Combining Ranking with Traditional Methods for Ordinal Class Imbalance

Ricardo Cruz[1(✉)], Kelwin Fernandes[1,2], Joaquim F. Pinto Costa[3], María Pérez Ortiz[4], and Jaime S. Cardoso[1,2]

[1] INESC TEC, Porto, Portugal
{rpcruz,kafc,jaime.cardoso}@inesctec.pt
[2] Faculty of Engineering, University of Porto, Porto, Portugal
[3] Faculty of Sciences, University of Porto, Porto, Portugal
jpcosta@fc.up.pt
[4] Universidad Loyola Andalucía, Córdoba, Spain
mariaperez@uloyola.es

Abstract. In classification problems, a dataset is said to be imbalanced when the distribution of the target variable is very unequal. Classes contribute unequally to the decision boundary, and special metrics are used to evaluate these datasets. In previous work, we presented pairwise ranking as a method for binary imbalanced classification, and extended to the ordinal case using weights. In this work, we extend ordinal classification using traditional balancing methods. A comparison is made against traditional and ordinal SVMs, in which the ranking adaption proposed is found to be competitive.

Keywords: Ordinal classification · Class imbalance · Ranking · SVM

1 Introduction

Ordinal classification, also known as ordinal regression, is a subset of multiclass classification problems where the target variable has an ordinal scale, and so it is possible to establish an order between any two classes. Often, it is desirable to punish more an error incurred from misclassification of an observation as an adjacent class than an error when the observation is misclassified as a more disparate classes. The extra ordinal constrains can, and have been used, to produce models that specifically optimize for these ordinal metrics.

Classification datasets, which feature a disproportion in the distribution of observations in each class, are said to be class imbalance. Traditional methods favor too much the majority classes. Furthermore, traditional metrics such as accuracy can produce apparently good results for models which consider only the majority class, and special metrics have been devised for the purpose of evaluating models applied to class imbalance problems. Much literature exists in the topic, but only one attempt has been made using ranking, by the same authors, in a binary [1] and ordinal [2] classification context.

I. Rojas et al. (Eds.): IWANN 2017, Part II, LNCS 10306, pp. 538–548, 2017.
DOI: 10.1007/978-3-319-59147-6_46

Pairwise scoring rankers are an attractive family of models, since the problem is solved in the space of the differences between classes, and so is inherently balanced for the binary case. However, when transporting the problem to a multi-class ordinal context, the imbalance problem arises again between pairs of classes. In this work, approaches of tackling this imbalance are proposed and evaluated.

The work is divided as follows. Section 2 overviews some existing methods for ordinal classification. Section 3 overviews some methods for class imbalance. Section 4 details our proposal of combining ranking with traditional methods for ordinal class imbalance. Section 5 provides empirical experiments and results. Section 6 concludes the work.

2 Ordinal Classification

Many ordinal classifiers currently exist. oSVM [3] takes advantage of the fact that the decision boundaries are necessarily parallel in a well-formed ordinal problem. It transforms the original ordinal problem into a binary problem by increasing the number of dimensions, after which the multiple decision boundaries can be recreated.

SVOR [4] encompasses SVORIM and SVOREM which differ on how the constraints are defined. The idea is to find $k-1$ parallel discriminant hyperplanes in order to properly separate the data into ordered classes by modelling ranks as intervals [4].

Herbrich et al. [5] addresses ordinal classification using pairs in the space of differences. Let $\mathcal{C}_1 < \mathcal{C}_2 < \cdots < \mathcal{C}_K$ be the K classes involved. Let $\mathcal{S}_k = \{\mathbf{x}_n^{(k)}\}$ be the set of N_k samples from $\mathcal{C}_k$, with $N = \sum_{k=1}^{K} N_k$. Construct the differences $\mathbf{x}_{mn}^{(k\ell)} = \mathbf{x}_m^{(k)} - \mathbf{x}_n^{(\ell)}$ with $\mathcal{C}_k < \mathcal{C}_\ell$. Like in the binary setting, solve the binary classification problem in the set of the differences $\{(\mathbf{x}_{mn}^{(k\ell)}, +1), (-\mathbf{x}_{mn}^{(k\ell)}, -1)\}$, where $+1$ and -1 are the labels of the samples $\mathbf{x}_{mn}^{(k\ell)}$ and $-\mathbf{x}_{mn}^{(k\ell)}$, respectively.

An issue with this approach arises when one of the classes is strongly misrepresented when compared with the others. The data from each class $\mathcal{C}_k$ is involved in $N_k(N-N_k)$ points in the set of the differences. If $N_k \ll N_\ell$ then also $N_k(N-N_k) \ll N_\ell(N-N_\ell)$. For instance, if $N_1 = 10$ and $N_2 = N_3 = 100$, then the data from $\mathcal{C}_1$ is contributing to 2000 elements in the new space, while the data from $\mathcal{C}_2$ or $\mathcal{C}_3$ is contributing to 11000. So, the new learning problem will be dominated by the samples from $\mathcal{C}_2$ and $\mathcal{C}_3$ and it is likely that $\mathcal{C}_1$ will be poorly estimated.

Traditional one-vs-rest or one-vs-all ensembles can also be used for ordinal classes, even if they do not take order in consideration. However, they do not take advantage of that extra information, and do not optimize for ordinal metrics. Furthermore, they may produce models whose decision boundaries make little sense in an ordinal context; for instance, decision boundaries should not cross in an ordinal context [6].

3 Traditional Methods for Class Imbalance

Several methods have been proposed in tackling class imbalance, which usually involve:

(a) Pre-processing;
(b) Training with costs;
(c) Ensembles.

(a) Pre-processing usually involves a mix of undersampling the majority class and creating new synthetic examples of the minority class [7,8]. *(b) Training with costs* involves the use of a cost matrix so that the cost of misclassifying a class is inversely proportional to its frequency, and therefore the estimation algorithm minimize an weighted loss function, rather than the original imbalanced loss function, so that the minority class contributes more to the loss than it would otherwise. *(c) Ensembles* by which each model within the ensemble is trained with balanced subsets of the data, coupled with the previous preprocessing techniques [9].

Some strategies of tackling multiclass problems such as one-vs-rest exacerbate the imbalance problem by training. Given K classes with N_k observations, this strategy solves the problem using an ensemble of binary classifiers, training each classifier i with N_i positively labeled data against the rest $N-N_i$ negatively labeled data. This creates an imbalance problem, even if all classes are equally represented [10].

Another strategy is known as one-vs-one, whereby each classifier i is trained K using N_i positively labeled data and N_j, $\forall j \neq i$, negatively labeled data, resulting in an ensemble of $K(K-1)$ classifiers. Even this strategy is not optimal for ordinal datasets because it generates decision boundaries that make little sense in the context of ordinal data due to the fact the decision boundaries are not parallel [6].

Work already exists to adapt these methods to be used with ordinal classifiers in imbalance situations [8]. In the rest of the work, we propose using these methods as-is combined with ranking.

4 Combining Ranking with Traditional Methods

Class imbalance has been previously addressed using pairwise scoring ranking in [1]. Ordinal classification seems like a natural extension because, unlike ordinary multi-class problems, any two classes can be compared, as in ranking applications.

4.1 Ranking for Binary Class Imbalance

In pairwise scoring ranking, a function $f\colon X \rightarrow \mathbb{R}$ is constructed so that $f(\mathbf{x}_i) > f(\mathbf{x}_j)$ when $\mathbf{x}_i \succ \mathbf{x}_j$ for every pair of observations $(\mathbf{x}_i, \mathbf{x}_j)$, where $\succ$ means "preferred".

In the case here considered, as in [5], a base estimator is trained using the space of differences. Consider two classes, $\mathcal{C}_1$ and $\mathcal{C}_2$, with a set $\mathcal{S}_1$ of N_1 examples from $\mathcal{C}_1$ and a set $\mathcal{S}_2$ with N_2 examples from $\mathcal{C}_2$. Construct all $N_1 \times N_2$ pairs $\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$ with $\mathbf{x}_i \in \mathcal{S}_1$ and $\mathbf{x}_j \in \mathcal{S}_2$. Solve the binary classification problem using an ordinary SVM estimator in the set of the differences

$$\{(\mathbf{x}_{ij}, +1), (-\mathbf{x}_{ij}, -1) \mid \mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j\},$$

where $+1$ and -1 are the labels of the samples $\mathbf{x}_{ij}$ and $-\mathbf{x}_{ij}$, respectively.

The big families of rankers are pointwise, pairwise and listwise. We focus on pairwise and, in particular, scoring pairwise rankers in order to produce a function $f\colon X \to \mathbb{R}$ so that we can afterwards build a threshold to convert back the ranking score to classes.

4.2 Ranking for Ordinal Class Imbalance

We have already suggested an initial ordinal class adaption of ranking in previous work [2]. Consider all K-tuples $(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(K)})$, with $\mathbf{x}^{(k)} \in \mathcal{S}_k$. There are $\prod_{k=1}^{K} N_k$ of such K-tuples. Generate all pairwise differences between ordered elements in the K-tuple: $\mathbf{x}^{(k)} - \mathbf{x}^{(\ell)}$. There are $K(K-1)$ pairs built from a K-tuple. Like before, learn a binary classifier from the $\frac{K(K-1)}{2} \prod_{k=1}^{K} N_k$ pairs positively labeled, and the corresponding symmetric differences negatively labeled.

Note that in this case, each class is present in exactly the same number of elements in the new space: $(K-1) \prod_{k=1}^{K} N_k$ times. The imbalance binary case presented initially is a special case of this formulation, obtained by setting $K = 2$.

This approach is however repeating the pairs multiple times. A pairwise difference could be constructed with pairs $(\mathcal{C}_\ell, \mathcal{C}_m)$, repeated $(\prod_{k=1}^{K} N_k)/N_\ell N_m$ times. This would, however, be impractical. Several alternatives will be considered.

4.3 Balanced Difference Pairs

In order to balance the pairs of differences, conventional approaches from class imbalance are used, as mentioned in Sect. 3:

(a) Pre-processing;
(b) Training with costs;
(c) Ensembles.

Classes are balanced using the median of the distribution, so that, for every class k, $N'_k = \tilde{N}$, where $\tilde{N}$ is the median. The difference in the number of observations for each class is $\Delta N = \tilde{N} - N_k$. Therefore, the training sampling is oversampled if $\Delta N > 0$ and undersampled if $\Delta N < 0$.

For *(a) Pre-processing*, SMOTE [7] and MSMOTE [11] are evaluated. These oversampling techniques work by creating new synthetic examples based on the minority classes using an average of the nearest neighbors. While the original

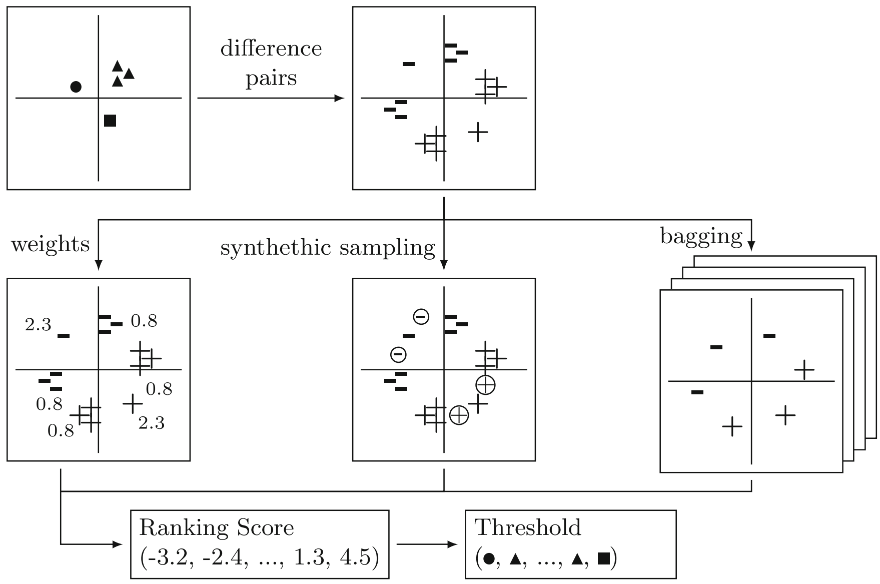

Fig. 1. Balanced ranking training.

paper [7] regards only binary problems, it is easily generalized to multiclass problems. MSMOTE is an extension that adds heuristics to identify points as outliers and refraining from oversampling using those.

(b) Training with costs involves using a cost matrix for each pair in the difference space defined to be inversely proportional to the frequency of each pair so that each pair contributes equally to the decision boundary.

When using *(c) Ensembles*, several strategies have been proposed which make use of either oversampling of the minority class or undersampling of the majority class so that each classifier is trained using balanced data [9]. Some work makes use of boosting where, in addition to fixing the imbalance problem, each classifier is trained to emphasize the most problematic cases. Boosting seems to muddle the causes of the balancing performance gain, and we suggest using bagging.

For the full picture of the method here proposed, with these several approaches, see Fig. 1.

4.4 Threshold for Ordinal Classes

After building the decision boundary from the difference pairs, the resulting score from the pairwise scoring ranker needs then to be transformed back to classes. Based on the training data, we obtain a ranking score s_i for each observation $\mathbf{x}_i$, which can be ordered, assuming that the score represents the order of y_i.

We here define the proposed threshold strategy recursively. Let s_i be the ordered score of observation i and k_i be the true class, we search the threshold left-to-right by invoking the function min_error with initial parameters $(s_0, k_0, 0)$.

$$\text{min\_error}(s_i, k_i, \hat{k}) = \begin{cases} \varepsilon_{k_i \hat{k}}, & \text{when } i = N, \\ \min\left\{\varepsilon_{k_i \hat{k}} + \text{min\_error}(s_{i+1}, k_{i+1}, \hat{k}),\ \text{min\_error}(s_i, k_i, \hat{k}+1)\right\} \end{cases}$$

where $\varepsilon = \left[\varepsilon_{k\hat{k}}\right]$ is a cost matrix. Informally, min_error tests whether, at any given time, it is less costly to continue assuming observation i to be of class $\hat{k}$ or if it less costly to make a threshold and start assuming observations are now $\hat{k}+1$. Notice observations are ordered by the score and classes are ordinal.

For the cost matrix, absolute costs have been used, $\varepsilon_{k\hat{k}} = \{|k - \hat{k}| \mid \forall k, \hat{k}\}$.

5 Experiments

Pairwise scoring ranking has already been experimented within a binary classification context [1]. It has been found that, in a multi-class context, the difference space is no longer balanced. In this section, we experiment with the different balancing approaches previously discussed.

The proposed method is contrasted against state-of-the-art methods: One-vs-Rest SVM, One-vs-Rest SVM with a balanced cost matrix, SVOR [4], and oSVM [3]. All of which also use an SVM as the base estimator.

Each model is cross-validated by grid-search with $C \in \{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$ using k-fold with $k = 3$. Final scores are obtained by 30-fold validation, using the same folds from [8].

Both linear and RBF kernels are tested. The proposed model with linear kernel is implemented by ourselves, while the RBF kernel version uses SVM$^{\text{rank}}$ by Thorsten Joachims[1]. His version has been modified to allow setting weights for each pair of differences. All implementations from our work including the dataset folds are made publicly available[2]. Python and scikit-learn were used.

5.1 Evaluation Metrics

Typically, in binary imbalance problems, special balanced metrics are used. The most popular are F_1 and G-mean. But these metrics are only well-established for binary settings. For ordinal classification, Mean Absolute Error (MAE) is widely used,

$$\text{MAE} = \frac{1}{N} \sum_i |k_i - \hat{k}_i|.$$

But this metric suffers from two problems. First, it treats an ordinal variable as a cardinal variable. Second, the metric is sensible to the per-class distribution of the magnitude of the errors, and is therefore not suitable for class imbalance. Like Pérez-Ortiz et al. [8], since the datasets are imbalance, we will contrast our imbalance ranking approach against conventional methods by using the Maximum Mean Absolute Error (MMAE) metric proposed by [12]. MMAE is defined as

$$\text{MMAE} = \max\{\text{MAE}_k \mid k = 1, \ldots, K\}.$$

[1] https://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html.
[2] http://vcmi.inescporto.pt/reproducible_research/iwann2017/OrdinalImbalance/.

Table 1. Datasets used in the experiments.

Dataset	N	#vars	K	IR
balance-scale	625	4	3	0.170
car	1728	21	4	0.054
contact-lenses	24	6	3	0.267
cooling	768	8	8	0.066
diabetes5	43	2	5	0.091
diabetes10	43	2	10	0.167
newthyroid	215	5	3	0.200
pyrim5	74	27	5	0.250
pyrim10	74	27	10	0.143
squash-stored	52	51	3	0.348
squash-unstored	52	52	3	0.167
stock10	950	9	10	0.131
toy	300	2	5	0.356
triazines5	186	60	5	0.081
triazines10	186	60	10	0.040

5.2 Data

The datasets used come from real problems. The ordinal classification datasets are extracted from the benchmark repositories UCI [13] and mldata.org [14]. Some were originally regression problems converted into ordinal classification, and were obtained from the website of Chu and Ghahramani [15].

Datasets from [8] were used for the experiments, see Table 1. Here, the Imbalance Ratio (IR) metric represents $\text{IR} = \frac{\min_k N_k}{\max_k N_k}$, i.e. the ratio between the number of elements of the minority class to that of the majority class. $\text{IR} \in [0, 1]$, ranging from very imbalance to balanced, respectively. This provides a sense of the imbalance in each dataset.

Tables are ordered alphabetically. For performance reasons, not all datasets are used for the RBF kernel.

5.3 Models

The SVM models tested are **WRank**, **BRank**, **SRank**, and **MSRank**, which correspond to the proposed ranking method balancing difference pairs through weights, bagging and oversampling through SMOTE and MSMOTE, respectively. **OvR** and **OvR/w** are traditional SVM without and with balanced weights. **SVOREX** and **SVORIM** are from [4], and **oSVM** from [3].

These models are compared using linear and RBF SVM kernels. Two tables are presented for each kernel using the metrics discussed above: MAE and MMAE. The average of these metrics for each dataset is exhibited for the 30-fold validation. The best scores are presented in bold. Also in bold are scores which are statistically identical to the best score, using a paired difference Student's t-test with a 95% confidence level.

5.4 Results

Linear kernel results are presented in Table 2 for the aforementioned MAE and MMAE metrics. RBF kernel results are presented in Table 3.

Table 2. Results for SVMs with Linear kernel.

MAE									
Dataset	**WRank**	**BRank**	**SRank**	**MSRank**	OvR	OvR/w	SVOREX	SVORIM	oSVM
balance-scale	0.12	0.52	0.11	1.00	0.20	0.19	**0.11**	**0.11**	0.12
car	0.09	0.11	0.09	1.07	0.12	0.09	0.14	0.12	**0.08**
contact-lenses	0.42	0.48	**0.38**	**0.39**	**0.42**	**0.44**	0.51	0.54	**0.42**
cooling	**0.41**	1.05	1.13	1.28	0.44	0.55	0.48	0.50	0.49
diabetes5	**0.64**	**0.67**	0.74	0.77	0.72	0.95	0.84	**0.67**	0.85
diabetes10	**1.68**	**1.72**	1.77	1.77	2.06	2.15	1.81	**1.69**	2.41
newthyroid	0.04	0.18	0.05	1.00	0.04	**0.03**	0.04	0.04	**0.03**
pyrim5	**0.58**	0.99	1.16	1.16	**0.58**	0.70	1.08	0.99	0.65
pyrim10	1.34	**1.26**	**1.29**	1.33	1.52	1.49	2.89	**1.32**	1.50
squash-stored	0.46	0.86	1.11	1.13	0.47	0.47	**0.41**	0.44	**0.38**
squash-unstored	**0.27**	**0.27**	**0.27**	**0.26**	0.30	**0.30**	**0.26**	0.26	0.33
stock10	0.64	0.66	0.67	1.02	**0.42**	0.42	0.68	0.63	0.70
toy	**0.84**	0.93	**0.87**	1.21	1.02	1.01	1.13	0.95	0.96
triazines5	0.70	1.31	0.98	1.08	0.69	**0.67**	**0.67**	**0.67**	**0.70**
triazines10	1.40	1.95	2.01	1.97	**1.33**	1.51	**1.37**	1.39	1.45
Average	0.64	0.86	0.84	1.10	0.69	0.73	0.83	0.69	0.74
Deviation	0.48	0.52	0.58	0.42	0.55	0.58	0.73	0.48	0.62
Winner	40%	26%	26%	13%	26%	26%	33%	33%	33%
MMAE									
Dataset	**WRank**	**BRank**	**SRank**	**MSRank**	OvR	OvR/w	SVOREX	SVORIM	oSVM
balance-scale	0.21	1.01	0.15	1.10	1.00	0.96	**0.17**	**0.14**	0.21
car	0.47	1.06	**0.28**	1.15	0.77	**0.29**	1.36	1.01	0.49
contact-lenses	**0.81**	1.20	**0.78**	**0.76**	**0.88**	**0.73**	0.97	1.04	**0.82**
cooling	2.32	**1.72**	**1.73**	1.81	2.24	**1.80**	2.98	2.88	2.07
diabetes5	**1.15**	**1.17**	**1.20**	**1.23**	1.48	1.52	1.51	1.30	1.43
diabetes10	**3.09**	**3.16**	**3.12**	3.26	3.82	3.98	3.47	**2.94**	4.33
newthyroid	0.14	1.00	**0.09**	1.04	0.16	0.13	0.14	0.14	**0.13**
pyrim5	**1.40**	1.83	2.26	2.31	**1.30**	1.62	3.00	2.00	1.87
pyrim10	**3.80**	**3.91**	**3.84**	**3.75**	**3.86**	**3.84**	6.37	4.33	4.18
squash-stored	0.83	1.23	1.42	1.46	1.06	0.94	0.76	0.83	**0.66**
squash-unstored	0.57	1.00	0.55	0.52	0.76	0.80	**0.46**	0.46	**0.57**
stock10	1.02	1.04	0.97	1.64	1.03	**0.85**	1.30	1.05	1.29
toy	**1.79**	2.17	**1.67**	2.49	1.92	**1.57**	3.00	2.00	1.82
triazines5	2.77	2.11	**1.73**	2.52	2.99	2.94	3.00	3.00	2.79
triazines10	6.14	**4.46**	**4.58**	**4.55**	6.58	6.35	7.00	6.83	6.80
Average	1.77	1.87	1.63	1.97	1.99	1.89	2.37	2.00	1.96
Deviation	1.58	1.08	1.30	1.12	1.63	1.64	2.02	1.74	1.81
Winner	40%	33%	66%	26%	20%	40%	13%	13%	26%

Table 3. Results for SVMs with RBF kernel.

MAE								
Dataset	**WRank**	**BRank**	**SRank**	**MSRank**	OvR	OvR/w	SVOREX	SVORIM
balance-scale	0.11	0.92	0.14	0.15	0.14	0.18	0.11	**0.05**
car	0.13	0.19	0.25	0.24	**0.12**	0.21	0.41	0.41
contact-lenses	0.52	0.51	0.45	0.46	0.47	**0.33**	1.31	0.98
cooling	0.62	0.61	0.63	0.63	**0.54**	0.62	0.58	**0.55**
diabetes5	**0.65**	P**0.68**	**0.64**	**0.65**	**0.65**	**0.67**	0.72	**0.69**
diabetes10	**1.35**	1.53	1.38	**1.37**	1.75	1.78	1.62	1.56
newthyroid	0.29	0.31	0.29	0.29	0.25	0.23	**0.16**	0.16
pyrim5	**0.47**	0.64	0.56	0.60	1.08	1.10	1.08	0.99
pyrim10	**1.02**	1.18	**1.03**	**1.06**	2.73	2.18	2.88	2.00
squash-stored	**0.57**	0.57	0.57	0.57	0.73	0.57	0.73	0.57
squash-unstored	0.54	0.54	0.54	0.54	**0.44**	0.44	0.49	0.50
stock10	1.35	1.38	1.33	1.30	0.18	**0.17**	0.27	0.26
toy	**0.03**	0.12	**0.03**	**0.04**	0.91	0.66	1.08	0.95
triazines5	**0.68**	1.10	0.88	0.87	**0.67**	1.18	0.67	0.67
triazines10	**1.28**	1.76	1.67	1.64	1.37	2.45	1.37	1.37
Average	0.64	0.80	0.69	0.69	0.80	0.85	0.90	0.78
Deviation	0.42	0.48	0.46	0.45	0.68	0.72	0.69	0.52
Winner	53%	6%	20%	26%	33%	20%	6%	20%
MMAE								
Dataset	**WRank**	**BRank**	**SRank**	**MSRank**	OvR	OvR/w	SVOREX	SVORIM
balance-scale	0.20	1.79	0.19	0.19	1.00	0.24	1.00	**0.13**
car	1.98	2.00	1.00	1.01	1.13	**0.27**	3.00	3.00
contact-lenses	1.27	1.97	1.23	1.22	0.88	**0.53**	1.82	1.32
cooling	1.26	1.22	**0.99**	**0.98**	1.77	**1.01**	2.00	2.00
diabetes5	**1.90**	**1.97**	**1.97**	**1.88**	2.00	2.00	2.00	2.00
diabetes10	3.57	3.77	3.63	3.57	4.27	4.27	**3.06**	**2.84**
newthyroid	1.00	1.03	1.00	1.00	1.00	0.97	**0.64**	0.64
pyrim5	**1.15**	1.78	**1.14**	**1.22**	3.00	2.90	3.00	2.00
pyrim10	**2.65**	3.65	2.82	2.82	6.63	5.67	6.30	4.93
squash-stored	**1.00**	1.00	1.00	1.00	2.00	1.00	2.00	1.00
squash-unstored	**1.00**	1.00	1.00	1.00	1.00	1.00	1.00	1.00
stock10	3.71	4.84	3.27	3.16	0.66	**0.45**	1.14	1.11
toy	**0.10**	1.00	**0.10**	**0.10**	1.83	1.15	2.80	2.00
triazines5	2.66	**1.93**	**1.97**	**1.91**	3.00	2.82	3.00	3.00
triazines10	5.83	**4.73**	**4.99**	**4.69**	7.00	5.66	7.00	7.00
Average	1.95	2.25	1.75	1.72	2.48	1.99	2.65	2.26
Deviation	1.47	1.29	1.32	1.25	1.95	1.81	1.76	1.71
Winner	40%	20%	40%	40%	0%	26%	13%	13%

Using absolute costs in the thresholds performs usually better, albeit inverse frequency costs offers some competitiveness, especially in weighted models using MMAE. Interestingly, the RBF kernel is more stable across the two metrics.

Weighted difference pairs seem to provide the best results, albeit sometimes surpassed by SMOTE. MSMOTE is not found to be a competitive variant of SMOTE; rarely performing better and suffering from higher validation deviation for the linear kernel. These results are similar to the binary experiments from [1].

Bagging with models trained using undersampled and simple oversample of pairs did not offer much compelling results.

One-vs-Rest SVM is sometimes competitive, possibly due to the fact that its decision boundaries are not parallel constrained like traditional ordinal models.

6 Conclusion

Four traditional approaches are used in improving imbalance datasets metrics: pre-processing, using cost matrices, post-processing and ensembles, and often combinations of these. In a previous work, we have suggested ranking as an unexplored alternative to imbalance problems [1], in particular pairwise scoring ranking. Pairwise ranking models use an underlying estimator training in the space of difference pairs, therefore a necessarily balanced dataset in the binary case.

In a follow-up, the ordinal case was addressed [2]. However, it was verified that the imbalance problem was co-occurring in the new space. In this work, the new space is balanced through traditional balancing approaches, with the application of weights being generally superior.

Acknowledgment. This work was funded by the Project "NanoSTIMA: Macro-to-Nano Human Sensing: Towards Integrated Multimodal Health Monitoring and Analytics/NORTE-01-0145-FEDER-000016" financed by the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, and through the European Regional Development Fund (ERDF), and also by Fundação para a Ciência e a Tecnologia (FCT) within PhD grant numbers SFRH/BD/122248/2016 and SFRH/BD/93012/2013.

References

1. Cruz, R., Fernandes, K., Cardoso, J.S., Pinto Costa, J.F.: Tackling class imbalance with ranking. In: International Joint Conference on Neural Networks (IJCNN). IEEE (2016)
2. Cruz, R., Fernandes, K., Pinto Costa, J.F., Perez Ortiz, M., Cardoso, J.S.: Ordinal class imbalance with ranking. In: Rojas, I., et al. (eds.) IWANN 2017, Part II. LNCS, vol. 10306, pp. 538–548. Springer, Cham (2017)
3. Cardoso, J.S., Costa, J.F.: Learning to classify ordinal data: the data replication method. J. Mach. Learn. Res. **8**(Jul), 1393–1429 (2007)
4. Chu, W., Sathiya Keerthi, S.: New approaches to support vector ordinal regression. In: Proceedings of the 22nd International Conference on Machine Learning, pp. 145–152. ACM (2005)
5. Herbrich, R., Graepel, T., Obermayer, K.: Support vector learning for ordinal regression. In: Ninth International Conference on Artificial Neural Networks, ICANN 1999, (Conf. Publ. No. 470), vol. 1, pp. 97–102. IET (1999)
6. Pinto Costa, J.F., Sousa, R., Cardoso, J.S.: An all-at-once unimodal SVM approach for ordinal classification. In: Ninth International Conference on Machine Learning and Applications (ICMLA), pp. 59–64. IEEE (2010)

7. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002)
8. Pérez-Ortiz, M., Gutiérrez, P.A., Hervás-Martínez, C., Yao, X.: Graph-based approaches for over-sampling in the context of ordinal regression. IEEE Trans. Knowl. Data Eng. **27**(5), 1233–1245 (2015)
9. Liu, X.-Y., Jianxin, W., Zhou, Z.-H.: Exploratory undersampling for class imbalance learning. IEEE Trans. Syst. Man Cybern. **39**(2), 539–550 (2009)
10. Sahare, M., Gupta, H.: A review of multi-class classification for imbalanced data. Int. J. Adv. Comput. Res. **2**(5), 160–164 (2012)
11. Hu, S., Liang, Y., Ma, L., He, Y.: MSMOTE: improving classification performance when training data is imbalanced. In: 2nd International Workshop on Computer Science and Engineering, WCSE 2009, vol. 2, pp. 13–17 (2009)
12. Cruz-Ramírez, M., Hervás-Martínez, C., Sánchez-Monedero, J., Gutiérrez, P.A.: Metrics to guide a multi-objective evolutionary algorithm for ordinal classification. Neurocomputing **135**, 21–31 (2014)
13. Lichman, M.: UCI Machine Learning Repository (2013). http://archive.ics.uci.edu/ml
14. PASCAL. Pascal (pattern analysis, statistical modelling and computational learning) machine learning benchmarks repository (2011). http://mldata.org/
15. Chu, W., Ghahramani, Z.: Gaussian processes for ordinal regression. J. Mach. Learn. Res. **6**(Jul), 1019–1041 (2005)