SYMBOLIC FEATURES IN NEURAL NETWORKS

Włodzisław Duch, Karol Grudziński and Grzegorz Stawski[1]
Department of Computer Methods, Nicolaus Copernicus University
ul. Grudziądzka 5, 87-100 Toruń, Poland

Abstract:
Neural networks and many other systems used for classification and approximation are not able to handle symbolic attributes directly. A method of replacing the symbolic values of attributes by numerical values is described here. The probabilistic Value Difference Metric (VDM) and similar functions used in the Nearest-Neighbor algorithms in most cases work better than functions that can handle only numerical attributes. Such metric functions may be directly applied to symbolic data. Our method is based on converting the symbolic attribute values into an array of conditional probabilities of getting each output class given the attribute that has a particular symbolic value. Benchmark tests on artificial and real datasets compare standard approach of converting symbolic features into numbers, which is often performed without any justification, to a method proposed in this paper.

1 INTRODUCTION

Neural networks, multivariate statistical methods, pattern recognition and machine learning methods need numerical values as inputs. Usually symbolic features are converted into numbers without any justification. Some symbolic values have natural ordered, for example *small, medium, large*, and the assignment of numbers to symbolic values should reflect it. In other cases symbolic values cannot be ordered and the assignment is done in a random way.

It is easy to see that the complexity of a classification system, such as a neural network, depends on this assignment. Consider for example the *color* attribute with 6 values: *red, orange, yellow, green, blue, violet*, ordered here according to the wavelength. Classification of the reception fields of the 3 cones in a photoreceptor, each with a peak sensitivity between red-orange, yellow-green and blue-violet, requires two cuts (or two units in a neural classifier) if the colors are ordered from 1 to 6, separating those < 3 and those > 4 from those in te middle. Any other assignment will require more separating cuts (planes in statistical discrimination methods or units in neural networks), for example if *red*=1, *orange*=4, *yellow*=2, *green*=5, *blue*=3 and *violet*=6 five cuts are needed. If there are subtypes of colors, for example “yelow bahama", they will not be handled easily by systems using random assignment of colors to numbers, lowering the accuracy of the system on the test cases. Several attributes with symbolic values assigned randomly may lead to a very difficult classification problem, while a proper assignment make greatly reduced the complexity of the problem. What is the “proper" way of assigning the numerical values to symbolic attributes?

[1] E-mails: {duch,kagru,staw}@phys.uni.torun.pl, WWW: http://www.phys.uni.torun.pl/kmk

An approach originally developed in machine learning community and used in the instance based learning [1] may be helpful to answer this question. The idea is to replace a symbolic feature with an array of conditional probabilities of obtaining each class given the attribute that has a particular symbolic value. In case of instance based methods like the k-Nearest-Neighbor (k-NN) algorithm the results obtained on the original symbolic data using the VDM [2] similarity function are equivalent to the results achieved on data replaced with conditional probabilities with distances computed with Euclidean function. For neural classifiers there is no such relationship. Many variants of VDM-like functions can be used to select the best similarity function on a training set and use it for the test set.

In the next section the methods designed to handle symbolic features in the domain of instance based learning are reviewed [1]. In section 3 our approach to use symbolic features in classifiers requiring numerical inputs is presented. In the subsequent section the method proposed is compared with random assignment of numerical values on several classification problems. Short discussion of the results concludes this paper.

2 SUPPORT FOR SYMBOLIC FEATURES IN INSTANCE BASED METHODS

In this section we briefly review some of the methods allowing to handle symbolic attributes in instance based learning (IBL). IBL is a paradigm of learning in which some or all instances (training cases) are stored in memory during learning and distance (similarity) functions are used to determine how close an unknown (query) input case is to each stored case and use one or more nearest instances to predict the output class. This approach to classification has gained wide attention and is commonly referred to as k-Nearest Neighbor classification (k-NN) or instance based learning. In the case of k-NN algorithm usually all training samples are stored, forming the reference set of prototype cases. The distance from each case $\mathbf{x}$ given for classification to all reference cases is computed and the k shortest distances are found. The predicted class of the unknown case $\mathbf{x}$ is found taking the most common class among the k nearest neighbors. IBL algorithms use smart indexing techniques and their reference memory is organized usually in a form of a tree to avoid computing all the distances from training samples. Parallel computing techniques may easily be exploited in IBL. Both k-NN methods as well as other IBL algorithms choose a very simple knowledge representation for cases, most often in a form of vectors of numerical or symbolic attributes.

Three families of distance functions used in IBL that provide support for symbolic attributes are presented below. The first metric, called 'Heterogeneous Euclidean-Overlap Metric' (HEOM) [6], is similar to the metric used in IB1, IB2 and IB3 systems [1]. The second, called the Value Difference Metric (VDM) metric [2], has several variants (DVDM, IVDM, HVDM) described in details by Wilson and Martinez [6]. The third, called the Minimum Risk Metric (MRM) metric, was introduced recently by Blanzieri and Ricci [7], but since it cannot be used with our approach it will only be briefly mentioned.

2.1 Heterogeneous Euclidean-Overlap Metric (HEOM)

This HEOM metric function is defined as

$$HEOM(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{a=1}^{m} d_a\left(x_a, y_a\right)^2} \tag{1}$$

where

$$d_a(x_a,y_a) = \begin{cases} 1, & \text{if } x_a \text{ or } y_a \text{ is unknown, else} \\ 1-\delta(x_a,y_a), & \text{if } a \text{ is symbolic, else} \\ rn\_df_a(x_a,y_a) & \end{cases} \tag{2}$$

The function $1-\delta(x_a,y_a)$ and the range-normalized difference $rn\_df_a(x_a,y_a)$ are defined as:

$$1-\delta(x_a,y_a) = \begin{cases} 0, & \text{if } x_a = y_a \\ 1, & \text{otherwise} \end{cases} \tag{3}$$

$$rn\_df_a(x_a,y_a) = \frac{|x_a - y_a|}{range_a} \tag{4}$$

The value $range_a = \max_a - \min_a$ is used to normalize the attributes, where $\max_a$ and $\min_a$ are the maximum and minimum values of the attribute a found in the training set.

2.2 VDM-type metrics

VDM metric is defined as the Minkovsky combination of :

$$D_V^q(\mathbf{x},\mathbf{y}) = \sum_{a=1}^{N} vdm_a(x_a,y_a)^q \tag{5}$$

where q is a positive constant (1 for Manhattan, or 2 for Euclidean metrics) and the distance between two values x and y of the attribute a is defined as:

$$vdm_a(x,y)^q = \sum_{c=1}^{C} \left| \frac{N_{a,x,c}}{N_{a,x}} - \frac{N_{a,y,c}}{N_{a,y}} \right|^q = \sum_{c=1}^{C} |P(c|a=x) - P(c|a=y)|^q \tag{6}$$

where

- C is the number of classes in $\mathcal{T}$;
- $N_{a,x,c}$ is the number of vector in $\mathcal{T}$ that have the value x for the attribute a and the vector belongs to the class c;
- $N_{a,x}$ is the number of vectors in the training set $\mathcal{T}$ that have the value x for the attribute a, equal to the sum of $N_{a,x,c}$ over all classes $c = 1 \ldots C$;
- $P(c|a=x)$ is the conditional probability that the output class is c given that attribute a has the value x. This probability is estimated using the training set $\mathcal{T}$ is $P(c|a=x) = N_{a,x,c}/N_{a,x}$.

Distance is defined here via a data-dependent matrix with the number of rows equal to the number of classes and the number of columns equal to the number of all attribute values. Generalization for continuous values requires a set of probability density functions $p_{ij}(x)$, with $i = 1, \ldots, C$ and $j = 1, \ldots, N$.

The $vdm_a(x, y)$ distance is zero if for the values x and y of the attribute a probabilities of obtaining the same class are identical. This is intuitively correct; in our example with 6 color values the data set should contain examples of the first class (cone 1) for the values red and orange, second class for yellow and green and third class for blue and violet, giving proper ordering. For several interacting attributes VDM metric cannot be easily justified.

Wilson and Martinez extended VDM to numeric attributes [6]. They essentially discretize the numeric attributes (calling it the DVDM metric) or smooth the histogram estimation of $P(c_i|x_a)$ by averaging (calling it IVDM metric). In neural networks, statistical methods or decision trees such approach does not seem to be useful.

Minimum Risk Metric (MRM) directly minimizes the risk of misclassification. If $\mathbf{x}$ is a training sample from the c_i class and $\mathbf{y}$ is its nearest neighbor the risk of misclassifying $\mathbf{x}$ is given by $P(c_i|\mathbf{x})(1 - P(c_i|\mathbf{y}))$. The total finite risk is the sum of the risks summed over all classes. Blanzieri and Ricci [7] propose to minimize directly the risk $r(x, y)$ which leads to the metric:

$$MRM(\mathbf{x}, \mathbf{y}) = r(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{C} P(c_i|\mathbf{x})(1 - P(c_i|\mathbf{y})) \tag{7}$$

Unfortunately this formula is not additive in respect to attributes. MRM may easily be extended to include more than one neighbor or sum over distance-weighted neighbors to estimate the misclassification risk in a more realistic way, but so far such extensions have not been tried.

3 CHANGING SYMBOLIC TO NUMERICAL VALUES

The data-dependent distance functions defined above are directly applicable to any discrete attributes, symbolic or numerical. Empirical tests [6, 7] showed that these metric functions are highly competitive, frequently given much better results than standard metric functions (usually Euclidean or Manhattan) applied to vectors with symbolic values of attributes converted to numerical values. These metrics define certain topography in the input space. One way to preserve this topography in Euclidean spaces is to create vectors with the same distance relations as those computed with the data-dependent distance functions. Using generalization of Minkovsky metric:

$$D^q(\mathbf{x}, \mathbf{y}) = \sum_i d(x_i, y_i)^q \tag{8}$$

we need numerical values replacing symbolic values such that the contribution from $d(x_i, y_i)$ should be the same as calculated from the data-dependent distance functions. As long as these functions are calculated as the sum over components, for example the VDM:

$$D_V^q(\mathbf{x}, \mathbf{y}) = \sum_{a=1} \sum_{c=1}^{C} |P(c|a = x) - P(c|a = y)|^q \tag{9}$$

each symbolic feature x of an attribute a may be replaced by a C-dimensional vector of conditional probabilities $\mathbf{x}n = (xn_1, xn_2 \ldots xn_C) = (P(1|a = x), P(2|a = x) \ldots P(C|a = x))$. Defining

$$d(x, y) = \sum_{l=1}^{C} |xn_l - yn_l|^q \tag{10}$$

the results of VDM are reproduced. The same approach may be used with HEOM metric but not with MRM, since it is not additive. Attributes with symbolic values are thus replaced by C-dimensional subvectors, while numerical values are left unchanged. For two classes instead of two components of the $\mathbf{x}n$ only one component is sufficient to reproduce results:

$$\begin{aligned} d(x, y) = \quad & |P(1|a = x) - P(1|a = y)|^q + |P(2|a = x)| - P(2|a = y)|^q = \\ & 2|P(1|a = x) - P(1|a = y)|^q \end{aligned} \tag{11}$$

Thus for two classes the dimension of the input vector does not change, while for problems with more than two classes the dimensions may grow significantly. We have developed a program that converts the data from symbolic to numerical values. In the next section experimental results comparing the performance of three classifiers on data converted using the VDM metric and with random assignment of numerical values are described. The results of k-NN the effect of converting data into probabilities does not change the results of the nearest neighbor classifier as long as Euclidean function is used.

4 EXPERIMENTAL RESULTS

Experiments were made with four datasets containing symbolic attributes. Results obtained with the FSM and the k-NN classifiers (k=1 was used) are summarized in the table below. Results obtained with C4.5 were not significantly better and they show similar trend, comparable to the other two classifiers, therefore they are not reported here. All data was taken from the UCI repository [8].

The Australian Credit dataset has 690 cases, each with 6 continuous and 8 symbolic values. There are 2 classes (majority rate is 55.5%), no missing values. 10-fold cross-validation test were run therefore the variance could be calculated. Both FSM network and k-NN algorithm achieved similar results (within statistical accuracy) on the raw data as on the converted data (the number of attributes did not change here).

On the Flags dataset, containing 194 samples, 3 continuous and 25 symbolic attributes, 8 classes (majority rate 30.8%) and no missing values, 10-fold cross-validation tests were performed. Large improvement is observed for the FSM network on the converted data despite the fact that the dimensionality grew up from 28 to 203. k-NN result was only slightly better on the original rather than on the converted dataset.

The Monks-3 dataset contains 122 training cases with 6 discrete attributes, two classes (majority rate 50.8%), and no missing values. The tests set contains 432 examples. Both the FSM network and the k-NN algorithm improved their prediction ability on converted dataset.

Dataset	**FSM**		k-NN	
	raw	**converted**	**raw**	**converted**
Australian Credit	84.8% ± 1.0%	84.0% ± 0.5%	65.3% ± 0.8%	66.3% ± 0.5%
Flags	51.3% ± 2.4%	61.5% ± 2.6%	41.4% ± 1.7%	40.4% ± 1.3%
Monk 3	96.3% ± 1.2%	96.8% ± 0.1%	91.0%	93.8%

5 CONCLUSIONS

The problem of converting symbolic to numerical features is important for neural and many other types of classifiers. The algorithm proposed here has its origin in the instance based learning based on calculation of the conditional probabilities using VDM metric. A few conclusions may be drawn from the computational experiments performed so far. The method improves the prediction ability of neural network and the nearest neighbor classifiers, at least for some data sets. The improvements may differ, depending on the classification system used.

In the two-class problems the number of features does not grow, and therefore the method seems to be especially suitable for classifiers separating a single class from the rest. In problems with many classes with many symbolic attributes the dimension of the input vectors may grow significantly. This fact faced with the small number of samples may lead to the “curse of dimensionality" problem [9]. One way to reduce the dimensionality after conversion of the symbolic to numerical attributes that we are investigating at the moment is to use the principal component analysis.

REFERENCES

[1] D. W. Aha, D. Kibler and M. K. Albert. Instance-Based Learning Algorithms. *Machine Learning*, Vol. 6, pp. 37-66, 1991.

[2] C. Stanfill and D. Waltz. Toward memory-based reasoning. *Communication of the ACM*, 29, pp. 1213-1229, 1986.

[3] W. Duch, G. H. F. Diercksen. Feature Space Mapping as a universal adaptive system. *Computer Physics Communications* 87, pp. 341-371, 1995.

[4] W. Duch, R. Adamczak, N. Jankowski. *New developments in the Feature Space Mapping model*, 3rd Conf. on Neural Networks, Kule, Poland, Oct. 1997, pp. 65-70

[5] J.R. Quinlan. *C4.5: Programs for machine learning.* San Mateo, Morgan Kaufman 1993

[6] D. R. Wilson and T. R. Martinez. Improved heterogeneous distance functions. *Journal of Artificial Intelligence Research*, 11, pp. 1-34, 1997.

[7] E. Blanzieri and F. Ricci. Advanced Metrics for Class-Driven Similarity Search. International Workshop on Similarity Search, Firenze, Italy, September 1999.

[8] C.J. Mertz, P.M. Murphy, UCI repository of machine learning databases, available at the address
http://www.ics.uci.edu/pub/machine-learning-databases;

[9] C. Bishop, *Neural networks for pattern recognition*. Clarendon Press, Oxford 1995.