2018 Joint IEEE International Conference on Development and Learning and
Epigenetic Robotics (ICDL-EpiRob)
Tokyo, Japan, September 16-20, 2018

Partitioning Sensorimotor Space by Predictability Principle in Intrinsic Motivation Systems

Melisa Idil Sener and Emre Ugur
Computer Engineering Department
Bogazici University
Istanbul, Turkey
Email: melisa.sener, emre.ugur@boun.edu.tr

***Abstract*—Inspired by infant development, intrinsic motivation (IM) guides the robot with intelligent exploration strategies, enabling efficient and effective learning in high-dimensional search spaces. A particular method in IM, namely Intelligent Adaptive Curiosity (IAC), adaptively partitions agents sensorimotor space ($\mathbb{SM}$) into regions of exploration, and guides the agent to select the regions that are in the moderate level of difficulty, and learns separate experts for different regions. Therefore, the means of partitioning the $\mathbb{SM}$ and the mechanisms behind region generation is of utmost importance. In this study, we propose a method for partitioning the space that allows maximizing the performances of the experts that will be responsible for learning skills. In brief, for each potential partitioning, the error of the experts are calculated and the partitioning that would generate the minimal error in the future is selected. Our method is evaluated in a setting with a simulated robot that learns predicting the next state given the current state and the action taken in an environment composed of regions with different properties. We verified the proposed method, $\mathbb{SM}$ is partitioned into more semantically meaningful regions adapting environment dynamics, the exploration of the robot in these regions can better exploit IM mechanisms and the system learn more efficiently and effectively i.e. with higher performance in a shorter time, compared to a baseline method.**

***Index Terms*—intrinsic motivation, autonomous mental development, reinforcement learning, active learning, developmental robotics**

I. Introduction

For many years, scientists generally have followed three main approaches to build intelligent systems. In the first one, an intelligent system is directly programmed to perform a given task. In the second, the computer is provided human-edited sensory data and runs a learning program specific to the task. Finally in the last, intelligent systems evolved by the principle of "survival of the fittest", i.e. the most competent races left their survival skills to successive generations [1].

Survival depends on lifelong learning and application of what has been learned. IM is regarded as a set of active learning mechanisms for developmental robots, improving learning in high-dimensional search spaces. Since IM demands the development of broad competence rather than immediate external goals [2], IM is a part of continuous and high-quality learning [3]. Autonomous mental development concept is defined as developing mental capabilities under the control of a learning agent's own developmental program, via the autonomous real-time interactions with the external environment with the agent's own sensors and actuators as well as its own internal environment with time [1]. Originated from the fact that IM mechanisms generate learning signals by observing the skills or knowledge level needed to be acquired by the agent [4], autonomous mental development concept were tried to be adapted to learning machines. Contrary to manual development involving running a program for a specific task with hand-engineered representations, autonomous development consists of two main phases. The first one (construction and programming phase), a developmental program is formed, controlling the autonomous development of the agent and not related a specific task and the agent's body is designed according to its operation environment. The beginning time of the execution of this program is considered the time that the agent was "born" and the second phase begins. In the second one (autonomous development phase) the agent starts interacting with the physical world and develops the skills required in that environment. Skills acquired early are used later by the robot to support the learning of new tasks [5].

Efficient and effective learning in high-dimensional spaces is hard and can be simplified by splitting $\mathbb{SM}$ into smaller regions. The regions with similar characteristics can be generalized effectively by an expert responsible only for these regions. Distributing learning task across the experts of smaller regions provides more accurate results and thus improves the overall quality of the learning. In order to make these experts proficient in their local regions, $\mathbb{SM}$ should be split wisely. A particular method of IM, namely IAC, provides a smart splitting scheme in such a high-dimensional $\mathbb{SM}$ and drives the learning agent to explore the regions by considering the competence level of the agent. The essential point that forms the core of study is the decision of how to split $\mathbb{SM}$ into specialized learning regions. Reflecting environment dynamics to the learning space and formation of child regions should be determined by the previously collected experience of to be split region. Thus, to determine the regions to be formed after the split, we are considering the potential success rates of the candidate regions. Our study performed better than the original study [6] (we will also briefly explain their idea in this paper) doesn't consider how the experts would perform in the generated sub-regions, in an experimental setting simulating the interaction of a robot with a simple environment. In this study, we propose a novel method to split the learning space into easy to learn regions and provide a more accurate way

of calculating intrinsic reward that the agent gets. As a result, our approach considers the future aspects of splitting process and reflects a more distilled way of using IM for exploration.

II. Related Work

The general structure of IM studies is composed of two main modules. According to Oudeyer et al. [6], the first module is defined as a learning machine **M** predicts the sensorimotor consequences of the execution of a given action in a given context and the second module is defined as a meta-learning machine **metaM** predicts the errors that machine **M** makes in its predictions. In [6] classify the existing methods according to their action selection mechanisms:

- **Error Maximization:** In this type of systems, (e.g., [2]) the action that **metaM** predicts the largest error in prediction of **M** then the robot directly use this prediction to choose which action to do.
- **Progress Maximization:** In this type of systems, (e.g., [7], [8]), in addition to **M** and **metaM** there exists a third module called **Knowledge Gain Assessor (KGA)**. **KGA** predicts and stores the mean error rate of **M** in the close future. The action, leading to the greatest decrease of the difference between the expected mean error rate of the predictions of **M** in the close future and the mean error rate in the close past, is selected.
- **Similarity-Based Progress Maximization:** In order to make Progress Maximization method work well in the real-world, the similarity between the situations should be considered. In this approach, the agent should compare its current error rate with the recent error rate of similar situations encountered before. A robot that employs this approach should be able to distinguish between the situations and relate them in their contexts by itself. Such systems (e.g., [6], [9]) are able to distinguish between various kind of activities and decide which one of them to be explored next.

In Schmidhuber's study [9] a curious model building system is defined as a system that directs the agent to the situations that it expected to learn some capabilities or knowledge from that context. That model is developed based on Watkins' Q-Learning algorithm [10]. Curiosity is defined as the desire for improvement of a system predicting the reactions of the environment and realized by reinforcement learning. Since the model is adaptive, predictions will be improved by the time. As a result, the agent will stop discovering regions that it can predict well and will move on to parts of the environment which haven't been explored yet.

Some of the studies in developmental robotics studies use IM as a supporting module works along with their main modules. For example, [4] incorporates Cognitive Sciences and Computer Vision fields and aimed the development of overt attention skills. In the experiment, they excepted the eye gaze of the agent to be directed to the relevant areas in the scene. In this study, they used artificial neural networks.

In Ugur's study [11], it is aimed to hierarchically structuring the affordances of the objects with different levels of complexity. In the study, to learn object affordances, object selection action was performed using a heuristic to increase the object variety. Thus, in each iteration robot continues the exploration by choosing the most "interesting" actions among the candidates and improves the learning progress (LP). In [12] they proposed a system that learns qualitative representations of states and predictive models in a bottom-up manner by discretizing the continuous variables. Konidaris et al. studied construction of symbols that are directly used as preconditions and effects of actions for generation of deterministic [13] probabilistic [14] plans in simulated environments.

Recently, in Forestier's [15] study, intrinsically motivated goal exploration processes (IMGEP) introduce self-generation of goals as parametrized problems and, allow the agent to automatically build a learning curriculum. IMGEP is a powerful framework that allows discovery of skills of increasing complexity without explicitly defining a goal. From this study, the idea of goal parametrization can be employed to our approach and our way of splitting the learning space may leverage the goal parametrization procedure.

While some of the related studies apply **IAC** to pre-split sensorimotor regions, the others [11],[15] apply autonomous split mechanisms by using similar approaches to the **IAC**. Our method belongs to the latter category and differs from the existing methods by splitting $\mathbb{SM}$ according to the success of the learning system and potential error rate. In particular, method in [6] belongs to similarity-based progress maximization group and is built upon the comparison of LP. Our proposed method is built upon the comparison of LP too but it differs by the discernment mechanism of the similar sensorimotor states.

III. Method

The proposed method is built on the **IAC** framework in [6]. In III-A we give a brief introduction to **IAC**, in III-B we formalize the input and the output of the system, in III-C we explain the method given in [6] and explain our approach of splitting $\mathbb{SM}$, in III-D we describe learning machines **M** and their contribution to splitting process, then in III-E we provide the formal explanation of LP and finally in III-F we clarify the action selection mechanism.

A. Summary

Intelligent adaptive curiosity (**IAC**) proposed by [6], adaptively splits $\mathbb{SM}$ into regions and uses LP of these regions for deciding which region to explore and learn next. Regions with large LP are primarily explored and learned. Since LP would be low in problems which are too easy or too complex or impossible to learn, it automatically works on problems that have moderate complexity before dealing with simple and hard learning problems. Thus, robot's actions become more complex gradually and developmental sequence organizes itself.

The flow of the **IAC** algorithm can be summarized as follows: Each experience encountered by the robot is recorded to the memory of the system as a vector (we'll call them as "exemplar" in order to be coherent with **IAC** [6]). An exemplar is a couple of current sensorimotor state and its outcome in sensory space $\langle SM(t), S(t+1)\rangle$. $\mathbb{SM}$ continuously split into regions when any region met C criterion. Note that C can be

any condition depending on the application, here we used a threshold value for the number of exemplars that a region is allowed to contain as in [6]. Each region has its own **M** and this machine is responsible for predicting the next sensory state given current sensorimotor state covered by that region. Each **M** is trained with the corresponding region's exemplars and when a prediction should be made, the **M** covering that exemplar is selected and used for the prediction. After the execution of the action in the given sensorimotor context, the difference between the actual outcome and the prediction is calculated and recorded into the corresponding region's error list. Afterward, this list is used for the evaluation of LP of that region. LP is the core of the IM in the system and used for the determination of the action which contribute most to the learning process.

B. Format

The input of each **M** is a vector $SM(t)$, the concatenation of current sensory state $S(t)$ and motor parameters $M(t)$ of the robot. Based on the current sensorimotor state $SM(t)$, **M** learns to predict the next sensory state $S(t+1)$.

C. Splitting the Sensorimotor Space

We aimed to improve the learning performance of the **IAC** by splitting $\mathbb{SM}$ according to the predictability principle. **IAC** splits $\mathbb{SM}$ and the mechanism behind this division is the essential part of our study. In our method, before the actual split performed, the regions are hypothetically split into two parts a predefined number of times by considering each $SM(t)$ dimension (feature) and potential learning success of each hypothetical region is calculated. This process clarifies our idea of determining which feature dimension and value will be used in that region's splitting procedure.

Splitting procedure can be summarized as follows: At the beginning, only one region (R_0) exists and when it meets C criterion, it's split into two new regions. This way, each region when it satisfies the C condition, is split into two child regions and stores the feature dimension used for splitting along with the corresponding cutting value in itself. Since the cutting dimension of a region corresponds to a feature of SM vectors, when a prediction is to be made, the corresponding region can easily be found by using the cutting dimension and value stored in regions. After splitting, exemplars contained by a parent region distributed across its children by considering the cutting information of the parent. For example: if the selected cutting dimension is the motor command and the determined cutting value is 0.5, all the exemplars inside the left child region would have their motor command value below 0.5 while all the exemplars inside the right region would have their motor command value above 0.5.

In the rest of this paper, we will refer our method based on the potential error calculation as **PE-IAC** and the method proposed in [6] based on variance as **V-IAC**.

V-IAC splits the set of exemplars into two sets so that the sum of the variances of $S(t+1)$ components of the exemplars of each set, weighted by the number of exemplars of each set, is minimal. Detailed explanation of it can be found in [6].

Our proposed method **PE-IAC** first hypothetically splits exemplar set into two by each feature dimension of SM vector predefined number (chosen arbitrarily) of times. From the hypothetical pair of regions, the pair with the lowest total potential error is selected. Thereby, instead of **V-IAC**, which does the splitting according to feature distribution in exemplars, splitting in **PE-IAC** is done by taking potential successes of candidate regions into account. Let each exemplar $SM(t)$ is a vector with length l and the decision how to split the region is done by following steps:

- Each parent region's exemplar set is sorted for each dimension index j by considering only that dimension.
- The sorted set of exemplars are split into two from different cutting values. Each hypothetical child region's **M** is trained with the set of exemplars contained by that region and corresponding errors are calculated. This process is executed incrementally. Sum of the errors of each child region is divided into the length of the length of the exemplar set and minimum of these two values is taken and stored as $pe_{j,i}$ (potential error by splitting j^{th} dimension from its i^{th} cutting value). From all the calculated error rates for that dimension $PE_j = \{pe_{j,1}, pe_{j,2} \dots pe_{j,i}\}$, the smallest value is selected $pe_j = \min(PE_j)$ and corresponding cut value is stored.
- Smallest potential error rate from all dimensions is selected and corresponding cutting dimension and cutting value is used for actual splitting.

D. Learning Machines (M)

Each region has a learning machine that is trained by that region's own exemplars. **M** of a region is responsible for prediction of next sensory state given sensorimotor input covered by the region. Any machine learning algorithm can be used for implementing **M**. For the sake of the integrity of the system, the same algorithm could be used for all **M** inside it. In this paper, selection of the learning algorithm doesn't depend on the method and any algorithm that is compatible with the given learning task could be used. In [6], **K-Nearest Neighbor(K-NN)** [16] is used as the regression algorithm. When a region is split, new regions' **M** can't use directly their parent's **M**. Thus, after each split, newly generated child regions should train their own **M** with their own set of exemplars.

The second main contribution of our proposed method is that each new **M** is trained by the exemplars of the corresponding region one-by-one and forms its own error list. Therefore, different from **V-IAC** where each child inherits parent's exemplars, in our method each child region and its corresponding expert considers only the errors made only by itself.

E. Calculating Learning Progress (LP)

Calculation of LP of each region is computed from its error list as in [6]. Let $S'(t+1)$ denote the prediction, $S(t+1)$ denote the actual outcome of $SM(t)$ vector and $e_n(t+1)$ denote the error. Then the error is mathematically:

$$e_n(t+1) = ||S(t+1) - S'(t+1)||^2 \quad (1)$$

Region R_n's error list E_n will be consist of:

$$e_n(t+1-\phi), e_n(t+1-\phi+\omega_1), e_n(t+1-\phi+\omega_2), \ldots, e_n(t+1)$$

Here $e_n(t+1-\phi)$ denotes the error of first exemplar covered by that region and inherited from the parent. Since the exemplars inherited from the parent doesn't follow a regular time pattern, $e_n(t+1-\phi+\omega_1)$ denotes the next exemplar covered by that region and $e_n(t+1)$ denotes the most recent prediction error.

LP of a region is calculated by taking the smoothed derivative of closest error curve and smoothed derivative of older closest error curve. Let θ denote the smoothing parameter and τ denote the time window parameter, mathematically :

$$\langle e_n(t+1)\rangle = \frac{\sum_{i=0}^{\theta} e_n(t+1-i)}{\theta} \quad (2)$$

$$\langle e_n(t+1-\tau)\rangle = \frac{\sum_{i=0}^{\theta} e_n(t+1-\tau-i)}{\theta} \quad (3)$$

$\langle e_n(t+1)\rangle$ and $\langle e_n(t+1-\tau)\rangle$ denote the smoothed derivative of closest errors and older closest errors respectively. The actual decrease in the prediction error rate is denoted by $D(t+1)$ and the LP calculated by:

$$L(t+1) = -D(t+1) = \langle e_n(t+1-\tau)\rangle - \langle e_n(t+1)\rangle \quad (4)$$

F. Action Selection

In IM systems, action selection is done by maximizing the intrinsic reward that the agent gains from executing the corresponding action. In our problem, since $\mathbb{SM}$ is continuous, next candidate $SM(t+1)$ vector is selected by random sampling inside this space. In a set consisting 100 sampled exemplars, each sample's corresponding region is found and LP of these regions are compared. With ϵ-greedy action selection mechanism, the sample covered by the region with the largest LP is selected and used as the next sensorimotor input of the system. Next, the input is executed, results are observed and the system is updated. **PE-IAC** and **V-IAC** use the same mechanism to calculate LP and to select the action.

IV. Experiment Setup

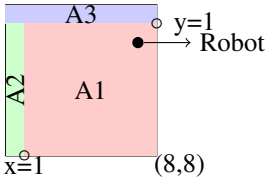


Fig. 1. Experiment environment is a 8×8 2-D environment consists of areas with different characteristics. Area-1 (A1) is slippery, Area-2 (A2) is sticky and Area-3 (A3) is completely random.

In the experiment setting, a robot that moves in an 8×8 2-D environment is simulated as shown in Figure 1. Experiment environment consists of three sub-areas with each of them has a different characteristic. Robot's actions' consequences depend on the area it stands and frequency of the sound emitted by the robot. The robot moves in vertical and horizontal directions and motor commands of this movement, namely h, v defined in $h, v \in \mathbb{R} \mid -1 < h, v < 1$. Without considering the effect of the area that the robot stands and the frequency of the sound, if the robot's horizontal motor command $h = 0.5$ and its x position at time t is $x = 1.2$ then at time $t+1$ the robot would be in $x = 1.7$ by executing the given action. Furthermore, it emits a sound with frequency $f \in \mathbb{R} \mid 0 \leq f < 1$ and the frequency affects the interaction of the robot with the environment. Without considering the effect of the area that the robot stands, if the emitted sound frequency value is $f1 \in [0, 0.33)$, reverse of the motor commands are executed (i.e., $\langle h = 0.1, v = 0.4\rangle \mapsto \langle h = -0.1, v = -0.4\rangle$. If it is $f2 \in [0.33, 0.66)$ regardless of their value, executed as $h = 0, v = 0$. If $f3 \in [0.66, 1)$ then the commands executed as they are.

After considering the effect of sound frequency on robot's interactions, h, v commands executed depending on the area it stands, i.e. for A1, multiplying both with 3; for A2 dividing both with 2 and for A3 movement of the robot will be completely random. Changes in the x, y position of the robot are calculated by adding h, v commands to current x, y position of the robot. In this setting $M(t) = (h, v, f)$ consists of horizontal speed, vertical speed and sound frequency respectively. Sensory vector consists of robot's x, y position: $S(t) = (x, y)$. In brief, robot maps the next sensory state to the current sensorimotor input: $SM(t) = (h, v, f, x, y) \mapsto (x', y') = S(t+1)$

A. Learning Flow

At time t, N possible SM vector is introduced by the system. Except for the time before the first split and ϵ-greedy action selection rule, next action is determined by IM namely, LP of the regions. (N depends on the environment dynamics and here we used $N = 100$.) Next, the robot makes a prediction about $S(t+1)$ with the given SM vector. To make it, the system finds the responsible region of that exemplar and the region's **M** is used for the prediction of the execution of this vector's outcome. And finally, this SM vector is stored in that region's exemplar set.

B. Experiment Parameters

The system is trained with 5000 iterations and the required number of exemplars for the splitting condition is 1000 (C condition) i.e. when a region collects 1000 exemplars, it is split into two new regions. For ϵ-greedy action selection rule, $\epsilon = 0.3$, for calculating LP smoothing parameter $\theta = 30$ and time window parameter $\tau = 5$ is used. Furthermore, to determine splitting dimension and value, 10 different split locations were used when computing the error rates of hypothetical regions. Results are compared with **V-IAC** by using the same parameters.

V. Results

We evaluated our method by analyzing the experimental results according to following: how well the autonomously generated sensorimotor regions reflect the underlying experimental setup (Section V-A); and whether our system allowed efficient and effective learning by analyzing the change in

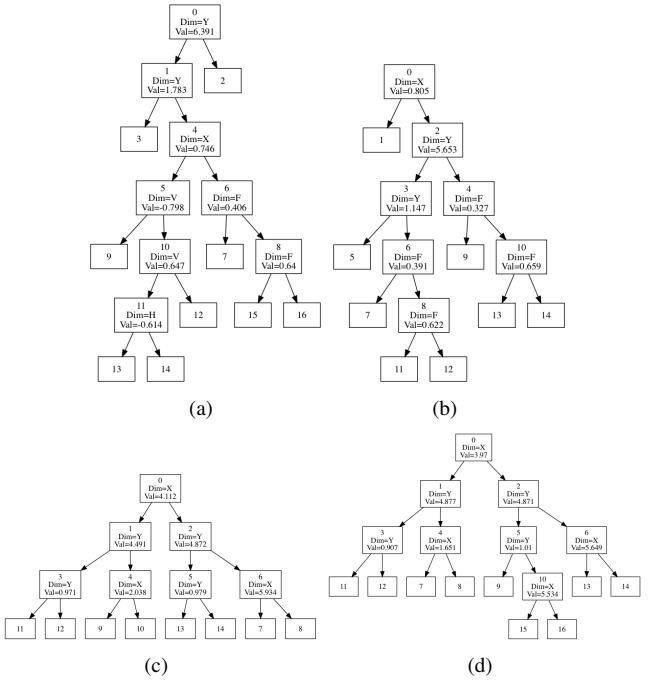


Fig. 3. $\mathbb{SM}$ region tree formed by **PE-IAC** (3(a),3(b)) and **V-IAC** (3(c),3(d)). Here first, second and third line corresponds to the ID, cutting dimension and cutting value of the region respectively. Arrows show the parent-child relationship between the regions. Here, while the leaves with parameter values correspond to the split regions, the others correspond to non-split regions.

LP (Section V-B), and the decrease in total error rate (Section V-C). We compared our results with the baseline method, **V-IAC** in our analysis.

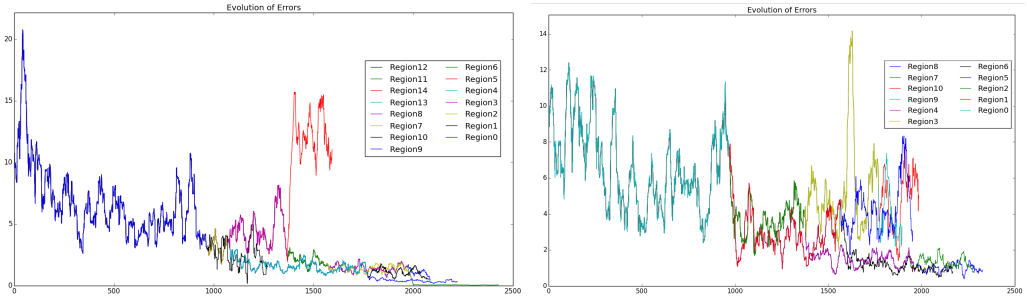
A. Generated sensorimotor regions

Our aim in this method is splitting $\mathbb{SM}$ from the points which split the space in a semantically sensible and useful manner to improve the learning rate of the system. In order to evaluate this, both **PE-IAC** and **V-IAC** were trained with 5000 iteration set. Due to the randomness, after a large number of runs, various distinctively structured trees were formed. Two representative trees formed during our experiments are shown in Fig. 3. In Fig. 3, different trees produced by **PE-IAC** with the same parameters are shown. In Fig. 3(a), the calculated cutting dimension of the 4^{th} region is compatible with the environment dynamics. The 6^{th} region covers A1 completely and the following splits are based on the F parameter, i.e. the system represents the effects of the F action parameter qualitatively. Another tree formed by **PE-IAC** method is shown in Fig. 3(b). In this tree, first split occurred at $X = 0.805$. In this case, the system prefers exploring the region with X parameter larger than the cutting value 0.08, which is a value close to the boundary (1.0) that separates A1 and A2. Next, the 3^{rd} region is split by $Y = 1.147$, which defines the other border between A1 and A3; then there was no observation of further exploration in region 5. This is because of the region is a non-learnable region since involved randomness. As can be seen in both two children of the 2^{nd} region, the system was successful in terms of splitting the region by F parameter from the points which make difference. As shown in Fig. 3(c) and 3(d), **V-IAC** couldn't split $\mathbb{SM}$ successfully taken into account the environment dynamics i.e. there were no trees formed by this method using F parameter as a cutting dimension.

We further analyzed how often which sound frequencies were explored by the robot for training different experts. When we compare plots in Fig. 4 , we observe that our method prefers using certain sound frequencies as the time passes. However, such distinction in **V-IAC** isn't discovered within 5000 iterations. Furthermore, it is observed that after 10000 iteration phase, this parameter has still not been discovered in **V-IAC**. Thus, **V-IAC** couldn't explicitly identify and represent the effect of sound frequencies on robot's interaction with the environment.

B. Comparison of Learning Progresses (LP) of Regions

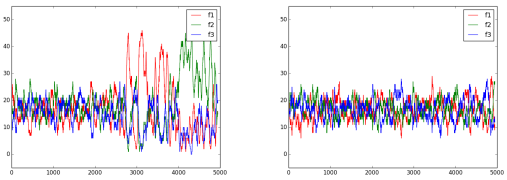
Here, we analyze the LP in each region and compare the LPs observed in **PE-IAC** and **V-IAC**. In Fig. 2(a), each child region's error curve is preceded by that region's parent's error curve, thus the decrease in the error after the split can be observed from the plot. First of all, as $\mathbb{SM}$ is split, the coherency of predictions is improved in general. However, the error rate of region 5 remains same (and high) as it corresponds to A3 that is completely random and therefore not learnable. The smoothed derivative error curves generated by **V-IAC** are shown in Fig. 2(b). In this method, for each child region, errors are completely inherited from the parent region and



(a) **PE-IAC** (b) **V-IAC**

Fig. 2. Smoothed derivative curves of errors. In **V-IAC** each child region completely inherits its parent's error list. However, in **PE-IAC**, each child region forms its own error list. For visualization, smoothed error curves are represented according to robot's encountering time of the experiences.

error change can be observed by considering this case. When the two methods are compared, it can be observed that with our method error drops quickly in all regions except unlearnable region and, with **V-IAC** while error drops in some regions, it remains same in other regions since those regions include parts from unlearnable A3.



(a) **PE-IAC** (b) **V-IAC**

Fig. 4. Preferred sound frequencies by the agent during one experiment. Here, x-axis represents the number of exemplars collected by the agent and y-axis represents the sound frequency values obtained by considering and smoothing all the frequencies inside a small time window.

C. Comparison of Decrease in Total Error Rate

For both methods, error rates of the systems that involve experiments with different training sizes are shown in Fig. 5. For this plot, we have trained both methods with the number of exemplars stated in the x-axis. Next, both methods were tested with 10 different test sets each of them including 2000 exemplars. Initially, there is no difference in performance between the two systems in the training set with 1000 training examplars, because the first split hadn't yet occurred, and the system hasn't made action selection by using its own IM. After training the system with 2000 exemplars, the gap between the performances of the methods becomes more clear.

VI. Conclusion

In this study, we have proposed a novel method for splitting $\mathbb{SM}$ to improve the learning performance and compared with an existing approach. Our contributions in this paper are: 1) we have brought in a new splitting mechanism which considers the successes a broad range of potential learning regions and 2) while evaluating the LPs that is used as the intrinsic reward of the agent, we consider only the errors made by that learning region. As a result, IM mechanism provided in our work splits $\mathbb{SM}$ more accurately and considers the future aspects of the splitting decisions. Also, our method is more memory efficient due to the fact that the child regions don't inherit its parent's error list. However, considering a broad range of candidate regions and re-evaluating their performances for each split decision, brings high computational overhead. As a future work, we would like to optimize this issue. Further, in this work, we set the total exemplar number that the agent will encounter during its lifetime. However, this mechanism should be autonomous as well. After some point, the splitting mechanism should be stopped when the sufficient level of competency is acquired by the agent. Aside from what we've presented, as a future work different intrinsic reward formulations can be employed to this algorithm.

We've evaluated our method in a simple experimental setting, we have observed that proposed method performs better than the existing method. As a future work, we are considering testing this method in more complex robotic problems after making improvements to the algorithm.

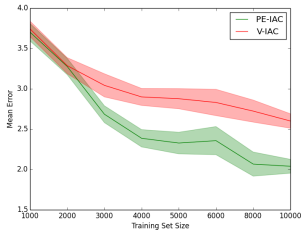


Fig. 5. Comparison of decrease in the mean error rates in **PE-IAC** and **V-IAC**. Lines correspond to mean error values of the test sets and shaded areas represent the variance.

References

[1] J. Weng, J. McClelland, A. Pentland, O. Sporns, I. Stockman, M. Sur, and E. Thelen, "Autonomous mental development by robots and animals," *Science*, vol. 291, no. 5504, pp. 599–600, 2001.

[2] A. G. Barto, S. Singh, and N. Chentanez, "Intrinsically motivated learning of hierarchical collections of skills," in *Proceedings of the 3rd International Conference on Development and Learning*. Citeseer, 2004, pp. 112–19.

[3] R. M. Ryan and E. L. Deci, "Intrinsic and extrinsic motivations: Classic definitions and new directions," *Contemporary educational psychology*, vol. 25, no. 1, pp. 54–67, 2000.

[4] V. Sperati and G. Baldassarre, "Learning where to look with movement-based intrinsic motivations: A bio-inspired model," in *Development and Learning and Epigenetic Robotics (ICDL-Epirob), 2014 Joint IEEE International Conferences on*. IEEE, 2014, pp. 461–468.

[5] J. Weng, "A theory for mentally developing robots," in *Development and Learning, 2002. Proceedings. The 2nd International Conference on*. IEEE, 2002, pp. 131–140.

[6] P.-Y. Oudeyer, F. Kaplan, and V. V. Hafner, "Intrinsic motivation systems for autonomous mental development," *IEEE transactions on evolutionary computation*, vol. 11, no. 2, pp. 265–286, 2007.

[7] J. M. Herrmann, K. Pawelzik, and T. Geisel, "Learning predictive representations," *Neurocomputing*, vol. 32, pp. 785–791, 2000.

[8] F. Kaplan and P.-Y. Oudeyer, "Motivational principles for visual know-how development," 2003.

[9] J. Schmidhuber, "Curious model-building control systems," in *Neural Networks, 1991. 1991 IEEE International Joint Conference on*. IEEE, 1991, pp. 1458–1463.

[10] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.

[11] E. Ugur and J. Piater, "Emergent structuring of interdependent affordance learning tasks using intrinsic motivation and empirical feature selection," *IEEE Transactions on Cognitive and Developmental Systems*, 2017.

[12] J. Mugan and B. Kuipers, "Autonomous learning of high-level states and actions in continuous environments," *IEEE Transactions on Autonomous Mental Development*, vol. 4, no. 1, pp. 70–86, 2012.

[13] G. Konidaris, L. P. Kaelbling, and T. Lozano-Perez, "Constructing symbolic representations for high-level planning." in *AAAI*, 2014, pp. 1932–1938.

[14] ——, "Symbol acquisition for probabilistic high-level planning," *Image (a1, Z0)*, vol. 1, p. Z0, 2015.

[15] S. Forestier, Y. Mollard, and P.-Y. Oudeyer, "Intrinsically motivated goal exploration processes with automatic curriculum learning," *arXiv preprint arXiv:1708.02190*, 2017.

[16] D. T. Larose and C. D. Larose, "k-nearest neighbor algorithm," *Discovering Knowledge in Data: An Introduction to Data Mining, Second Edition*, pp. 149–164, 2005.