

Imprecise Selection and Fitness Approximation in a Large-Scale Evolutionary Rule Based System for Blood Pressure Prediction

Erik Hemberg
MIT CSAIL
hembergerik@csail.mit.edu

Kalyan Veeramachaneni
MIT CSAIL
kalyan@csail.mit.edu

Franck Deroncourt
MIT CSAIL
francky@csail.mit.edu

Mark Wagy
University of Vermont
mark@gmail.com

Una-May O'Reilly
MIT CSAIL
unamay@csail.mit.edu

ABSTRACT

We present how we have strategically allocated fitness evaluations in a large-scale rule based evolutionary system called ECStar. We describe a strategy that culls potentially weaker solutions early, then later only compete with solutions which have equivalent fitness evaluations, as they are evaluated on more fitness cases. Despite incurring some imprecision in fitness comparison, which arises from not evaluating on all the fitness cases or even the same ones, the strategy allows our system to make effective progress when the resources at its disposal are unpredictably available.

Categories and Subject Descriptors

D.1.2 [Programming Techniques]: Automatic Programming

Keywords

Genetic Programming, Rule Based, Age Layering, Distributed, Large Scale

1. INTRODUCTION

Large repositories of data offering the potential for inferential analysis via Machine Learning (ML) are becoming more ubiquitous. For example, in this contribution, we are referencing a large-scale medical database called MIMIC¹ consisting of time series digital versions of physiological signals, and detailed clinical and bedside information from an ICU setting. Physiological data alone is about 4TB. Our current contribution attempts to solve the problem of predicting arterial blood pressure by learning from waveforms in the database.

In general, to knowledge mine such repositories, a ML system needs to handle the volume of data efficiently and accurately. We are using an genetics-based ML (GBML) approach to predict arterial blood pressure in ICU patient waveform. As we use evolutionary computation our system requires multiple passes through the training data. Training

data is repeatedly presented to different, evolving, candidate solutions to evaluate their fitness.

Our system, ECStar is a distributed learning classifier system, see [3]. It uses a large number of volunteer compute nodes as “Evolutionary Engines”. An Evolutionary Engine locally executes an evolutionary algorithm (EA) (iterative population-based fitness evaluation, selection and reproduction) in its spare cycles. ECStar’s design is suited to using volunteer compute nodes and exploits their low cost. However, using them, while also learning from a large data set, imposes some conditions challenges:

Constrained RAM and disk space: Given that volunteers offer no more than 0.5GB RAM and very modest disk space, it is infeasible to keep the entire training set at each evolutionary engine.

Unpredictable volunteer availability: It is impossible assure a particular evolutionary engine uses a particular split of the fitness suite and all evolutionary engines provide full fitness suite coverage. An evolutionary engine is served random fitness cases from the fitness case server.

Potentially high latency communication: Since volunteers are widely dispersed geographically, communication to and from the fitness case server needs to use packets which are sized for minimal latency while not so small that their frequent requests cause network congestion.

2. ECStar

ECStar can strategically allocate whatever number of fitness evaluations it has available, over a fixed duration, in order to assure fast learning progress, see Fig. 2.

Our strategy has a number of steps: Locally at each evolutionary engine, we first cull individuals that perform relatively worse on the basis of a small fraction of fitness cases, while allowing those that are relatively better to survive. These survivors are next “seasoned” locally on more fitness cases. During seasoning, they are used for breeding. At the end of seasoning, if they pass a minimum fitness standard, they become “graduates” of their evolutionary engine and are passed to the hub of the ECStar system called the “Evolutionary Coordinator”. Figure 2 shows the number of evaluation (age) thresholds λ_1 , λ_2 , and λ_3 , on the breeding pool.

The Evolutionary Coordinator archives the best graduates it receives and randomly sends them out to evolutionary en-

¹<http://mimic.physionet.org>

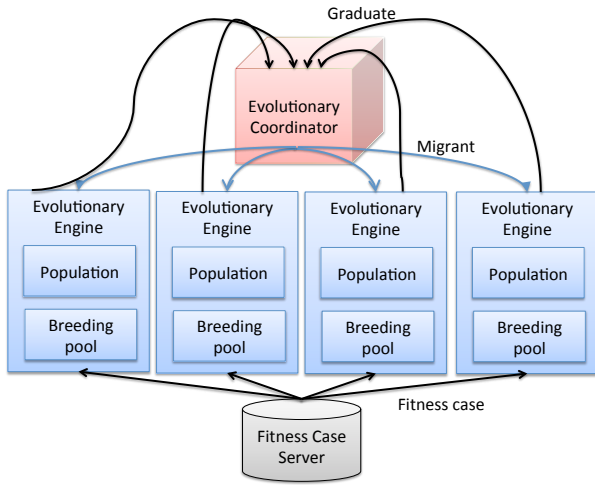


Figure 1: The hub and spoke model of ECStar.

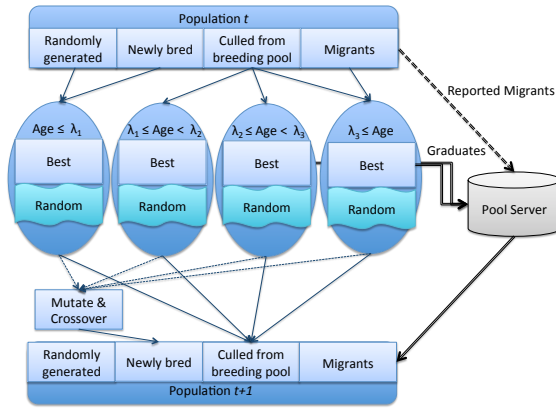


Figure 2: Evolution at the evolutionary engine.

gines as *migrants* for further fitness evaluations. Migrants are returned periodically by evolutionary engines but how long it will take is unpredictable. Some migrants may therefore receive many further fitness evaluations if they are fortunate enough to come and go to evolutionary engines which are frequently available while others may take a very long time to come back from just one evolutionary engine. The Evolutionary Coordinator has to strategically deal with this evaluation quantity mismatch while it exerts selection pressure on *migrants* to keep their quantity in check. To do this, its archive is a set of fixed size brackets each spanning successively higher intervals of fitness evaluations. When an individual arrives at the Evolutionary Coordinator, its bracket is identified by consulting how many fitness cases it has been evaluated on. Unless the bracket is not full, it must be better than the lowest ranked individual in the bracket, by relative fitness, to enter the archive. A truly fit individual moves up the brackets as it is evaluated on more fitness cases. Eventually it reaches the topmost bracket and is sequestered off the system as a high performing solution.

In fact, at any point in time, the archive can be consulted for the best performing individuals of the system.

The strategy is effective at assuring fair competition when individuals local to an evolutionary engine have fewer evaluations because these individuals have been evaluated on the same test cases. After individuals become migrants, competition at the Evolutionary Coordinator is fair in terms of fitness case quantity but somewhat noisy due competitors likely not having been evaluated on the same fitness cases. However this noise decreases in higher brackets because when individuals have been tested on more fitness cases, there is a higher likelihood that some of their fitness cases are in common, i.e. “symmetric”. The most important parameter of the method is how many fitness cases are used at the evolutionary engine for the first culling step. If that quantity is too low, the system would be vulnerable to making poor early choices because fitness estimates at this step are too unreliable. If the quantity is too high, fitness evaluations are wasted on individuals that are truly fit enough to breed, migrate and provide a good solution.

Our bracketing strategy ensures that only individuals which have been evaluated on the same number of fitness cases compete. It is important not to confuse this with age layering as in ALPS, [1]. Age in ALPS denotes how many generations an individual descends from.

3. EXPERIMENTS

Our goal is to predict the arterial blood pressure (ABP) of patients in an Intensive Care Unit. The signals include multiple leads of ECG, Arterial Blood Pressure, and Pulmonary Artery Pressure [2]. We select data for a subset of 47 patients and set the thresholds for different classes based on Mean Arterial Pressure MAP. We use a lag of 100 beats, future time window is set to 30 min and the forecast window is set to 10 min.

We employ a beat onset detection algorithm provided by the MIMIC II database and extract the samples that correspond to a beat and extract the features. We filter the data and extract and label features. The signal for each patient is divided into data packages with 1500 rows. From preliminary results for runs lasting 1h with 3 fold cross validation the mean prediction accuracy of the top solutions were 82%.

For future work one improvement of the design would be to have different selection pressure at the different layers.

Acknowledgements

Thank you to the Li Ka Shing Foundation.

References

- [1] Hornby, G.: Alps: the age-layered population structure for reducing the problem of premature convergence. In: Proceedings of the 8th annual conference on Genetic and evolutionary computation. pp. 815–822. ACM (2006)
- [2] Saeed, M., Lieu, C., Raber, G., Mark, R.: Mimic ii: a massive temporal icu patient database to support research in intelligent patient monitoring. In: Computers in Cardiology, 2002. pp. 641–644. IEEE (2002)
- [3] Urbanowicz, R., Moore, J.: Learning classifier systems: a complete introduction, review, and roadmap. Journal of Artificial Evolution and Applications 2009, 1 (2009)