Parallel Stochastic Search for Protein Secondary Structure Prediction

V. Robles[1], M.S. Pérez[1], V. Herves[1], J.M. Peña[1], P. Larrañaga[2]

[1] Department of Computer Architecture and Technology, Technical University of Madrid, Madrid, Spain
[2] Department of Computer Science and Artificial Intelligence, University of the Basque Country, San Sebastián, Spain

Abstract. Prediction of the secondary structure of a protein from its aminoacid sequence remains an important and difficult task. Up to this moment, three generations of Protein Secondary Structure Algorithms have been defined: The first generation is based on statistical information over single aminoacids, the second generation is based on windows of aminoacids –typically 11-21 aminoacids– and the third generation is based on the usage of evolutionary information. In this paper we propose the usage of naïve Bayes and Interval Estimation Naïve Bayes (IENB) –a new semi naïve Bayes approach– as suitable third generation methods for Protein Secondary Structure Prediction (PSSP). One of the main stages of IENB is based on a heuristic optimization, carried out by estimation of distribution algorithms (EDAs). EDAs are non-deterministic, stochastic and heuristic search strategies that belong to the evolutionary computation approaches. These algorithms under complex problems, like Protein Secondary Structure Prediction, require intensive calculation. This paper also introduces a parallel variant of IENB called PIENB (Parallel Interval Estimation Naïve Bayes).

1 Introduction and Related Work

Stochastic search algorithms are founded on the idea of selective and heuristic exploration over the complete space of possible solutions. These algorithms evaluate only a sample of this space and, using some heuristics, select future candidates in terms of their possibilities to improve current solutions. This is a very important issue for the cases in which the evaluation of each candidate is expensive in terms of computation. Although only a (relatively) small set of candidates is evaluated, the number of evaluations for a very complex problem could be very high. There are different efforts to make this kind of techniques to perform faster. The parallel nature of these algorithms sets a clear strategy to deal with this problem.

One of the best known stochastic algorithms are Genetic Algorithms (GAs) [8]. GAs have also been designed as parallel algorithms in three different ways [1, 2, 15]: (i) as master-slave problem with a single population, the master node computes all the genetic operators and the evaluation of the fitness of the individuals is calculated by slave processors, (ii) multiple-population algorithms, independent problems are executed with its own population, these populations exchange best individual according to

some migration rules (this model has been called *island model* [26, 17]) and (iii) fine-grain parallel GAs, consistent in a spatially-structure population with a single individual per node and neighborhood restrictions for genetic crossover.

The most interesting, both in terms of practical application and theoretical contribution, is the island model. The performance gained using this approach comes twofold. First, the global population is split into smaller sub-populations and the offspring of new individuals is also divided by the number of nodes of the computation. Although the computation performance is probably better, as the size of the population decreases the quality of the solution could also be reduced due to the lack of diversity in each of the subpopulations. This is solved by the migration of individuals between populations. Second, there are researchers who claim the possibility to reach superlineal speedups in this kind of algorithms, achieving better result with less number of total individual evaluated. Although there are many controversial discussions [21] some studies about the increment of the selection pressure [2] provide an appropriate answer.

Our contribution deals with the extension of the ideas already developed for parallel GAs towards another stochastic paradigm (EDAs [14]) and apply them to the optimization of the Interval Estimation Naïve Bayes performance. Afterwards IENB will be used to deal with the PSSP problem.

The outline of this paper is as follows. Section 2 is an introduction to the semi naïve Bayes approach IENB. Section 3 describes our parallel version of this approach. Section 4 analyzes naïve Bayes and IENB as suitable methods for PSSP. Section 5 shows the results of the evaluation of these methods in PSSP. Finally, section 6 enumerates the conclusions and outlines further future work.

2 Interval Estimation Naïve Bayes

The naïve Bayes classifier [5, 7] is a probabilistic method for classification. It can be used to determine the probability that an example belongs to a class given the values of the predictor variables. The naïve Bayes classifier guarantees optimal induction given a set of explicit assumptions [4]. However, it is known that some of these assumptions are not compliant in many induction scenarios, for instance, the condition of variable independence respecting to the class variable. Improvements of accuracy has been demonstrated by a number of approaches, collectively named semi naïve Bayes classifiers, which try to adjust the naïve Bayes to deal with a-priori unattended assumptions.

Previous semi naïve Bayes classifiers may be divided into three groups, depending on different pre/post-processing issues: (i) to manipulate the variables to be employed prior to application of naïve Bayes induction [11, 13, 18], (ii) to select subsets of the training examples prior to the application of naïve Bayes classification [10, 12] and (iii) to correct the probabilities produced by the standard naïve Bayes [25, 6].

In this work, to deal with the problem of Protein Secondary Structure Prediction, we have used a new semi naïve Bayes approach named *Interval Estimation Naïve Bayes (IENB)* [22] that belongs to approaches that correct the probabilities produced by the standard naïve Bayes. In this approach, instead of calculating the point estimation of the conditional probabilities from data, as simple naïve Bayes does, confidence intervals are calculated. After that, by searching for the best combination of values into these intervals, it is aimed to break the assumption of independence among variables the simple naïve Bayes does. This search is carried out by a heuristic search algorithm and is guided by the accuracy of the classifiers.

To deal with the heuristic search EDAs –estimation of distribution algorithms– have been selected. EDAs [14] are non-deterministic, stochastic and heuristic search strategies that belong to the evolutionary computation approaches. In EDAs, a number of solutions or individuals are created every generation, evolving once and again until a satisfactory solution is achieved. In brief, the characteristic that most differentiates EDAs from other evolutionary search strategies, such as GAs, is that the evolution from a generation to the next one is done by estimating the probability distribution of the fittest individuals, and afterwards, by sampling the induced model. This avoids the use of crossing or mutation operators, and, therefore, the number of parameters that EDAs requires is reduced considerably.

While IENB improves naïve Bayes accuracy, its biggest problem is the running time. This problem is worst in the case of the protein dataset due to its size (about 70000 instances). Thus, we have decided the development of a parallel version of this algorithm in order to improve its performance. This parallelization is described in the next section.

3 Parallel IENB

With the aim of increasing the performance and accuracy of IENB, we have developed a parallel version of IENB, named PIENB. This approach is based on the simultaneous execution of the IENB code on different nodes of a cluster, exchanging the best individuals achieved in the nodes each N generations. PIENB uses the island model, described in the first section. The algorithm takes into account the following aspects:

1. Every node generates and improves an independent population, but each N generations, the best M individuals of this population are migrated in a round-robin fashion. The algorithm checks if a concrete individual has been already sent to the target node. Nodes only send individuals that are not included in the destination. This migration implies a faster convergence to the solution, because of the feedback process between the nodes. N and M are configuration parameters, which depends on the population size and the number of nodes. The migrated individuals replace the worst individuals in the destination population.
2. PIENB takes advantage of the higher processor capacity of a cluster of several nodes. Therefore, PIENB may achieve better results in a shorter time. Typically, for a cluster of n nodes, the speedup is near to n.

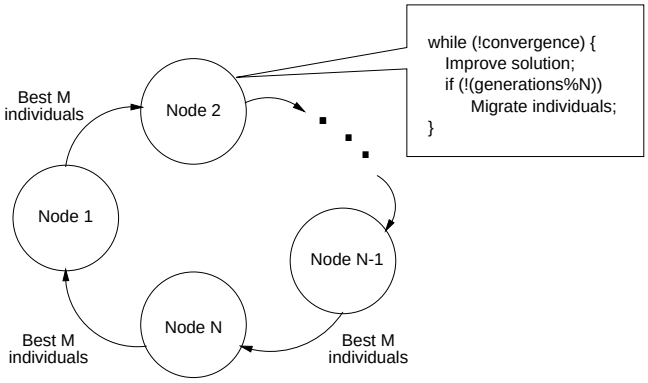

Fig. 1. PIENB flow control

Figure 1 shows the PIENB flow control. The pseudocode implemented in every node is also shown. The arrows represent the messages sent and received by every node, describing the relationship between the nodes. Nevertheless, it is possible to use different network topologies.

When one node has converged, it does not finish, because it has links with other nodes. In this case, this node takes the role of "bridge", receiving and sending messages from and to the corresponding nodes in the topology. Only when all the nodes have converged, the application finishes, belonging the solution to the last node.

To implement PIENB, MPI [16] has been used, mainly because of the following reasons:

1. It is an standard message-passing interface, which allows different processes to communicate among them through the usage of messages.
2. It is widely used in cluster of workstations.
3. It enhances the solution performance, because of its capacity for parallel programming.
4. It provides primitives for changing the network topology.

MPI is used as communication framework in the migration and bridge process.

4 Protein Secondary Structure Prediction with IENB

Prediction of a secondary structure of a protein from its aminoacid sequence remains an important and difficult task. Successful predictions provide a starting point for direct tertiary structure modelling, and also can significantly improve sequence analysis and sequence-structure threading for aiding in structure and function determination [24].

Since early attempts to predict secondary structure, most effort have focused on development of mappings from a local window of residues in the sequence to the structural state of the central residue in the window, and a large number of methods for estimating such mappings have been developed.

Methods predicting protein secondary structure have improved substantially in the 90's through the use of machine learning methods and evolutionary information [23]. At the alignment level, the increasingly size of databases and the ability to produce profiles that include remote homologs using PSI-BLAST have also contributed to performance improvement [9, 19, 20].

In this section we present a novel approach to protein secondary structure prediction (PSSP) based on the usage of naïve Bayes, IENB and its parallel version (PIENB). Most of the state-of-the-art PSSP methods are based on a three layer fashion: a first layer that maps from sequence to structure, a second layer from structure to structure and a third layer that corrects the obtained structure [23, 9]. In this case, we have developed only the first layer with really promising results (see next section).

In order to make the predictions, we have used a window of 13 aminoacids. To be able to use the evolutionary information (profiles) in naïve Bayes, IENB and PIENB we have adjusted the naïve Bayes formula:
Example of protein: A,R,N,S,T,V, ...
Example of protein profile: A80 S20, R50 S45 T5, N75 D5 C5 Q10, ...

Naïve Bayes classification formula (window of n aminoacids):

$$P(C = c|X_1 = x_1, \ldots, X_n = x_n) \propto P(C = c) \prod_{k=1}^{n} P(X_k = x_k|C = c) \quad (1)$$

Naïve Bayes classification formula for proteins profiles (window of n aminoacids):

$$P(C = c|X_1 = x_1, \ldots, X_n = x_n) \propto P(C = c) \prod_{k=1}^{n} \big(\sum_{j=1}^{20} pr_j P(X_k = x_j|C = c) \big) \quad (2)$$

where pr_j is the probability that the aminoacid in position k would be mutated into value x_j.

5 Experimental Results

For the experimentation with PSSP the datasets CB513 [3] has been used. For all the proteins in the dataset the evolutionary information has been included using the program PSI-BLAST from the database PIR-NREF. This database has been filtered to take out low complexity, coiled-coil and transmembrane regions. To generate the learning

cases we used a window of 13 aminoacids, obtaining a total of approximately 70000 instances. For obtaining the accuracy prediction a *leave-one-out* validation is performed.

The experimentation has been done with a 8 nodes cluster with Intel Xeon 2MHz, 1GB of RAM and connected by a Gygaethernet.

Several classification mechanism have been performed with this dataset. Table 1 shows the results of all of these executions. First, Naïve Bayes algorithm with no evolutionary information and, second, using this information. An important improvement is achieved as well as an increment in the execution time. This increment is due to (i) the larger number of attributes the algorithm has to estimate, (ii) the more expensive training and evaluation calculation and (iii) the bigger size of the input data (with vs. without profile information).

Table 1. Experimental Results for Protein Secondary Structure Prediction

Algorithm	**Accuracy**	**Time**
Naïve Bayes without evolutionary information	61.22	3 seconds
Naïve Bayes	67.58	80 seconds
IENB	70.16	40 days
PIENB	70.33	5 days

The last two rows of the table retrieve the results for both the sequential and parallel versions of the Interval Estimation Naïve Bayes (this last execution has been done 5 times, the showed value is the average). As it is shown a better classification accuracy is achieved but with a difference in execution time of several orders of magnitude. A further analysis of these two cases follows.

The parameters used to performs these experiments have been:

1. **IENB**:
   - Population size: 1000 individuals per generation
   - Offspring: 2000
   - Other options: elitism
2. **PIENB**:
   - Population size: 1000 individuals per generation (125 for each of the subpopulations)
   - Offspring: 2000
   - Migration rate: 10 individuals every 5 generations
   - Migration topology: Unidirectional ring (round-robin)
   - Other options: elitism
   - Migration replacement: Best migrated individuals replace worst

The better performance reached by the parallel version can be possible because of two reasons, first the speedup factor is close to 8 because of the ratio between communication and processing is very low. Second, the exploration of solutions using quasi-independent populations provided by the island model improves the quality of the solution and skips sub-optimal maximums. In order to analyze this bias a representation of

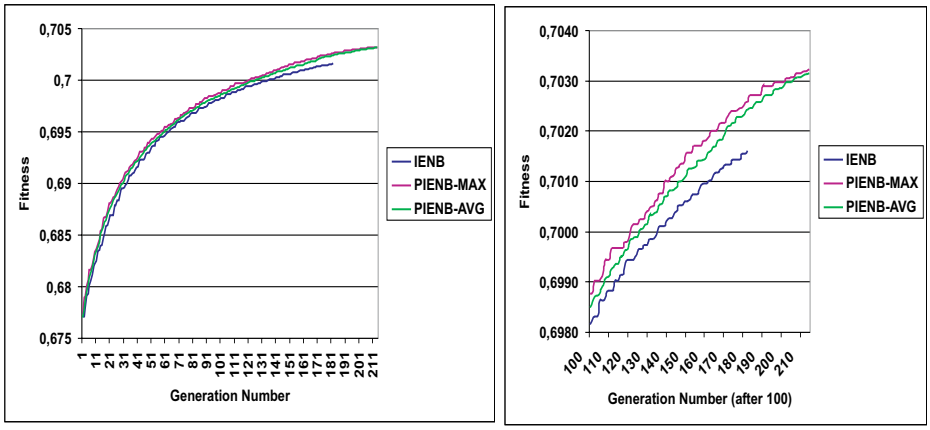

Fig. 2. Fitness value for IENB and PIENB depending on the number of generations

the best fitness (in the case of the sequential version) and the best and averaged fitness of each of the subpopulations (for the parallel one) is pictured in graph 2.

6 Conclusions and Further Work

On this contribution a new parallel semi-Naïve Bayes classifier has been presented. This new algorithm is based on stochastic search of the best combination of conditional probabilities. This approach has been designed as a very complex optimization problem, thus a parallel version of the algorithm has been implemented. This parallel version both reduces the execution time and improves the overall fitness of the algorithm. Our method is a single-layer classification approach that is very competitive with state-of-the-art classifiers [9]. And our future interests are addressed to design a second/third layer to perform structure-structure prediction.

The parallel algorithm presented here is a first experiment in the application of multi-population schemas for EDAs algorithms, different topologies [2], different policies and a combination of migration parameters are open to continue researching here.

References

1. T.C. Belding. The distributed genetic algorithm revisited. In *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 114–121, 1995.
2. E. Cant-Paz. *Efficient and accurate parallel genetic algorithms*. Kluwer Academic Publishers, 2001.
3. J. A. Cuff and G. J. Barton. Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *Proteins: Struct. Funct. Genet.*, pages 508–519, 1999.

4. P. Domingos and M. Pazzani. Beyond independence: conditions for the optimality of the simple Bayesian classifier. In *Proceedings of the 13th International Conference on Machine Learning*, pages 105–112, 1996.
5. R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, 1973.
6. J.T.A.S. Ferreira, D.G.T. Denison, and D.J. Hand. Weighted naive Bayes modelling for data mining. Technical report, Deparment of mathematics, Imperial College, May 2001.
7. D.J. Hand and K. Yu. Idiot's Bayes - not so stupid after all? *International Statistical Review*, 69(3):385–398, 2001.
8. J.H. Holland. Genetic algorithms and the optimal allocation of trials. *Journal on Computing*, 2(2):88–105, 1973.
9. D.T. Jones. Protein secondary structure prediction based on decision-specific scoring matrices. *Journal of Molecular Biology*, 292:195–202, 1999.
10. R. Kohavi. Scaling up the accuracy of naïve-Bayes classifiers: a decision-tree hybrid. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 202–207, 1996.
11. I. Kononenko. Semi-naive Bayesian classifier. In *Sixth European Working Session on Learning*, pages 206–219, 1991.
12. P. Langley. Induction of recursive Bayesian classifiers. In *European Conference on Machine Learning. Berlin: Springer-Verlag*, pages 153–164, 1993.
13. P. Langley and S. Sage. Induction of selective Bayesian classifiers. pages 399–406, 1994.
14. P. Larrañaga and J.A. Lozano. *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*. Kluwer Academic Publisher, 2001.
15. David Levine. *A Parallel Genetic Algorithm for the Set Partitioning Problem*. PhD thesis, Illinois Institute of Technology, Mathematics and Computer Science Division, Argonne National Laboratory, 1994.
16. Message Passing Interface Forum. *MPI: A Message-Passing Interface Standard*, May 1994.
17. G. Michaelson and N. Scaife. Parallel functional island model genetic algorithms through nested skeletons. In *Proceedings of 12th International Workshop on the Implementation of Functional Languages*, pages 307–313, September 2000.
18. M. Pazzani. Searching for dependencies in Bayesian classifiers. In *Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics*, pages 239–248, 1996.
19. G. Pollastri, D. Przybylski, B. Rost, and P. Baldi. Improving the prediction of protein secondary strucure in three and eight classes using recurrent neural networks and profiles. *Proteins*, 47:228–235, 2002.
20. D. Przybylski and B. Rost. Alignments grow, secondary structure prediction improves. *Proteins*, Submitted, 2001.
21. W.F. Punch. How effective are multiple populations in genetic programming. In *Genetic Programming, Proceedings of the Third Annual Conference*, 1998.
22. V. Robles, P. Larañaga, J.M. Peña, O. Marbán, J. Crespo, and M.S. Pérez. Collaborative filtering using interval estimation naïve bayes. *Lecture Notes in Artificial Intelligence (Advances in Web Intelligence)*, (2663):46–53, May 2003.
23. B. Rost and C. Sander. Prediction of protein secondary structure at better than 70% accuracy. *Journal of Molecular Biology*, 232:584–599, 1993.
24. S.C. Schmidler, J.S. Liu, and D.L. Brutlag. Bayesian segmentation of protein secondary structure. *Journal of Computational Biology*, 7(1/2):233–248, 2000.
25. G.I. Webb and M.J. Pazzani. Adjusted probability naïve Bayesian induction. In *Australian Joint Conference on Artificial Intelligence*, pages 285–295, 1998.
26. Darrell Whitley, Soraya B. Rana, and Robert B. Heckendorn. Island model genetic algorithms and linearly separable problems. In *volutionary Computing, AISB Workshop*, pages 109–125, 1997.