Feature Selection for Ordinal Text Classification

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani

Abstract—*Ordinal classification* (also known as *ordinal regression*) is a supervised learning task that consists of automatically determining the implied rating of a data item on a fixed, discrete rating scale. This problem is receiving increased attention from the sentiment analysis / opinion mining community, due to the importance of automatically rating increasing amounts of product review data in digital form. As in other supervised learning tasks such as (binary or multiclass) classification, feature selection is needed in order to improve efficiency and to avoid overfitting. However, while feature selection has been extensively studied for other classification tasks, is has not for ordinal classification. In this paper we present four novel feature selection metrics that we have specifically devised for ordinal classification, and test them on two datasets of product review data against three metrics previously known from the literature, using two learning algorithms from the "support vector regression" tradition. The experimental results show that all four proposed metrics largely outperform all of the three baseline techniques, on both datasets and for both learning algorithms.

Index Terms—Ordinal regression, ordinal classification, feature selection, product reviews.

1 Introduction

Text management tasks such as *ad hoc* text retrieval, text clustering, and text classification, are usually tackled by representing the textual documents in vector form. The resulting vector spaces are always characterized by a high dimensionality (often in the range of the tens, sometimes hundreds of thousands dimensions), since words (or word stems) are normally used as features, and since many thousands of them occur in any reasonably-sized document space. This very high dimensionality is not very problematic in *ad hoc* retrieval, where the basic operation (computing the distance of two vectors in the vector space) can be performed quickly, thanks to the sparse nature of the two vectors. It is instead problematic in other tasks involving supervised or unsupervised learning, such as text classification or clustering.

For instance, in text classification many supervised learning devices, such as neural networks, do not scale well to large numbers of features, and even those learning devices that do scale well have a computational cost at least linear in the dimensionality of the vector space. While this negatively impacts on efficiency, effectiveness suffers too, since if the ratio of the number of training examples to the number of features is low, overfitting occurs. For all these reasons, in text managements tasks involving learning, the high dimensionality of the vector space may be problematic. Several techniques for reducing the dimensionality of a vector space for text learning tasks have been investigated, the main one being *feature selection* (FS). This latter consists in identifying a subset $S \subset T$ of the original feature set T such that $|S| \ll |T|$ ($\xi = |S|/|T|$ being called the *reduction level*) and such that S reaches the best compromise between (a) the effectiveness of the resulting classifiers and (b) the efficiency of the learning process and of the classifiers (which is, of course, inversely proportional to $|S|$).

While feature selection mechanisms have been extensively investigated for text classification (see e.g., [2], [3]), and to a lesser extent for text clustering (see e.g., [4]), they have not for a related and important text learning task, namely, ordinal text classification. *Ordinal classification* (OC – also known as *ordinal regression*) consists in estimating a *target function* $\Phi : X \rightarrow R$ which maps each object $x_i \in X$ into exactly one of an ordered sequence (that we here call *rankset*) $R = \langle r_1 \prec \ldots \prec r_n \rangle$ of *ranks* (aka "scores", or "labels", or "classes"), by means of a function $\hat{\Phi}$ called the *classifier*[1]. This problem is somehow intermediate between *single-label classification*, in which R is instead an unordered set, and *metric regression*, in which R is instead a continuous, totally ordered set (typically: the set $\mathbb{R}$ of the reals). A key feature of ordinal regression is also that the "distances" between consecutive ranks may be different from each other.

OC is of key importance in the social sciences, since human judgments and evaluations tend to be expressed on ordinal (i.e., discrete) scales; an example of this is customer satisfaction data, where customers may evaluate a product or service on a scale consisting of the values Disastrous, Poor, Fair, Good, Excellent.

In this paper we address the problem of feature selection for OC. Here of course we are only interested in "filter" approaches to FS, i.e., greedy approaches in which a mathematical function f is applied to each feature in T in order to compute its expected contribution to solving the classification task, after which only the $x = |S|$ top-scoring features are retained [5] (x may be

This is a revised and substantially extended version of a paper appeared as [1].

• *All authors are with Istituto di Scienza e Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche, 56124 Pisa, Italy.*
E-mail: firstname.lastname@isti.cnr.it

1. Consistently with most mathematical literature we use the caret symbol (ˆ) to indicate estimation.

a predetermined number or may, more typically, be expressed as a percentage of $|T|$). "Wrapper" approaches, in which entire subsets of x features are evaluated as a whole through the actual learn-and-classify process, are instead not feasible in text-related applications, due to the high dimensionality of the feature space[2]. We here present four novel feature selection metrics that we have specifically devised for ordinal classification, and thoroughly test them on two datasets of product review data by using two SVM-based algorithms for ordinal regression[3].

The paper is organized as follows. In Section 2 we discuss related work, and in Section 3 we present our four FS algorithms for OC. Section 4 reports the results of experiments we have conducted on these methods, while Section 5 concludes.

2 Related work

To the best of our knowledge, there are only two papers [7], [8] that address feature selection for ordinal regression.

Mukras et al. [8] propose an algorithm, called *probability redistribution procedure* (PRP), that takes as input the distribution of the feature t_k across the ranks (as deriving from the distribution across the ranks of the training examples containing t_k) and modifies it, according to the notion that, when t_k occurs in (a document belonging to) a rank r_j, it is "also taken to occur", to a degree linearly decaying with the distance from r_j, in the ranks close to r_j. The modified distribution is then used in selecting features through a standard application, as in binary classification, of the information gain function.

Baccianella et al. [7] describe two methods called *minimum variance* (here noted Var) and *round robin on minimum variance* ($RR(Var)$). The basic idea underlying Var is that of measuring the variance of the distribution of a feature t_k across the ranks, and retaining only the features that have the smallest variance. The intuition behind Var is that a feature is useful iff it is capable of discriminating a small, contiguous portion of the rankset from the rest, and that features with a small variance are those which tend to satisfy this property.

$RR(Var)$ is instead based on the observation (originally discussed in [9] for binary text classification and for functions other than Var) that Var might select many features that discriminate well some of the ranks, while selecting few or no features that discriminate well the other ranks. In order to solve this problem, in $RR(Var)$ one provisionally "assigns" each feature t_k to the rank $r(t_k)$ closest to its average rank value (which is the rank that t_k presumably best discriminates), orders for each rank the features assigned to it, and then has the n ranks take turns, according to a "round robin" (RR) policy, in picking features from the top-most elements of their rank-specific orderings.

2. Interestingly, the literature on FS for metric regression seems to have mostly, if not only, investigated "wrapper" approaches [6].

3. The source code of all the techniques discussed in this paper (both the ones proposed here and the ones from the literature which we use as baselines) can be downloaded at http://patty.isti.cnr.it/~baccianella/FS4OR/sourcecode/.

3 Feature Selection for Ordinal Classification

Let us fix some notation. Let $R = \langle r_1 \prec \ldots \prec r_n \rangle$ be an ordered sequence of ranks, or *rankset*. Given any $j \in \{1, \ldots, (n-1)\}$, $\underline{R}_j = \{r_1, \ldots, r_j\}$ will be called a *prefix* of R, and $\overline{R}_j = \{r_{j+1}, \ldots, r_n\}$ will be called a *suffix* of R. As mentioned in Section 1, our task is to estimate (from a training set Tr) a target function $\Phi : D \rightarrow R$ which assigns each document $d_i \in D$ to exactly one rank in R by means of a classifier $\hat{\Phi}$, to be evaluated on a test set Te. Our feature selection methods will consist of (a) scoring each feature $t_k \in T$ by means of a scoring function $Score$ that measures the predicted utility of t_k for the classification process (the higher the value of $Score$, the higher the predicted utility), and (b) given a predetermined reduction level ξ, selecting the $|S| = \xi \cdot |T|$ features with the highest $Score$. For convenience we will often express reduction levels as percentages, e.g., writing 1% in place of 0.01.

3.1 The Var*IDF method

$Var * IDF$ is a variant of the Var method described in [7]. Recall from Section 2 that Var is based on the intuition of retaining the features that minimize the variance of their distribution across the ranks. For instance, a feature t_k that occurs only in (training documents belonging to) rank r_j has, in the training set, variance $Var(t_k)$ equal to zero; this feature seems obviously useful, since its presence in a test document d_i can be taken as an indication that d_i belongs to r_j. By the same token, a feature t_1 that occurs 90% of the times in r_j and the remaining 10% in a rank *contiguous to* r_j has lower variance than a feature t_2 that occurs 90% of the times in r_j and the remaining 10% in a rank *faraway from* r_j. Feature t_1 is also more useful than t_2 since the presence of t_1 in a test document d_i tends to indicate that d_i belongs to r_j or its vicinity, while t_2 gives a less clearcut indication. In sum, we are interested in retaining features with low variance and discarding ones with high variance.

However, we here note that a feature t_k that occurs only once in the entire training set (e.g., in rank r_j) is trivially such that $Var(t_k) = 0$, but is not useful, since the fact that it occurs exactly in r_j might be due to chance. The features that we are really interested in are those that have low variance *and* high frequency of occurrence (in the training set), since this high frequency of occurrence (a) lends statistical robustness to the estimated value of their variance, and (b) tends to guarantee that the feature will be encountered often in the test set, and will thus contribute to the classification of *many* test documents.

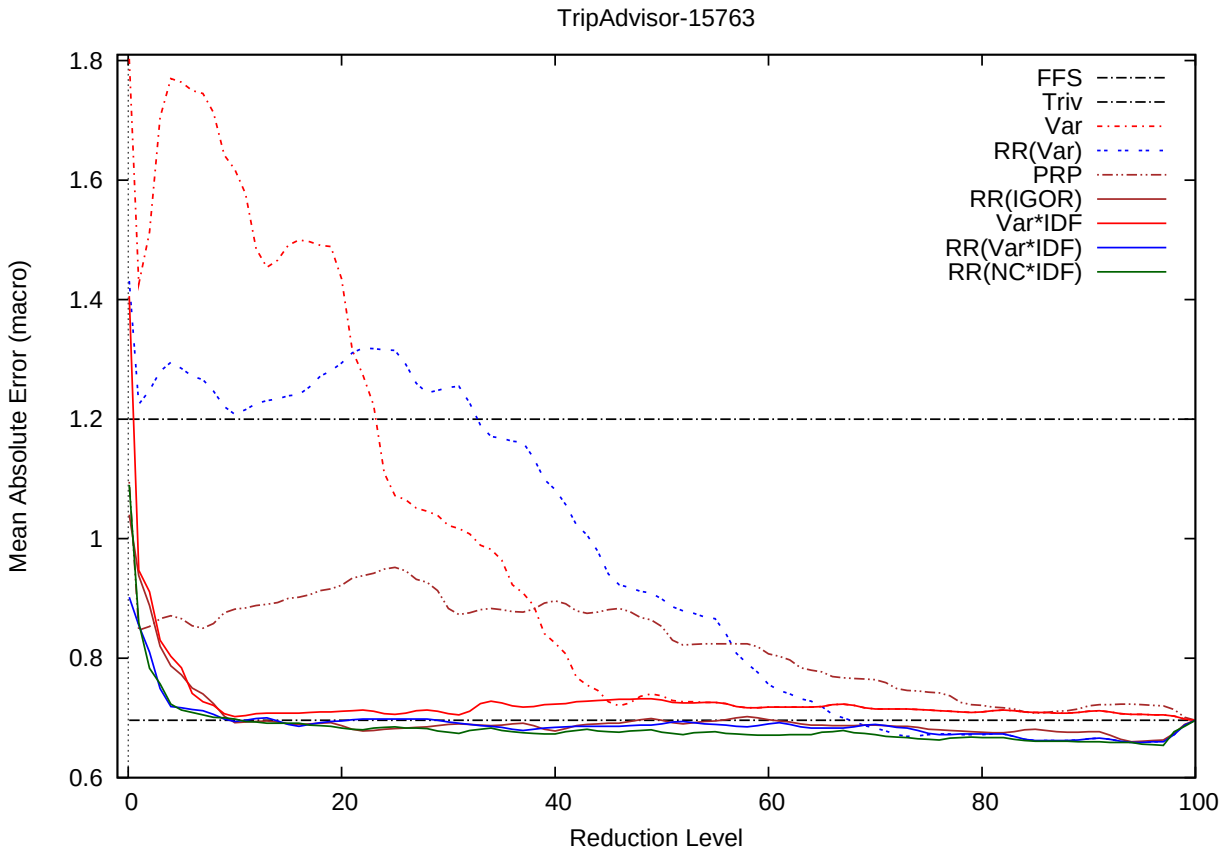

Fig. 1. Results obtained with three baseline feature selection techniques (coloured dotted lines) and our four novel techniques (coloured solid lines) on the TripAdvisor-15763 dataset with the ϵ-SVR learner. Results are evaluated with MAE^M; lower values are better. "FFS" refers to the full feature set (i.e., $\xi = 1$), while "Triv" refers to uniform assignment to the trivial class.

We formalize this by defining

$$Score(t_k) = -(Var(t_k) + \epsilon) * (IDF(t_k))^a \qquad (1)$$

where (i) IDF is the standard inverse document frequency, defined as $IDF(t_k) = \log \frac{|Tr|}{\#_{Tr}(t_k)}$ (where $\#_{Tr}(t_k)$ denotes the number of training documents that contain feature t_k), and (ii) a is a parameter (to be optimized on a validation set) that allows to fine-tune the relative contributions of variance and IDF to the product. Note that, when $Var(t_k) = 0$, we still have $Score(t_k) = 0$, irrespectively of the value of $IDF(t_k)$, which is undesirable. As a result, we smooth $Var(t_k)$ by adding to it a small value ϵ prior to multiplying it by $IDF(t_k)$.

The x features with the highest $Score$ value are retained while the others are discarded.

3.2 The RR(Var*IDF) method

Similarly to Var, the $Var * IDF$ method runs the risk of exclusively catering for a certain rank and disregarding the others. If all the retained features mostly occur in rank r_j and its neighbouring ranks, the resulting feature set will exclusively contain features good at discriminating r_j and its neighbouring ranks from the rest, while other ranks might not be adequately discriminated by any of the remaining features.

Similarly to the $RR(Var)$ method hinted at in Section 2, in order to pick the best x features the $RR(Var * IDF)$ method thus (i) provisionally "assigns" each feature t_k to the rank closest to the mean of its distribution across the ranks; (ii) orders, for each rank r_j, the features provisionally assigned to r_j in terms of their value of the $Score$ function of Equation 1, with the highest-scoring ones at the top of the ordering; and (iii) enforces a "round robin" policy in which the n ranks take turns in picking their favourite features from the top-most elements of their rank-specific orderings, until x features are picked. In this way, for each rank r_j the final set of selected features contains at least[4] the $\frac{x}{n}$ features that are best at discriminating r_j and its neighbouring ranks, which means that all the ranks in the rankset R are adequately championed in the final feature set S.

4. The "at least" here means that, since the same feature may be a top-scoring one for more than one rank, strictly more than $\frac{x}{n}$ features per rank may eventually get selected. Similar considerations apply to the $RR(IGOR)$ and $RR(NC * IDF)$ techniques to be discussed in Sections 3.3 and 3.4, where a round robin phase is also present.

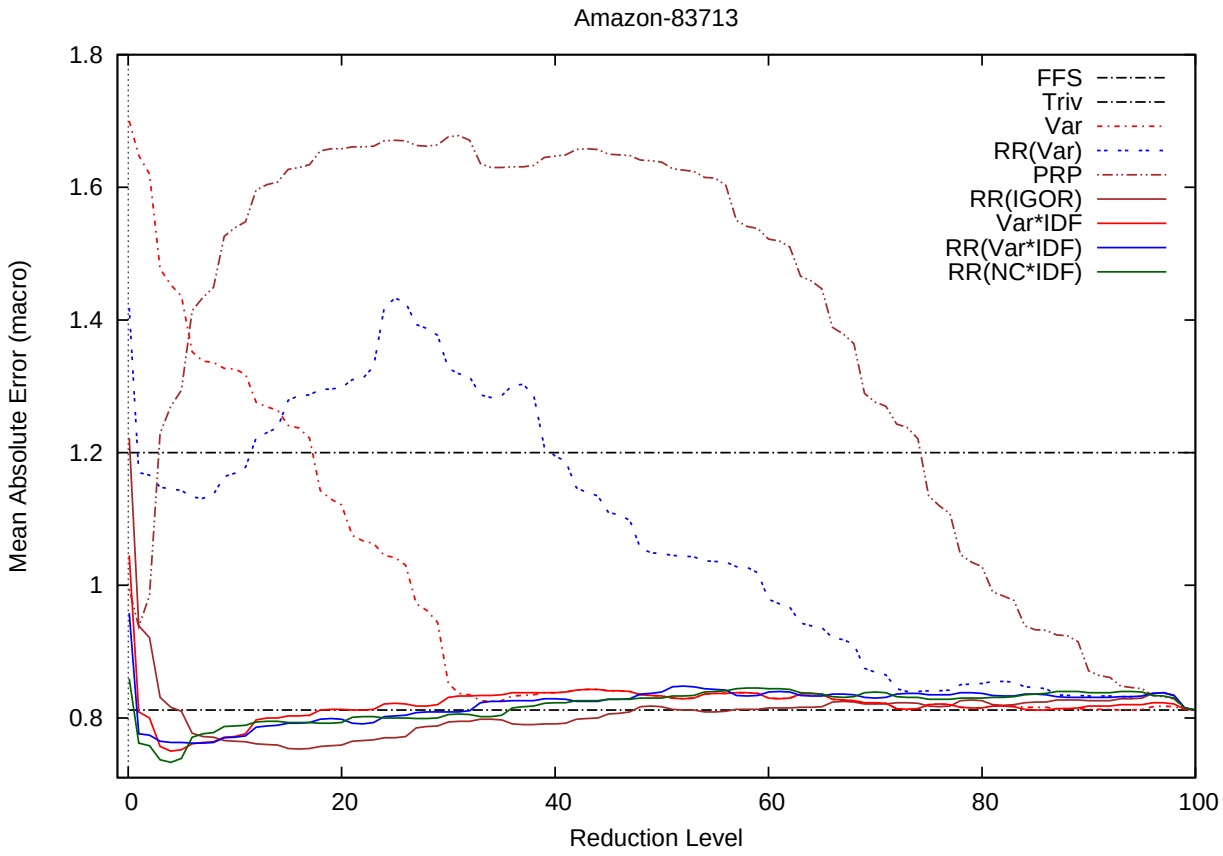

Fig. 2. Same as Figure 1 but on Amazon-83713 instead of on TripAdvisor-15763.

3.3 The RR(IGOR) method

The *round robin on information gain for ordinal regression* ($RR(IGOR)$) method is based on the idea of adapting *information gain*, a function routinely employed in feature selection for binary classification (see e.g. [3]), to ordinal regression[5].

In a binary classification task in which we need to separate class c_j from its complement $\overline{c}_j$ we may perform feature selection by scoring each feature t_k with the function

$$\begin{aligned} IG(t_k, c_j) &= H(c_j) - H(c_j|t_k) = \\ &= \sum_{c\in\{c_j,\overline{c}_j\}} \sum_{t\in\{t_k,\overline{t}_k\}} P(t,c) \log \frac{P(t,c)}{P(t)P(c)} \end{aligned}$$

where $H(\cdot)$ indicates entropy and $H(\cdot|\cdot)$ indicates conditional entropy. $IG(t_k, c_j)$ measures the reduction in the entropy of c_j obtained as a result of observing t_k, i.e., measures the information that t_k provides on c_j; for binary classification the x features with the highest $IG(t_k, c_j)$ value are thus retained, while the others are discarded.

5. Any function routinely used for feature selection in binary classification, such as chi-square or odds ratio, could have been used here in place of information gain.

$RR(IGOR)$ is based on the idea of viewing ordinal regression on rankset $R = \langle r_1 \prec \ldots \prec r_n \rangle$ as the simultaneous generation of $(n-1)$ binary classifiers $\ddot{\Phi}_j$, each of which is in charge of deciding whether a document belongs to (one of the ranks in) prefix $\underline{R}_j = \{r_1 \prec \ldots \prec r_j\}$ or to suffix $\overline{R}_j = \{r_{j+1} \prec \ldots \prec r_n\}$, for $j = 1, \ldots, (n-1)$. For each feature t_k we thus compute $(n-1)$ different $IG(t_k, c_j)$ values, by taking $c_j = r_1 \cup \ldots \cup r_j$ and $\overline{c}_j = r_{j+1} \cup \ldots \cup r_n$, for $j = 1, \ldots, (n-1)$.

Similarly to the $RR(Var*IDF)$ method of Section 3.2, in order to pick the best x features we (i) sort, for each of the $(n-1)$ binary classifiers $\ddot{\Phi}_j$ above, the features in decreasing order of their $IG(t_k, c_j)$ value; and (ii) enforce a round robin policy in which the $(n-1)$ classifiers $\ddot{\Phi}_j$ above take turns, for at least $\frac{x}{n-1}$ rounds, in picking their favourite features from the top-most elements of their classifier-specific orderings. In this way, for each classifier $\ddot{\Phi}_j$ the final set of selected features contains the $\frac{x}{(n-1)}$ features that are best at discriminating the rankset prefix $\underline{R}_j$ from the rankset suffix $\overline{R}_j$, which means that all the classifiers $\ddot{\Phi}_j$ of rankset R are adequately championed in the final feature set S.

Of course the intuition here is that, if test document d_i belongs to rank r_j, classifiers $\ddot{\Phi}_1, \ldots, \ddot{\Phi}_{j-1}$ will be represented in S by features that indicate d_i to belong to their corresponding rankset *suffixes* $R_1, \ldots, R_{j-1}$, while

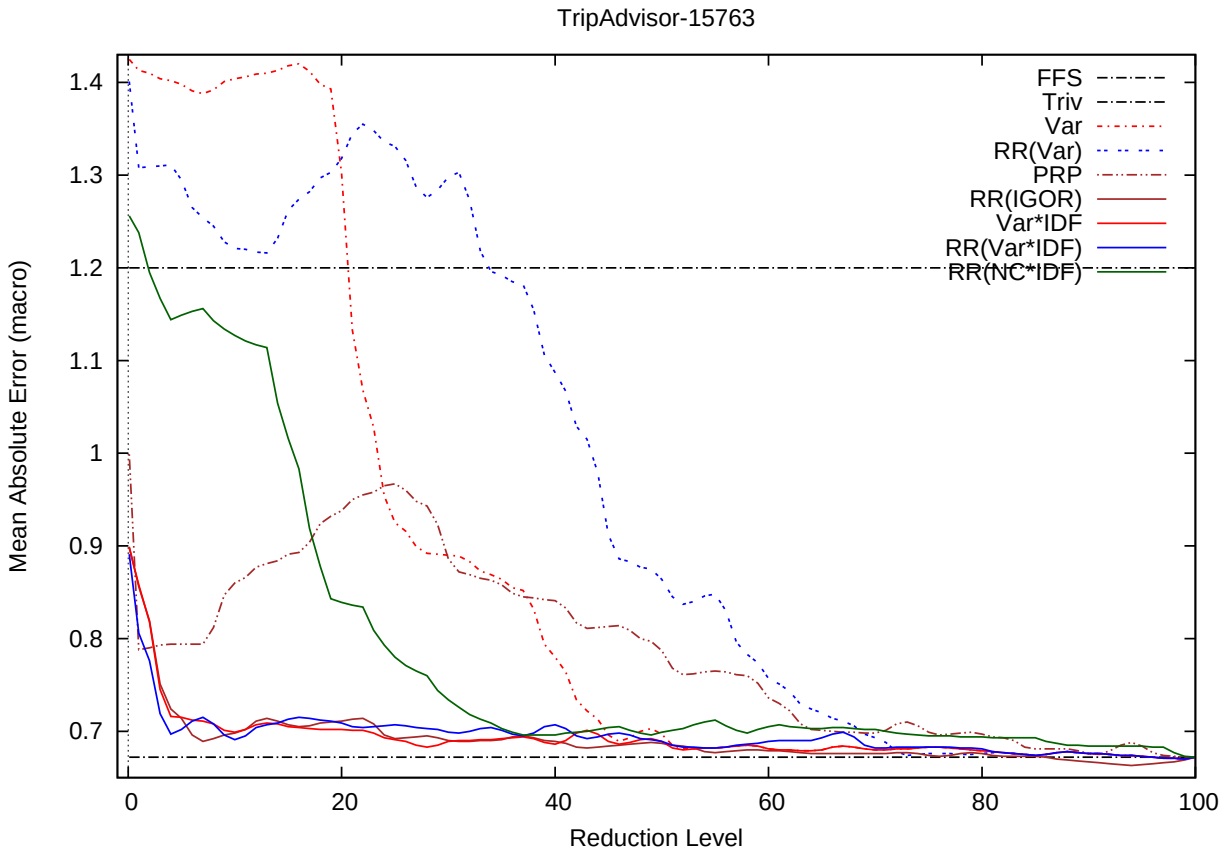

Fig. 3. Same as Figure 1 but with the SVOR learner instead of with the ϵ-SVR learner.

classifiers $\ddot{\Phi}_j, \ldots, \ddot{\Phi}_{n-1}$ will be represented in S by features that indicate d_i to belong to their corresponding rankset *prefixes* $\overline{R}_j, \ldots, \overline{R}_{n-1}$.

3.4 The RR(NC*IDF) method

A problem with the methods we have proposed up to now, and with the ones mentioned in Section 2, is that none of them depends on (i.e., optimizes) the specific evaluation measure chosen for evaluating ordinal regression. The $RR(NC * IDF)$ method tries to address this shortcoming by including the chosen evaluation measure as a parameter, and directly optimizing it.

Assume that E is the chosen evaluation measure, and that $E(\hat{\Phi}, d_i)$ represents the error that classifier $\hat{\Phi}$ makes in classifying d_i. For example, if $\hat{\Phi}(d_i) = r_1$, $\Phi(d_i) = r_2$ and E is absolute error (see Section 4.1.2), then $E(\hat{\Phi}, d_i) = |r_1 - r_2|$. Let us define the *negative correlation* of a feature t_k with a rank r_j in the training set Tr as

$$NC_{Tr}(t_k, r_j) = \frac{\sum_{\{d_i \in Tr \mid t_k \in d_i\}} E(\tilde{\Phi}_j, d_i)}{|\{d_i \in Tr \mid t_k \in d_i\}|} \quad (2)$$

where $\tilde{\Phi}_j$ is the "trivial" classifier that assigns all documents to the same rank r_j. In other words, $NC_{Tr}(t_k, r_j)$ measures how bad an indicator of membership in rank r_j feature t_k is, where "bad" is defined in terms of the chosen error measure; in fact, $NC_{Tr}(t_k, r_j)$ is equal to zero (the best possible value) when, by trivially classifying under r_j all training examples that contain t_k, the error E that we would make amounts to zero.

It would now be tempting to define $Score(t_k, r_j)$ as $-NC_{Tr}(t_k, r_j)$, i.e., as the opposite of the negative correlation between t_k and rank r_j, since $-NC_{Tr}(t_k, R(t_k))$ measures how well t_k identifies r_j. While this is in principle reasonable, for the same reasons as outlined in Section 3.1 we need to compensate for the fact that NC does not pay attention to frequency-of-occurrence considerations; this method might thus select features whose estimates are not statistically robust.

This leads us to defining the $Score$ of a feature t_k with respect to rank r_j as

$$Score(t_k, r_j) = -(NC_{Tr}(t_k, r_j) + \epsilon) * (IDF(t_k))^a \quad (3)$$

where the ϵ and a parameters serve the same purpose as in Equation 1. Similarly to what happens in the $RR(Var * IDF)$ and $RR(IGOR)$ methods, in order to select the best x features we now apply a round robin policy in which each rank r_j is allowed to pick the (at least) $\frac{x}{n}$ features with the best $Score(t_k, r_j)$, so that each rank in

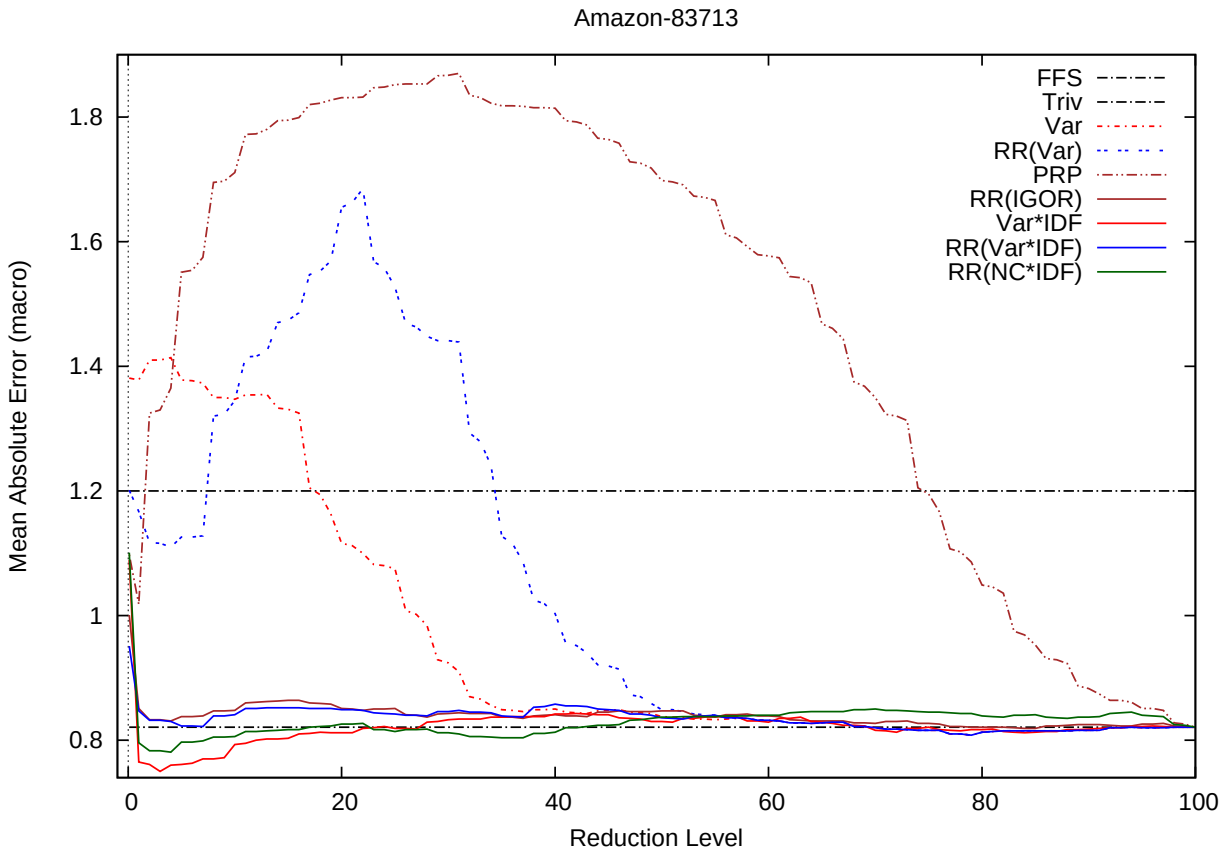

Fig. 4. Same as Figure 1 but on Amazon-83713 instead of on TripAdvisor-15763 *and* with the SVOR learner instead of with the ϵ-SVR learner.

Dataset	$\|Tr\|$	$\|Te\|$	1 Star	2 Stars	3 Stars	4 Stars	5 Stars
TripAdvisor-15763	10,508	5,255	3.9%	7.2%	9.4%	34.5%	45.0%
Amazon-83713	20,000	63,713	16.2%	7.9%	9.1%	23.2%	43.6%

TABLE 1
Main characteristics of the two datasets used in this paper; the last five columns indicate the fraction of documents that have a given number of "stars".

the rankset is well catered for by the final set of features[6].

3.5 A note on rank distance

As remarked in the introduction, the distances between consecutive ranks are not always equal. This may be the case, e.g., when a rankset consists of ranks Poor, Good, Very Good, Excellent, since it might be argued that the distance between Poor and Good is higher than the distance between Very Good and Excellent. Note that all our FS functions, with the exception of $RR(IGOR)$, allow bringing to bear these distances in the feature selection phase. In fact, $Var*IDF$ and $RR(Var*IDF)$ are inherently sensitive to these distances because variance is; and $RR(NC*IDF)$ is also sensitive to them, since one only needs to plug these distances into the E measure used to define negative correlation.

6. The version of the $RR(NC*IDF)$ method that was originally presented in the earlier version of this paper [1] was slightly different from the one presented here, in that it included an intermediate step in which feature t_k was assigned to the rank $R(t_k)$ that was least negatively correlated with it. We have now removed this step since it did not allow t_k to be picked, in the round robin phase, for a rank r_j different from $R(t_k)$, which penalized r_j, since t_k was not necessarily picked for $R(t_k)$. As a result, the current version of the method is simpler and (as the experiments have shown) more effective than the previous version.

4 Experiments

4.1 Experimental setting

4.1.1 The datasets

We have tested the proposed measures on two different datasets, whose characteristics are reported in Table 1.

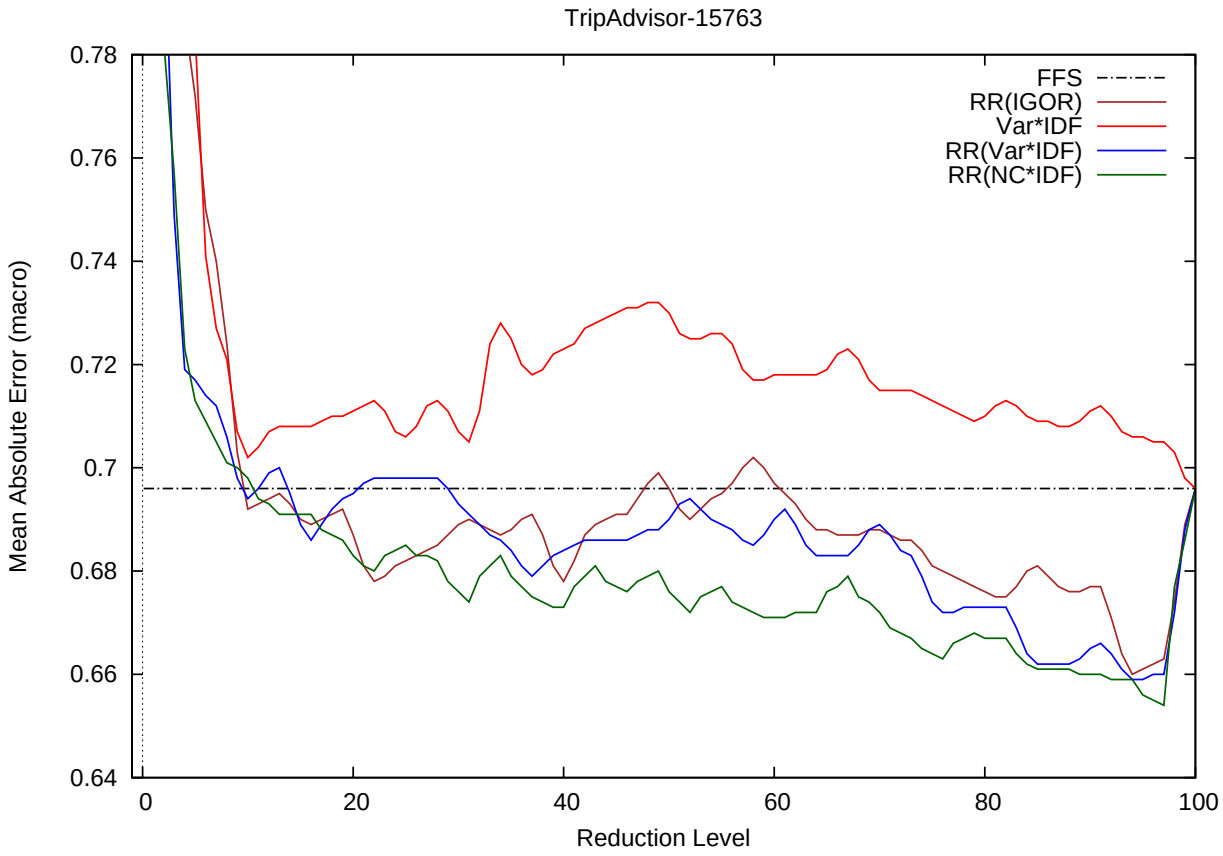

Fig. 5. Close-up on the results obtained with four novel feature selection techniques on the TripAdvisor-15763 dataset with the ϵ-SVR learner. "FFS" refers to the full feature set (i.e., $\xi = 1$).

The first is the TripAdvisor-15763 dataset built by Baccianella et al. [7], consisting of 15,763 hotel reviews from the TripAdvisor Web site[7]. We use the same split between training and test documents as in [7], resulting in 10,508 documents used for training and 5,255 for test; the training set contains 36,670 unique words. From the 10,508 training documents we have randomly picked 3,941 documents to be used as a validation set for parameter optimization.

The second dataset is what we here call Amazon-83713, consisting of 83,713 product reviews from the Amazon Web site; Amazon-83713 is actually a small subset of the Amazon dataset (consisting of more than 5 million reviews) originally built by Jindal and Liu for spam review detection purposes [10], and contains all the reviews in the sections `MP3`, `USB`, `GPS`, `Wireless 802.11`, `Digital Camera`, and `Mobile Phone`. We have randomly picked 20,000 documents to be used for training, and use the remaining 63,713 documents for test; the training set contains 138,964 unique words. From the 20,000 training documents we have randomly selected 4,000 documents to be used as a validation set. To the best of our knowledge, Amazon-83713 is now the largest dataset ever used in the literature on ordinal text classification.

Both datasets consist of reviews scored on a scale from 1 to 5 "stars"; both datasets are highly imbalanced (see Table 1), with positive and very positive reviews by far outnumbering negative and very negative reviews.

4.1.2 Evaluation measures

As our main evaluation measure we use the *macroaveraged mean absolute error* (MAE^M) measure proposed in [11], and defined as

$$MAE^M(\hat{\Phi}, Te) = \frac{1}{n}\sum_{j=1}^{n}\frac{1}{|Te_j|}\sum_{d_i \in Te_j}|\hat{\Phi}(d_i) - \Phi(d_i)| \quad (4)$$

where Te_j denotes the set of test documents whose true rank is r_j and the "M" superscript indicates "macroaveraging". As argued in [11], the advantage of MAE^M over "standard" mean absolute error (defined as

$$MAE^{\mu}(\hat{\Phi}, Te) = \frac{1}{|Te|}\sum_{d_i \in Te}|\hat{\Phi}(d_i) - \Phi(d_i)| \quad (5)$$

where the "μ" superscript stands for "microaveraging") is that it is robust to rank imbalance (which is useful, given the above-mentioned imbalanced nature of our

7. The dataset is available for download from http://patty.isti.cnr.it/~baccianella/reviewdata/

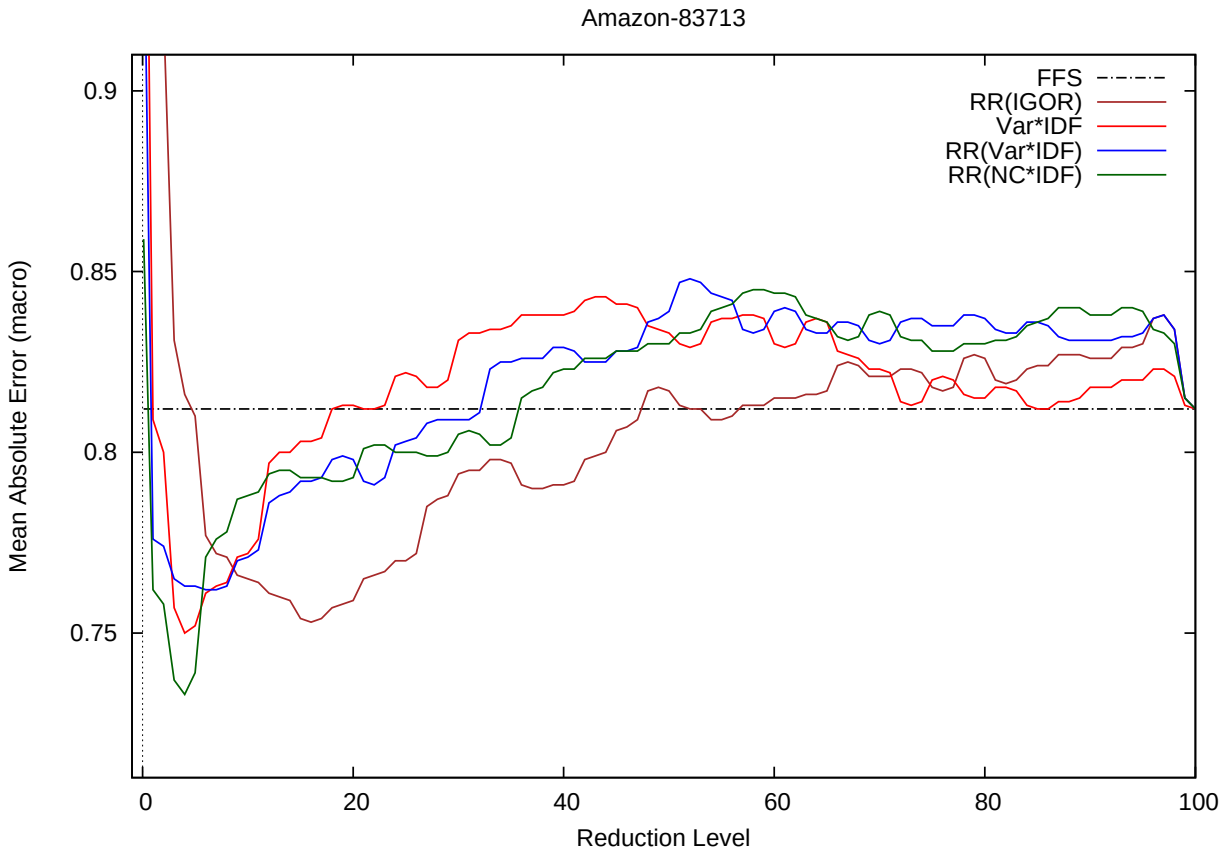

Fig. 6. Same as Figure 5 but on Amazon-83713 instead of on TripAdvisor-15763.

datasets) while coinciding with MAE^{μ} on perfectly balanced datasets (i.e., datasets with exactly the same number of test documents for each rank). For completeness we have computed, additionally to MAE^{M}, also MAE^{μ} results for all our experiments.

4.1.3 Learning algorithms

We have tested our methods with two different SVM-based learning algorithms for ordinal regression: ϵ-SVR [12], originally devised for linear regression and which we have adapted to solve ordinal regression problems, and SVOR [13], which was specifically devised for solving ordinal regression.

ϵ-support vector regression (ϵ-SVR), is the original formulation of support vector regression as proposed in [12]; we have used the implementation from the freely available LibSvm library[8]. ϵ-SVR can be adapted to the case of ordinal regression by (a) mapping the rankset onto a set of consecutive natural numbers (in our case we have simply mapped the sequence [1 Star, ..., 5 Stars] onto the sequence $[1, \ldots, 5]$), and (b) rounding the real-valued output of the classifier to the nearest natural number in the sequence.

SVOR [13] consists instead in a newer algorithm that tackles the ordinal regression problem without using any *a priori* information on the ranks, and by finding $n-1$ thresholds that divide the real-valued line into n consecutive intervals corresponding to the r ordered ranks. The authors propose two different variants: the first (nicknamed SVOREX, for "Support Vector Ordinal Regression with EXplicit constraints") takes into account only the training examples of adjacent ranks in order to determine the thresholds, while the second (SVORIM, for "Support Vector Ordinal Regression with IMplicit constraints") takes into account all the training examples from all of the ranks. Given that the authors have experimentally shown SVORIM to outperform SVOREX, the former (in the implementation available from the authors of [13][9]) is the variant we have adopted for our experiments.

Both learning algorithms use the sequential minimal optimization algorithm for SVMs [14], and both map the solution on the real-valued line. The main difference between them is the use of *a priori* information. In fact, when using ϵ-SVR the user needs to explicitly specify a mapping of the rankset onto a sequence of naturals and to set the thresholds between these latter, while SVOR automatically derives all the needed information from the training set.

We have optimized the γ and C parameters of both

8. http://www.csie.ntu.edu.tw/~cjlin/libsvm/

9. http://www.gatsby.ucl.ac.uk/~chuwei/svor.htm

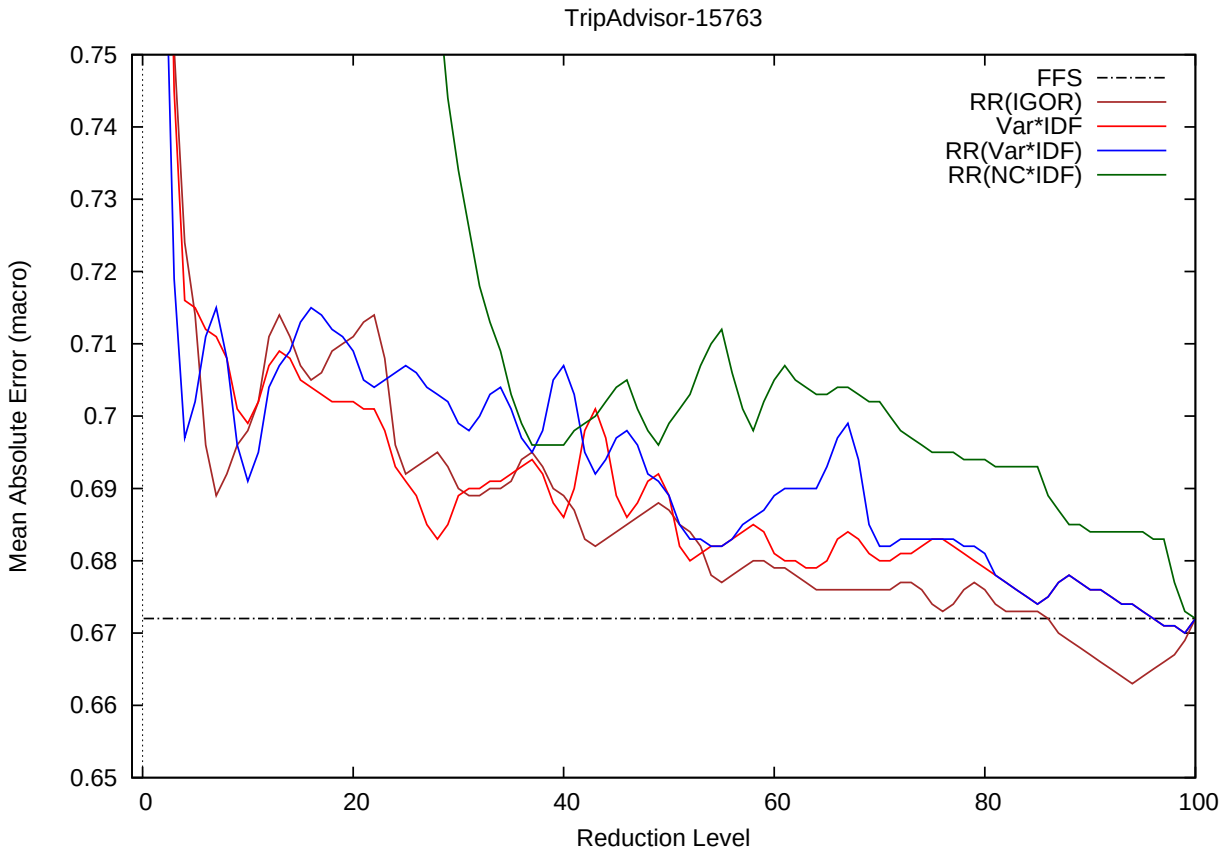

Fig. 7. Same as Figure 5 but with the SVOR learner instead of with the ϵ-SVR learner.

learners on the validation sets described in Section 4.1.1; the validation has been carried out with the full feature set ($\xi = 1$), and the values which have proven optimal have later been used for all reduction levels. The γ parameter has been optimized by testing all values from 2^{-15} to 2^{3} with step 2 in the exponent, while the C parameter has been optimized by testing all values from 2^{-11} to 2^{9} also with step 2 in the exponent; these ranges and steps are the ones recommended by the creators of LibSvm in the "readme" file.

4.1.4 Baselines

For all our experiments we have used three different baseline methods: the first is the PRP method of [8], while the second and third are the Var and $RR(Var)$ methods of [7] (see Section 2 for details).

We also draw comparisons with the "trivial baseline", i.e., with the method that consists in trivially assigning all test documents to the "trivial class", defined as follows. For a given (binary, ordinal, or other) classification problem a *trivial classifier* $\tilde{\Phi}_j$ may be defined as a classifier that assigns all documents to the same class r_j; accordingly, the *trivial class* $\tilde{r}$ may be defined as the class that minimizes the chosen error measure E on the training set Tr across all trivial classifiers, i.e., $\tilde{r} = \arg\min_{r_j \in R} E(\tilde{\Phi}_j, Tr)$. [11] shows that for both MAE^M and MAE^μ the trivial class $\tilde{\Phi}_k$ need not be the majority class, as instead happens for single-label classification when Hamming distance (aka "error rate") is the chosen error measure. For instance, for both the TripAdvisor-15763 and Amazon-83713 datasets, 4 Stars is the trivial class for MAE^μ, since we obtain lower MAE^μ ($MAE^\mu = 0.805$ for TripAdvisor-15763 and $MAE^\mu = 1.171$ for Amazon-83713) in assigning all training documents 4 Stars than by assigning all of them 5 Stars, which is the majority class. [11] also shows that the trivial class(es) for MAE^M are always the middle class $r_{\frac{n+1}{2}}$ (when there is an odd number of classes) or the middle classes $r_{\lfloor \frac{n+1}{2} \rfloor}$ and $r_{\lceil \frac{n+1}{2} \rceil}$. In both the TripAdvisor-15763 and Amazon-83713 datasets there is a single middle class, 3 Stars, which is then the trivial class (uniform assignment to the trivial class yields $MAE^M = 1.200$ for both datasets).

4.1.5 Experimental protocol

As a vectorial representation, after stop word removal (and no stemming) we use standard bag-of words with cosine-normalized $tfidf$ weighting. Note that, as discussed in detail in [7], bag-of words is certainly not the optimal method for generating the internal representations of product reviews: two expressions such as "A great hotel in a horrible town!" and "A horrible hotel

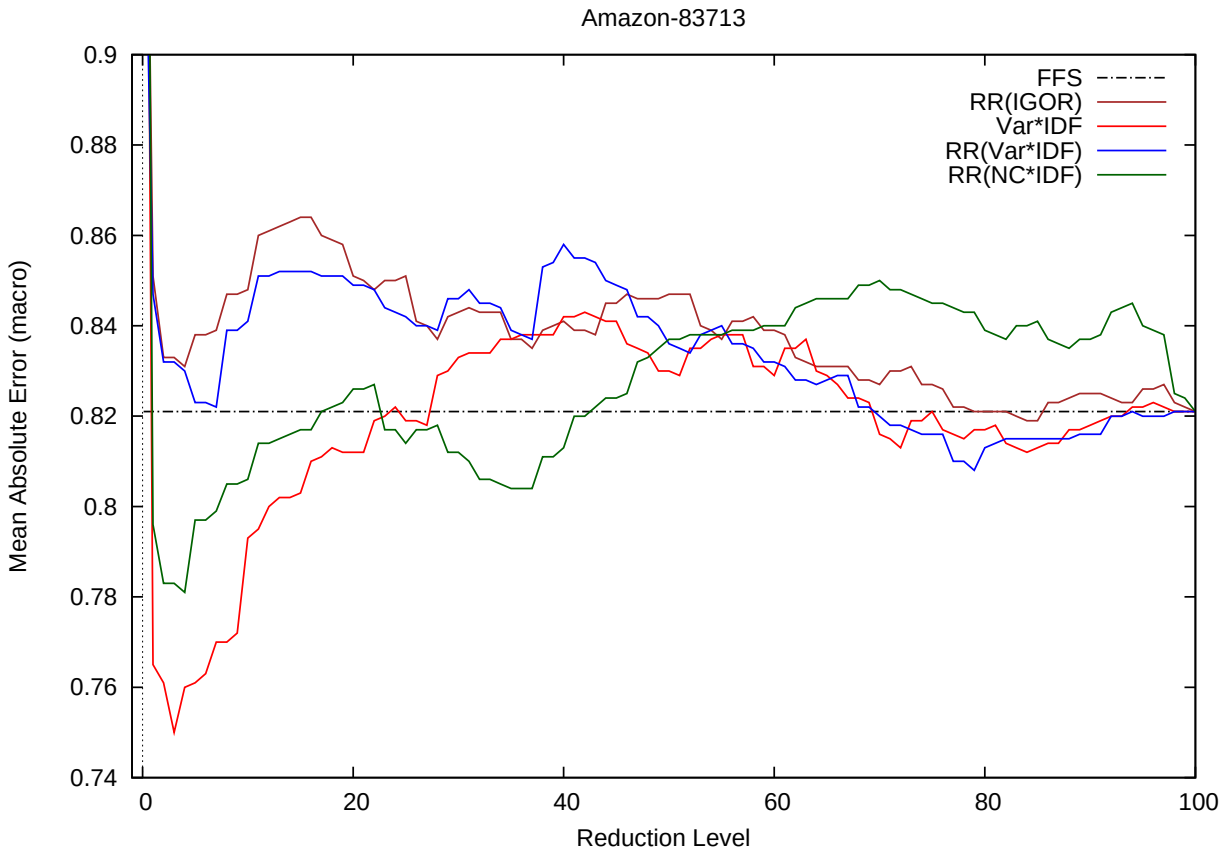

Fig. 8. Same as Figure 5 but on Amazon-83713 instead of on TripAdvisor-15763 *and* with the SVOR learner instead of with the ϵ-SVR learner.

in a great town!" would receive identical bag-of-words representations, while expressing opposite evaluations of the hotel being reviewed. The reason why in this paper we have chosen bag-of-words (instead of the more complex and linguistically richer representations we have championed in [7]) is to guarantee easier replicability by other researchers of the results presented here.

We have run all our experiments for all the 100 reduction levels $\xi \in \{0.001, 0.01, 0.02, 0.03, \ldots, 0.99\}$. For the $Var * IDF$, $RR(Var * IDF)$ and $RR(NC * IDF)$ methods we have set the smoothing parameter ϵ to 0.1. For the same methods we have (individually for each method) optimized the a parameter on the validation sets described in Section 4.1.1 and then re-trained the optimized classifier on the full training set (i.e., including the validation set). During validation, all integer values in the range [1,20] were tested (values smaller than 1 had already shown a dramatic deterioration in effectiveness, and were thus not investigated in detail), and the best value for a given method was retained. For all three methods this optimization was performed with $\xi = 0.10$ (since this is a "paradigmatic" reduction level in much of the literature on feature selection for text classification), and the value that proved optimal was chosen for all feature reduction levels; previous experiments on the validation set had shown anyway that in all cases the chosen parameter value was optimal for any $\xi \in [0.001, 0.20]$.

For $RR(NC * IDF)$, the E error measure was taken to be $|\hat{\Phi}(d_i) - \Phi(d_i)|$ (i.e., absolute error), given that it is the document-level analogue of both MAE^M and MAE^μ.

4.2 Results

The results of our experiments are displayed in Figures 1 to 2, in which the effectiveness of each feature selection policy is plotted as a function of the tested reduction level. The two horizontal lines indicate the effectiveness (measured via MAE^M) obtained (a) by using the full feature set ($\xi = 1$), or (b) by assigning all test documents to the trivial class.

The first observation that comes natural by observing Figures 1 to 2 is that the three baselines are dramatically inferior to the four novel techniques proposed in this paper, except for a few isolated cases (e.g., bad performance of $RR(NC * IDF)$ on Amazon-83713 with SVOR and with $\xi < .30$). PRP is somehow comparable with our novel techniques for very aggressive reduction levels (e.g., $\xi = 0.01$), but is drastically inferior to them for all other reduction levels, even underperforming, on Amazon-83713, the trivial baseline in the range $\xi \in$

		Baselines				Our new techniques			
	ξ	Trivial	Var	$RR(Var)$	PRP	$Var * IDF$	$RR(Var * IDF)$	$RR(IGOR)$	$RR(NC * IDF)$
MAE^M	.01	1.200	1.466	1.213	0.898	0.845	**0.817**	0.856	0.863
	.10	1.200	1.424	1.234	1.248	**0.741**	0.747	0.753	0.767
MAE^μ	.01	0.988	1.027	1.251	0.656	0.644	**0.621**	0.640	0.643
	.10	0.988	0.978	0.949	0.831	0.605	**0.597**	0.618	0.620

TABLE 2
Performance of different feature selection functions at two representative reduction levels (.01 and .10), as averaged across the two different datasets and the two different learners used in this paper. **Boldface** indicates the best performer.

$[0.05, 0.70]$ with both learners (it performs somehow better, bust still worse than our four proposed techniques, on TripAdvisor-15763). Var is, instead, comparable with our techniques for the less aggressive reduction levels (i.e., $\xi \in [0.4, 1.0]$), but it yields very bad results for the more aggressive ones, even below the trivial baseline if $\xi \in [0.001, 0.15]$; this is likely due to the fact that the top-scoring features for the Var method (i.e., the only ones that get selected when the reduction level is very aggressive) are features with very low frequency of occurrence (some of them maybe occurring in a single training document), while when the reduction level is less aggressive also "good" features (i.e., with low variance *and* high frequency of occurrence) are selected. $RR(Var)$ performs instead uniformly worse than the proposed techniques for all reduction levels and on both datasets. On a marginal note, it should be noted that for the more aggressive levels of reduction $RR(Var)$ performs better than Var; this is likely due to the fact that the round robin step is very important when the features are few, since its presence allows each rank to be represented by at least some features that are highly discriminative for it.

From now on we will thus largely ignore the three baseline techniques and focus on discussing our four novel techniques and their differences. In order to do this we will analyze Figures 5 to 8, which present the same results of Figures 1 to 4, respectively, in close-up view, zooming-in on our four techniques.

A second observation is that our proposed techniques are fairly stable across $\xi \in [0.05, 1.0]$, and deteriorate, sometimes rapidly, only for the very aggressive levels, i.e., for $\xi \in [0.001, 0.05]$). This is especially evident on the Amazon-83713 dataset (by far the larger of the two), for both learners. This is in stark contrast with the instability of the baselines; e.g., as noted above, both PRP and Var perform reasonably well for some reduction levels but disastrously for others.

For $\xi \in [0.05, 1.0]$ the accuracy is, for each of our four techniques, more or less comparable to the accuracy obtained with the full feature set (i.e., with no feature selection). The full feature set tends to be, although by a very small margin, the best choice in the TripAdvisor-15763 dataset, while the situation is less clearcut in the much larger Amazon-83713 dataset, with the proposed techniques slightly underperforming the full feature set for $\xi \in [0.3, 1.0]$, and outperforming it for $\xi \in [0.01, 0.3]$. This latter is very good news, since it indicates that one can reduce the feature set by an order of magnitude (with the ensuing benefits in terms of training-time and – especially important – testing-time efficiency) and obtain an accuracy equal or even slightly superior (roughly a 10% improvement, in the best cases) to that obtainable with the full feature set. Incidentally, this is clearly reminiscent of the results obtained by Yang and Pedersen, who, in their seminal paper on feature selection for binary text classification [3], showed that the best feature selection techniques could allow exactly that (i.e., an improvement in effectiveness of about 10% when the size of the feature set is reduced by one order of ($\xi = .10$), which was the reduction level at which the best performance was obtained).

$RR(NC * IDF)$ even marginally outperforms the full feature set when the size of the feature set is reduced by two orders of magnitude ($\xi = .01$), and only marginally underperforms it when the reduction is by three orders of magnitude ($\xi = .001$), regardless of the learner used; we think this is a striking performance.

It is not easy instead to decide which of the four techniques we have proposed is the best, since none of them

consistently outperforms the others regardless of dataset and learner. For instance, for $\xi < .40$ $RR(NC*IDF)$ has a strangely bad performance on TripAdvisor-15763 when SVOR is used, but it performs very well when ϵ-SVR is used or when Amazon-83713 is the dataset. And on the Amazon-83713 dataset, when using ϵ-SVR, $RR(IGOR)$ is clearly the best when $\xi \in [0.10, 0.70]$, while when $\xi \in [0.001, 0.10]$ the best performer is $RR(NC*IDF)$. In order to get a clearer sense of the relative merits of these four techniques, in Table 2 we report the performance of all the techniques discussed for two representative reduction levels (.01 and .10), averaged across the two datasets and the two learners used. The results reported show that, when MAE^M is used as the evaluation measure, $RR(Var*IDF)$ is clearly the best performer for $\xi = .01$, and is close to being the best performer also for $\xi = .01$ (where $RR(Var*IDF)$ outperforms it by a narrow margin).

Finally, note that we have evaluated all the experiments reported here also according to the MAE^μ measure; we do not report these results for reasons of space, and because (as fully argued in) we believe MAE^M to more faithfully represent what a user wants from an ordinal regression device[10]. Suffice it to say that the MAE^μ results substantially confirm the MAE^M results, with the difference that our FS functions manage to outperform the full feature set ($\xi = 1$) even more frequently than with MAE^M.

5 Conclusions

In this paper we have proposed four novel feature selection techniques for ordinal classification, and we have tested them against three baseline techniques from the literature on two datasets of product review data; one of these datasets is the largest dataset of product review data ever tested for ordinal classification purposes, and is being presented here for the first time.

The experiments, that we have carried out with thorough parameter optimization and for an extensive range of reduction levels, have clearly shown that all our four techniques are clearly superior to all three baselines, on both datasets. The experiments on the Amazon-83713 dataset (by far the larger of the two) seem to indicate that all techniques deliver a fairly stable performance across the range [0.05,1.0] of reduction levels, and that performance tends to peak close to the 0.10 level; this indicates that it is viable to downsize the feature set by one order of magnitude while at the same time retaining, and sometimes even moderately improving upon, the effectiveness delivered by the full feature set.

10. A spreadsheet with detailed figures for all the 28×100 experiments conducted, along with large-size versions of all the plots, for both MAE^M and MAE^μ, can be downloaded at http://patty.isti.cnr.it/~baccianella/FS4OR/. Note that the results obtained with the techniques involving a round-robin step with the ϵ-SVR package are different from the analogous results presented in the earlier version of this paper [1] because of a bug in our software that we detected after [1] had gone to print. The bug had the consequence that fewer features than actually declared were being selected by the techniques.

Acknowledgments

We thank Nitin Jindal and Bing Liu for kindly sharing with us their "Amazon" dataset, and Salvatore Ruggieri for an important pointer to the literature.

References

[1] S. Baccianella, A. Esuli, and F. Sebastiani, "Feature selection for ordinal regression," in *Proceedings of the 25th ACM Symposium on Applied Computing (SAC'10)*, Sierre, CH, 2010, pp. 1748–1754.

[2] G. Forman, "An extensive empirical study of feature selection metrics for text classification," *Journal of Machine Learning Research*, vol. 3, pp. 1289–1305, 2003.

[3] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *Proceedings of the 14th International Conference on Machine Learning (ICML'97)*, Nashville, US, 1997, pp. 412–420.

[4] M. Dash, K. Choi, P. Scheuermann, and H. Liu, "Feature selection for clustering - a filter solution," in *Proceedings of the 2nd IEEE International Conference on Data Mining (ICDM'02)*, Maebashi City, JP, 2002, pp. 115–122.

[5] G. H. John, R. Kohavi, and K. Pfleger, "Irrelevant features and the subset selection problem," in *Proceedings of the 11th International Conference on Machine Learning (ICML'94)*, New Brunswick, US, 1994, pp. 121–129.

[6] A. Miller, *Subset selection in regression*, 2nd ed. London, UK: Chapman and Hall, 2002.

[7] S. Baccianella, A. Esuli, and F. Sebastiani, "Multi-facet rating of product reviews," in *Proceedings of the 31st European Conference on Information Retrieval (ECIR'09)*, Toulouse, FR, 2009, pp. 461–472.

[8] R. Mukras, N. Wiratunga, R. Lothian, S. Chakraborti, and D. Harper, "Information gain feature selection for ordinal text classification using probability re-distribution," in *Proceedings of the IJCAI'07 Workshop on Text Mining and Link Analysis*, Hyderabad, IN, 2007.

[9] G. Forman, "A pitfall and solution in multi-class feature selection for text classification," in *Proceedings of the 21st International Conference on Machine Learning (ICML'04)*, Banff, CA, 2004, pp. 38–45.

[10] N. Jindal and B. Liu, "Review spam detection," in *Proceedings of the 16th International Conference on the World Wide Web (WWW'07)*, Banff, CA, 2007, pp. 1189–1190.

[11] S. Baccianella, A. Esuli, and F. Sebastiani, "Evaluation measures for ordinal text classification," in *Proceedings of the 9th IEEE International Conference on Intelligent Systems Design and Applications (ISDA'09)*, Pisa, IT, 2009, pp. 283–287.

[12] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," in *Proceedings of the 9th Conference on Neural Information Processing Systems (NIPS'96)*, Denver, US, 1997, pp. 155–161.

[13] W. Chu and S. S. Keerthi, "Support vector ordinal regression," *Neural Computation*, vol. 19, no. 3, pp. 145–152, 2007.

[14] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in kernel methods: Support vector learning*, B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge, US: MIT Press, 1999, pp. 185–208.