A NEW HEURISTIC OF THE DECISION TREE INDUCTION

NING LI, LI ZHAO, AI-XIA CHEN, QING-WU MENG, GUO-FANG ZHANG

Key Lab. of Machine Learning and Computational Intelligence, College of Mathematics and Computer Science, Hebei University, Baoding, 071002, Hebei, China
EMAIL:lining@cmc.hbu.cn

Abstract:

Decision tree induction is one of the useful approaches for extracting classification knowledge from a set of feature-based instances. The most popular heuristic information used in the decision tree generation is the minimum entropy. This heuristic information has a serious disadvantage-the poor generalization capability [3]. Support Vector Machine (SVM) is a classification technique of machine learning based on statistical learning theory. It has good generalization. Considering the relationship between the classification margin of support vector machine(SVM) and the generalization capability, the large margin of SVM can be used as the heuristic information of decision tree, in order to improve its generalization capability.

This paper proposes a decision tree induction algorithm based on large margin heuristic. Comparing with the binary decision tree using the minimum entropy as the heuristic information, the experiments show that the generalization capability has been improved by using the new heuristic.

Keywords:

Decision tree; Support vector machines; SMO; Clustering; large margin; Generalization

1. Introduction

Decision tree induction is one of the most important branches of inductive learning. It is one of the most widely used and practical methods for inductive inference. It creates a decision tree from the instance table according to heuristic information and classifies some instances by using a set of rules, which is transformed from the decision tree. Many methods have been developed for constructing decision trees from collections of instances. Given a training set, a general procedure for generating a decision tree can be briefly described as follows. The entire training set is first considered as the root node of the tree. Then the root node is split into several sub-nodes based on some heuristic information. If the instances in a sub-node belong to one class, then the sub-node is regarded as a leaf node, else we continue to split the sub-node based on the heuristic information. This process repeats until all leaf nodes are generated. The most popular heuristic information used in the decision tree generation is the minimum entropy. This heuristic information has many advantages such as small number of leaves and less computational efforts. However, it has a serious disadvantage-the poor generalization capability [3].

Support vector machines (SVMs) are a classification technique of machine leaning based on statistical learning theory. Considering a 2-class classification problem, SVMs are to construct a hyper-plane from a training dataset as a classifier to classify unseen instances. The SVM hyper-plane has the maximum margin which is defined as the distance between support vectors and the separate hyper-plane. According to Vapnik statistical leaning theory, the maximum of margin implies that the separate hyper-plane is optimal in the sense of expected risk minimization. The optimal characteristics of separate hyper-planes have resulted in an excellent generalization capability and some other good performances of SVM classifiers [3].

The inverse problem is how to split a given dataset into two clusters such that the margin between the two clusters attains maximum. Initially suppose we do not know the class for each individual instance. It is expected that instances of the dataset are assigned with class-labels (+1 or –1) such that the margin defined according to the separate hyper-plan of SVMs attains maximum [3].

This paper investigates a new decision tree generation procedure to improve the generalization capability of existing decision tree programs based on minimum entropy heuristic. Due to the relationship between the margin of SVMs and the generalization capability, the split with maximum margin is considered as the new heuristic information for generating decision trees. In the process of induction, the inverse problem of SVMs is required to be solved. Compared with the decision tree programs based on minimum entropy heuristic, the training accuracy and testing accuracy has been improved.

The rest of this paper is organized as follows. In section 2, the basic concept of support vector machines and

978-1-4244-3703-0/09/$25.00 ©2009 IEEE

its solving algorithm is reviewed briefly. Section 3 proposes the inverse problem of SVMs and its solving method. In section 4, the algorithm of inducing decision tree based on large margin is developed. Section 5 introduces the induction algorithm of binary decision tree based on minimum entropy. Section 6 shows some experimental results and finally Section 7 draws conclusions.

2. Support vector machines

2.1. The basic problem of SVMs[3]

Let $S=\{(x_1,y_1),(x_2,y_2),\ldots,(x_N,y_N)\}$ be a training set, where $x_i \in R^n$ and

$y_i \in \{-1,1\}$ for $i=1,2,\ldots,N$ .The optimal hyper-plane of S is defined as $f(x)=0$,

where

$$f(x)=(w_0 \cdot x)+b_0 \tag{1}$$

$$w_0=\sum_{j=1}^{N} y_j \alpha_j^0 x_j \tag{2}$$

$(w_0 \cdot x)=\sum_{i=1}^{n} w_0^i \cdot x^i$ is the inner product of the two vectors, where $w_0=(w_0^1,w_0^2,\ldots,w_0^n)$ and $x=(x^1,x^2,\ldots,x^n)$. The vector W_0 can be determined according to the following quadratic programming.

Maximum

$$W(\alpha)=\sum_{i=1}^{N}\alpha_i-\tfrac{1}{2}\sum_{i,j=1}^{N} y_i y_j \alpha_i \alpha_j (x_i \cdot x_j)$$

Subject to

$$\sum_{j=1}^{N} y_j \alpha_j = 0; \quad C \ge \alpha_i \ge 0, \quad i=1,2,\ldots,N \tag{3}$$

where C is a positive constant. The constant b_0 is given by

$$b_0 = y_i - \left(x_i \sum_{j=1}^{N} y_j \alpha_j^0 x_j \right) \tag{4}$$

Substituting (2) for w_0 in (1), we have

$$f(x)=\sum_{i=1}^{N} y_i \alpha_i^0 (x_i x) + b_0 \tag{5}$$

We can know the separability of two subsets through checking whether the following inequalities

$$y_i(w_0 x_i - b) \ge 1; \quad i=1,2,\ldots,N \tag{6}$$

hold well.

2.2. Generalization in feature space[3]

Practically the performance of SVMs based on the previous section may not be very good for the nonlinear-separable cases in the original space. To improve the performance and to reduce the computational load for the nonlinear separable datasets, Vapnik extended the SVMs from the original space to the feature space.

The key idea of the extension is that an SVM first maps the original input space into a high-dimensional feature space through some nonlinear mapping, and then constructs an optimal separating hyper-plane in the feature space. Without any knowledge of the mapping, the SVM can find the optimal hyper-plane by using the dot product function in the feature space. The dot function is usually called a kernel function. According to Hilbert–Schmidt theorem, there exists a relationship between the original space and its feature space for the dot product of two points.

That is

$$(z_1 \cdot z_2) = K(x_1,x_2) \tag{7}$$

where it is assumed that a mapping Φ from the original space to the feature space exists, such that $\Phi(x_1)=z_1$ and $\Phi(x_2)=z_2$, and $K(x_1,x_2)$ is conventionally called a kernel function satisfying the Mercer theorem. Usually the following three types of kernel functions can be used: polynomial with degree p, radial basis function and sigmoid function. Replacing the inner product $(x_1 \cdot x_2)$ in (5) with the kernel function $K(x_1,x_2)$, the optimal separating hyper-plane becomes the following form:

$$f(x)=\sum_{i=1}^{N} y_i \alpha_i^0 K(x_i,x)+b_0 \tag{8}$$

It is worth noting that the conclusion of section 2.1 is still valid in the feature space if we substitute $K(x_1,x_2)$ for the inner product $(x_1 \cdot x_2)$.

2.3. Solving the basic problem of SVMs with SMO

Due to the solving of the basic problem of SVMs need to be called for many times, so a fast algorithm—Sequential

Minimal Optimization(SMO) is used in our method.

The SMO algorithm searches through the feasible region of the dual problem and maxmizes the objective function:

$$\begin{aligned} &max \quad W(\alpha)=\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j K(x_i,x_j) \\ &s.\text{t.} \quad \sum_{i=1}^{l}\alpha_i y_i=0, \\ &\qquad C\geq\alpha_i\geq 0, \qquad i=1,\dots,l. \end{aligned} \tag{9}$$

Sequential Minimal Optimization (SMO) is a simple algorithm that can quickly solve the SVM QP problem without any extra matrix storage and without using numerical QP optimization steps at all. SMO decomposes the overall QP problem into QP sub-problems, using Osuna's theorem to ensure convergence.[8]

Unlike other methods, SMO chooses to solve the smallest possible optimization problem at every step. For the standard SVM QP problem, the smallest possible optimization problem involves two Lagrange multipliers, because the Lagrange multipliers must obey a linear equality constraint. At every step, SMO chooses two Lagrange multipliers to jointly optimize, finds the optimal values for these multipliers, and updates the SVM to reflect the new optimal values.

The advantage of SMO lies in the fact that solving for two Lagrange multipliers can be done analytically. Thus, numerical QP optimization is avoided entirely.

3. An inverse problem of SVMs[3]

3.1. The description of the inverse problem of SVMs

For a given dataset of which no class labels are assigned to instances, we can randomly split the dataset into two subsets. Suppose that one is the positive instance subset and the other is the negative instance subset, we can calculate the maximum margin between the two subsets according to Procedure 1 where the margin is equal to 0 for the non-separable case. Obviously the calculated margin depends on the random split of the dataset. Our problem is how to split the dataset such that the margin calculated according to Procedure 1 attains the maximum.

It is an optimization problem. We mathematically formulate it as follows. Let

$S=\{x_1,x_2,\dots,x_N\}$ be a dataset and $x_i\in R^n$ for $i=1,2,\dots,N$, Ω={f | f is a function from S to {-1,1}}. Given a function $f\in\Omega$, the dataset can be split to two subsets and then the margin can be calculated by Procedure 1. We denote the calculated margin (the functional) by Margin (f). Then the inverse problem is formulated as

$$\text{Maximum}_{f\in\Omega}(\text{Margin}(f)). \tag{10}$$

3.2. The solving of the inverse problem of SVMs

In reference [3], the authors use genetic algorithm to solve the inverse problem of SVMs. Due to the high time complexity, we can use clustering to solve this problem.

Given a training sample set with n samples. Solving the inverse problem need to partition them into two classes randomly, i.e. a class label (1 or -1) is assigned to each sample. The partition will be gotten after the class label assignment. For each partition, the basic problem of SVMs needs to be solved to get the margin value. If the enumerating method is used, there will be 2^n assignment, the partition will be $2^{n-1}-1$ by ignoring the symmetrical partition and the partition that all the samples are assigned with 1 or -1. That means the basic problem of SVMs need to be solved for $2^{n-1}-1$ times, the time complexity is very high.

So the clustering is used in our algorithm. All the samples are clustered into m clusters at first, and then the class label will be assigned to each cluster, i.e. the samples in the same cluster will be assigned with the same class label. So there will be 2^m assignment and the corresponding partion will be $2^{m-1}-1$, this make the times for solving the basic problem of SVMs decrease sharply.

The algotithm is as follows:

1. The training set is devided into m clusters by k-means clustering.

2. The samples in each cluster are assigned with class label, let $Maxm\arg in$ be the max margin and set theinitial to be 0.

3. for each partition $P_i(i=1,\dots,2^{m-1}-1)$,

(1)Solve the basic problem of SVMs, record the corresponding margin.

(2)If $\text{m}\arg\text{in} > \text{Maxm}\arg\text{in}$

Then $\text{Maxm}\arg\text{in} = \text{m}\arg\text{in}$

4. Return $\text{Maxm}\arg\text{in}$ and the corresponding partition.

4. The induction of decision trees based on large margin heuristic

4.1. The main idea

Due to the high generalization capability of SVMs, we can apply the SVMs inverse problem to the decision tree induction process. We construct a decision tree in the process of maximizing the margin of each split. For a given training set, initially we do not consider its class labels, the dataset can be split into two subsets by solving the inverse problem of SVM. The subsets can be considered as the new branches. The instances labeled with -1 can be considered as the left branch, and the instances labeled with +1 can be considered as the right branch. The hyper-plane function with the maximum margin is recorded. It is used as the decision function. When we classify the new instances, the outcomes of the decision function will decide the branches they belong to. If the outcome of the new coming instance is less than 0, it belongs to the left branch, else it belongs to the right one.

4.2. Induction algorithm

Given a training set where the attributes have numerical values and a truth level threshold β, the induction process consists of the following steps:

Step1: Create a root node that contains all the instances for the tree. Split the instances into two subsets as the new branches according to the procedure of solving the inverse problem of SVMs. The hyper-plane with the maximum margin is recorded at the root node.

Step2: For each branch of the above node, calculate the proportions of all objects within the branch belonging to each class. If the proportion is above the given threshold β, terminate the branch as a leaf. Otherwise, continue to split the branch using the method in step1.

Step3: Repeat step2 for all newly generated decision nodes until no further growth is possible, the decision tree then is complete.

Take the two-class problem for example, the specific algorithm is as follows:

```
MaximumMargin(Examples, $\beta$)
Examples is the training examples set, $\beta$ is the threshold of the proportion for the examples belonging to each class. The decision tree that can classify the given examples correctly will be returned.
● Create the Root node of the decision tree
● If the proportion of the Examples with positive class is bigger than $\beta$, then return the tree consisted of Root node with label =+.
● If the proportion of the Examples with negative class is bigger than $\beta$, then return the tree consisted of Root node with label =-.
● Otherwise,
    ● The solving of the inverse problem of SVMs is performed on Examples and the partition with the max margin will be gotten.
    ● Let the hyperplane be $f(x)$, if $f(x) \le 0$, set $c(x) = -1$, if $f(x) \ge 0$, set $c(x) = +1$.
    ● Add two branches under the Root node, they are $f(x) \le 0$ and $f(x) > 0$ respectively.
    ● Set $Examples_{-1}$ is the subset which satisfies $f(x) \le 0$ in Examples.
      Set $Examples_{+1}$ is the subset which satisfies $f(x) > 0$ in Examples.
    ● Add the subtrees MaximumMargin($Examples_{-1}$, $\beta$)
      and MaximumMargin( $Examples_{+1}$ , $\beta$ )
      under the two branches.
● End.
Return Root.
```

5. The induction algorithm of binary decision tree based on minimum entropy heuristic

Suppose D is the training set in the form of numerical, the induction algorithm is described as follows[7]:

Step1 If all the instances in D belong to one class or satisfy other terminating rule, for example, the proportion of all instances within the branch belonging to one class is above a given threshold, D is terminated as a leaf node labeled that class. Otherwise, turn to step2;

Step2 A non-leaf node is produced, D is split into n (for binary decision tree, n=2) example sets, labeled with D_i ($i = 1, 2, \ldots n$). For each D_i , execute step1.

The heuristic information used in step2 is minimum entropy. Let $A_1, A_2, \ldots, A_m$ be all the attributes, $v_{i1}, v_{i2}, \ldots, v_{in}$ be the values for A_i .For each value of each attribute v_{ii} , we can split D into two subsets- D_1 and D_2 , D_1 is the subset of instances with the value for A_i less than or equal to v_{ii} , D_2 is the subset of instances with the value for A_i more than v_{ii} .So we can obtain n_i splits for attribute A_i ,where n_i is the number of values of A_i . For each split, we can calculate the entropy. We choose the minimum entropy which is labeled E_i for attribute A_i , then the attribute with the minimum entropy, i.e. $\min(E_i)$, is chosen to be the expended attribute.Experiments and discussion

6. The experiment and discussion

We apply the SVM into the decision tree induction, i.e. using the max margin as the heuristic to induce the decision tree. This algotithm and the one that uses minimal entropy as the heuristic are both for the data with real type. We have implemented the two decision tree induction system.

We choose Iris, Rice, Pima, Image Segment from UCI database and Table 1 shows the features of each database. Using these data we perform the experiment and compare the test accuracy. For the multi-class database, we just choose two of them. The experimental steps are as follows:

1. Data preprocessing. Let the class attribute of the data be -1 and 1, and divide them into two parts randomly, 70% as the training set and 30% as the testing set.
2. Set the corresponding parameters, the parameters and their values are listed in Table 2.
3. Train the data using the induction algorithm based on large margin, and induce the decision tree.
4. Use the testing data to test, get the testing accuracy.

Table 3 shows the experimental results, i.e. the comparison of the average testing accuracy between the two algorithms, where the algorithm1 represents the binary decision tree induction algorithm introduced in section 5 and the algorithm2 represents the one introduced in section 4.2, the time is the time spent by algorithm 2.

Table 1. The features in the database

The database	The Sample number	The attribute number
Iris	100	5
Rice	98	6
Image Segment	661	19
Pima	768	9

Table 2. The parameters of the algorithm

k =4	Clusters number
two_sigma_squared=2	RBF kernel function parameters
$tolerance = 0.001$	KKT tolerance degree
$eps = 0.001$	the mulitipliers updating threshold in SMO
$\beta = 0.9$	The threshold of leave nodes
C=10000	The punishment factor

Table 3. The comparison of the testing algorithm

Database	Algorithm 1	Algorithm 2	Time
Iris	97.8%	100%	31ms
Rice	79.4%	88.89%	625ms
Image Segment	100%	100%	3421ms
Pima	66.09%	73.0%	600000ms

The result illustrates that the training accuracy and testing accuracy has been improved by using the maximum margin heuristic. The generalization ability has been improved at some extent.

The concept of margin of a classifier is the central to the success of a new classification learning algorithm [3]. Generalization capability of SVMs depends strongly on the margin [3]. Increasing the margin has been shown to be able to improve the generalization capability of SVMs through data-dependent structural risk minimization [5].In the induction process we use the maximum margin as the heuristic information, i.e., The margin of the split during the induction of decision tree attains the maximum., So the split with the maximum margin has good generalization.

7. Conclusions

In order to generate decision trees with higher generalization capability, this paper proposes a decision tree induction algorithm based on large margin. The clustering algorithm for solving the inverse problem of SVMs is chosen during the induction. The main disadvantage of this algorithm is still its large time complexity. Practically it is expected to give an improved version or another algorithm with time complexity reduction [3].

Ackonwlegements

This research is supported by the natural science foundation of Hebei Province (F2008000635) and is supported in part by Hebei Provincial Education Department (No. 2008306)

References

[1] Quinlan J R. “Induction of decision tree”, Machine Learning, 1986, 1: 81~106.

[2] T. M. Mitchell, “Machine Learning”, The McGraw-Hill Co.,414p, 1997.

[3] Xi-zhao Wang, Qiang He, De-Gang Chen, Daniel Yeung, “A genetic algorithm for solving the inverse problem of support vector machines”, Neurocomputing 68(2005):225-238.

[4] V.N. Vapnik, “Statistical Learning Theory”, Wiley, New York, ISBN 0-471-03003-1, 1998.

[5] V.N. Vapnik, “An overview of statistical learning theory”, IEEE Trans. Neural Networks 10 (5) (1999): 88–999.

[6] V.N. Vapnik, “The Nature of Statistical Learning Theory”, Springer, New York, ISBN 0-387-98780-0, 2000.

[7] Jie Yang, Chen-zhou Ye, Yue Zhou , Nian-yi Chen, “On Upper Bound of VC Dimension of Binary Decision Tree Algorithms”, Computer Simulation, 2005,2(22):74-78.

[8] John C. Platt, “Sequential Minimal Optimization A Fast Algorithm for Training Support vector Machines”[R]. Cambridge: Microsoft Research, 1998.