ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1662

Not Being at Odds with a Class: A New Way of Exploiting Neighbors for Classification

Myriam Bounhas[1] and **Henri Prade**[2] and **Gilles Richard**[3]

Abstract. Classification can be viewed as a matter of associating a new item with the class where it is the least at odds w.r.t. the other elements. A recently proposed oddness index applied to pairs or triples (rather than larger subsets of elements in a class), when summed up over all such subsets, provides an accurate estimate of a global oddness of an item w.r.t. a class. Rather than considering all pairs in a class, one can only deal with pairs containing one of the nearest neighbors of the item in the target class. Taking a step further, we choose the second element in the pair as another nearest neighbor in the class. The oddness w.r.t. a class computed on the basis of pairs made of two nearest neighbors leads to low complexity classifiers, still competitive in terms of accuracy w.r.t. classical approaches.

1 Introduction

Several classification methods rely on the idea that a new item x should be classified in the class w.r.t. which it appears to be the least at odds. Logical proportions [7] are Boolean expressions that link four Boolean variables through equivalences between similarity or dissimilarity indicators pertaining to pairs of these variables. Among logical proportions, the heterogeneous ones [7] provide a natural basis to build a global oddness measure of an item w.r.t. a multiset, by cumulating the oddness index w.r.t. different triples of elements in the multiset for the different features [3]. This has suggested the simple procedure of classifying a new item x into the class $\mathcal{C}$ which minimizes this global oddness measure. Moreover, it has been noticed that the oddness measure of an item w.r.t. a triple can be generalized to a subset of any size, as well as to numerical features [3]. We investigate here the use of oddness measures based on *selected* pairs, since our aim is to show that we can keep on with the same idea of oddness while lightening the computational cost. Doing so, we preserve the accuracy while further reducing the complexity by constraining the set of considered pairs. The idea is to constrain the choice of the *two* elements in the pairs. We study the options of using pairs including one nearest neighbour, and then two nearest neighbors [4].

2 Oddness of an item w.r.t. a multiset of values

In order to build an oddness index for a Boolean x w.r.t. a multiset of Boolean values $\{a_i \mid i \in [1,n]\}$, we look for a formula $F(a_1,\ldots,a_n,x)$ holding iff $a_1 = \ldots = a_n = 0$ and $x = 1$, or if $a_1 = \ldots = a_n = 1$ and $x = 0$. The oddness $odd(\{a_i|i \in [1,n]\}, x)$ of a *Boolean* x w.r.t. a multi-set of Boolean values $\{a_i \mid i \in [1,n]\}$ can be summarized as:

$$\neg(\bigvee_{i\in[1,n]} a_i \equiv x) \wedge \neg(\bigwedge_{i\in[1,n]} a_i \equiv x) \quad (1)$$

It is clear that *odd* holds true only when the value of x is seen as being *at odds* among the other values: x is the intruder in the multiset of values. In the case $n = 2$, $odd(\{a_1, a_2\}, x)$ is 0 if and only if the value of x is among the majority value in the multiset $\{a_1, a_2, x\}$. When $n = 3$, $odd(\{a_1, a_2, a_3\}, x)$ does not hold true only in the situations where there is a majority among values in a_1, a_2, a_3, x and x belongs to this majority (e.g. $odd(\{0, 1, 0\}, 0) = 0$), or when there is no majority at all (e.g. $odd(\{0, 1, 1\}, 0) = 0$).

Extension to *real values* is quite straightforward. Assuming that numerical features are renormalized between 0 and 1, we use the standard definitions of the logical connectives in the $[0, 1]$-valued Łukasiewicz logic [8]. Then, a graded counterpart to formula (1) is:

$$\min(|x - \max\{a_1,\ldots,a_n\}|, |x - \min\{a_1,\ldots,a_n\}|) \quad (2).$$

A natural extension $odd(\{\overrightarrow{a_i}|i \in [1,n]\}, \overrightarrow{x})$ to *vectors* with m features is to consider the sum componentwise of the *odd* values computed via expression (1) or (2) as:

$$\Sigma_{j=1}^{m} odd(\{a_i^j|i \in [1,n]\}, x^j) \quad (3),$$

where x^j is the j-th component of $\overrightarrow{x}$ and the a_i^j's are the j-th components of the vectors $\overrightarrow{a_i}$ respectively. If $odd(\{\overrightarrow{a_i}|i \in [1,n]\}, \overrightarrow{x}) = 0$, it means that no feature indicates that $\overrightarrow{x}$ behaves as an intruder among the a_i's. On the contrary, high values of $odd(\{\overrightarrow{a_i}|i \in [1,n]\}, \overrightarrow{x})$ means that, on *many* features, $\overrightarrow{x}$ appears as an intruder.

3 Global oddness measure for classification

Given a set $\mathcal{C} = \{\overrightarrow{a_i}|i \in [1,n]\}$ of vectors gathering examples of the same class, one might think of computing $odd(\mathcal{C}, \overrightarrow{x})$ as a way of evaluating how much $\overrightarrow{x}$ is at odds w.r.t. $\mathcal{C}$. An immediate classification algorithm would be to compute $odd(\mathcal{C}, \overrightarrow{x})$ for every class and to allocate to $\overrightarrow{x}$ the class which minimizes this number. Nevertheless, this number is not really meaningful when the size of $\mathcal{C}$ is large. Indeed, we have to be careful because then $\{\overrightarrow{a_i}|i \in [1,n]\}$ is summarized by two vectors made respectively by the minimum and the maximum of the feature values among the examples of $\mathcal{C}$ (due to expressions (2) and (3)). These two vectors have high chance to be *fictitious* in the sense that, usually, they are not elements of $\mathcal{C}$. Approximating our knowledge of the set $\mathcal{C}$ using only the maximal ranges of the feature values over the members of the set seems very crude. An idea is then to consider small subsets S of the class $\mathcal{C}$, then compute $odd(S, \overrightarrow{x})$ and finally add all these atomic oddness indices to get a global measure of oddness of $\overrightarrow{x}$ w.r.t. $\mathcal{C}$. This approach leads to the following initial formula: $\Sigma_{S\subseteq\mathcal{C},|S|=n}\, odd(S, \overrightarrow{x})$. To take into account the relative size of the different classes, it is fair to introduce a normalization factor and our final definition is:

$$Odd_n(\mathcal{C}, \overrightarrow{x}) = \frac{1}{\binom{|\mathcal{C}|}{n}} \Sigma_{S\subseteq\mathcal{C},|S|=n}\, odd(S, \overrightarrow{x})$$

[1] LARODEC Laboratory, ISG de Tunis, Tunisia & Emirates College of Technology, Abu Dhabi, United Arab Emirates, email: myriam_bounhas@yahoo.fr
[2] IRIT, University of Toulouse, France, & QCIS, University of Technology, Sydney, Australia, email: prade@irit.fr
[3] IRIT, University of Toulouse, France, & BITE, London, UK, email: richard@irit.fr

A classification algorithm follows, where we allocate to a new item x the class $\mathcal{C}$ minimizing $Odd_n(\mathcal{C}, \overrightarrow{x})$ for a given n, thus defining a family of classifiers also denoted Odd_n.

Algorithm 1 Oddness-based algorithm

Input: a training set TS of examples $(\overrightarrow{z}, cl(\overrightarrow{z}))$
  a new item $\overrightarrow{x}$,
  an integer n,
Partition TS into sets $\mathcal{C}$ of examples with the same label c.
for each $\mathcal{C}$ **do**
  Compute $Odd_n(\mathcal{C}, \overrightarrow{x})$ for subsets of size n
end for
$cl(\overrightarrow{x}) = argmin_c(Odd_n(\mathcal{C}, \overrightarrow{x}))$
return $cl(\overrightarrow{x})$

4 Experimentations

The experimental study is based on 15 datasets taken from the U.C.I. machine learning repository [6], applying standard 10 folds cross-validation technique. On these datasets, we compare classifiers Odd_1, Odd_2, Odd_3 accuracy to the one of state-of-the-art classifiers C4.5, SVMs, JRip and IBk (shown in Table 1). Our first experiments

Table 1. Classification accuracy of state-of-the-art classifiers

Datasets	C4.5	SVM	SVM	JRip	IBk
		Poly-Kernel	PUK-Kernel		(k=5,k=15)
Balance	78	90	89	76	84,88
Car	95	91	87	91	92,76
Monk1	99	75	100	98	100,100 -
Monk2	95	67	67	73	64,67
Monk3	100	100	100	100	99,98
Spect	81	81	83	81	80,79
Voting	96	96	96	95	93,91
Diabetes	74	77	77	76	73,74
Cancer	96	97	96	96	97,94
Heart	77	84	81	81	78,81
Magic	76	77	81	77	78 ,77
Ionosphere	91	88	94	88	85,84
Iris	96	96	96	95	95,96
Wine	94	98	99	93	95,96
Sat. Image	94	94	95	93	95,94

lead to rather poor performances due to the huge number of subsets considered in each class, giving them equal importance, while a lot of them blur the accuracy of oddness measure through the summation.

In our first experiments, considering the average accuracy on all 15 datasets, it appears that Odd_2 is a better performer than Odd_1 and Odd_3. We then focus on Odd_2, trying to privilege subsets including elements of particular interest such as nearest neighbors in the classes. Since selecting one element of a pair as a nearest neighbour of the item x in the target class leads to good accuracy rates, we also consider the option of taking the second element as another nearest neighbour in the class, with the normalization factor chosen accordingly as $\frac{1}{\binom{k}{2}}$, leading to $Odd_2(NN, NN)$. This is also beneficial from a complexity viewpoint.

In Tables 2 and 3, we provide classification results respectively for $Odd_2(NN, Std)$ (using only one nearest neigbour), and $Odd_2(NN, NN)$ (using two nearest neigbours) for different values of k (k being the number of nearest neigbours used).

In the last column of Table 3, we assign respectively a positive $'+'$, negative $'-'$, or neutral $'.'$ mark if the $Odd_2(NN, NN)$ is respectively better, worse or similar to $Odd_2(NN, Std)$ for $k = 15$. This comparison shows that the two classifiers have similar efficiency for most datasets, except for Monk2 where $Odd_2(NN, NN)$ outperforms not only $Odd_2(NN, Std)$, but also SVM and IBk.

We can also note that $Odd_2(NN, NN)$ performs more or less in the same way as the best known algorithms. Especially we compared this classifier to IBk for k=15 using the Wilcoxon Matched-Pairs Signed-Ranks [5]. We get a p-value = 0.024 which shows that $Odd_2(NN, NN)$ is significanly better than IBk.

Table 2. Classification accuracies given as mean and standard deviation with $Odd_2(NN, Std)$ classifier for $k = 3, 7, 13, 15$

Datasets	$Odd_2(NN, Std)$			
Value of k	**3**	**7**	**13**	**15**
Balance	75.61±4.99	85.65±4.11	86.23±3.59	86.33±3.57
Car	87.73±3.99	92.68±3.17	90.85±2.99	90.1±2.86
Monk1	99.59±2.95	99.76±2.19	99.38±2.87	99.33±2.87
Monk2	36.39±8.57	53.03±5.85	61.34±5.34	64.28±4.4
Monk3	99.82±0.72	99.87±0.66	99.6±1.31	99.14±1.82
Spect	82.98±5.17	84.08±4.7	83.82±5.76	83.38±5.76
Voting	92.96±4.29	94.33±3.38	94.98±2.63	94.39±3.82
Diabetes	73.61±3.99	75.1±3.49	75.5±3.03	75.93±3.98
W. B. Cancer	95.79±2.32	96.31±1.82	96.37±1.91	96.49±2
Heart	81.55±3.94	82.52±5.39	82.1±5.34	82.34±5.55
Magic	78.39±3.27	78.32±3.03	78.83±3.18	78.94±3.09
Ionosphere	91.22±4.11	90.56±4.92	91.58±5.4	91.69±5.6
Iris	94.53±5.47	94.53±5.47	95.21±4.68	95.21±4.68
Wine	97.58±2.71	98.24±2.44	97.67±2.87	97.58±2.87
Sat. Image	95.1±1.48	94.59±1.36	93.93±1.77	93.84±1.67

Table 3. Classification accuracies given as mean and standard deviation with $Odd_2(NN, NN)$ classifier for $k = 3, 7, 13, 15$

Datasets	$Odd_2(NN, NN)$				
Value of k	**3**	**7**	**13**	**15**	
Balance	56.37±6.56	83.63±3.89	83.72±4.12	84.47±3.54	(-)
Car	75.38±4.53	92.39±3.89	91.26±2.89	90.36±3.83	(.)
Monk1	85.95±4.66	99.76±0.24	99.56±0.74	99.57±0.94	(.)
Monk2	53.72±8.1	59.7±4.6	67.93±4.23	67.73±3.13	(+)
Monk3	94.49±4.97	99.95±0.05	99.91±0.71	99.59±1.25	(.)
Spect	76.52±6.02	82.77±5.09	83.82±3.68	84.51±4.80	(+)
Voting	90.69±2.92	93.36±2.85	94.3±3.57	94.86±3.03	(.)
Diabetes	68.48±5.2	73.06±3.22	74.55±3.48	74.84±3.03	(-)
W. B. Cancer	94.05±2.48	95.86± 2.88	96.05±2.38	96.02±2.32	(.)
Heart	74.29±4.72	79.46±8.26	81.44±6.07	82.58±7.60	(.)
Magic	75.25±3.41	78.68±3.73	79.24±2.53	79.52±3.15	(+)
Ionosphere	91.17±4.2	91.64±2.88	91.81±2.0	91.85±2.12	(.)
Iris	94.59±5.62	94.59±5.62	94.7±5.63	94.98±5.24	(.)
Wine	97.64±4.81	97.81±2.65	97.77±2.78	97.74±2.78	(.)
Sat. Image	95.51±1.75	95.29±1.1	94.79±1.69	94.77±1.81	(+)

5 Conclusion

In this paper, we suggest a new way to evaluate the oddness of an item w.r.t. a class. Since using subsets of pairs (Odd_2) provides better accuracy results than singletons (Odd_1) or triples (Odd_3), we further investigate this option by filtering candidate pairs. We first choose one item in a pair as a nearest neighbour, then two elements in a pair as nearest neighbours in the class. Experiments show that we are still competitive with state of the art classifiers (k-NN, SVM) while having drastically decreased complexity. These results suggest that it may be beneficial to consider *pairs of nearest neighbours* and then to minimize an oddness measure, which departs from the k-NN view dealing with nearest neighbours in isolation.

REFERENCES

[1] M. Bounhas, H. Prade, and G. Richard, 'Analogical classification: A new way to deal with examples', in *Proc. 21st Europ. Conf. on Artificial Intelligence, Praha, ECAI'14*, volume 263, pp. 135–140. IOS Press, (2014).

[2] M. Bounhas, H. Prade, and G. Richard, 'Analogical classification: Handling numerical data', in *Proc. of Scalable Uncertainty Management - 8th Int. Conf, SUM*, eds., U. Straccia and A. Calì, volume 8720 of *LNCS*, pp. 66–79. Springer, (2014).

[3] M. Bounhas, H. Prade, and G. Richard, 'Oddness-based classifiers for boolean or numerical data', in *KI 2015: Advances in Artificial Intelligence - 38th Annual German Conference on AI, Dresden, Germany, September 21-25, 2015, Proceedings*, pp. 32–44, (2015).

[4] *Nearest Neighbor: Pattern Classification Techniques (Nn Norms : Nn Pattern Classification Techniques)*, ed., B. V. Dasarathy, IEEE Computer Society Press, 1990.

[5] J. Demsar, 'Statistical comparisons of classifiers over multiple data sets', *Journal of Machine Learning Research*, **7**, 1–30, (2006).

[6] J. Mertz and P.M. Murphy, 'Uci repository of machine learning databases', *Available at: ftp://ftp.ics.uci.edu/pub/machine-learning-databases*, (2000).

[7] H. Prade and G. Richard, 'Homogenous and heterogeneous logical proportions', *IfCoLog J. of Logics and their App.*, **1**(1), 1–51, (2014).

[8] N. Rescher, *Many-valued Logic*, McGraw-Hill Inc., New York, 1969.