An Optimal Weighting Criterion of Case Indexing for Both Numeric and Symbolic Attributes

Takao Mohri and Hidehiko Tanaka
Information Engineering Course, Faculty of Engineering
The University of Tokyo
7-3-1 Hongo Bunkyo-ku, Tokyo 113, Japan
{mohri,tanaka}@MTL.T.u-tokyo.ac.jp

Abstract

Indexing of cases is an important topic for Memory-Based Reasoning(MBR). One key problem is how to assign weights to attributes of cases. Although several weighting methods have been proposed, some methods cannot handle numeric attributes directly, so it is necessary to discretize numeric values by classification. Furthermore, existing methods have no theoretical background, so little can be said about optimality. We propose a new weighting method based on a statistical technique called Quantification Method II. It can handle both numeric and symbolic attributes in the same framework. Generated attribute weights are optimal in the sense that they maximize the ratio of variance between classes to variance of all cases. Experiments on several benchmark tests show that in many cases, our method obtains higher accuracies than some other weighting methods. The results also indicate that it can distinguish relevant attributes from irrelevant ones, and can tolerate noisy data.

Introduction

Indexing of cases is an important topic for both Case-Based Reasoning(CBR) and Memory-Based Reasoning(MBR) (Stanfill & Waltz 1986). Indexing is especially important in MBR because of the lack of case adaptation phase, and the nearest instances' classes are directly mapped to a new instance. Usually, similarity is calculated by summing up weights of matched attributes. Therefore, weighting attributes is one of the key points of case indexing. Good attribute weighting can eliminate the effects of noisy or irrelevant attributes.

Several weighting methods for attributes have been proposed, but few address the two following points:

- Handling both numeric and symbolic attributes.
- Defining optimality criteria.

We propose a new weighting method based on a statistical technique called Quantification Method II. This method solves the two problem: It can handle both numeric and symbolic attributes in the same framework, and produces weights that are optimal as they maximize the ratio of variance between classes to variance of all cases.

Typical Weighting Methods for Attributes

Per-/Cross-Category Feature Importance

Before explaining our method, let us review some typical attribute-weighting methods.

Per-category feature importance and *cross-category feature importance* (in short, PCF/CCF) were proposed in (Creecy *et al.* 1992). Both weighting methods are based on conditional probabilities. In the case of PCF/CCF, a symbolic attribute with N values is converted to a set of N binary attributes.

Suppose u means a case stored in the database, and v is a case for query. c_u denotes the class of the case u. u is stored in the database, so the class of u is known already. N_C is the number of classes,and N_a is the number of attributes. Then the definitions of similarity using the per-/cross-category feature importance are as follows:

Per-category feature importance:

$$Similarity(u,v) = \sum_{a}^{N_a} w(c_u,a)\delta(u_a,v_a)$$

$$where \quad w(c,a) = P(c|a)$$

$$\delta(x,y) = \begin{cases} 0 & if(x=y) \\ 1 & if(x \neq y) \end{cases}$$

Cross-category feature importance:

$$Similarity(u,v) = \sum_{a}^{N_a} w(a)\delta(u_a,v_a)$$

$$where \quad w(a) = \sum_{i=1}^{N_C} P(c_i|a)^2$$

When an attribute a and a class c have high correlation, the conditional probability $P(c|a)$ is high, therefore, the weights of the attributes also become high.

While *per-category feature importance* addresses the classes of training data, *cross-category* neglects them, and uses an average (in fact, summation of square) as weight.

We have applied these *per-/cross-category weighting methods* for a weather prediction task using memory-based reasoning (Mohri, Nakamura, & Tanaka 1993). In that paper, we show *per-category importance* is too sensitive to the proportion of classes, and has a tendency to answer the most frequent class too often.

Value Difference Metric

Stanfill and Waltz proposed the *Value Difference Metric* (in short, VDM)(Stanfill & Waltz 1986), and applied it to the English word pronunciation problem. VDM defines distance between cases as follows:

$$distance(u,v) = \sum_{a=1}^{N_a} w(a,u_a)\delta(a,u_a,v_a)$$

$$where \quad w(a,p) = \sqrt{\sum_{c=1}^{N_c}\left(\frac{C_a(p,c)}{C_a(p)}\right)^2}$$

$$\delta(a,p,q) = \sum_{c=1}^{N_c}\left(\frac{C_a(p,c)}{C_a(p)} - \frac{C_a(q,c)}{C_a(q)}\right)^2$$

where p and q are possible values of an attribute, $C_a(p)$ is the total number of times that value p occurred at an attribute a, and $C_a(p,c)$ is the frequency that p was classified into the class c at an attribute a.

In VDM, the distance between values is calculated for every pair of values for every attribute. Each attribute is weighted by $w(a,u_a)$.

Recently, Cost and Salzberg proposed MVDM (Modified VDM) (Cost & Salzberg 1993). MVDM omits the attribute-weighting (the term $w(a,u_a)$) of VDM, and introduces weighting of cases. MVDM performs well for some tests including prediction of protein secondary structure and pronunciation of English text. However, we have not tested the weighting of cases, because we believe that weighting of cases should be discussed separately from weighting of attributes.

IB4

Aha's IB4(Aha 1989; 1992) is an incremental instance-based learning algorithm. IB4 has many features including noise-tolerance, low storage requirement, and an incremental attribute-weighting function. Attribute weights are increased when they correctly predict classification and are otherwise decreased. Weights are calculated for each concept and each attribute, so its similarity is concept-dependent.

IB4 can handle both numeric and symbolic attributes. Numeric values are linearly normalized to $[0,1]$. The distance between symbolic values is the Hamming distance. The similarity function of IB4 is as follows:

$$Similarity(c,u,v) = -\sqrt{\sum_{a=1}^{N_a} Weight_{c_a}^2 \cdot diff(u_a,v_a)^2}$$

$$diff(u_a,v_a) = \begin{cases} \delta(u_a,v_a) & if\ attribute\ a \\ & is\ symbolic \\ |u_a - v_a| & if\ attribute\ a \\ & is\ numeric \end{cases}$$

Problems of These Weighting Methods

The weighting methods described above have some drawbacks:

- The way to handle both symbolic and numeric attributes

  PCF/CCF and VDM are basically for symbolic attributes, so in order to handle numeric attributes, values must be quantized into discrete intervals. A subsequent difficulty is that the discretization method tends to be ad hoc. Moreover, when a discretization method is used, total order defined by the numeric attribute is lost. For example, an interval $(0,10)$ is divided into 10 intervals. Before discretizing, the distance between 0.5 and 1.5 is 9 times smaller than that between 0.5 and 9.5. After discretizing, however, the distance between the interval (0,1) and (1,2) is 1. It is equal to the distance between (0,1) and (9,10).

  IB4 can handle both numeric and symbolic attributes, but the distance between symbolic values are merely the Hamming distance. Meanwhile, PCF/CCF calculates different weights for each symbolic value, and VDM defines similarity from the frequency of symbolic attributes and classified results.

- No statistical optimality criteria

  Although the procedure to calculate weights is clear, there is no explanation about whether they are optimal in any sense. Only benchmark tests support any claim for utility. In fact, Bayesian classification is an optimal method. However, if independence between attributes is not assumed, then probabilities for combinations of any values in the all attributes must be known. Therefore, strict Bayesian classification is not practical. Although the independence between attributes is usually assumed, this assumption is often not satisfied.

A New Attribute-Weighting Method based on Quantification Method II

Basic Explanation of Quantification Method II

Quantification Method II[1](in short, QM2) is a kind of multivariate analysis used to analyze qualitative data,

[1]In Japan, the Quantification Method is famous and has been used since the 1950s. In Europe, USA and other

and used for supervised learning problems. Basically, its input and output variables are symbolic ones, but an extension is easy to use numeric variables as input data.

The strategy of QM2 is as follows:

1. quantify attributes
2. define a linear expression to calculate a criterion variable (in terms of multivariate analysis) for each case
3. decide the coefficients of the linear expression that makes the ratio of the variance between classes and the variance of all cases
4. At the prediction phase, QM2 predicts a criterion value of a query by using the linear equation and calculated coefficients. It selects the nearest class by comparing the criterion value with classes' averages of each class.

Let us show how QM2 works by a simple example. Consider possible values of symbolic *attr1* as {YES,NO}, *attr2* as {A,B,C}. *Attr3* is numeric. Suppose class is in {YES, NO}.

Table 1: A Simple Example

class	attr1	attr2	attr3
YES	YES	A	10.3
YES	NO	B	12.5
YES	YES	A	8.6
NO	NO	B	9.4
NO	NO	C	8.4

At the quantification phase of attributes, each symbolic attribute is replaced with multiple new attributes, the number of which is that of possible values of the original attribute. If the value of the original attribute is the i-th symbol, then the value of the i-th new attribute is set to 1, and the remaining new attributes are 0. That is to say, N binary attributes are used instead of a symbolic attribute with N values. (In fact, N-th binary attribute is redundant. When solving an eigenvalue problem explained later, N-1 binary attributes are used). Numeric attributes are not changed at all. In the case of the example, Table 1 is converted to Table 2.

Next, a linear equation is assumed to calculate criterion variable y_{c_i} for all cases. Cases can be divided into M groups by the class to which they belong. Suppose c is an index of such groups, i is an index of a case in a group, and n_c is the size of each group. w_a is a coefficient for the a-th attribute.

countries, it may be better known as "correspondence analysis" (J.P.Benzécri et al. 1973) by J.P.Benzécri of France (Hayashi, Suzuki, & Sasaki 1992)

Table 2: A Simple Example (After Quantification)

class	attr1		attr2			attr3
	u_1	u_2	u_3	u_4	u_5	u_6
1	1	0	1	0	0	10.3
1	0	1	0	1	0	12.5
1	1	0	1	0	0	8.6
0	0	1	0	1	0	9.4
0	0	1	0	0	1	8.4

For each case, a criterion variable y_{c_i} is calculated by the following equation:

$$y_{c_i} = \sum_{a=1}^{N_a} w_a u_a$$

where N_a is the number of quantified attributes. In the example above, $N_a = 6$.

Then, from y_{c_i} of these cases, the variance of all cases σ^2 and the variance between groups σ_B^2 can be calculated as follows:

$$\eta^2 = \frac{\sigma_B^2}{\sigma^2}$$

$$\sigma^2 = \frac{1}{N}\sum_{c=1}^{M}\sum_{i=1}^{n_c}(y_{c_i} - \overline{y})^2$$

$$\sigma_B^2 = \frac{1}{N}\sum_{c=1}^{M} n_c(\overline{y}_c - \overline{y})^2$$

$$where \begin{cases} \overline{y}_c \equiv \frac{1}{n_c}\sum_{i=1}^{n_c} y_{c_i} \\ \overline{y} \equiv \frac{1}{N}\sum_{c=1}^{M} n_c \overline{y}_c \end{cases}$$

The Quantification Method II provides the coefficient w_a, which maximizes the ratio η^2. This problem results in an eigen value problem of a $N_a \times N_a$ square matrix calculated by u_a of all instances. The elements of the eigen vector become w_a. For detailed calculation, please refer to (Kawaguchi 1978).

New Classification Methods based on Quantification Method II

We introduce three methods based on Quantification Method II. The first two (QM2m, QM2y) are instance-based methods, and the rest is the original QM2 itself.

- **QM2m** (QM2 + matching)
  Each attribute has its weight calculated by QM2. Numeric attributes are directly used for matching. In the case of symbolic attributes, quantified binary attributes are used for matching. The absolute values of w_a are used as weights. The similarity between two cases is calculated as follows:

$$Similarity(u, v) = \sum_{a=1}^{N_a}(|w_a| \cdot |u_a - v_a|)$$

- **QM2y** (QM2 + matching by y_{c_i})
  The values y_{c_i} are computed for all training cases before testing. When testing, y_{c_i} of the test case is calculated and used for matching. The case whose y_{c_i} is the nearest to that of the test case is selected, and its class is assigned to the test case. The similarity between two cases is calculated as follows:

$$\begin{aligned} Similarity(u,v) &= |y_u - y_v| \\ &= \left| \sum_{a=1}^{N_a} w_a u_a - \sum_{a=1}^{N_a} w_a v_a \right| \end{aligned}$$

- **QM2**
  QM2 is the direct use of the naive Quantification Method II, and is not an instance-based method. For each class, the average of the criterion variable $\overline{y_c}$ is calculated from training cases beforehand. For a query, a criterion variable y of the test case is calculated, and the class whose criterion value is the nearest is selected as an answer.

QM2m uses the absolute values of the coefficient w_a as weights of attributes. This seems useful because the coefficients indicate the importance of the attributes. In general, the coefficient may be a positive or negative value, so we take their absolute values. The advantages of these QM2-based methods over other weighting methods are as follows:

- It supports both numeric and symbolic attributes in the same framework.
  No symbolization or normalization of numeric values are necessary. All weights for both numeric and symbolic attributes are calculated by the same procedure, so it can identify useless attributes.
- It is based on a statistical criterion.
  The weights of attributes are optimal in the sense that they maximize the variance ratio η^2. Therefore, they have a theoretical basis and clear meaning.

Experiments

The experimental results for several benchmark data are shown in Table 3. Four data sets (*vote, soybean, crx, hypo*) were in the distribution floppy disk of Quinlan's C4.5 book (Quinlan 1993). The remaining four data sets (*iris, hepatitis, led, led-noise*) were obtained from the Irvine Machine Learning Database (Murphy & Aha 1994).

Including our 3 methods,VDM, PCF, CCF, IB4, and C4.5 are compared. Quinlan's C4.5 is a sophisticated decision tree generation algorithm, and used by default parameters. The accuracies by pruned decision trees are used in the experimental results.

All accuracies are calculated by 10-fold cross-validation (Weiss & Kulikowski 1991). The specifications of these benchmark data are shown in Table 4. *led* has 7 relevant Boolean attributes, and 17 irrelevant (randomly assigned) Boolean attributes. The attributes of *led-noise* is same as *led*, but 10 % of attribute values are randomly selected and inverted(i.e., noise is 10%).

In the case of PCF, CCF and VDM, numeric attributes must be discretized. In the following experiments, normal distribution is assumed for data sets, and the numeric values are quantized into five equal-sized intervals.

In the case of QM2 methods, eigen values are calculated, and each eigen vector is treated as a set of coefficients w_a. At the weight calculations of *soybean, led, led-noise* data, multiple large eigen values were found. The number of such eigen values are 18, 7 and 7 respectively. In the case of such data sets, w_a for each eigen value are used to calculate scores, and those scores were summed to a total score. In the current program, the largest eigen value is used and, if the next largest eigen value is larger than half of the former one, w_a derived from this eigen value is also used. This setting is ad hoc.

Discussion

In the experimental results, accuracies of the QM2 family (QM2m, QM2y, QM2) is higher or comparable to other weighting methods. Especially, QM2 becomes the best method at four benchmark tests, and scores among the top three at six tests. However, QM2 gets an extremely low accuracy at the *hypo* test. The analysis of this result is undergoing. One more claim is that in *led*, 100% accuracy could be achieved by our methods. It indicates that the QM2 family can distinguish relevant attributes from irrelevant ones. The result of *led-noise* shows that the QM2 family can tolerate noisy data.

One obvious drawback is that weight calculation of our methods is computationally expensive. Therefore, it is not economical to use the QM2 family incrementally. Table 5 shows the calculation time for weights. Our methods take between about ten to a hundred times longer than PCF/CCF and VDM. However, let us note two points. Firstly, for such benchmarks, calculation time is at most several seconds. It is not excessive, especially when weights can be calculated beforehand. Secondly, our method achieve better accuracy, so weight calculation time may be a reasonable cost to bear.

Table 5: Weight Calculation Time [sec]

Method	iris	vote	soy-bean	crx	hypo	hepa-titis
VDM	0.00	0.02	0.08	0.03	0.22	0.01
PCF/CCF	0.00	0.01	0.07	0.02	0.19	0.01
QM2	0.03	0.97	6.45	1.79	5.20	0.22

Table 3: Experimental Results

	iris		vote		soybean		crx		hypo		hepatitis		led		led-noise	
1	QM2	98.0	QM2	95.0	QM2y	93.4	IB4	86.7	C4.5	99.6	CCF	81.3	QM2y	100.0	QM2	74.3
2	QM2y	96.7	IB4	94.7	QM2	93.4	CCF	83.7	PCF	92.3	QM2y	80.6	QM2m	100.0	IB4	68.9
3	QM2m	95.3	QM2m	94.3	CCF	92.2	VDM	83.3	QM2m	91.0	QM2	80.6	QM2	100.0	QM2m	66.5
4	CCF	94.7	VDM	93.0	VDM	91.8	QM2m	82.9	VDM	90.7	C4.5	80.0	VDM	100.0	VDM	65.5
5	C4.5	94.7	QM2y	92.3	QM2m	91.2	QM2	82.9	CCF	89.2	PCF	79.4	C4.5	100.0	QM2y	65.0
6	VDM	94.7	C4.5	92.3	IB4	90.3	C4.5	82.7	QM2y	85.8	VDM	79.4	IB4	99.8	C4.5	64.3
7	PCF	92.7	CCF	91.0	C4.5	89.6	QM2y	82.4	IB4	67.7	IB4	78.7	CCF	94.8	PCF	54.7
8	IB4	81.3	PCF	88.0	PCF	51.0	PCF	80.0	QM2	50.2	QM2m	76.8	PCF	65.1	CCF	54.7

Table 4: Specification of Benchmark Data

	iris	vote	soybean	crx	hypo	hepatitis	led(-noise)
# of data	150	300	683	490	2514	155	1000
# of numeric attributes	4	0	0	6	7	6	0
# of symbolic attributes	0	16	35	9	29	13	24

Our method has a statistical optimality criterion which is to maximize η^2, the ratio of variance between groups σ_B^2 to total variance σ^2. Although this criterion seems to be close to the optimality of accuracy, it's not obvious. To clarify the relation between this criterion and accuracy needs further research.

Conclusions and Future Work

We proposed a new attribute-weighting method based on a statistical approach called Quantification Method II. Our method has several advantages including:

- It can handle both numeric and symbolic attributes within the same framework.
- It has a statistical optimal criterion to calculate weights of attributes.
- Experimental results show that it's accuracy is better than or comparable with other weighting methods, and it also tolerates irrelevant, noisy attributes.

Many things are left as future work including experiments on other benchmark tests, to clarify the bias of these algorithms(i.e. the relation between our algorithms and natures of problems), combination of case-weighting methods, and so on.

Acknowledgements

Special thanks to David W. Aha for helpful advice and giving us a chance to use his IBL programs. Also thanks to the anonymous reviewers for many useful comments to improve our draft. This research is supported in part by the grant 4AI-305 from the Artificial Intelligence Research Promotion Foundation in Japan.

References

Aha, D. W. 1989. Incremental, instance-based learning of independent and graded concept descriptions. In *Proceedings of the Sixth International Machine Learning Workshop(ML89)*, 387–391.

Aha, D. W. 1992. Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms. *Int. J. Man-Machine Studies* 36:267–287.

Cost, S., and Salzberg, S. 1993. A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning* 10:57–78.

Creecy, R. H.; Masand, B. M.; Smith, S. J.; and Waltz, D. L. 1992. Trading MIPS and memory for knowledge engineering. *Communications of the ACM* 35(8):48–63.

Hayashi, C.; Suzuki, T.; and Sasaki, M., eds. 1992. *Data Analysis for Comparative Social Research.* North-Holland. chapter 15, 423–458.

J.P.Benzécri et al. 1973. L'analyse des données 1,2(paris: Dunod)(in French).

Kawaguchi, M. 1978. *Introduction to Multivariate Analysis II (in Japanese).* Morikita-Shuppan.

Mohri, T.; Nakamura, M.; and Tanaka, H. 1993. Weather forecasting using memory-based reasoning. In *Second International Workshop on Parallel Processing for Artificial Intelligence (PPAI-93)*, 40–45.

Murphy, P. M., and Aha, D. W. 1994. UCI repository of machine learning databases. Irvine, CA: University of California, ftp://ics.uci.edu/pub/machine-learning-databases.

Quinlan, J. R. 1993. *C4.5: Programs for Machine Learning.* Morgan Kaufmann.

Stanfill, C., and Waltz, D. 1986. Toward memory-based reasoning. *Communications of the ACM* 29(12):1213–1228.

Weiss, S. M., and Kulikowski, C. A. 1991. *Computer Systems That Learn.* Morgan Kaufmann.