

An Estimation of Distribution-like Algorithm based on a Denoising Autoencoder

Alex, Sid, and Chrisantha

School of Electronic Engineering and Computer Science
Queen Mary, University of London
{a.churchill,ssgd,c.t.fernando}@qmul.ac.uk

Abstract. In this paper we present a novel neural-based optimisation algorithm. The algorithm follows the traditional generate-update methodology of Estimation of Distribution algorithms, using a denoising autoencoder to learn the structure of promising solutions within its hidden layer, with the output neurons defining a probability distribution that is sampled from to produce new solutions. The algorithm is shown to outperform a canonical Genetic Algorithm on several combinatorial problems, including the multi dimensional 0/1 knapsack problem, MAXSAT and the Hierarchical If and Only If. Analysis shows that the neural network is able to learn interesting structural features of the search space, while the sampling method employed supports continued exploration, enabling optimal solutions to be found on NP-hard problems.

1 Introduction

Estimation of Distribution Algorithms (EDAs) are a growing field in Evolutionary Computation, which attempt to build a statistical model of sections of a search space in order to uncover underlying structure and guide search in an efficient manner [1]. At the heart of an EDA lies a model-building algorithm. Examples include Bayesian Networks [2], Markov Networks [?] and K-Means clustering [?]. In this paper we introduce a novel neural-based method for modelling, a denoising autoencoder. An auto encoder is a feed forward neural network, consisting of at least one hidden layer, which is trained to reproduce its inputs from its outputs. Over the course of training the hidden layer learns a representation of the data, which has been used for reconstructing missing data [?] and dimensionality reduction [?]. The algorithm introduced in this paper trains a single autoencoder with promising solutions from a population, from which structure of the search space is learnt by the representation in the hidden layer. This differs from traditional EDAs such as PBIL [?] or ECGA [?] as an explicit statistical model is not produced. However, the learnt structure can be leveraged to produce new solutions by inputting an existing or randomly generated solution into the network and sampling from the output neurons using a binomial distribution. Results presented in Section 6 show that the autoencoder method is able to outperform a canonical Genetic Algorithm on a range of combinatorial and hierarchical problems.

2 Background

Early work on EDAs concentrated on methods that explicitly modelled the probabilities of independent alleles occurring in a population of genotypes. These include the compact Genetic Algorithm [?], PBIL [?] and Univariate Marginal Probability methods[?]. Improved success was found by using clustering algorithms (ECGA) [?], Bayesian Networks [?], Markov Networks [?] and tree structures [?]. These methods have used greedy search in order to build models of the conditional probabilities between variables. The use of the autoencoder model in this paper is motivated by its potential to learn high-dimensional dependencies in data, while still maintaining a low computational cost in terms of training time. Recently there has been interest in neural-based methods in an EDA context for multi-objective optimisation. These include Growing Neural Gas (GNG) [?] and Restricted Boltzmann Machines (RBM) [?]. An autoencoder is another method for unsupervised learning of features that has potential to model solution structure and has hitherto not been applied to combinatorial optimisation.

Another motivation for this approach is to investigate methods in which Evolutionary Algorithms can be implemented in Neural Networks. The *Neural Replicator Hypothesis* [?] proposes that evolutionary processes could operate in the brain at phylogenetic timescales. In [3] a network of spiking neurone combined with Hebbian learning, which enabled linkages to be found between features, was applied to combinatorial problems and solved the 128-bit HIFF problem.

3 Methods

3.1 Denoising Autoencoder

Sid's stuff here.

3.2 Optimisation Algorithm

Alex's stuff on the optimisation algorithm.

4 Experiments

Experiments intro blurb.

4.1 0/1 Knapsack Problems

4.2 Hierarchical If and only If

4.3 Royal Road

4.4 MAXSAT

5 Results

6 Discussion

7 Conclusion

Acknowledgments. The heading should be treated as a subsubsection heading and should not be assigned a number.

References

1. Pelikan, M., Sastry, K., Cantú-Paz, E.: Scalable optimization via probabilistic modeling. Springer (2006)
2. Pelikan, M.: Hierarchical Bayesian optimization algorithm. Springer (2005)
3. Fernando, C., Goldstein, R., Szathmáry, E.: The neuronal replicator hypothesis. *Neural Computation* **22**(11) (2010) 2809–2857