

Genetic Algorithms and Feature Weighting

Introduction

Genetic algorithms are a mode of computation inspired by evolutionary processes, and they have applications in a wide range of domains. One task that GAs are particularly effective in performing is feature weighting for classifier systems. Researchers in music information retrieval should be knowledgeable about genetic algorithms and their use in classifier systems. Therefore, this paper offers a brief overview of genetic algorithms (GAs), feature selection and weighting for classification, and the use of GAs to accomplish these tasks.

Genetic Algorithms

Following the concept of evolution by Charles Darwin in the nineteenth century and early computer simulations of evolution in the mid-twentieth century, genetic algorithms were formally invented in the 1970's by John Holland (1975). Works by De Jong (1975) and Goldberg (1989) further advanced research on genetic algorithms and led to widespread use in many domains (Coley 1999).

It is possible to view genetic algorithms as performing optimization problems for which maxima are hard to find deterministically but one may easily guess possible solutions and evaluate them against each other. A solution to the problem is represented as a chromosome, where each chromosome contains several constituent genes. A set of chromosome values constitute a population, and this population interbreeds and mutates according to a simplified evolutionary model (Coley 1999).

In the simplest case, after initializing a random population, a genetic algorithm involves the repetition of an execution loop that may go as follows: Each member of the population is evaluated to determine a numerical fitness value, which is related the performance of the problem solution encoded in that population member. The fitness values are used to select population members to "mate" with each other via crossover, in which offspring are produced by means of swapping the parents' genes on either side of a randomly chosen locus point. The offspring chromosomes may then be mutated with a given probability. This process is repeated a certain number of times, or until the population meets certain characteristics. At the termination of the algorithm, the fittest individual (or individuals) are taken as solutions to the problem (Coley 1999).

Many variations on this simple algorithm are used in practice. The methods of selection, mutation, crossover, and encoding a solution in a chromosome, as well as the population size, all impact the algorithm's performance (Wolpert and Macready 1997). Additionally, genetic algorithms may be implemented in parallel to achieve faster execution and higher quality solutions (Cantú-Paz 2000).

Feature Selection and Feature Weighting

Feature selection and feature weighting are two related tasks that are very relevant to classification problems in music information retrieval. In this context, features are any numerical or binary values representing attributes of an instance to be classified. Cory McKay's (2004) Bodhidharma MIDI classification system, for example, uses features such as "number of unpitched instruments," "range of highest line," and "initial time signature."

Even though many features may be available to a classifier, it is not necessarily desirable to actually use all of them. The "curse of dimensionality" is that the size of the classifier's training set must grow exponentially with the dimensionality of the feature space (McKay 2004). Two common approaches to addressing the selection of relevant features are dimensionality reduction (e.g., using principle component analysis) and experimentation (i.e., choosing subsets and evaluating them by training a classifier to use them) (McKay 2004). Experimentation poses a problem in that the number of potential feature subsets is exponentially related to the number of available features, so an exhaustive search is infeasible for large numbers of features (Siedlecki and Sklansky 1989).

Knowledge of the subset of useful features is only one step toward an optimal classifier system. In any feature subset, some features may be more relevant than others. In a K-Nearest Neighbor classifier, for example, this means that weighting some features relative to others can "warp" the dimension space to maximize the separation of classes and improve classifier performance (Punch et al. 1993). However, the time complexity problem of feature selection is only magnified in feature weighting.

Using GAs for Feature Selection and Weighting

Siedlecki and Sklansky (1989) demonstrated that GAs are useful in performing feature selection accurately and within a reasonable amount of time. Their algorithm involved a population of feature subsets represented by binary vectors. The fitness of each member was evaluated by training a classifier using its encoded feature subset, where better classifier performance yielded higher fitness. (The fitness was also affected by the size of the subset, as Siedlecki and Sklansky were interested in finding the smallest subset with acceptable performance rather than the optimal subset per se.) They compared the performance of a genetic algorithm to exhaustive search, branch and bound, and (p,q)-search on a K-Nearest Neighbor classification problem with 24 features, and they found that the GA performed favorably. The feature sets selected by the GA performed closer to those selected by the exhaustive search than those selected by the other algorithms, and the time necessary to reach a solution was less for the GA than for exhaustive search and branch and bound.

Punch et al. (1993) built on the work of Siedlecki and Sklansky by using GAs for K-Nearest Neighbor classifier feature weighting. In this work, each chromosome was a vector of real-valued weights for each feature. They found that the best approach was to follow binary selection with feature weighting. However, computation time was a problem, with results taking 14 days to compute. They noted a near-linear speedup when fitness evaluation was performed in parallel on multiple processors. Notably, the parallel implementation of the GA still involved a single population, so the quality of the solution

was not affected by parallelization. More recently Minaei-Bidgoli et al. (2004) have shown that GAs are also useful for feature weighting in multiple classifier systems.

Applications to Music Information Retrieval

The above results are applicable to any problem in music information retrieval dealing with classification using moderate to high numbers of features, such as the ACE project (Fujinaga 2005). Further knowledge of genetic algorithm behavior, GA feature weighting schemes, and GA parallelization strategies can contribute to such systems becoming more accurate and efficient.

References

- Cantú-Paz, E. 2000. *Efficient and accurate parallel genetic algorithms*. Boston: Kluwer Academic Publishers.
- Coley, D. 1999. *An introduction to genetic algorithms for scientists and engineers*. Singapore: World Scientific Publishing.
- De Jong, K. 1975. *An analysis of the behavior of a class of genetic adaptive systems*. Ph.D. Dissertation. University of Michigan.
- Fujinaga, I. 2005. *Autonomous Classification Engine Project*.
<http://coltrane.music.mcgill.ca/ACE/>.
- Goldberg, D. 1989. *Genetic algorithms in search, optimization and machine learning*. Reading, MA: Addison-Wesley.
- Holland, J. 1975. *Adaptation in natural and artificial systems*. Ann Arbor, MI: University of Michigan Press.
- McKay, C. 2004. *Automatic genre classification of MIDI recordings*. M.A. Thesis. McGill University, Canada.
- Minaei-Bidgoli, B., G. Kortemeyer, and W. Punch. 2004. Optimizing classification ensembles via a genetic algorithm for a web-based educational system. *Proceedings of the International Workshop on Syntactical and Structural Pattern Recognition and Statistical Pattern Recognition*, 397–406.
- Punch, W., E. Goodman, M. Pei, L. Chia-Shun, P. Hovland, and R. Enbody. 1993. Further research on feature selection and classification using genetic algorithms. *Proceedings of the 5th International Conference on Genetic Algorithms*, 557–64.
- Siedlecki, W., and J. Sklansky. 1989. A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters* 10 (5): 335–47.
- Wolpert, D., and W. Macready. 1997. No-free-lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation* 1 (1): 67–82.