Parallel Data Mining of Bayesian Networks from Gene Expression Data

Longde Yin[1], Chun-Hsi Huang[2], Sanguthevar Rajasekaran[3]

Keywords : Gene Regulatory Networks, Genetic algorithm, Parallel Data Mining

DNA microarrays allow monitoring gene expression for tens of thousands of genes in parallel and are already producing huge amounts of valuable Gene Expression Data. Uncovering gene/protein interaction and key biological feature of cellular systems from these data is a major challenge in computational biology. Bayesian network (BN) is a promising method to describe relationships between genes in a genetic regulatory network. However, learning Bayesian network structure is an optimization problem in the space of directed acyclic graphs [1]. The number of such graphs is super-exponential in the number of variables. Therefore, we need to develop high-performance parallel search algorithms.

In the work described here the problem is to find the best Genetic Regulatory Network in a very large solution space of all possible Bayesian networks. Since the problem is NP-hard [3], a heuristic search technique must be used. This leads to the employment of Genetic algorithm (GA), since GA has been shown to be a robust and effective search method requiring very little information about the problem to explore a large search space. The GA works on a population of solutions, which change as the algorithm cycles through a sequence of generations, until a satisfactory solution has been found. Solutions are directed graphs; viable solutions (those which will be scored and allowed to breed in the next generation) are directed acyclic graphs. The scoring function used was the Bayesian scoring metric (BSM) [1], in which the best fit to the experimental data is calculated using Bayesian techniques. High scoring structures have a greater chance of being selected as parents for the next generation [2].

Due to the sheer volume of data involved in data mining, the time required to execute genetic algorithms and the intrinsic parallel nature of genetic algorithms, we decide to parallelize the Genetic algorithm (PGA) and plan to take two different approaches to parallelizing the GA.

[1] Dept. of Computer Science & Engineering, University of Connecticut, USA , E-mail: yin@engr.uconn.edu

[2] Dept. of Computer Science & Engineering, University of Connecticut, USA. E-mail: huang@engr.uconn.edu

[3] Dept. of Computer Science & Engineering, University of Connecticut, USA. E-mail: rajasek@engr.uconn.edu

The first approach is to implement GA in master-slave model Breeding (reproduction, crossover and mutation) was carried out in parallel. In fact the scoring was also implemented in parallel. The selection had to be implemented sequentially and thus remained on the master (the root processor which is the controller, and is connected to the host). This was necessary, as all of the structures from the new generation needed to be re-mixed to form new parents from the gene pool before distribution to the slaves for breeding. The remaining processors are utilized as slaves, which carry out the breeding in parallel and report the new structures and their scores to the master (root processor) [2].

The second approach is to divide the population into subpopulations, run a conventional GA in each subpopulation and allow the periodic communication of information between subpopulations that helps in the search for the solution. The information usually exchanged between subpopulations is a subset of the fittest individuals of each subpopulation. This exchange of individuals is known as migration. This parallel version of GA actually is a distributed GA.

Although this project is still in progress, it is anticipated that the PGA should enhance the efficiency of genetic search and has higher probability to get the optimal solution than the GA. Since the PGA can make use of multiple computing resources at the same time and can divide the large problem into several smaller ones.

In the poster session, presentation will address the problem of parallelization of Genetic Algorithm for the Mining of Bayesian Network based on Gene Expression Data. Two approaches to parallelizing the GA will be presented in detail. The results of the performance study of PGA and PGA's applications in Data Mining will be presented too.

References

[1] Nir Friedman and et al. 2000. Using Bayesian Networks to Analyze Expression Data, *J Comput Biol.*, 7(3-4):601-20.

[2] Roy Sterritt and et al. 2000. Parallel Data Mining of Bayesian Networks from Telecommunications Network Data, *IPDPS Workshops 2000*: 415-426.

[3] Chickering D.M. and D. Heckerman. 1994. Learning Bayesian network is NP-hard, *Microsoft Research*, MSR-TR-94-17.