WEIGHTS OF TESTS

Vesela Angelova

Abstract: *Terminal test is subset of features in training table that is enough to distinguish objects in classes. Tests are used as supporting sets in classification algorithms of type estimate calculation or voting. We propose ideas for valuation of votes and for reasoning of different weights of votes. The modifications for weights are proposed that are based on representativeness of test, distribution of votes from objects, dependency of features and coding of tests.*

Keywords: *classification, voting, tests, combinatory, coding theory.*

1. Introduction

1.1. Classification as part of machine learning (or pattern recognition), attempts to assign each input record to one of given classes. One type of learning procedure uses a training set of pre-existing patterns or instances, which have been labeled by hand with the correct class. The data-set used for training consists of vectors of observed features (variables) and class-label. This training set produces a rule (or classifier) which can evaluate a class-label for new input records of variable measurements.

One of the classification methods with training table uses a subset of the variables such that the class-label of the records in the training data can be distinguished only by them. We call such subsets tests. If the classes are disjoint, the full set of variables is a test.

Irreducible or *terminal test* (TT) is a set of variables, such that it is a test and removing any variables from it would make the resulting set non-test [Zhuravlev,1980]. *Length* of TT is the number of variables in the set. TT is a fraction, sensitive to class-label, partial and minimal description of differentiation for all classes (2 or more). TT makes alone assumption for affiliation (membership) of record to class and this is a property learned by the training data.

If a new vector with the same variables is given by measurements of this variables, according to a number of coincidences for all TTs, we can establish the similarity to each class and choose join (recognize) to top rated class, with voting scheme. In the standard approach one coincidence between the pattern to be classified (new object) and the object of training table for TT is one vote for class labeled this object.

1.2. For this type of discrete heuristic algorithm one problem to overcome is to find the full set TTs. This is done only once and brute force approaches perform well even for tens of features in the training vectors. Doing this would be useful anyway when there are not enough records for statistical measurements.

A problem we discuss is the weight of the votes for short and long TTs. Long tests are intersecting and numerous. This creates excess in their votes over a test which consists of one variable, although the former distinguishes classes alone. We search in several directions for numerical expression of TT' weights as far as reasoning for different weights. Indeed one justification is proposed in [Angelova,1987], where one vote of test reflects in proportion on weights of relevant variables.

2. Tests according to its representativeness

The set of all subsets of variables may view as flower bud:

n number of subsets with length 1

$\frac{n(n-1)}{2}$ number of subsets with length 2

...

n number of subsets with length *(n-1)*

1 number of subsets with length *n*.

The votes from TTs should not be with equal weights, because each one has different base of reducible or non-terminal tests, which part is this TT. With standard voting TT with length one and all its extensions distinguish classes, but have negligible contribution for belonging to class. This type of pattern recognition independent of the length is unbalanced. One instrument for measure the weight of TT is to count all tests, representing by TT.

Let have training table with *n* variables for objects, labeled to classes.

For TT with length 1 (subset of one variable) we can find extended non-terminal tests representing by TT:

with length *2* -- *(n-1)* different choices for the second variable,

with length *3* -- $\frac{(n-1)(n-2)}{2}$ different choices for two variables,

....

with length *k* -- $\frac{(n-1)(n-2)\ldots(n-(k-1))}{(k-1)\ldots 2.1}$,

…

with length 1+*(n-2)* -- $\frac{(n-1)(n-2)\ldots(n-(n-2))}{(n-2)\ldots 2.1} = n-1$,

with length 1+*(n-1)* = *n* -- one choice or full set of variables.

TT with length *k* represents such pyramid of tests with truncated top of length *k* :

with length *к+1* -- *(n-k)* different choices for adding another variable to set –TT,

with length *к+2* -- $\frac{(n-k)(n-k-1)}{2}$,

...

with length *k+(n-k)* = *n* , one choice.

The picture is like this: the TT' variables set is extended by one until it reaches the base of full variable set. Thus every TT is peak (acute or truncated) of open leaf of bud flower. This point of view gives metric concerning TT representativeness with regard to reducible (non-terminal) tests.

Statement: The numerical expression of representativeness of TT with length *k* is 2^{n-k}.

Really, let count for each TT' length number of representing tests in whole leaf:

For TT with length 1: $1 + (n-1) + \frac{(n-1)(n-2)}{2} + \ldots + (n-1) + 1 = 2^{n-1}$

For TT with length 2: $1 + (n-2) + \frac{(n-2)(n-3)}{2} + \ldots + (n-2) + 1 = 2^{n-2}$

…

For TT with length k: $1+(n-k)+\frac{(n-k)(n-k-1)}{2}+\ldots+(n-k)+1=2^{n-k}$.□

Ratio is not depending on the number of TTs $b_1, b_2, \ldots, b_k$ respectively with length 1, 2,..., κ so we can normalize obtaining weights:

$2^{n-k}/x = b_1.2^{n-1} + b_2.2^{n-2}+\ldots b_k.2^{n-k})/100$ gives the weight of the lightest and longest TT (with length 2^{n-k}) and the percentage when $2^{k-1} : 2^{k-2} : \ldots : 2:1$.

At first glance it seems the weight of short TT is greater, but long ones are more and partly repeating due to intersecting of variables. Thus relative differences according to aggregate weights are more acceptable.

Example 1:

```
class K1          (1,1,1,0,1,0,1,1,0,0,1),
                  (1,0,1,1,0,1,1,1,1,0,1),
                  (0,1,1,0,0,0,1,0,1,1,0),
class K2          (0,1,0,0,1,0,0,1,0,1,1),
                  (1,0,0,1,0,1,0,0,0,1,1),
                  (0,0,0,1,1,0,0,1,0,0,0).
New pattern       (0,0,1,1,1,0,1,0,0,0,1)
```

Number of all TTs is 16, sorted by length:

	1	2	3	4	5	6	7	8	9	10	11	length	weight
01	0	0	1	0	0	0	0	0	0	0	0	– 1	8
02	0	0	0	0	0	0	1	0	0	0	0	– 1	8
03	1	0	0	0	0	0	0	1	0	0	0	– 2	4
04	0	0	0	0	0	0	0	0	0	1	1	– 2	4
05	1	1	0	0	0	0	0	0	1	0	0	– 3	2
06	1	0	0	1	0	0	0	0	1	0	0	– 3	2
07	1	0	0	0	1	0	0	0	1	0	0	– 3	2
08	1	0	0	0	1	0	0	0	0	1	0	– 3	2
09	1	0	0	0	0	1	0	0	1	0	0	– 3	2
10	1	0	0	0	0	0	0	0	1	1	0	– 3	2
11	0	1	0	0	1	0	0	0	0	1	0	– 3	2
12	0	1	0	0	0	0	0	0	1	1	0	– 3	2
13	0	0	0	1	1	0	0	0	0	1	0	– 3	2
14	0	0	0	1	0	0	0	0	1	1	0	– 3	1
15	0	1	0	0	0	1	0	1	0	1	0	– 4	1
16	0	0	0	1	0	1	0	1	0	1	0	– 4	1

Total: 16 Sum of weights: 46

When each vote is regular (weight=1) the new pattern obtains 9 votes from K1 and 12 - from K2 (the possibility is from 3 objects to receive maximal 16 votes or total 48 votes from class), therefore is recognized as belonging to class K2.

When each vote is according representativeness of TT, the same new pattern obtains 60 votes from K1 and 24 votes - from K2 (maximum is 3.46=138 votes from class), therefore is recognized as belonging to K1 according to coincidences with measurements for TT consisting of one variable (variable 3 and variable 7).

3. Distribution of the objects in class

When a new object is joining to a class some short TTs may disappear and spread into few longer ones. This is another circumstance supporting our idea for alignment weight of short TT to weights of several derived from him, **as if they tend to keep total weight**.

How is the new object recognized before joining is essential question. If it is recognized with an advantage to a class by the votes of all objects in this class, then the number of TTs is changing less (respectively sum of weights), the over-fit is hold. If it is recognized to a class by average votes, but major differences between objects in this class, the increment of TT' number is larger. We can imagine in space model like adding to a group a new point, which is out of this group and near some participants. Adding this new point to a group expands differences (TTs) by number and lengths.

The recognition quality would be better when fewer and shorter TTs are used (therefore weighty), with uniform distribution of objects' votes. If these requirements are not fulfilled, the questions raise about more classes or distant characteristics of objects.

Example 2:

The new objects `(1,0,0,1,0,0,1,1,0,1,0)` is recognized according above mentioned training table, receiving:

- from class K1 - average 12,6 weighted votes (14,12,12), and non-weighted 2,3 (2,3, 2)

- from class K2 - average 14 weighted votes (8,26,8) , and non-weighted 3 (1,10,1).

If this new object is joining to K2, because of votes by only one participant, the TTs' number raises more than joining to K1. In both cases the new object breaks one short TT (length one).

We can conclude for this new object together with nearest object to make new class.

4. How to use dependency (intersecting) between TT

It is clear that only TTs with length one are independent, another ones form some groups of intersecting variables. Two approaches are possible for this problem – to represent depending feature like **common unit**, or to focus on the features out of this unit like **unique base** for weight of TT.

4.1. We can choose some subsets of variables, which are part of intersecting TTs, but are not TT itself. In these we find representative sets – i.e. different combinations, typical for the class, which are not found in other classes. The example shows the most frequent occurrence of a subset of features 1 and 9, and the measurement for these features (1, 1) and (0, 1) are typical of class K1, and (0, 0) is typical for K2. (But (1, 0) for the same features 1 and 9 occurs in both classes, this measurement is not representative sets, that's why the features 1 and 9 do not form TT). If there is not typical measurement for one class, anti-closeness is possible decision.

Quite often are found subsets {2, 10}, {4, 10}, {5, 10}, {9, 10}.

There are subsets that are not found in any test and may be excluded from consideration - for example {2, 4}, {5, 6} or (5, 8}.

4.2. Another approach is to determine the coverage of each TT upon reducible tests from level 4-features, but those that are covered in one way by TT. In this way we do slit of all subsets of some level, in the case of 4-

variables tests and give weight of TT on top of each of these sets, but without overlap. The task is solved by software.

№	3	7	11	8	6	2	4	5	**1**	**9**	10	#4-variables, uniquely covered	#3-variables
01	1	0	0	0	0	0	0	0	0	0	0	60	34
02	0	1	0	0	0	0	0	0	0	0	0	60	34
04	0	0	1	0	0	0	0	0	0	0	1	14	7
03	0	0	0	1	0	0	0	0	1	0	0	14	7
09	0	0	0	0	1	0	0	0	**1**	**1**	0	1	1
05	0	0	0	0	0	1	0	0	**1**	**1**	0	1	1
06	0	0	0	0	0	0	1	0	**1**	**1**	0	1	1
08	0	0	0	0	0	0	0	1	**1**	**1**	0	1	1
10	0	0	0	0	0	0	0	0	**1**	**1**	1	0	1
08	0	0	0	0	0	0	0	1	1	0	1	1	1
13	0	0	0	0	0	0	1	1	0	0	1	2	1
14	0	0	0	0	0	0	1	0	0	1	1	2	1
11	0	0	0	0	0	1	0	1	0	0	1	2	1
12	0	0	0	0	0	1	0	0	0	1	1	2	1
15	0	0	0	1	1	1	0	0	0	0	1	1	0
16	0	0	0	1	1	0	1	0	0	0	1	1	0

The dependencies between the features become apparent after length 2 of TT when trying to line-up of the triangular matrix and the found coating on level 4 and 3- variables indicate that TT with length 1 and 2 are crucial for classification (left table). On the other hand the unique base of TT (right columns) consists of very different number of reducible tests.

5. Code-words for tests

Another point of view is to represent TT like codeword. The *coding* is transformation to another alphabet; our goal is to compress the description of TTs (see the table in example 1). We have a table, each row is TT with 1 for involved variable and 0 for not-involved. This table is binary even for k-meaning variables. Thus the length of TT (number of variables) is *Hamming weights* of codeword and we look for big weight of TT with small Hamming weight of its code-word.

Assume that there is at least one 1 for variable, otherwise this variable is removed. No one description is part of other by definition of TT.

Statement: The *Hamming distance* between two code-words is at least 2.

Really, if some description t_i is part of (or involved in) another t_j, then t_j is not TT. Therefore t_i has at least one 1, for which t_j has in the same place 0. Because opposite statement is also true, they have at least two mismatches.□

Let sort the rows by its Hamming weights or number of ones. We can cut short each description to last 1, and the set of code-words is prefix-free binary code, by definition of TT no one contains in another. But we can truncate these descriptions or code-words more forward by rearrangement of variables (or columns of table). If we put first the variables with 1 for the shortest TT and cut the description after this 1, and if we continue in the same manner for the next lengths of TTs, we receive shortest code-words for TT with highest weight. Code-words for TT with weight 1 (for example its number is k) have different length (from 1 for the first TT to k, k≤n for the last with length 1). We continue with length 2 taking into account to put forward the variables with more contributions or 1s in column. By increasing the length of TT more and more their 1s are pulled and the lengths of code-words like to be the same (restrict by n) but the code-words differ in 2 positions at least. Constructing a tree confirms the necessary of different weights for TTs, namely *weighted path length (depth)* of code is minimal when codeword length is short for maximum weight (usually proportional to probabilities).

Statement: If we take into account proposed weights, precisely $p_1 \geq p_2 \geq \ldots \geq p_s$. (Indeed $p_i=2.p_j$ for test *i* with length by 1 shorter than length of test j), we can generate shortest description of set of all TTs.

Really, we can swap TT with equal length because of equal weight, although the code-word lengths are not equal. By finite number of attempts this swapping produces minimal total length of code-words in 3-dimensional space because of distance (unlike Huffman code in the plane where two code-words have distance one).□

Thus TTs minimal coding shows that the dividing hyper-surface between classes consists of units, low-dimensional are high-significant.

Example 3:

For our training table and table of TTs, one possible coding is:

	3	7	11	10	1	8	9	5	2	4	6
01	1										
02	0	1									
04	0	0	1	1							
03	0	0	0	0	1	1					
10	0	0	0	1	1	0	1				
07	0	0	0	0	1	0	1	1			
08	0	0	0	1	1	0	0	1			
05	0	0	0	0	1	0	1	0	1		
11	0	0	0	1	0	0	0	1	1		
12	0	0	0	1	0	0	1	0	1		
06	0	0	0	0	1	0	1	0	0	1	
13	0	0	0	1	0	0	0	1	0	1	
14	0	0	0	1	0	0	1	0	0	1	
15	0	0	0	1	0	1	0	0	1	0	1
16	0	0	0	1	0	1	0	0	0	1	1

This procedure provides minimal prefix-free coding of TTs, but not to be presented each other or to perform actions.

6. Conclusion

Updated software with proposed weight of TT in section 2 can be used for voting in such areas as recognition quality assessment software [Eskenazi,1990] or survey [Angelova,2008], where there are insufficient standards for statistics. The ideas from section 3, 4, 5 indicate directions for future work.

Acknowledgements

The paper is published with financial support by the project ITHEA XXI of the Institute of Information Theories and Applications FOI ITHEA (www.ithea.org) and the Association of Developers and Users of Intelligent Systems ADUIS Ukraine (www.aduis.com.ua).

Bibliography

[Zhuravlev,1980] Yu. I. Zhuravlev et al., "Recognition and classification problems with standard testing information", J. of Computing and mathematical Physics (Jurnal vichislitel'noi matematiki i matematicheskoi fisiki), 1294-1309, vol.20 No 5, 1980 (in Russian).

[Angelova,1987] V. Angelova, "Weighted estimation of characteristics", Mathematics and mathematical education, XVI UBM spring conference, 1987, pp.301-305 (in Bulgarian).

[Eskenazi,1990] A. Eskenazi, V. Angelova, "A New Method for Software Quality Evaluation", J. of New Generation Computing Systems, 3 (1990) 1, 47-53.

[Angelova,2008] V. Angelova, A. Eskenazi, "Application of classification methods to a problem related to specific groups of e-government users", Serdica J. Computing 2 (2008), 101-112.

Authors' Information

Vesela Angelova –*Software Engineering Department, Institute of Mathematics and Informatics*

Acad. Bonchev Str., Bl. 8, Sofia-1113, Bulgaria; e-mail: vaa@math.bas.bg

Major Fields of Scientific Research: Software technologies, Pattern recognition