ORIGINAL RESEARCH

About the sensitivity of ordinal classifiers to non-monotone noise

Irena Milstein*[1], Arie Ben-David[1], Rob Potharst[2]

[1] Technology Management, Holon Institute of Technology, Israel
[2] Econometric Institute, Erasmus School of Economics, The Netherlands

Received: February 24, 2015 **Accepted:** April 29, 2015 **Online Published:** July 3, 2015
DOI: 10.5430/air.v4n2p83 **URL:** http://dx.doi.org/10.5430/air.v4n2p83

Abstract

Ordinal classifiers have become quite popular in recent years. However, no one has systematically tested yet how sensitive they are to noise. This research investigates for the first time the effect of non-monotone noise on the accuracy related rankings of ten classifiers in a controlled manner. The findings of this experiment are reported here. They clearly show that some models are more sensitive than others to non-monotone noise. Some classifiers which ranked higher in absence of noise performed poorly when the noise level increased even modestly. Others, which ranked relatively low in noiseless datasets, ranked much better when the noise levels increased. Two classifiers which assure monotone classifications became practically useless at relatively low levels of noise, while other classifiers' accuracies deteriorated at a much slower pace. Three alternative accuracy-related measures were used: Accuracy, Kappa and the Gini Index, and all were subjected to statistical tests. The lesson to be learned from this experiment is that it is very important to measure and report, among other things, the levels of noise which are present in datasets used for the evaluation of classification models.

Key Words: Ordinal classification, Monotone classification, Non-monotone noise, Non-monotonicity index

1 Introduction

Monotone decision-making is very common in human daily life. This type of problems is typified by the fact that the output (*i.e.*, the decision to be made) is expected to be either a non-decreasing or a non-increasing function of the input. For example, all other things being equal, a company with a better liquidity ratio should score at least a similar bond rating as one with a worse ratio. On a similar line of reasoning, a healthier applicant for life insurance is expected not to pay a higher premium than one with worse health condition. A candidate for a certain type of job is expected to get at least a similar fitness score as another who has made worse impression during an interview if all their other characteristics of the two applicants are identical. The list of domains where this monotonicity assumption makes sense is virtually endless. It includes bankruptcy predictions, consumer preferences, various health related decisions, investment decisions and so on.

The abundance of ordinal problem domains in which monotone decisions are expected has not gone unnoticed by data mining researchers and various classification models were proposed over the years. Other ordinal classifiers do not make this assumption, since there are problem domains which are ordinal in nature, but in which the monotonicity assumption does not make sense. These models are discussed in the next section. Virtually all ordinal models (*i.e.*, those with the monotonicity assumption and those without) need to accommodate non-monotone examples while learn-

*__Correspondence:__ Irena Milstein; Email: irenam@hit.ac.il; Address: Technology Management, Holon Institute of Technology, 52 Golomb St., P.O. Box 305, Holon 5810201, Israel.

ing, since real-world data is typically not pure monotone. Ordinal datasets usually include examples that violate the monotonicity assumption. Such examples are in fact non-monotone noise which should be dealt with somehow during the learning phase. We refer to this common phenomenon as *non-monotone noise*. Researchers have noticed this fact, and took various measures to deal with it. Surprisingly, however, there is not a single research report that systematically checks how sensitive these classifiers are to various levels of non-monotone noise.

We report here the results of an experiment which measures the accuracy of some known classifiers in presence of various levels of non-monotone noise. Some of the results are very interesting. We have found that some classifiers, which provide excellent learning performance for noiseless or almost noiseless datasets, have deteriorated very rapidly as the non-monotone noise level increases; while others seemed to be less sensitive.

In this experiment we were mainly interested in checking how sensitive classifiers are to non-monotone noise. In other words, we have not tried to find the most accurate classifier, but rather to test each classifier's sensitivity to increasing levels of non-monotone noise given a predefined (not necessarily the optimal) set of parameters.

The experiment was limited to ordinal monotone datasets, in which there is a single dependent variable and several attributes. It was also assumed that all the attributes as well as the dependent variable are expressed in ordinal terms. This assumption is very often reasonable in the context of human decision-making, since as human beings we tend to think in symbolic ordinal terms. When granting a credit line, for instance, it usually does not make any difference whether the applicant's net assets are worth 868,000 USD or 866,000 USD. Instead, we tend to think about this feature in ordinal terms such as "low", "average", "high", *etc*. Otherwise, the decision maker will be overloaded with useless information. Also, it is assumed here that an example with certain attribute values can have only one class value (*i.e.*, it cannot take several class values, each with some probability).

The experiment was also restricted to what we nickname "real-world" human decision-making problems that are problems which human beings can solve while taking all the attributes into account simultaneously. Various researchers in human decision-making such as Miller[1] and Ganzach[2] have shown that even experienced decision-makers do not take more than 7 plus or minus two attributes into account simultaneously (also known as "Miller's Magical Numbers"). In other words, this research does not deal with datasets which have 100 or 1000 attributes. Rather it is focused on problems which have one dependent variable and seven or less attributes. Each attribute has 2 to 7 possible ordinal values, and so does the dependent (*i.e.*, the class) variable.

The coming section discusses related work (Section 2) followed by a description of the experiment in Section 3, which includes: The datasets and their characteristics (3.1), the classifiers which were used in the experiment (3.2), the settings of the experiment and the three measures for assessing the accuracy of the models (3.3). The results are presented and discussed in Section 4. Conclusions and suggestions for further research are given in Section 5.

2 Related work

Ordinal classifiers have become increasing popular in recent years. McCullagh's Ordinal Regression (OR) is perhaps the most well known among them.[3] OR does not require that the datasets to be learned from are purely monotone. On the other hand, OR does not guarantee monotone classifications afterwards. OR is an example of an ordinal classifier in which the monotonicity assumption does not apply. It is, therefore, particularly suited for application domains in which this assumption does not hold. In contrast, the Ordinal Learning Model (OLM) is an early and simple ordinal classier that makes this assumption.[4] Similar to OR, it can work with non-monotone datasets, but unlike OR, the OLM assures monotone classifications.

Monotone decision tree models have been introduced by Makino *et al.*[5] and later by Potharst and Bioch,[6] Cao-Van and De Baets,[7] Feelders and Pardoel,[8] and others. They all require purely monotone datasets to learn from, and they assure the monotonicity of subsequent classifications. Lievens *et al.*'s probabilistic Ordinal Stochastic Dominance Learner (OSDL) can learn from noisy ordinal datasets and also provides monotone classifications thereafter.[9] Monotone neural network classifiers were introduced by Daniels and Velikova.[10] Their algorithms can learn from non-monotone datasets, but they do not guarantee monotone classifications. Other approaches towards ordinal classifications include, but are not limited to, Ben-David's hybrid approach,[11] Frank and Hall's Ordinal Class Classifier (OCC) meta-model,[12] and Popova and Bioch classification by function decomposition.[13] The last three models can cope with non-monotone noisy data but do not guarantee subsequent monotone classifications. This is unlike the rule ensembles proposed by Dembczynski *et al.*,[14] which can work with noisy data, and provide monotone classifications. While a detailed discussion of each individual model is beyond the scope of this paper, the interested reader can find a reference to them all in the Bibliography. Some of these models are used in the experiment to be described in the coming section.

As has been mentioned above, some classifiers, such as those which were proposed by Makino *et al.*,[5] Potharst and Bioch,[6] Cao-Van and De Baets,[7] and Feelders and Pardoel[8] can only learn from datasets which are purely monotone. Since most real-world datasets include non-monotone noise, the first phase of these classifiers typically aims at eliminating all non-monotone occurrences from the datasets. Three papers have been published so far on the generation of monotone artificial datasets. The paper by De Loof *et al.* is about generating completely random monotone datasets.[15] It uses the computationally intensive Markov Chain Monte Carlo method. The paper by Potharst *et al.* made use of a simpler computation, and proposed a method to incorporate an underlying structure into the artificial monotone datasets.[16] The third paper by Milstein *et al.* proposed a new algorithm for generating purely monotone as well as noisy ordinal datasets.[17] The user of this algorithm can control the levels of non-monotone noise in the resulting datasets. This algorithm was used for generating the datasets in the experiment to be reported in the coming section.

There have been several proposals in the literature suggesting how to define and measure non-monotone noise in ordinal datasets. In order to describe these proposals one needs to re-introduce some formal definition:

Let D be a dataset with k ordinal attributes $A_1, \cdots, A_k$ and class variable Y which has C possible ordinal values. The dataset consists of n examples x. A partial ordering $\preceq$ on D is defined as

$$x \preceq x' \Leftrightarrow A_j(x) \leq A_j(x') \,\text{for}\, j = 1, \cdots, k \tag{1}$$

Thus, two examples x and x' in space D are *comparable*, if either $x \preceq x'$ or $x' \preceq x$, otherwise x and x' are *incomparable*. *Identical* examples denoted as $x = x'$, and *non-identical* as $x \neq x'$.

Having this notation in mind, we call a pair of comparable examples (x, x') monotone if

$$x \preceq x' \wedge x \neq x' \wedge Y(x) \leq Y(x') \tag{2}$$

or

$$x = x' \wedge Y(x) = Y(x') \tag{3}$$

A dataset consisting of n examples is *monotone* if all possible pairs of examples are either monotone or incomparable.

Example x from D *clashes* with example x' from D if

$$x \preceq x' \wedge x \neq x' \wedge Y(x) > Y(x') \tag{4}$$

or

$$x = x' \wedge Y(x) \neq Y(x') \tag{5}$$

Furthermore, we use the following notation: if x is an example from dataset D, then $NClash(x)$ is the number of examples from D that clash with x. $Clash(x)$ = 1 if x clashes with some examples in D, and 0 otherwise. If $Clash(x)$ = 1, x is called a *non-monotone* example.

Following Daniels and Velikova,[10] we call the first, and most obvious, index of non-monotonicity to be introduced here *NMI*1, the number of clash-pairs divided by the total number of pairs of examples in the dataset. So,

$$NMI1 = \frac{1}{n(n-1)} \sum_{x \in D} NClash(x) \tag{6}$$

Horvath *et al.* suggested dividing by the number of all comparable pairs in the dataset.[18]

The second index is called here *NMI*2, the number of non-monotone examples divided by the total number of examples. So,

$$NMI2 = \frac{1}{n} \sum_{x \in D} Clash(x) \tag{7}$$

The third index, *NMI*3, is the minimum number of class label changes needed to make a dataset monotone, divided by the total number of examples.[10,19] The lowest value of all these three indices is 0, when the dataset is purely monotone. The highest value of the first two indices is 1, when every example in the dataset clashes with all the others (*i.e.*, all the examples are non-monotone with respect to each other).

*NMI*1 has been chosen for this experiment due to its intuitiveness rather than its simplicity. Unlike *NMI*2 and *NMI*3, *NMI*1 reflects the fact that non-monotonicity occurs in pairs of examples, since it counts all the clashing pairs in the dataset relative to the total number of pairs.

3 The experiment

The purpose of the experiment was to check the sensitivity of various classifiers to non-monotone noise. Of particular interest was to answer questions such as: Are classifiers' rankings changed when the noise level increases? Are some classifiers more sensitive to non-monotone noise than others?

A description of the datasets which were used in the experiment is given in the coming section, followed by a list of the classifiers which were used. The experiment settings are discussed later on.

3.1 The datasets

All the datasets which were used in this experiment were generated using an algorithm which is described by Milstein *et al.*[17] Since the algorithm was presented and discussed there, it will not be re-iterated here in detail. It should only be mentioned that the algorithm artificially generates datasets with user-specified monotone patterns as well as user-defined *NMI*1 non-monotone noise index levels.

Ten distinct datasets, each containing 1000 examples, were generated for the experiment. Their key characteristics are shown in Table 1. The number of attributes, k, in each dataset is shown in the second leftmost column. The number of attributes' possible ordinal values, V_j, is shown in the middle column. To simplify the experiment the values of V_j were identical for all the attributes in the respective dataset ($i.e., V_j = V$). The attribute values were randomly selected by the algorithm from Uniform distributions. The number of possible ordinal values of the dependent variable (*i.e.*, the number of ordinal class values), C, is shown in second rightmost column. As has been mentioned in the Introduction section, the selected values of k, V, and C are not trivial on one hand, yet they are still comprehensible by decision-makers (*i.e.*, each is in the range 2 to 7). The monotone functions which were used for generating the dependent (*i.e.*, the class) values are shown in the rightmost column, were A_j denotes the value of the j^{th} attribute.

Table 1: The major characteristics of the datasets

Dataset number	**Number of attributes, *k***	**Number of attribute values, *V***	**Number of class values, *C***	**Function for class numerical value**
1	5	4	6	$\sum j(A_j)^j$
2	4	5	7	$\sum j(A_j)^3$
3	7	3	6	$\sum[j^2 + (A_j)^j]$
4	7	4	4	$\sum(A_j)^j$
5	3	7	2	$\sum e^{A_j}$
6	4	6	2	$\sum A_j^{(2j+1)}$
7	6	3	5	$\sum \sqrt{A_j}$
8	3	6	5	$\sum 2^{A_j}$
9	6	4	7	$\prod(A_j + 1)^2$
10	5	5	3	$\sum(A_j)!$

Since our major purpose in this experiment was to check how sensitive classifiers are to non-monotone noise, seven versions of each dataset were generated. Each dataset was initially generated without any non-monotone noise (*i.e.*, purely monotone). Later the algorithm incrementally generated noisy examples up to the following *NMI*1 noise index levels: 1%, 2%, 5%, 10%, 15% and 20%. These values were selected based on our and others' observations of real ordinal datasets, in which the monotonicity assumption made sense. Milstein *et al.*[17] have measured the *NMI*1 values of four real world ordinal datasets: ESL, ERA, LEV, and SWD which can be found at the Weka project web site.[20] It was found that the *NMI*1 values ranged from about 1% to 4%. Daniels and Velikova who have checked two datasets found that they were almost monotone.[10] On the other hand, Horvath *et al.* found only 5 purely monotone datasets out of 40.[18] Based on these observations we have chosen the values of *NMI*1 within a similar range, but also allowed it higher values in order to check how the classifiers perform when encountering nosier datasets.

3.2 The classifiers

Ten data mining classifiers which are included in Weka were chosen for the experiment: Three non-ordinal (*i.e.*, classifiers which do not take the ordinal order into account), and seven ordinal. The first category included: (a) An implementation of the well-known C4.5,[21] nicknamed J48 in Weka, (b) Logistic Regression (LOGISTIC) which is also widely used in data mining,[22] and (c) Sequential Minimal Optimization (SMO),[23] a famous fast version of Support Vector Machines.

Table 2: The classifiers and their major characteristics

	Classifier	**Ordinal**	**Monotone classification**
1	C4.5 (J48)	no	no
2	Logistic Regression (LOGISTIC)	no	no
3	Ordinal Class Classifier / J48 (OCC/J48)	yes	no
4	Ordinal Class Classifier / LOGISTIC (OCC/LOGISTIC)	yes	no
5	Ordinal Class Classifier / OLM (OCC/OLM)	yes	no
6	Ordinal Class Classifier / OSDL (OCC/OSDL)	yes	no
7	Ordinal Class Classifier / SMO (OCC/SMO)	yes	no
8	Ordinal Learning Model (OLM)	yes	yes
9	Ordinal Stochastic Dominance Learner (OSDL)	yes	yes
10	Sequential Minimal Optimization *(*SMO*)*	no	no

Many ordinal classifiers were mentioned in the related work section, several of which were used in this experiment: Two ordinal classifiers which assure monotone classifications, the OSDL and the OLM, and five versions of the OCC. Being a meta-model the OCC was operated above the three non-ordinal classifiers (J48, LOGISTIC, and SMO) and the two ordinal classifiers (OSDL and OLM) as the base models. The ordinal classifiers which were used in the experiment were all mentioned with their respective references in the Related Work section earlier. The non-ordinal classifiers are all well described in the literature and will not be re-iterated here.

Table 2 shows the classifiers which were used in the experiment, their short nicknames, whether they are ordinal (*i.e.*, use the ordinal order within the data while learning), and whether or not they assure monotone classification thereafter.

3.3 Experiment settings

The experiment was conducted using Weka 3.7.10 with the default values of the classifiers' parameters. The default values of the parameters rather than their optimal values were chosen to make the computation manageable. As has previously been mentioned, the aim of the experiment was not finding a "winner" (as most publications do) among the tested classifiers, but rather to check how sensitive each classifier is to increasing levels of non-monotone noise, given a set of (not necessarily optimal) parameters.

Ten fold cross validation was used throughout. The results of each fold were recorded using Weka's Experimenter module. For each fold the following data was collected: The accuracy (*i.e.*, the "hit ratio", nicknamed ACC here), Cohen's Kappa (KAPPA), and the Area Under the ROC Curve (AUC). These three accuracy-related indices were recorded since there is still no consensus among researchers which is the "best" one, and all three are used in data mining research.[24]

While assessing the performance of a classifier, one is usually interested in measuring its added value relative to a random one. Both AUC and Kappa measure this added value (each in its own way), but on different scales: When the class values are evenly distributed (as is the case of our experiment), a random classifier results zero KAPPA and 0.5 AUC. The accuracy (ACC) of a random classifier depends on the number of class values, C. A perfect classifier scores 1 for ACC, KAPPA and the AUC. In order to use identical scales, the ACC and the KAPPA results were recorded as is directly from Weka's Experimenter. This has been done since KAPPA compensates for ACC's random successes anyway. However, the AUC was converted to the well known Gini index (GINI) by the known formula: $GINI = 2(AUC - 0.5)$. This way all three accuracy-related indices, i.e. ACC, KAPPA, and GINI, ranged from 0 to 1.

As has been mentioned in Section 3.1, ten distinct purely monotone artificially generated datasets were available in this experiment, each having seven versions (0%, 1%, 2%, 5%, 10%, 15%, and 20% of *NMI*1 noise). Since the experiment tested the performance of ten classifiers using ten-fold cross validation, the total number of learning-classification cycles was 7,000.

4 Results

The average values of ACC, KAPPA and GINI for each classifier over all the ten datasets are shown graphically in Figure 1. The horizontal axes indicate the *NMI*1 non-monotone noise indices. Figure 1 shows that some classifiers are more sensitive indeed to non-monotone noise than others. It can be seen in Figure 1, for instance, that by all three accuracy-related measures, the performance of the OLM and the OSDL deteriorated at a faster pace than that of J48 and LOGISTIC regression. This is particularly noted at the lower range (0% to 5%) of the *NMI*1 noise.

When the noise level increased modestly from 0% (*i.e.*, purely monotone datasets) to 1%, the OLM's KAPPA, for example, decreased from 0.895 to 0.515 (a loss of 42.5%), and that of the OSDL from 0.904 to 0.341 (a loss of 62.3%). For comparison, LOGISTIC Regession's KAPPA was reduced from 0.912 to 0.787 (a loss of only 13.7%). A look at the comparable results of the GINI index reveals very similar picture: OLM's GINI was reduced from 0.896 to 0.514 (a loss of 42.6%), OSDL's GINI – from 0.904 to 0.341 (a loss of 62.3%), and LOGISTIC Regression's GINI – from 0.978 to 0.891 (a loss of only 8.9%). While looking at the wider range of the *NMI*1 noise index, it is worthwhile to mention that the negative KAPPA and GINI values indicate worse than random classifier performance. A look at the average performance of the classifiers at 5% noise level reveals that four out of the ten classifiers (OCC/OLM, OCC/OSDL, OSDL, and OLM) became virtually useless for any practical application, since both their KAPPA and GINI values deteriorated below 0.2. Virtually all the classifiers performed as poorly as random classifiers (or very close to that) when the level of noise reached 10% and above.

Figures 2-4 present the classifiers' rankings by ACC, KAPPA and GINI respectively for the different values of the *NMI*1 noise indices. These rankings are based upon the average ranks of the classifiers across the ten datasets. The noise levels are shown on the horizontal axes, and the rankings on the vertical, where 1 indicates the best performing classifier, 2 - the second best, and so on. Perhaps one of the most interesting observations here is the fact that not a single classifier ranked the best across all the *NMI*1 noise index range. LOGISTIC regression, for instance, which ranked second in noiseless datasets according to KAPPA, was only the fourth when the noise level increased to 1%. It was surpassed by SMO, for instance, which ranked in the seventh position in purely monotone datasets by KAPPA.

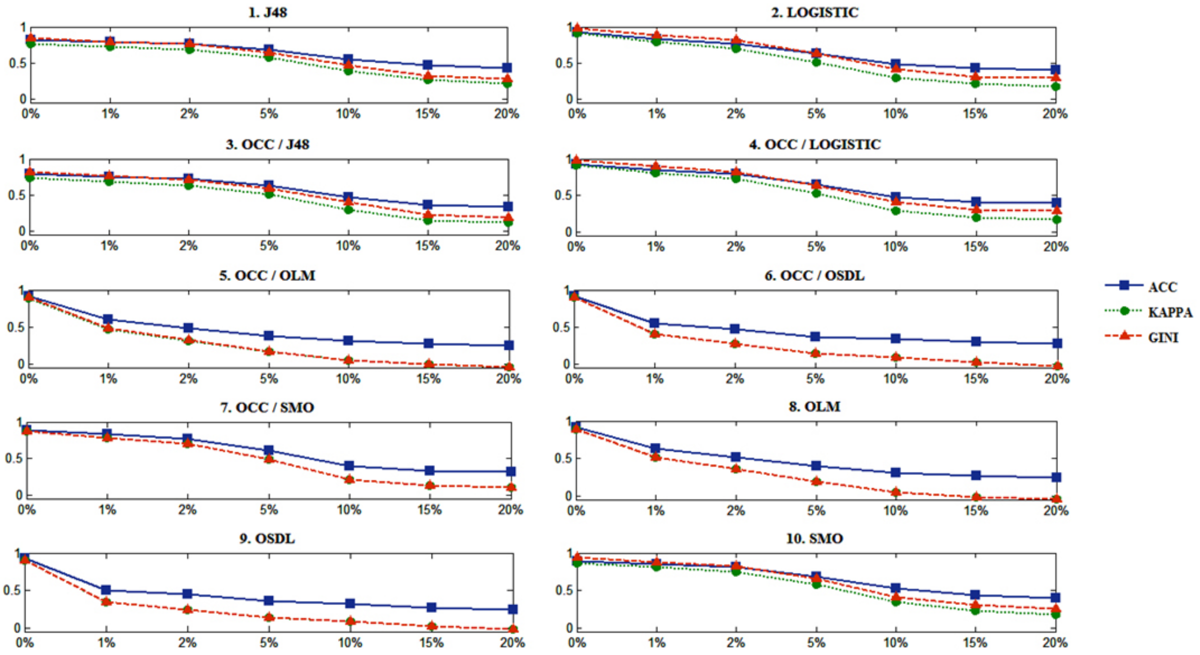

Figure 1: Average ACC, KAPPA, and GINI versus *NMI*1 noise index

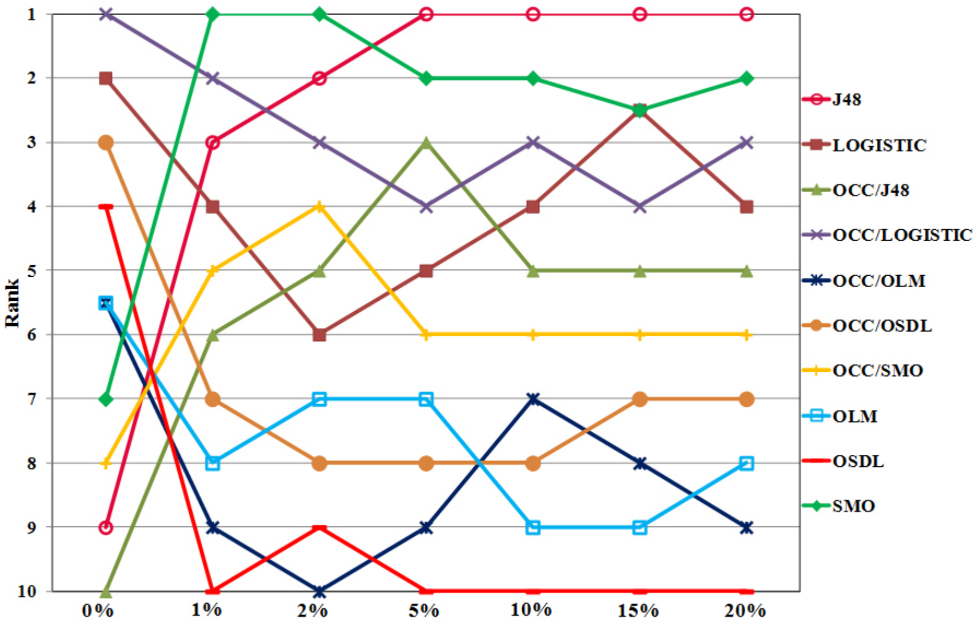

Figure 2: Ranking by ACC as a function of a noise index

Despite of the expected differences in some rankings according the different accuracy-related measures, there are some notable similarities between them. Generally speaking, the OLM and the OSDL (*i.e.*, those classifiers which assure monotone classifications) got above-average rankings in noiseless datasets according to the three measures. However, their rankings deteriorated quite rapidly in presence of even low levels of noise. On the other hand, the non-ordinal classifiers, such as J48 and SMO were the most successful at the higher end of the *NMI*1 noise scale.

It worth mentioning some other key observations as well: Figures 2-4 show that the rank of J48 has improved significantly when the noise in the datasets increases: By ACC, for instance, from the 9th position for the monotone datasets to the 3rd for 1% noisy datasets, and then to the 2nd for 2%

noisy datasets. Furthermore, J48 ranked the first for 5% to 20% noisy datasets by both ACC and KAPPA. The rank of SMO by both ACC and KAPPA also improved as the noise level increased: From the 7^{th} position by ACC in the monotone datasets to the first for the datasets with 1% and 2% of noise, and then it concedes to J48. These results are quite similar according to Kappa. According to GINI, SMO was at a fairly competitive and quite a stable position throughout the entire noise range.

Generally speaking, the rankings of the various classifiers by both ACC and KAPPA were quite similar, with some minor exceptions: In particular, at the high end of the noise levels, where all the classifiers performed quite poorly in absolute terms. The rankings according to the GINI index are somehow different from those of ACC and KAPPA. For example, the improvement in the rank of J48 is not so pertinent when the ranking is based on GINI. This is since it ranked relatively higher, the sixth instead of the ninth, for the monotone datasets in the first place. Similar to ACC and KAPPA rankings, according to GINI, LOGISTIC and OCC/LOGISTIC also ranked high, whereas OSDL, OCC/OSDL, OLM and OCC/OLM ranked relatively low.

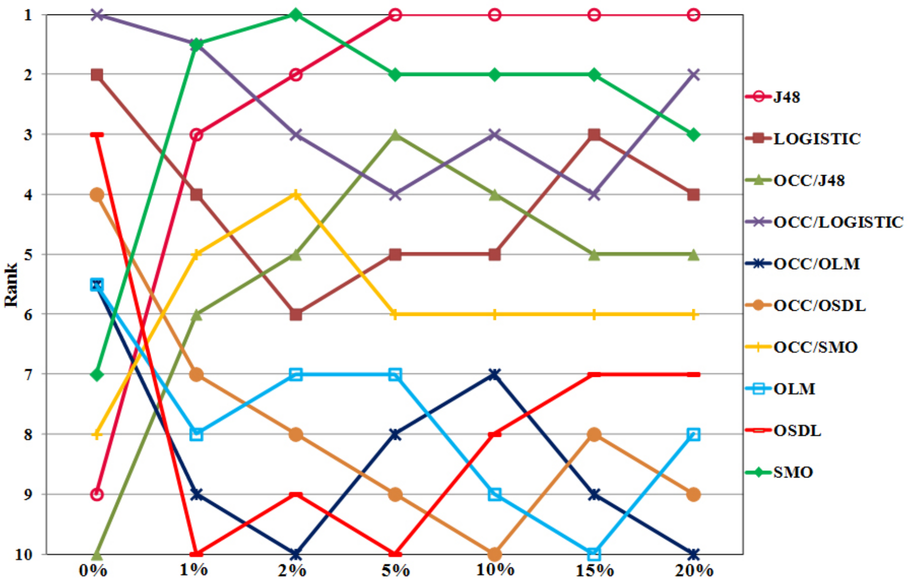

Figure 3: Ranking by KAPPA as a function of a noise index

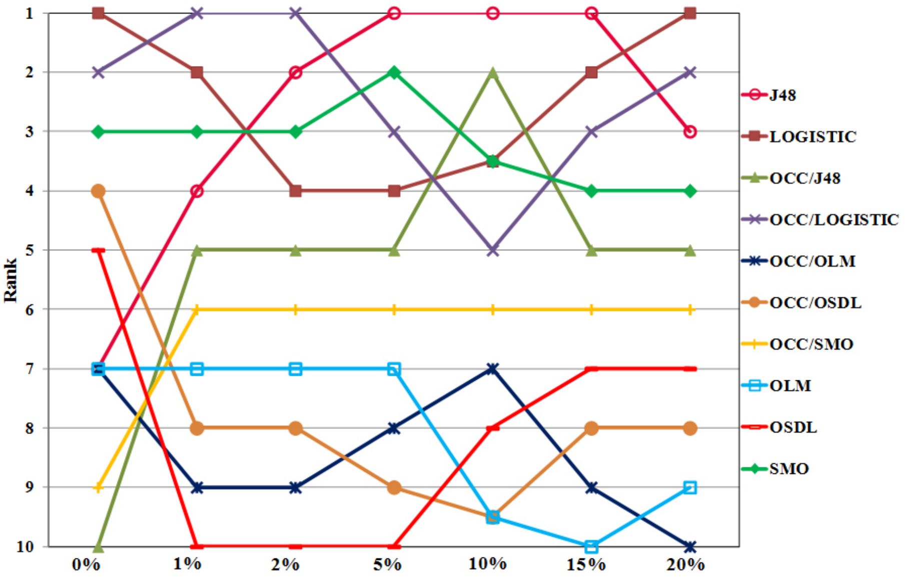

Figure 4: Ranking by GINI as a function of a noise index

Table 3 presents two statistics, Friedman and its correction suggested by Iman and Davenport,[24] for the different levels of noise. The Friedman test checks whether the measured average ranks are significantly different from the mean rank, as expected under null-hypothesis. With 10 classifiers and 10 datasets, the critical value for F(9, 81) at α = 0.05 is 1.998. That is, the null-hypothesis (*i.e.*, that all the algorithms are equivalent) is rejected for any noise level and for all the accuracy-related measures.

Table 3: Friedman statistic, χ_F^2, and its correction, F_F

	Statistic	***NMI*1 noise level**						
		0%	**1%**	**2%**	**5%**	**10%**	**15%**	**20%**
ACC	χ_F^2	49.05	46.37	56.38	60.75	61.50	60.85	59.34
	F_F	10.78	9.57	15.10	18.69	19.42	18.79	17.42
KAPPA	χ_F^2	43.18	46.37	56.32	60.57	63.11	65.46	63.19
	F_F	8.30	9.57	15.05	18.53	21.13	24.01	21.21
GINI	χ_F^2	34.06	61.71	62.06	62.80	64.19	67.78	66.10
	F_F	5.48	19.63	19.99	20.78	22.38	27.46	24.90

Friedman test with Iman and Davenport correction does not make pairwise comparison among pairs of classifiers. Tables 4 and 5 present the average pairwise ranking differences by Kappa for purely monotone datasets and for 5% noise level, respectively. The results of Nemenyi post-hoc test for all three metrics at the different levels of noise are available from the authors upon request. With 10 classifiers, the critical value at α = 0.05 is 3.164 (at α = 0.10 it is 2.920).[24] Therefore, the critical differences are 4.284 for α = 0.05 (3.954 for α = 0.10). Significant differences for α = 0.05 are underlined and highlighted in gray background.

Table 4: Average pairwise ranking differences by KAPPA for 0% *NMI*1 noise index

	OCC/ LOGISTIC	**LOGISTIC**	**OSDL**	**OCC/ OSDL**	**OCC/ OLM**	**OLM**	**SMO**	**OCC/ SMO**	**J48**	**OCC/ J48**
OCC/LOGISTIC	-									
LOGISTIC	0.2	-								
OSDL	1.4	1.2	-							
OCC/OSDL	1.45	1.25	0.05	-						
OCC/OLM	2.75	2.55	1.35	1.3	-					
OLM	2.75	2.55	1.35	1.3	0	-				
SMO	3.1	2.9	1.7	1.65	0.35	0.35	-			
OCC/SMO	3.3	3.1	1.9	1.85	0.55	0.55	0.2	-		
J48	5.75	5.55	4.35	4.3	3	3	2.65	2.45	-	
OCC/J48	6.3	6.1	4.9	4.85	3.55	3.55	3.2	3	0.55	-

Table 5: Average pairwise ranking differences by KAPPA for 5% *NMI*1 noise index

	J48	**SMO**	**OCC /J48**	**OCC/ LOGISTIC**	**LOGISTIC**	**OCC/ SMO**	**OLM**	**OCC/ OLM**	**OCC/ OSDL**	**OSDL**
J48	-									
SMO	0.5	-								
OCC/J48	1.3	0.8	-							
OCC/LOGISTIC	1.6	1.1	0.3	-						
LOGISTIC	2.1	1.6	0.8	0.5	-					
OCC/SMO	2.7	2.2	1.4	1.1	0.6	-				
OLM	5.7	5.2	4.4	4.1	3.6	3	-			
OCC/OLM	5.9	5.4	4.6	4.3	3.8	3.2	0.2	-		
OCC/OSDL	6.05	5.55	4.75	4.45	3.95	3.35	0.35	0.15	-	
OSDL	6.15	5.65	4.85	4.55	4.05	3.45	0.45	0.25	0.1	-

5 Conclusions and further research

In general, with the exception of OCC/LOGISTIC, the non-ordinal classifiers are found to have high rank than the ordinal ones. This result may be regarded as counter intuitive, since one can expect classifiers which exploit the ordinal order to do better than those which do not. These seemingly odd findings are quite consistent with an earlier work,[25] where the performance of several ordinal and non-ordinal classifiers was assessed on real-world ordinal datasets without measuring or controlling their noise levels, as we have done in this experiment. These findings, however, should be taken with a grain of salt, since better rankings do not necessarily mean "more accurate". As can be seen, for instance, in Table 4, OCC/LOGISTIC which ranked first while learning purely monotone datasets, was more accurate, according to KAPPA, than J48 and OCC/J48 with 95% level of confidence. It was, however, statistically indistinguishable from all the other classifiers. When the noise level increased to 5%, J48 ranked the first, but it was statistically indistinguishable from OCC/LOGISTIC (and also from SMO, OCC/J48, LOGISTIC, and OCC/SMO).

Another interesting observation which emerges from this experiment relates to those classifiers which assure monotone classifications, the OSDL and the OLM. According to KAPPA, they were statistically indistinguishable from OCC/LOGISTIC and LOGISTIC, which ranked the first and the second respectively in noiseless datasets (see Table 4). However, both the OSDL and the OLM were found to be (statistically) worse than J48 and SMO, which ranked first and second when the noise level increased to 5% (see Table 5). A similar, though not identical, result is obtained when pairs of classifiers are compared by GINI. Both OSDL and OLM are statistically indistinguishable from LOGISTIC and OCC/LOGISTIC while learning from purely monotone datasets, but (statistically) inferior to J48, SMO, OCC/LOGISTIC, and LOGISTIC when the noise level was 5%. By all three accuracy related measures, both the OSDL and the OLM were not as accurate as their counterparts (which do not assure monotone classification) as the noise level increased. This phenomenon can be explained by the OSDL and OLM's unique (and frequently important) feature, the generation of monotone classifications. This feature imposes extra constraints on their learning process. An immediate practical conclusion one can draw from this experiment is that both the OSDL and the OLM become quite useless in terms of accuracy while learning from datasets which have *NMI*1 noise level of 5% and above.

It has been shown here that classifiers rankings are dependent, among other things, upon the noise levels in the datasets. More work has to be done in the future to fully understand how non-monotone noise affects the accuracy of the various classifiers. Adding more classifiers to experiments such as the one which has been reported here, where the noise levels are controlled, is one way to go. Another open question is whether the assignment of optimal values to each classifier at each noise level will affect the findings of this work. We see no reason to suspect that the general picture will change (*i.e.*, the rankings will remain stable when the optimal parameter values will be used), but this hypothesis is to be tested as well. Using real-world data instead of artificial data, while maintaining controlled levels of non-monotone noise, is another challenge. It is also of interest to study the effects of the number of examples which are used for learning on accuracy (while controlling the noise levels), and how the selection of monotone functions (see Table 1) affects the various classifiers when the datasets become noisier.

Currently most publications do not report the noise levels in the datasets which were used in the experiments. This omission may lead the reader to wrong conclusions, since, as this work demonstrates for the first time, classifiers' rankings are dependent, among other things, on non-monotone noise levels. We therefore hope that this paper will encourage researchers to measure and report the noise levels in the datasets they use while comparing the accuracies of classifiers in scientific publications. As this paper demonstrates, non-monotone noise does affect classifiers' performance and rankings.

References

[1] Miller GA. The magical number seven, plus or minus two: Some limits on our capacity for processing information. Psychological Review. 1956; 63(2): 81-97. http://dx.doi.org/10.1037/h0043158

[2] Ganzach Y. Goals as determinants of nonlinear noncompensatory judgment strategies. Organizational Behavior and Human Decision Processes. 1993; 56: 422-440. http://dx.doi.org/10.1006/obhd.1993.1062

[3] McCullagh P. Regression models for ordinal data. Journal of the Royal Statistical Society. 1980; 42(2): 109-142.

[4] Ben-David A, Sterling L, Pao YH. Learning and classification of monotonic ordinal concepts. Computational Intelligence. 1989; 5: 45-49. http://dx.doi.org/10.1111/j.1467-8640.1989.tb00314.x

[5] Makino K, Suda T, Ono H, *et al*. Data analysis by positive decision trees. IEICE Transactions on Information Systems. 1999; E82-D (1): 76-88.

[6] Potharst R, Bioch JC. Decision trees for ordinal classification. Intelligent Data Analysis. 2000; 4: 97-111.

[7] Cao-Van K, De Baets B. Growing decision trees in an ordinal setting. International Journal of Intelligent Systems. 2003; 18: 733-750.

[8] Feelders A, Pardoel M. Pruning for monotone classification trees. Lecture Notes in Computer Science. 2003; 2810: 1-12. http://dx.doi.org/10.1007/978-3-540-45231-7_1

[9] Lievens S, De Baets B, Cao-Van K. A probabilistic framework for the design of instance-based supervised ranking algorithms in an ordinal setting. Annals of Operations Research. 2006; 163: 115-142. http://dx.doi.org/10.1007/s10479-008-0326-1

[10] Daniels HAM, Velikova MV. Derivation of monotone decision models from noisy data. IEEE Transactions on Systems, Man and Cybernetics - Part C. 2006; 36: 705-710.

[11] Ben-David A. Monotonicity maintenance in information-theoretic machine learning algorithms. Machine Learning. 1995; 19: 29-43. http://dx.doi.org/10.1007/BF00994659

[12] Frank E, Hall M. A simple approach to ordinal classification. The 12th European Conference on Machine Learning. 2001: 145-156.

[13] Popova V, Bioch JC. Monotone classification by function decomposition. Lecture Notes in Computer Science. 2005; 3735: 203-214. http://dx.doi.org/10.1007/11563983_18

[14] Dembczynski K, Kotlowski W, Slowinski R. Learning rule ensembles for ordinal classification with monotonicity constraints. Fundamenta Informaticae. 2009; 94: 163-179.

[15] De Loof K, De Baets B, De Meyer H. On the random generation of monotone datasets. Information Processing Letters. 2008; 107: 216-220. http://dx.doi.org/10.1016/j.ipl.2008.03.007

[16] Potharst R, Ben-David A, van Wezel M. Two algorithms for generating structured and unstructured monotone ordinal datasets. Engineering Applications of Artificial Intelligence. 2009; 22: 491-497.

[17] Milstein I, Ben-David A, Potharst R. Generating noisy monotone ordinal datasets. Artificial Intelligence Research. 2014; 3(1): 30-37.

[18] Horvath T, Eckhardt A, Buza K, *et al*. Value-transformation for monotone prediction by approximating fuzzy membership functions. The 12th IEEE International Symposium on Computational Intelligence and Informatics. 2011: 367-372.

[19] Rademaker M, De Baets B, De Meyer H. Optimal monotone relabelling of partially non-monotone ordinal data. Optimization Methods and Software. 2012; 27(1): 17-31.

[20] Hall M, Frank E, Holmes G, *et al*. The WEKA data mining software: An update. SIGKDD Explorations. 2009; 11(1). http://www.cs.waikato.ac.nz/ml

[21] Quinlan JR. Programs for machine learning, Morgan Kaufmann. 1993.

[22] Le Cessie S, Houwelingen JC. Ridge estimators in logistic regression. Applied Statistics. 1990; 41(1): 191-201.

[23] Platt J. Fast training of support vector machines using sequential minimal optimization, in Advances in Kernel Methods - Support Vector Learning, eds. Schoelkopf, B., Burges, C., and Smola, A., MIT Press. 1998.

[24] Demsar J. Statistical comparisons of classifiers over multiple datasets. Journal of Machine Learning Research. 2006; 7: 1-30.

[25] Ben-David A, Sterling L, Tran T. Adding monotonicity to learning algorithms may impair their accuracy. Expert Systems with Applications. 2009; 36(3): 6627-6634.