CS7641 – Machine Learning

Assignment 2 – Randomized Optimization

Mason J. Kelchner  
mason.kelchner@gatech.edu

# 

# INTRODUCTION

The purpose of this report and the analyses presented is to explore random search optimization algorithms and their application to supervised machine learning techniques. The random optimization algorithms investigated include random hill climbing, simulated annealing, genetic algorithms, and MIMIC [1]. Two experiments are performed for these random optimization methods. The first experiment used each algorithm to find solutions to three varying complexity discrete combinatorial problems including: Travelling Salesman, Random Bit Matching, and the Four Peaks. These optimization algorithms and problems were implemented using custom functions and modules as well as the *mlrose-hiive* implementations [2] [3]. Other optimization problems were investigated including the Knapsack optimization problem and One Peak optimization; however, for the purpose of highlighting the advantages of simulated annealing, genetic algorithms, and MIMIC these were not included. The second experiment compared the performance of random hill climbing, simulated annealing, and genetic algorithms with varying hyperparameters on the optimization of the weights in a neural network in place of backpropagation. Performance of these various algorithms was determined by looking at a combination of factors including the accuracy of predictions from the neural network formed from the weights set by each algorithm, the F1 score (utilizing the python module sklearn.f1\_score), and the wall clock time required to solve the optimization of weights [4]. Various activation functions, hidden nodes, and other hyperparameters were tuned for each algorithm to explore the effects of each on the performance of the optimization algorithms. For this second experiment, a mushroom classification data set was used with 20 features and 61,069 instances classifying mushrooms either as edible (e) or poisonous (p). Each experiment and all data preparation will be explained in the following sections.

# Randomized Optimization Experiments

Each of the following optimization problems were selected to highlight advantages and disadvantages of each of the random optimization algorithms introduced earlier.

## Travelling Salesman Problem

## Random Bit Batching Problem

\*MIMIC does well with structure (chain problems like this one). Price of MIMIC is time (time complexity is high). Good when cost of fitness function is high/expensive. MIMIC still gets caught in local optima.

## Four Peaks Problem

# Randomized Optimization For Neural Network

# Summary

# References

|  |  |
| --- | --- |
| [1] | J. De Bonet, C. Isbell and P. Viola, "MIMIC: Finding Optima by Estimating Probability Densities," *NIPS,* pp. 424-430, 1997. |
| [2] | G. Hayes, "mlrose: Machine Learning, Randomized Optimization and SEarch package for Python," https://github.com/gkhayes/mlrose. |
| [3] | A. Rollings, "). mlrose: Machine Learning, Randomized Optimization and SEarch package for Python, hiive extended remix," https://github.com/hiive/mlrose. |
| [4] | F. Pedregosa, Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research,* vol. 12, pp. 2825-2830, 2011. |