F-008

An Novel Soft Margin Classifier Using Genetic Algorithm

Yuji Mizuno, Goutam Chakraborty, Kazuhiko Yamashita
Iwate Prefectural University information science department

keywords : Genetic Algorithm, Soft Margin

Abstract - In many high dimensional classification problems, though the different class data are not linearly separable, we prefer a linear classifier over an over-trained high variance boundary. By limiting to linear boundary, in many practical applications, we can ignore noisy samples. Linear support vector machine is such a Soft margin classifier. In this work, we propose a genetic algorithm approach to construct the soft margin classifier. We simulated the algorithm and did several experiments with synthetic and real life data. The results are evaluated and found to be very near to optimum. The quality of the results and computation costs are compared with existing algorithms.

1 Introduction

Single layer perceptron can only create linear boundaries between classes. Yet, most of the real-life sample data from different classes are not linearly separable. It is possible to have non-linear boundaries using multilayer perceptron trained by error back propagation. Recently support vector machine is gaining popularity as classifier. In general application, the samples are first transformed to a higher dimensional space, where they are linearly separable and then an optimum linear boundary is achieved by training the SVM. The main notion is to minimize the risk of misclassification in case of unseen test data.

In many practical application, a linear support vector machine serves the purpose well. Linear support vector machine can not achieve zero error on training samples, if they are not linearly separable. The linear boundary is then set such that the sum of distances of mis-classified samples from the boundary line for two different classes are same (in SVM we basically consider two class problem).

Soft-margin is such a boundary, where, though there are mis-classified sample because the two class samples are not linearly separable, the risk of mis-classification of unknown sample is minimized.

There are different ways of achieving that. In this paper, we introduce a genetic algorithm approach to get a soft-margin linear boundary. The rest of the paper is as follows. Section 2 gives a short description of genetic algorithm. In addition we explain how we decide the fitness function of the genetic algorithm. Section 3 gives the details of the experiments and results. Section 4 is the conclusion.

2 Genetic Algorithm

Genetic algorithm imitates the way different species came to existence. The basic idea is survival of the fittest. The general procedure is as follows:

1. And initial population of many random solutions of the problem are generated. As these solutions are randomly generated, most of them are bad solutions, with qualities differing over a wide range. This is called the initial generation.
2. Now, the solutions from the present generation are selected in such a way that, better solutions survive to the next generation with higher probabilities. There are different ways to execute this selection, e.g., roulette selection, tournament selection, rank selection etc.
3. Now, from the selected solutions, randomly pairs of solutions are chosen and an operation called crossover is executed on chosen pairs. By crossover, parts of the solutions are swapped. This is like parents chromosomes being crossed over to make children chromosome. Sometimes, the solutions thus generated are better than the parent solutions.
4. With low probability, small parts of the solutions are changed to create completely new solutions. This operation is called mutation.
5. After crossover and mutation operations are over, we get a new set of solutions, we call the solutions of the next generation.
6. The above steps are repeated until an acceptable solution is reached, or the time is over.

The main problem in applying genetic algorithm to solve a problem is first to code the problem in a binary form - we call chromosomes. There should be a one-to-one mapping from a solution to a code. The second problem dependent part is how to calculate the fitness of different solutions. Once these two are defined, the rest is routing. Of course, depending on the complexity of the problem, the optimum population size, crossover and mutation probabilities etc. varies. In the next section we will explain how we used GA to solve soft-margin problem.

3 Description of the Algorithm

As we are looking for linear boundary, the boundary could be expressed as $y = ax + b$, where a is slope of the line and b the length cut at y-axis. So, basically we are looking for optimum values for a and b so that the line makes the soft-margin.

To implement this search for optimal value using GA, first we need to decide how to code the GA chromosomes. As we need to find the optimum value of two real numbers, the chromosome also consists of two decimal numbers. We also tried using binary coding of the decimal numbers. But using real numbers as genes proved to be more efficient for this problem. The population size 100 was used. The search usually converged after a few thousand generations, and we fixed the number of generations to be 10,000.

Tournament selection was used. For a good balance of diversity during the initial generations of searching, the initial tournament size was fixed at 2. During the end generations, to improve the efficiency of search, tournament size was increased to 5.

As the number of gene was only 2, single point crossover was used. Crossover probability was set at high value of 0.6. Mutation probability was set at 0.03, i.e., only 3% of the population underwent mutation at every generation. We also used elite preservation, i.e., the best chromosome until present generation is saved.

The fitness function is calculated as follows. The algorithm should work as maximum margin linear classifier in case the two classes are linearly separable. Therefore, in fitness function, we used a term so that the nearest correctly classified samples of the two classes are equidistant from the boundary. When the ratio of these two distances are different from one, the fitness is low. The value of fitness function increases as the ratio approaches 1. There is another part in the fitness function for overlapping samples. With linear boundary there will be mis-classified samples. Let the number of misclassified samples from class 1 and class 2 are K_1 and K_2 respectively. The other part of fitness function is to ensure that $K_1 = K_2 = K$, K is minimized, and the total distances of mis-classified samples of the two classes from the boundary line is same.

4 Experiments, Data and Results

For our experiment we used a synthetic two dimensional data of two classes. On a two dimensional space, first two points were randomly selected. They are the center of the two classes of samples. Now, other samples from the two classes were generated with Gaussian distribution around the centers. The standard deviation of one class was set to be much bigger than the other class. Also, the two standard deviation was such that there will be region of overlapping samples near the boundary. The data is shown in Fig. 1

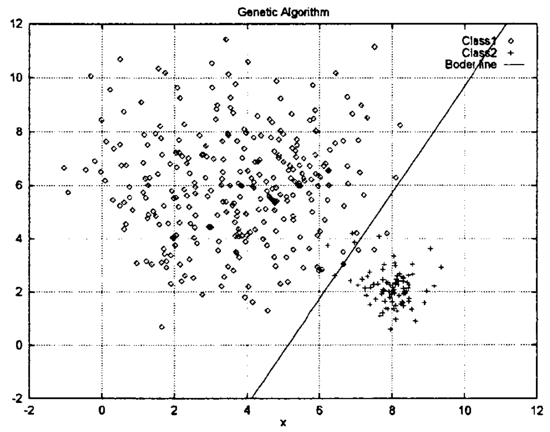

Figure 1: The two class data and the boundary line as obtained by the proposed algorithm

The boundary line, as created by the proposed GA based algorithm is shown. We can see that a total of 4 samples from each class have been mis-classified. We can also verify that the total distances of those mis-classified samples from the linear boundary are same for both the classes. Thus the condition for soft-margin is satisfied.

5 Conclusions and future work

We proposed a genetic algorithm approach to solve the problem of soft margin classifier, i.e., linear support vector machine. In the experiment, the proposed method could find the correct boundary line as required by the soft-margin classifier.

We did not compare the efficiency of our approach with other existing methods. Moreover, we used a simple problem of only two dimensions. We are now working with other high-dimensional data, where the number of GA parameters are more, making the search more complex.

References

[1] Hiroaki Kitano, "Genetic Algorithm", Japanese Society for Artificial intelligence, vol.7 No.1, pp26-37, 1991.