an efficient Bayesian Q-Learning approach [3]. Before we explain the training algorithm, we need to explain the basic components of the MDP framework represented in the form of the 4-tuple (S, A, T, R) shown in Table 2.

TABLE 2. The elements of an assumed MDP to formulate LCSC

States (S)	$S_{LCSC} = \bigoplus_{i=1}^{L} (D_{LDE_i} \,\|\|\, [0]_{1\times\|C\|})$ L = number of Local Decision Experts D_{LDE_i} = the degree of support from ith local decision expert's perspective: $D_{LDE_i} = [P(DD\_Action_j \| s_{LDE_i})]_{j=1}^{\|C\|}$. |C| = the number of classes that can be announced as final decision. $\oplus$ is the element-by-element sum operator that adds the degree of support of each class on different classifiers. || is bitwise OR. If one local decision expert is consulted, its degree of support is combined with that of others already consulted. Otherwise, the state is kept unchanged.
Actions (A)	$A_{LCSC} = DD\_Actions \cup IA\_Actions$ $DD\_Actions = C =$ *the set of possible class labels* $IA\_Actions = \{Consult\ LDE_1, \ldots, Consult\ LDE_L\}$
Transition Function (T)	$S^{next}_{LCSC} = S^{current}_{LCSC} \oplus D_{LDE_i}$, *if action* $\in$ *IA_Action* $(action = ConsultLDE_i)$ $S^{next}_{LCSC} =$ *Terminal, if action* $\in$ *wrong DD_Actions* $S^{next}_{LCSC} =$ *Goal, if action = Correct DD_Actions*
Reward Function (R)	R = High Positive, *if action = Correct DD_Actions* R = High Negative, *if action* $\in$ *wrong DD_Actions* R = (Low Negative) $\times$ (number of consultations have already done), *if action* $\in$ *IA_Actions*

The approach we employed to construct a state variable for LCSC out of local classifiers' decisions is very similar[3] to *Decision Profile* proposed in [5]. The main idea of such a

[2]An MDP environment assumes that the next state of an agent is solely dependent on the current state and does not depend on the information encoded in previous states.

[3]The difference is that we combined the decision vectors using Σ operator rather than making an extended matrix.

representation is that the continuous outputs provided by a classifier for a given class can be interpreted as the degree of support given to that class where it is usually accepted as an estimate of the posterior probability for that class. We used this approach to represent the decision of LCSC.

The learning algorithm of LCSC is as follows: The ***training phase*** starts with a null state for CLCS. An IA-Action is performed at the beginning. Either a random or a general decision make[4] is consulted at the beginning to find a rough view of the state CLCS is in. In the agnostic partitioning version of the approach, an arbitrary expert by chance is consulted where we use a general practitioner is consulted with in the pre-knowledge-based version of the approach.

Next, CLCS tries to make a decision. If it decides to consult another expert, the state is updated accordingly and updating of the knowledge in our Bayesian Q-Learning core takes place according to both the cost of such consultation and also that of the information which is acquired in return. It is noticeable that according to the reward function shown in the last row of Table 2, we have tried to make CLCS to consult as few experts as possible. As the number of consultations increases, the amount CLCS has to pay rises accordingly.

On the other hand, if the decision is of diagnosis declaration type, the decision is evaluated by the critic and the reward/punishment is assigned in return. Again, updating of the knowledge takes place, but this time with the new reinforcement signal received for the selected DD-Action. After performing a diagnosis declaration action, one step of learning is finished.

Now, let's explain the ***Stop Criterion***. The algorithm continues until all training samples are observed once.We repeat the process for reasonable number of training epochs. When an epoch is finished the error on the validation set is evaluated. The training process is stopped when the error on validation set has increased for consecutive epochs.

3.4. **Evaluation measures.** In addition to the correct classification rate which evaluates the optimality of the decision in the classification domain, we can also evaluate the approach in terms of number of local experts selected for consultation to reach such a decision as well. Naturally, we prefer to pay the least possible cost for consultation while being able to make a truthful decision. If we have more information about the design of local experts on hand (as in pre-knowledge based approach) we can even evaluate the distribution of consultations (i.e. rejected classifiers due to their lack of knowledge or those who found more knowledgeable). The number of consulted local experts implicitly shows the number of features involved in the decision making process[5]. Reduction of required features means a potential benefit.

4. **Case Study: Realizing LCSC on Arrhythmia Recognition.** In this research, we used 452 ECG recordings from the UCI "Arrhythmia" dataset [13]. This dataset includes about 0.33% missing attribute values and 22 unclassified instances (unknown). These shortcomings show that the prediction accuracy of any approach on this dataset cannot be perfect, however, these also make this dataset more comparable to the dynamic real-world situations. Each record consists of a set of clinical parameters measured on ECG signals and some personal information about the subjects. The dataset is divided into two broad groups, labeled as Normal (regular heartbeat) and Abnormal (arrhythmia). The abnormal group itself is divided into 14 finer classes. The ECG signals includes 5

[4]E.g. a general practitioner.

[5]*Number-of-features-involved=Avg. number-of-local-experts-consulted (after learning)* $\times$ *Avg. Number-of-features-considered-by-each-local-expert*

parameters: QRS duration, PR interval, QT interval, T interval and P interval. The personal information available includes age, height, weight and sex. There are 245 cases in the normal group and 207 cases in the abnormal group.

In order to realize the approach on this dataset we need to realize all three phases of our approach which are described in Table 3.

Table 3. Steps of LCSC implementation on arrhythmia

<table>
<tr><th colspan="2">Phase 1: Partitioning the input feature space</th></tr>
<tr><th>Random Partitioning (Agnostic)</th><th>Partitioning based on Pre-knowledge</th></tr>
<tr><td colspan="2">The set of features: $F = \{f_1, f_2, \ldots, f_{279}\}$, Number of features $= |F| = 279$
The set of output classes: $C = \{c_1, c_2, \ldots, c_{16}\}$, Number of classes $= |C| = 16$</td></tr>
<tr><td>Rank the features according to the effect on CCR using Forward Selection (with k-NN).
Make 20 Bins out of sorted features each with approximately 14 features.
Assume we want 30 classifiers to make redundancy. So, select 20 features from these 20 Bins randomly (to bring locality) to assign to these local classifiers.</td><td>Features to be considered by Expert1 (General Practitioner) = age, sex, Height, Weight, QRS duration, P-R interval, Q-T interval, T interval, P interval, QRS, T, P, QRST, J, Heart rate.
Features to be considered by other 12 experts (Expert2 to Expert13): the rest of features[6] recorded from 12 channels (DI, DII, DIII, AVR, AVL, AVF, V1, V2, V3, V4, V5 and V6).</td></tr>
<tr><td colspan="2">$L_{Agnostic} = 30$
$L_{Pre-Knowledge} = 13$</td></tr>
<tr><th colspan="2">Phase 2: Training Local Decision Experts</th></tr>
<tr><td colspan="2">We have to train L local decision experts in this phase. We have selected the k-NN classification algorithm. So, we find the optimal K for each of L individual local decision experts using the validation set. With their optimal K, these act as weak classifiers with the CCRs slightly better than chance.
The outputs of local k-NNs are transformed into the probabilistic form (in order to represent them in the form of the degree of support.)</td></tr>
<tr><th colspan="2">Phase 3: Training Learner of Classification and Sequence of Consultations</th></tr>
<tr><td colspan="2">The same setting as shown in Table 2 , with
|C| = 16, L = 30 (agnostic)
|C| = 16, L = 13 (pre-knowledge)</td></tr>
</table>

The results taken on this dataset is displayed in Table 4. We divided the whole dataset into three parts for training ($\sim$80% = 353 instances), validation ($\sim$10% = 47 instances) and test ($\sim$10% = 52 instances). We have repeated the whole training algorithm (including partitioning step, training local classifiers, training LCSC) for 5 times and averaged the result. The other benchmark approaches evaluated on the same dataset are also studied and reported here to measure how efficient our approach has act on the problem.

Table 4 shows that LCSC, although unable to outperform EDBND and AdaBoost, works almost as good as ensemble-based approaches [15] that have been evaluated on this dataset. Furthermore, it even adds a number of benefits: Its first benefit is the reduction of number of consultations and thus number of features used for recognition as learning

[6] *Q wave, R wave, S wave, R' wave, S' wave, Number of intrinsic deflections, Existence of ragged R wave, Existence of diphasic derivation of R wave, Existence of ragged P wave, Existence of diphasic derivation of P wave, Existence of ragged T wave, Existence of diphasic derivation of T wave, JJ wave, Q wave, R wave, S wave, R' wave, S' wave, P wave, T wave, QRSA, QRSTA.*

TABLE 4. Benchmarking the results of CLCS

Approach		Results (averaged in 5 runs)		
Algorithm	Description	Final Number of classifiers consulted (Test)	Correct Classification Cate	
EDBND	Ensembles of Balanced Nested Dichotomies for Multi-Class Problems [14]	–	72%	
AdaBoost	Described in [15]	–	71%	
Ensemble-based	Ensemble of kNN and SVM classifiers [15]	–	68%	
LCSC	Agnostic	$25 \rightarrow 14 \rightarrow 8 \rightarrow 16/30$	Test (Avg): 68% Min: 65%, Max: 70% std = 4.3×10^{-4}	Train (Avg) = 81%
	Pre-knowledge	$12 \rightarrow 10 \rightarrow 5 \rightarrow 7.7/13$	Test (Avg): 66% Min: 63%, Max: 68% std = 4.3×10^{-4}	Train (Avg) = 75%

progresses. In addition, the consultation policy is not fixed as is in common decision fusion methods; rather, it is learnt based on the observed situations. Another point to mention is that the number of required epochs to complete the learning process (before the stop criterion condition is met) is 4. This indirectly implies acceptable learning speed of LCSC. Furthermore, as it is shown in Table 4, the low standard deviation of LCSC results is a sign of its robustness to the applied splitting mechanisms into training/validation/test parts.

Also, another key point to add which has opened a path to the future steps is that the pre-knowledge based partitioning strategy could not outperform the agnostic design strategy. This can be considered as a sign of inadequate exploited knowledge on partitioning mechanism which will be discussed more in the future works.

5. **Conclusions.** In this paper, we proposed a three-phased learning approach named LCSC in the category of ensemble-based systems. We tested its applicability in Arrhythmia recognition. It concurrently learns two coupled issues: primarily the classification task and at the same time the sequence of the most beneficial consultations that help to reach formed decision. There are different advantages for LCSC. We can categorize the benefits in some broad classes from various perspectives:

- High dimensionality in feature space: in order to come up with the size of feature space, our approach utilizes the principal of *divide-and-conquer*. Therefore, the big problem is split into small manageable pieces of decision making problems with reasonable number of features.
- Non-homogeneity in learning algorithms of decision makers: the common language of decision space (i.e. degree of support) we utilized in the proposed framework is the main key to handle non-homogeneity in learning algorithms. This brings another major merit: Utilization of different possible sources of decision information (which can range from a naïve classifier to a human-level expert consultation) is thus made possible. If there exists a source of information (for the problem at hand) that can communicate in decision language we can also include it in the set of our local decision experts.
- From the perspective of representation in an alternative space (here, the decision space instead of the perceptual space), our approach can be considered as a clustering

method which makes a granular and compact decision space. In fact, CLCS is able to potentially form soft decision boundaries like fuzzy C-means [16] does.

- From the fusion perspective, it does not need a hand-designed strategy for combination of decisions. Instead, it learns based on the situation, which decisions to consider and how to purposefully combine them. This is done by considering reasonable costs for each action and employing continuous Q-learning. Two similar learning-based studies albeit with different goals have been proposed in [17,18]. In [17] an iterative decision method for optimal multiuser detection is given and in [18] a Q-learning based method for container transfer scheduling has been proposed. Both these studies are based on learning and do not act in a hand-designed manner.
- From the perspective of classification according to measures proposed in [10], we can mention some benefits for LCSC. It can deal with continuity of decision space using the efficient core of Bayesian framework. Moreover, its transparency of knowledge (i.e. explanability) is also an added superiority. For every new instance, it can give not only the output class label but also the sequence of consultations to reach such judgment. Last but not the least is that LCSC tries to keep its consultation ratio minimized which implicitly affects on reducing the number of features involved in the decision making process.

6. **Future Works.** As future steps, we are going to rectify some of the deficiencies in our approach. The most critical one is that when the number of output classes or the number of our local experts increases, the state representation tends to be inefficient. Therefore, to test our approach in the problems with higher number of output classes, we need to test if we can extract the implicit existing hierarchy in the goal space. A method to tackle this issue is proposed in [19]. This hierarchy can be extracted and used in the detailed design of local decision experts. This seems to be helpful in compacting the decision space, speeding up the learning process and thus improving the results.

Another issue is to test the approach in another domain with non-homogenous feature spaces. This actually brings about the necessity of other more powerful learning algorithms; such as neural-based methods or SVMs. Moreover, by exploiting the medical knowledge on the feature grouping mechanisms we are to design more helpful consultants thus improve the performance of the proposed learning approach.

Last but not least is to verify the robustness to the detailed design of classifiers. To do so, we have added misleading classifiers to test if CLCS can detect them or not. This is the focus of our current research to improve the proposed method's robustness to such un-avoidable design time problems.

REFERENCES

[1] M. S. Mirian, H. Firouzi, M. N. Ahmadabadi and B. N. Araabi, Concurrent learning of task and attention control in the decision space, *Proc. of the IEEE Advanced Intelligent Mechatronics*, Singapore, pp.1353-1358, 2009.

[2] R. S. Sutton and A. G. Barto, Reinforcement learning, *Cognitive Neuroscience*, vol.11, pp.126-134, 1999.

[3] B. N. Firouzi, M. N. Ahmadabadi and B. N. Araabi, A probabilistic reinforcement-based approach to conceptualization, *International Journal of Intelligent Technology*, vol.3, pp.48-55, 2008.

[4] R. Polikar, Ensemble based systems in decision making, *IEEE Circuits and Systems Magazine*, vol.6, pp.21-45, 2006.

[5] L. I. Kuncheva, J. C. Bezdek and R. P. Duin, Decision templates for multiple classifier fusion: An experimental comparison, *Pattern Recognition*, vol.34, pp.299-314, 2001.

[6] R. E. Schapire, The boosting approach to machine learning: An overview, *Lecture Notes in Statistics*, pp.149-172, 2003.

[7] S. A. Danziger, J. Zeng, Y. Wang, R. K. Brachmann and R. H. Lathrop, Choosing where to look next in a mutation sequence space, *Bioinformatics*, vol.23, pp.i104-i114, 2007.

[8] R. Ebrahimpour, E. Kabir, H. Esteky and M. R. Yousefi, View-independent face recognition with mixture of experts, *Neurocomputing*, vol.71, pp.1103-1107, 2008.

[9] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, New York, 2003.

[10] S. B. Kotsiantis, Supervised machine learning: A review of classification techniques, *Informatica*, vol.31, pp.249-268, 2007.

[11] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, The MIT Press, London, 1998.

[12] M. S. Mirian, M. N. Ahmadabadi, B. N. Araabi and R. R. Siegwart, Comparing learning attention control in perceptual and decision space, *Lecture Notes in Artificial Intelligence*, pp.242-256, 2009.

[13] C. L. Blake and C. J. Merz, *UCI Repository of Machine Learning Databases*, 1998.

[14] L. Dong, E. Frank and S. Kramer, Ensembles of balanced nested dichotomies for multi-class problems, *Lecture Notes in Computer Science*, vol.3721, pp.84-95, 2005.

[15] N. García-Pedrajas, Constructing ensembles of classifiers by means of weighted instance selection, *IEEE Trans. on Neural Networks*, vol.20, 2009.

[16] Y. Kusunoki, M. Inuiguchi and J. Stefanowski, Rule induction via clustering decision classes, *International Journal of Innovative Computing, Information and Control*, vol.4, no.10, pp.2663-2677, 2008.

[17] W. Liu and S. Ding, *Performance Analysis of the Iterative Decision Method for Optimal Multiuser Detection*, vol.3, 2009.

[18] Y. Hirashima, A Q-learning system for container transfer scheduling based on shipping order at container terminals, *International Journal of Innovative Computing, Information and Control*, vol.4, no.3, pp.547-558, 2008.

[19] Y. Wang, Fuzzy clustering analysis by using genetic algorithm, *ICIC Express Letters*, vol.2, no.4, pp.331-337, 2008.