RESEARCH ARTICLE OPEN ACCESS

Comparision of methods for combination of multiple classifiers that predict behavior patterns

Anuja V. Deshpande*, Prof. Dharmeshkumar Mistry**
*(Department of Computer Science, D.J Sanghvi College of Engineering, Mumbai-56)
** (Department of Computer Science, D.J Sanghvi College of Engineering, Mumbai-56)

ABSTRACT
Predictive analysis include techniques fromdata mining that analyze current and historical data and make predictions about the future. Predictive analytics is used in actuarial science, financial services, retail, travel, healthcare, insurance, pharmaceuticals, marketing, telecommunications and other fields.Predicting patterns can be considered as a classification problem and combining the different classifiers gives better results. We will study and compare three methods used to combine multiple classifiers. Bayesian networks perform classification based on conditional probability. It is ineffective and easy to interpret as it assumes that the predictors are independent. Tree augmented naïve Bayes (TAN) constructs a maximum weighted spanning tree that maximizes the likelihood of the training data, to perform classification.This tree structure eliminates the independent attribute assumption of naïve Bayesian networks. Behavior-knowledge space method works in two phases and can provide very good performances if large and representative data sets are available.
Keywords– Bayesian networks, Behavior-knowledge space, TAN.

I. INTRODUCTION

There is an increasing demand for predicting behavior in all industries. Ecommerce, Credit risk, Insurance fraud are just some of the applications which benefit from behavior prediction. Predicting behavior patterns can be seen as a classification problem. It has been observed that a combination of different classifiers produces better results as compared to that of a single classifier. Different classifiers usually have different methodologies and features, which generally complement each other. Hence, cooperation between these classifiers can be optimized to reduce errors and increase the performance of classification. The paper is divided as follows. We first understand the problem of combining different classifiers, in section 2, we discuss the various methods for combining classifiers and in section 3, we make a comparative study of these methods. Finally, in section 4 conclusions are given.If e_k denotes classifier k (where k=1,….K) and k is the total number of classifiers[4]. Let $C_{1,}...C_M$ be mutually exclusive and exhaustive set of patterns and M represents the total number of pattern classes[4]. A= {1, …., M} be a set which consists of all class index numbers.[4] x is the unknown input pattern and $e_k(x)=j_k$ means classifier k assigns input x to class $j_k \in A \cup \{M+1\}$. If $j_k \cup A_1$, it means classifier k accepts x, otherwise it rejects x[4]. The combination problem can then be stated as,"When k classifiers give their individual decisions to the unknown input, what is the method which can combine them efficiently and produce the final decision?"[4]. It can be formulated as:

$$\text{given} \quad \left.\begin{array}{rcl} e_1(x) & = & j_1 \\ e_2(x) & = & j_2 \\ & \vdots & \\ e_K(x) & = & j_K \end{array}\right\} \xrightarrow{?} E(x) = j \qquad (1)$$

Where E is the panel of multiple classifiers which gives z one definitive class j[4].

Various methods have been proposed to combine classifiers. In this paper we will study and compare three types of classifier combining methods i.e. Bayesian networks, tree augmented naïve Bayes (TAN) and Behavior-Knowledge Space method.

II. Methods for Combining Multiple Classifiers

2.1 Naïve Bayes Network

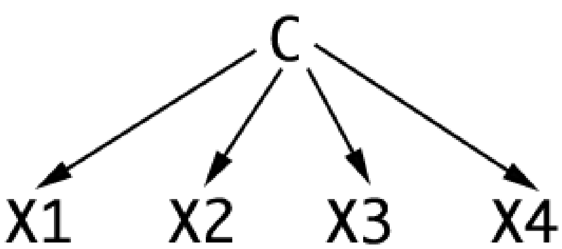

Fig 1. Naïve Bayes Network

A Bayesian network is also called as Bayes network, belief network, Bayesian model or probabilistic directed acyclic graphical model. It is a probabilistic graphical model (a type of statistical model) that represents a set of random variables and their

conditional dependencies via a directed acyclic graph (DAG).For example For example, a Bayesian network could represent the probabilistic relationships between diseases customer behavior and their predilection to buy a product. Given certain attributes, the network can be used to compute whether a customer is likely to buy a given product.
Conventionally, Bayesian networks are DAGs whose nodes represent random variables. These variables may be observable quantities, latent variables, unknown parameters or hypotheses. Edges represent conditional dependencies and nodes that are not connected represent variables that are conditionally independent of each other. The probability function associated with each node takes as input a particular set of values for the node's parent variables and gives the probability of the variable represented by the node. For example, the probability function could be represented by a table of 2^m entries, one entry for each of the 2^m possible combinations of its parents being true or false, if the parents are m Boolean variables. Advantages of this approach are that it is easy to understand and fast to train. The probability of classifier selection of class j where $1 \le j \le M$ as its classified class when true class was class i where $1 \le i \le M$ is defined as:

$$P(x \in C_i \mid e_k(x) = j) = \frac{n_{ij}(k)}{\sum_{i=1}^{M} n_{ij}(k)} \quad (2)$$

Where $e_k(x)$ is a class label selected by classifier k as the true class for an input x[3]. The belief function for class i can be expressed by the sum of conditional probabilities as follows:

$$\mathrm{BEL}(i) = \eta \prod_{k=1}^{K} p(x \in C_i \mid e_k(x) = j),$$
$$\text{for } i = 1, \ldots, M \quad (3)$$

where η is a normalization coefficient that satisfies $\sum_{i=1}^{M} \mathrm{BEL}(i) = 1$ [3].The belief function BEL(i) is the product of the contributions from all classifiers for class i, and represents the total validity for class i.[3] Taking the class label whose BEL value is the largest makes the final decision[3]. The combining rule is shown below:

$$F(x) = \begin{cases} j & \text{if } \mathrm{BEL}(j) = \max_{i \in \Lambda}(\mathrm{BEL}(i)) \text{ and } \\ & \mathrm{BEL}(j) \ge \alpha (0 < \alpha \le 1) \\ \text{reject} & \text{otherwise} \end{cases} \quad (4)$$

This approach assumes that all features are independent of each other. No structure learning procedure is required and hence this structure is easy to construct and works efficiently. To allow more complex networks, the Tree Augmented Naïve Bayes (TAN) network is proposed.
2.2 Tree Augmented Naïve Bayes Networks (TAN)

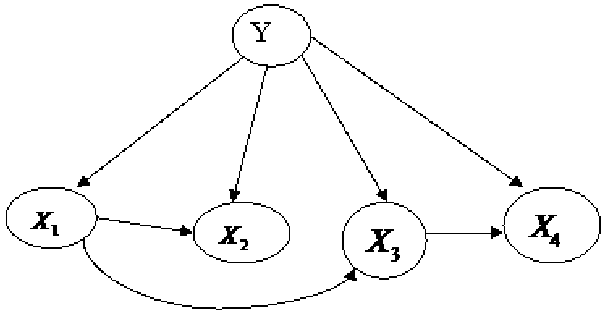

Fig 2. A TAN Bayes net

The figure depicts a TAN Bayes net for combining multiple classifiers. Here, X1, X2, X3, X4 represent the different rules. The TAN model, while retaining the basic structure of Naïve Bayes, also permits each attribute to have at most one other parent, allowing the model to capture dependencies between attributes[1]. Which arcs to include in the 'augmented' network is decided by the algorithm by making a complete graph between all the non-class attributes, where the weight of each edge is given as the conditional mutual information between those two attributes. A maximum weight spanning tree is constructed over this graph, and the edges that appear in the spanning tree are added to the network[1]. Given the independence assumptions in the tree T , the posterior probability is:

$$P(C_i|x_1, \ldots, x_n) \propto P(C_i) \prod_k P(x_k C_i) \quad (5)$$

where $x_j(k)$ stands for the parent of variable x_k in the tree T , and x0 for the null. We now need to keep a counter for the number of training instances, a counter for each class label, and a counter for each attribute value, parent value and class label triplet[2]. TAN maintains the computational simplicity if Naïve Bayes while increasing the performance.
2.3 Behavior-knowledge Space Method.
This method has two stages (1) the knowledge modeling stage, responsible for extracting knowledge from behavior of classifiers and constructing a K-dimensional behavior-knowledge space; and (2) the operation stage that is carried out for each test sample and which combines decisions generated from individual classifiers, enters a specific unit of the constructed space, and makes a final decision by a rule which utilizes the knowledge inside the unit [4].
A behavior-knowledge space (BKS) is a K-dimensional space where each dimension

corresponds to the decision of one classifier and has M+1 possible decision values from the set {1,2, …., M+1}[4]. If the decision of the classifier belongs to the set {1, …., M}, the classifier accepts the input sample, else the classifier rejects the input sample [4]. Each unit of the BKS contains three kinds of data (1) the total number of incoming samples, (2) the best representative class and (3) the number of samples belonging to each class [4].

The first i.e. the knowledge modeling stage, uses the learning set of samples with the expected class labels and the recognized class labels to construct the BKS [4]. The values of T and R are computed as follows:

$$T_{e(1)\ldots\ldots e(K)} = \sum_{m=1}^{M} n_{e(1)\ldots\ldots e(K)}(m)$$

$$R_{e(1)\ldots\ldots e(K)} = \{j \mid n_{e(1)\ldots\ldots e(K)}(j) = \max_{1 \le m \le M} n_{e(1)\ldots\ldots e(K)}^{(m)}\}$$

(6)

Where,

$n_{e(1)\ldots\ldots e(K)}^{(m)}$= the number of incoming samples belonging to class m in BKS

$T_{e(1)\ldots\ldots e(K)}$=total number of samples in BKS

$R_{e(1)\ldots\ldots e(K)}$= best representative class for the BKS[4].

In the operation stage, the final decision is made by the following rule:

$$E(x) = \begin{cases} R_{e(1)\ldots\ldots e(K)}, & \text{when } T_{e(1)\ldots\ldots e(K)} > 0 \text{ and } \dfrac{n_{e(1)\ldots\ldots e(k)}(R_{e(1)\ldots\ldots e(K)})}{T_{e(1)\ldots\ldots e(k)}} \ge \lambda, \\ M+1, & \text{otherwise} \end{cases} \quad (7)$$

Where λis a threshold which controls the reliable degree of the decision [4].

Table 1.Two dimensional behavior knowledge space

e(1)/*e*(2)	1	…	*j*	…	11
1	(1,1)	…	(1,*j*)	…	(1,11)
:	:	:	:	:	:
i	:	:	(*i*,*j*)	:	:
:	:	:	:	:	:
11	(11,1)	…	(11,*j*)	…	(11,11)

each cell in the table means the intersection of the decision values from the individual classifiers and becomes a basic unit of computation in BKS approach[3].

III. A Comparative Study

3.1. Independent clause assumption

A naïve Bayesian network needs the clauses to be independent. There may be strong dependencies among clauses that are not realized. Tree augmented naïve Bayes (TAN) allows us to capture the dependencies between different attributes and thus, improves performance of the former. But, even TAN has limitations. It allows each attribute to have at most one parent, thus restricting dependency. Behavior-knowledge space approach does not assume independencies among classifiers at all and hence, performs better in most cases than Bayesian networks and Tree augmented Bayesian networks.

3.2. Working

In naïve Bayesian networks, the outcome of each clause is represented as a random variable, the value of which depends on the examples classification. The tree augmented naïve Bayes algorithm works by making a complete graph between all the non-class attributes, where the weight of each edge is given as the conditional mutual information between those two attributes [1]. A maximum weight spanning tree is constructed over this graph, and the edges that appear in the spanning tree are added to the network [1]. In contrast to these approaches which rely on making trees, the Behavior-knowledge space approach works in two stages. First, by constructing a knowledge space and second, by making the final decision in the operation stage.

3.3. Performance

Naïve Bayesian networks assume classifier independency and are thus easy to interpret but inefficient. They are also too cautious about classifying something as positive [1]. TAN excel in handling imprecise rules and provide an advantage in situations with imprecise rules and a preponderance of negative examples, such as these link discovery domains [1]. The Behavior-knowledge space method can provide very good performances if large and representative data sets are available [5]. Otherwise over fitting is likely to occur, and the generalization error quickly increases [5].

IV. Conclusion

The assumptions, working and performance of three different approaches has been discussed. The Naïve Bayesian network, easiest to understand is very ineffective. The tree augmented naïve Bayes improves the performance of naïve Bayes to some extent, but still does not completely eliminate classifier independency assumption. The Behavior-knowledge space method overcomes this limitation and also has adaptive learning ability. It can also automatically find out the best threshold for a given required performance. Future works can include attempts to combine the different methods suggested here to achieve better results.

REFERENCES

Journal Papers:

[1] Jesse Davis, Vitor Santos Costa, Irene M. Ong, David Page and Ines Dutra, Using Bayesian Classifiers to Combine Rules

[2] Josep Roure Alcobé, Incremental Learning of Tree Augmented Naive Bayes Classifiers

[3] Eunju Kim, Wooju Kim, Yillbyung Lee, Combination of multiple classifiers for the customer's purchase behavior prediction, *Decision Support Systems 34* (2002) 167–175

Proceedings Papers:

[4] Huang, Y.S., Suen, C.Y., The behavior-knowledge space method for combination of multiple classifiers,*Computer Vision and Pattern Recognition, 1993. Proceedings CVPR '93., 1993 IEEE Computer Society Conference*, 1993, 347-352.

Chapters in Books:

[5] Terry Windeatt, Fabio Roli, The Behavior Knowledge Space Fusion Method: Analysis of Generalization Error and Strategies for Performance Improvement, *Multiple Classifier Systems*, (4th International Workshop, MCS 2003 Guildford, UK, June 11–13, 2003 Proceedings, Springer Berlin Heidelberg).