Bold Hearts 2004

Bottom-up and Combined Skill Learning via the SIVE Decomposition for 2D and 3D Simulation League

Daniel Polani

Adaptive Systems Research Group
Dept. of Computer Science
University of Hertfordshire, UK

Abstract. In our current teams Bold Hearts 2004 (2D and 3D), we seek to further the development of the SIVE method used in Bold Hearts 2003 to learn skills. The ultimate goal is to develop a methodology that will allow agents to learn interpreting sensors, generating actions, then skills and ultimately tactics and strategies from basic principles with little prior knowledge.

1 Introduction

Many approaches to construct powerful RoboCup agents concentrate on creating specific skills and capabilities. These are often constructed with knowledge about specific properties of the world physics as simulated by the soccer server. Use of explicit knowledge in agents, however, limits the flexibility and robustness of the approach should the simulated world change and it makes it difficult to generalize to other fields. In addition, the capability of learning is one of the key questions in Artificial Intelligence, and it would thus be desirable to address this question in the RoboCup context.

Learning has been part of the RoboCup endeavour for a long time [11, 12], in particular in combination with reinforcement learning [1, 9]. As we argued in [3], reinforcement learning methods are attractive for learning approaches because they are highly general, mathematically accessible and well understood. This generality, however, comes at a price. In large search spaces, the learning algorithms are slow and their robustness and generalizability is not well controlled. To alleviate that, dedicated decompositions of the representation of the state space have to be performed that deconstruct the task hierarchically into manageable parts [2]. Still, a large number of learning steps has to be taken to learn a more complex task. In addition, convergence problems can arise in continuous domains (as RoboCup) [13].

In [3], we suggested a different different approach. We introduced SIVE, which was inspired by many different sources. Its original motivation stems from the observation that humans are able to attain a much steeper learning performance than computers when faced with a new task. As mentioned in [3], when facing an autonomous agent team, a human team playing with the OpenZeng interface at the GermanOpen 2001, while being technically and tactically inferior, showed a rapidly improving performance.

This, although the human accuracy in estimating ball position and performing actions was nowhere as accurate as that of the autonomous team. This is a clear indication that the "exhaustive learning" character exhibited by typical learning algorithms is inadequate to obtain the directedness and generalization power that we find in human learning. In SIVE, we desire to mimic some of the properties exhibited by human learning: extremely fast generalization and adaptation, "holistic" learning and the capability to combine skills. For this purpose, the SIVE method has been introduced that combines ideas and approaches from different areas. It has first been used in Bold Hearts 2003 to train the passing skill. In this year's teams, we will develop the SIVE methodology further, for two purposes:

1. to build up fundamental skills for the new 3D Bold Hearts team and
2. to further develop the skills for the 2D Bold Hearts team.

2 The SIVE framework

2.1 SIVE recapitulation

We will briefly recapitulate SIVE (for more details, see [4]). There are several important ideas in SIVE:

1. train specific skills as opposed to specific tasks;
2. skills are successively combined to attain goals;
3. skills are learnt in a "holistic" fashion, as complete patterns, unlike in Reinforcement Learning, where they are split up into time slices;
4. training concentrates on the limits of capabilities, i.e. cases where a there is a degree of uncertainty whether a particular goal can or can not be attained are probed in particular detail. Cases where the prediction certainty is high, will not be analyzed in same detail;
5. while different heuristics may be used in implementing the actual pattern learning (e.g. Self-Organizing Maps or Support Vector Machines), the philosophy of SIVE is information-theoretical. The guiding principles are the separation of factors of influence and of invariance. These principles have been formulated in terms of information-theoretic criteria [4]

2.2 Earlier Work

The SIVE method is inspired by many different sources. Although motivated by classical reinforcement learning [13, 17], it does not follow the standard paradigm of time partitioning and thus sequences of actions into equally distributed intervals; in fact, even reinforcement learning models with continuous time [16] still do structure time. So do approaches for analysis that make use of the temporal structure of a behaviour [18].

SIVE does not do that. Instead, it captures the whole behaviour sequence as a whole, classifying that behaviour according to its outcome, e.g. *success* or *failure*, respectively. This (external) classification can be considered a minimal reinforcement, or else, analogous to a classification problem. While the support vector machine formalism [5, 15] would have provided a transparent approach to solve such a classification problem, for a number of conceptual reasons it was decided to develop the approach itself closely guided by information-theoretical perspectives, reducing the support vector machine formalism to a usable heuristics that can be replaced by other mechanisms for this purpose. It should be mentioned that the structural risk minimization propagated in [15] has information-theoretic ramifications; among those, the VC dimension is obtained from the size of the space of the possible classificator configurations. The SIVE framework was furthermore inspired by the explicit model to compute an optimal scoring strategy developed in [8], though it strives to significantly more generality.

Although the concrete implementation may resort to heuristical approaches, like Self-Organizing Maps and Support Vector Machines, these are incidental; central aspects of SIVE are to be viewed in the an information-theoretic framework. In [10], it was shown that purely information-theoretic measures are sufficient to reconstruct the sensoric structure of an agent (an AIBO, in this case) without any prior knowledge on the structure of the sensoric space.

In [6], agent behaviour is constructed by optimizing information-theoretic criteria, in particular certain types of information flow. From these optimizations, a variety of behaviours can be constructed in a way that is largely independent of the concrete scenario.

2.3 The SIVE Method: Extensions

Introduction In Sec. 2.1, we gave a brief overview over the SIVE methodology. We will not go into the SIVE method in detail, but we will repeat some of the issues from [3] that are relevant for the Bold Hearts 2004 teams. It should be mentioned that below explanation is not intended to be self-contained, but only for getting an outline of the idea. For a full explanation, we refer to aforementioned reference.

The SIVE method incorporates several aspects. It consists of training individual skills in explicitly given scenarios which correspond to scenarios set up by coaches for human teams. Behaviours are trained by classifying the overall outcomes and the training concentrates on critical (risky) behaviours, where the outcome can not be predicted safely. Finally, a representation is sought for the behaviours that attempts to capture the properties that affect the outcome of a behaviour the *most* and those that affect it the *least*. We illustrate SIVE in the concrete setting of a particular RoboCup scenario, although it is by no means limited to neither this scenario, nor to RoboCup and will be applied elsewhere.

Classification and Critical Cases As opposed to classical reinforcement learning, SIVE distinguishes the two cases of capture and loss, but does not a priori attribute utilities to the cases, as classical MDP learning methods. Thus, this enables considering different types of situations using the same framework. E.g., capturing the ball may be considered “success” if the catcher is our team mate or “failure” if the catcher is our opponent.

Another aspect of SIVE is that only “critical” (“interesting”, “risky”) behaviours are considered. E.g., in our scenario, SIVE ignores kicks where the ball is always captured or always lost by the catcher. Instead, SIVE concentrates on kicks where the outcome (capture or loss) cannot be safely predicted. By concentrating the actions on the critical region, SIVE aims at establishing the boundary between the different outcome classes. As opposed to classical classification methods, SIVE exploits the fact that the agent can actively probe the critical regions and does not have to restrict itself to a given sample of training data.

Variants and Invariants On this set of critical behaviours, SIVE proceeds to find additional structure. To be able to generalize, it is useful to know which are the parameters that most predictive for the outcome of an action (*variants*) and which are most insensitive to that outcome (*invariants*). The variants tell the agent which aspect of an action is most important for achieving a particular outcome. Note that the invariants are of specific interest for the SIVE method. Without being able to presently go into more detail in the present team description, it should be noted that they contain important additional information about the structure of the problem that can be used to compress information attained in one context to be used in a different context.

Given a set of "critical" data modeled by a random variable X and its outcome class modeled by a random variable R, the variants and invariants are obtained by seeking an information-preserving transformation $T : \mathcal{X} \to \mathcal{Z}_v \times \mathcal{Z}_i$ such that, if we write $T(X) = Z = (Z_v, Z_i)$, then the transformed random variable Z_v becomes a variant and the transformed random variable Z_i becomes an invariant. Ideally, we would require T such that

$$I(X;Z) = I(X;Z_v,Z_i) = H(X) \qquad \text{(information preservation),} \qquad (1)$$

$$I(Z_v;R) = I(X;R) \qquad \text{(variant property)} \qquad (2)$$

$$I(Z_i;R) = 0 \qquad \text{(invariant property)} \qquad (3)$$

$$I(Z_v;Z_i) = 0 \qquad \text{(independence)} \qquad (4)$$

The *information preservation* property requires that the transformation will not lose any information about the structure of the original data. The *variant* property requires that the transformation identifies a *variant* parameter Z_v which is as predictive about the outcome of the action as the original data. The *invariant* property separates an *invariant* parameter Z_i which is completely insensitive to the outcome. Finally, the variant and invariant are to be completely *independent* to remove redundancies.

It cannot be expected that in a real-world constellation all these properties be fulfilled or even a suitable transformation be practically identifiable. So, for a practical application the equations (1)-(4) are to be modified as to seek a transformation that maximizes the left-hand side of (1) and (2) and minimizes the left-hand side of (3) and (4). Since X is in general a continuous-valued variable, for the maximization of (1) to make sense, a suitable normalization of the transformation (e.g. fixed variance) has to be assumed.

Even in the form of an optimization task it is often still not possible to fulfil all the properties at once. There are different possible approaches to solve that problem. One way is to formulate a Lagrangian optimization problem not unlike that of the Information Bottleneck scenario of [14]. The other approach is to use a multiobjective optimization method, e.g. Evolutionary Algorithms.

In [3], the situation was so simple that a straightforward heuristic approach was chosen to perform an approximative SIVE decomposition along the requirements of Eqs. (1)-(4). This does not limit the generality of the principle.

3 Current developments in the SIVE framework

The framework above has only been started to be applied in Bold Hearts 2003. The full generality has not yet been used. The acquisition of the passing skill required a 2-dimensional separator manifold in a 3-dimensional space to implement the variant/invariant separation. However, in general, one will need higher-dimensional separators. To achieve that, the present skills will be implemented using hierarchical Self-Organizing Maps or support vector machines.

Another goal is to probe the generalization capability of SIVE. One of the advantages of the approach is that the variants/invariants decomposition is expected to separate out what properties are essential for solving a task and which are incidental. Bold Hearts 2003 had been trained only against a single team. This year's 2D team will also use a different team to train against and thereby study how the difference in strategy will reflect in the variants and invariants. This, on the other hand, will shed light on how generalization can be automated in the learning process in agents.

We will use the SIVE framework also to develop the 3D team skills. It is envisaged to base even the 3D world model reconstruction in the framework of SIVE (similarly to [10]). If that turns out to be infeasible, then for this year's team the world model will be reconstructed by hand, and the passing skill will be trained using SIVE, similarly to last year's team.

A new addition to the SIVE framework is the *empowerment* concept which is currently being developed in conjunction with the information-theoretical agent analysis studies in [6, 10]. Here, one attempts to optimize the information flow from actuators to sensors, which can give rise to the "discovery" of new (even higher-level) modes of behaviour. The goal is to attempt to use this empowerment concept in the development of new skills and in the refinement of existing skills for RoboCup agents.

4 Acknowledgements

We wish to thank the UvA team for providing the basis code (UvA 2002) [7] on which the present team is based and which allowed us to concentrate on the development of the SIVE method. In addition, we would like to thank Oliver Obst, Pat Riley, Tom Howard and all the other developers of the 3D simulator; their merit in advancing the Simulation League and RoboCup itself is appreciated to the highest degree.

Bibliography

[1] Buck, S., and Riedmiller, M., [2000]. Learning situation dependent sucess rates of actions in a robocup scenario. In *Proceedings of PRICAI '00, Melbourne, Australia, 28.8.-3.9.2000*, 809.

[2] Dietterich, T. G., [1999]. Hierarchical reinforcement learning with the MAXQ value function decomposition. *Submitted to Machine Learning*.

[3] Franco, S., and Polani, D., [2004]. Skill Learning Via Information-Theoretical Decomposition of Behaviour Features. In Polani, D., Browning, B., Bonarini, A., and Yoshida, K., editors, *RoboCup 2003: Robot Soccer World Cup VII*, vol. 3020 of *LNCS*. Springer. Team Description (CD supplement).

[4] Franco, S., and Polani, D., [2004]. Skill Learning via Information-Theoretical Decomposition of Behaviour Features. Technical Report 399, Dept. of Computer Science, University of Hertfordshire.

[5] Haykin, S., [1999]. *Neural networks: a comprehensive foundation*. Prentice Hall.

[6] Klyubin, A. S., Polani, D., and Nehaniv, C. L., [2004]. Organization of the Information Flow in the Perception-Action Loop of Evolved Agents. Technical Report 400, Department of Computer Science, Faculty of Engineering and Information Sciences, University of Hertfordshire. Submitted.

[7] Kok, J., and de Boer, R., [2002]. UvA Trilearn. Software.
`http://carol.wins.uva.nl/ jellekok/robocup/`, October 2003

[8] Kok, J. R., de Boer, R., and Vlassis, N., [2002]. Towards an optimal scoring policy for simulated soccer agents. In Gini, M., Shen, W., Torras, C., and Yuasa, H., editors, *Proc. 7th Int. Conf. on Intelligent Autonomous Systems*, 195–198. Marina del Rey, California: IOS Press.

[9] Lauer, M., and Riedmiller, M., [2000]. An Algorithm for Distributed Reinforcement Learning in Cooperative Multi-Agent Systems. In *Proc. 17th International Conf. on Machine Learning*, 535–542. Morgan Kaufmann, San Francisco, CA.

[10] Olsson, L., Nehaniv, C. L., and Polani, D., [2004]. Sensory Channel Grouping and Structure from Uninterpreted Sensor Data. Technical Report 401, Department of Computer Science, Faculty of Engineering and Information Sciences, University of Hertfordshire. Submitted.

[11] Stone, P., [2000]. *Layered Learning in Multiagent Systems: A Winning Approach to Robotic Soccer.* MIT Press.

[12] Stone, P., and Veloso, M., [1998]. A layered approach to learning client behaviors in the RoboCup soccer server. *Applied Artificial Intelligence*, 12.

[13] Sutton, R. S., and Barto, A. G., [1998]. *Reinforcement Learning*. Cambridge, Mass.: MIT Press.

[14] Tishby, N., Pereira, F. C., and Bialek, W., [1999]. The Information Bottleneck Method. In *Proc. 37th Annual Allerton Conference on Communication, Control and Computing, Illinois*.

[15] Vapnik, V., [1995]. *The Nature of Statistical Learning Theory*. New York: Springer.

[16] Vollbrecht, H., [1998]. Three Principles of Hierarchical Task Composition in Reinforcement Learning. In Niklasson, L., Bodén, M., and Ziemke, T., editors, *Proc. of the 8th International Conference on Artificial Neural Networks, Skövde, Sweden, 2-4 September 1998*, vol. II, 1121–1126. Springer.

[17] Watkins, C. C. J. H., and Dayan, P., [1992]. Q-learning. *Machine Learning*, 8(3):279–292.

[18] Wünstel, M., Polani, D., Uthmann, T., and Perl, J., [2001]. Behavior Classification with Self-Organizing Maps. In Stone, P., Balch, T., and Kraetzschmar, G., editors, *RoboCup-2000: Robot Soccer World Cup IV*, 108–118. Berlin: Springer Verlag. Winner of the RoboCup 2000 Scientific Challenge Award.