Feature Selection for Ordinal Regression

Stefano Baccianella, Andrea Esuli and Fabrizio Sebastiani
Istituto di Scienza e Tecnologie dell'Informazione
Consiglio Nazionale delle Ricerche
Via Giuseppe Moruzzi, 1 – 56124 Pisa, Italy
firstname.lastname@isti.cnr.it

ABSTRACT

Ordinal regression (also known as *ordinal classification*) 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 increasing attention from the sentiment analysis and 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 regression. In this paper we present four novel feature selection metrics that we have specifically devised for ordinal regression, and test them on two datasets of product review data.

Keywords

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 [4, 11], and to a lesser extent for text clustering, they have not for a related and important text learning task, namely, ordinal regression for text. *Ordinal regression* (OR – also known as *ordinal classification*) 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.

OR 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 OR. Here of course we are only interested in "filter" approaches to FS, i.e., 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 [7] (x may be a predetermined number or may,

SAC'10 March 22-26, 2010, Sierre, Switzerland.

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

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 regression, and test them on two datasets of product review data.

The paper is organized as follows. In Section 2 we discuss related work, and in Section 3 we present our four FS algorithms for OR. 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 [2, 9] that address feature selection for ordinal regression.

Mukras et al. [9] propose an algorithm, called *probability redistribution procedure* (PRP), that takes as input the distribution of the feature across the ranks (as deriving from the distribution across the ranks of the training examples containing it) and modifies it, according to the notion that, when a feature t_k occurs in (a document belonging to) a rank r_j, it is "taken to also 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. [2] 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 across the ranks, and retaining only the t 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 ordered scale 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 idea (originally presented in [5] for binary text classification) 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, in $RR(Var)$ one provisionally "assigns" each feature t_k to the rank closest to its average rank value, 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.

3. FEATURE SELECTION FOR ORDINAL CLASSIFICATION

Let us fix some notation. As mentioned in Section 1, our task is to learn (from a training set Tr) and evaluate (on a test set Te) a target function $\Phi : D \rightarrow R$ which classifies each document $d_i \in D$ into exactly one of an ordered sequence $R = \langle r_1 \prec \ldots \prec r_n \rangle$ of ranks by means of a classifier $\hat{\Phi}$. Our feature selection methods will consist of scoring each feature t_k 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 selecting the features based on their $Score$ value.

3.1 The Var*IDF method

$Var * IDF$ is a variant of the Var method described in [2].

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 lends statistical robustness to the estimated value of their variance.

We formalize this by defining

$$Score(t_k) = -Var(t_k) * (IDF(t_k))^a \tag{1}$$

where 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 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 $\epsilon = 0.1$ 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, for $\frac{x}{n}$ rounds, in picking their

[2] Interestingly, the literature on FS for metric regression seems to have mostly, if not only, investigated "wrapper" approaches [8].

favourite features from the top-most elements of their rank-specific orderings. In this way, for each rank r_j the final set of selected features contains 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.

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* (also known as *mutual information*), a function routinely employed in feature selection for binary classification (see e.g. [11]), to ordinal regression[3].

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 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.

$RR(IGOR)$ 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 $R_j = \{r_1, \ldots, r_j\}$ or to one of the ranks $\overline{R}_j = \{r_{j+1}, \ldots, r_n\}$, for $j = 1, \ldots, (n-1)$. We call R_j and $\overline{R}_j$ a *rankset prefix* and a *rankset suffix*, respectively. 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) order, for each of the $(n-1)$ binary classifiers $\ddot{\Phi}_j$ above, the features in terms of their value of the $IG(t_k, c_j)$ function, with the highest-scoring ones at the top of the ordering; and (ii) enforce a round robin policy in which the $(n-1)$ classifiers $\ddot{\Phi}_j$ above take turns, for $\frac{x}{n-1}$ 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 $\frac{x}{n}$ features that are best at discriminating the rankset prefix 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 in 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 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; e.g., 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

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

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.

Let us define the *rank* $R(t_k)$ *associated to a feature* t_k as

$$R(t_k) = \arg \min_{r_j \in R} NC_{Tr}(t_k, r_j)$$

i.e., as the rank that is least negatively correlated with the feature. It would now be tempting to define $Score(t_k)$ as $-NC_{Tr}(t_k, R(t_k))$, i.e., as the opposite of the negative correlation between t_k and the rank $R(t_k)$ associated to it. This would seem reasonable since, given that $R(t_k)$ is the rank that t_k best identifies, $-NC_{Tr}(t_k, R(t_k))$ measures how well it identifies it. 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 as

$$Score(t_k) = -NC_{Tr}(t_k, R(t_k)) * (IDF(t_k))^a \qquad (2)$$

where the a parameter serves 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, among the features such that $R(t_k) = r_j$, the $\frac{x}{n}$ features with the best $Score$.

Finally, note that for ranksets in which the distances between consecutive ranks are not always equal (this may be the case for a rankset consisting 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), $RR(NC * IDF)$ is the only technique (among those discussed in this paper, including those from the literature) that allows bringing to bear these distances in the feature selection phase: one only needs to plug these distances into the E measure used to define negative correlation.

[3] 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.

Dataset	$\|Tr\|$	$\|Te\|$	1	2	3	4	5
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".

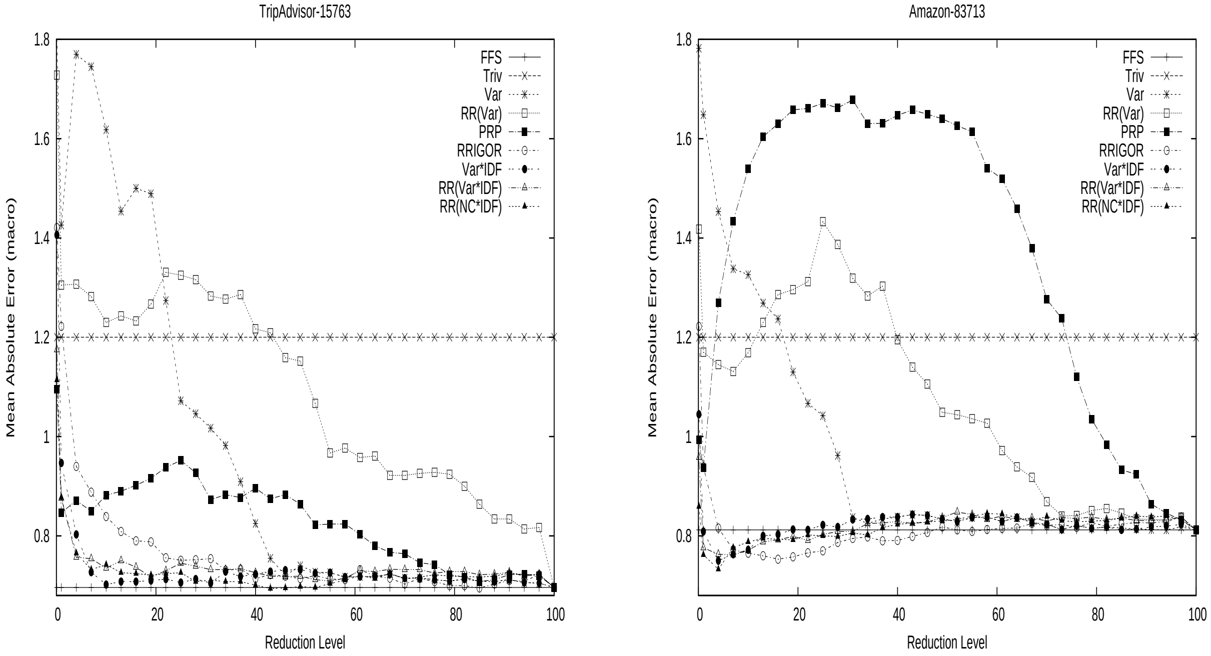

Figure 1: Results obtained with three baseline + four novel feature selection techniques on the TripAdvisor-15763 dataset (left) and on the Amazon-83713 dataset (right). Results are evaluated with MAE^M; lower values are better. "FFS" refers to the full feature set (i.e., no feature selection), while "Triv" refers to uniform assignment to the trivial class.

4. EXPERIMENTS

4.1 Experimental setting

4.1.1 *The datasets*

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

The first is the TripAdvisor-15763 dataset built by Baccianella et al. [2], consisting of 15,763 hotel reviews from the TripAdvisor Web site[4]. We use the same split between training and test documents of [2], resulting in 10,508 documents used for training and 5,255 used for test; the training set contains 36,670 unique words. From the 10,508 training documents we have randomly picked 3,941 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 [6], 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 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 rating product reviews.

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

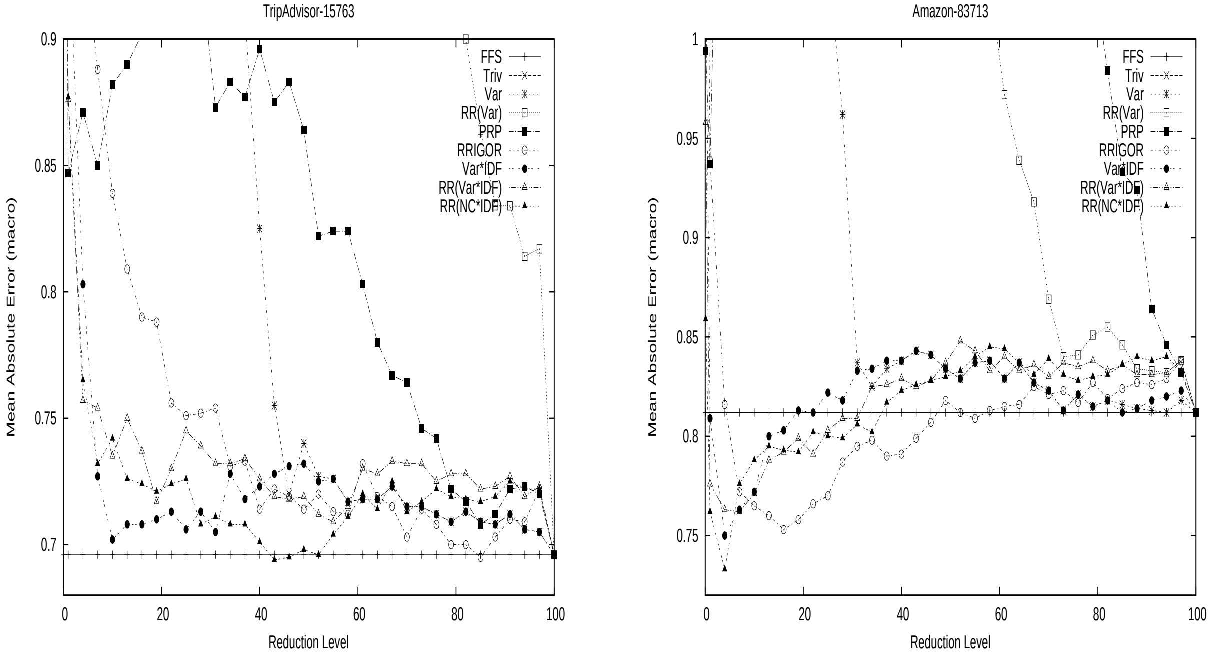

Figure 2: "Close-up" on the best results of Figure 1.

4.1.2 Evaluation measures

As our evaluation measure we use the *macroaveraged mean absolute error* (MAE^M) measure proposed in [1], and defined as

$$MAE^M(\hat{\Phi}, Te) = \frac{1}{n} \sum_{j=1}^{n} \frac{1}{|Te_j|} \sum_{d_i \in Te_j} |\hat{\Phi}(d_i) - \Phi(d_i)| \quad (3)$$

where Te_j denotes the set of test documents whose true rank is r_j and the "M" superscript indicates "macroaveraging". As argued in [1], the advantage of MAE^M over "standard" mean absolute error (defined as

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

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 datasets) while coinciding with MAE^μ on perfectly balanced datasets (i.e., datasets with the same number of test documents for each rank).

4.1.3 Baselines

For all our experiments we have used three different baseline methods: the first is the PRP method of [9], while the second and third are the Var and $RR(Var)$ methods of [2] (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 on the training set Tr across all trivial classifiers, i.e., $\tilde{r} = \arg\min_{r_j \in R} E(\tilde{\Phi}_j, Tr)$, where E is the chosen error measure. As argued in [1], the trivial class(es) for MAE^M are always the middle classes, i.e. $r_{\lfloor \frac{n+1}{2} \rfloor}$ and $r_{\lceil \frac{n+1}{2} \rceil}$, which coincide with the 3 stars class in both the TripAdvisor-15763 and Amazon-83713 datasets (assignment to the trivial class yields $MAE^M = 1.200$ for both datasets).

4.1.4 Experimental protocol

As a learning device for ordinal regression we have use ϵ-*support vector regression* (ϵ-SVR) [10] as implemented in the freely available LibSvm library [3]. As a vectorial representation, after stop word removal (and no stemming) we use standard bag-of words with cosine-normalized $tfidf$ weighting. 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 (individually for each method) optimized the a parameter on the validation set 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 $\xi = 0.10$; further experiments on the validation set showed that in all cases the optimized parameter was optimal anyway for any $\xi \in [0.001, 0.20]$.

For $RR(NC * IDF)$, the E evaluation 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 MAE^M.

4.2 Results

The results of our experiments are displayed in Figures 1 and 2, in which the effectiveness of each feature selection policy is plotted as a function of the tested reduction level[5]. The two horizontal lines indicate the effectiveness obtained (a) by using the full feature set (i.e., no feature selection), or (b) by assigning all test documents to the trivial class.

In the discussion that follows we refer by default to the MAE^M results on the Amazon-83713 dataset, since the test set of the Amazon-83713 dataset is more than 16 times larger than that of the TripAdvisor-15763 dataset, and we thus considers the results obtained on it somehow more reliable.

The first observation that comes natural by observing Figure 1 is that the three baselines are dramatically inferior to the four novel techniques proposed in this paper. 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 the trivial baseline in the range $\xi \in [0.05, 0.70]$ (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. 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 Figure 2, which presents the same results of Figure 1 in close-up view, zooming-in on the best performing techniques.

A second observation (that comes especially evident by looking at Figure 1) 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 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 not by a wide 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 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 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 text classification [11], showed that the best feature selection techniques could allow exactly that (i.e., an improvement in effectiveness of about 10% at $\xi = 0.10$, which was the level at which the best performance was obtained).

It is not easy instead to decide which of the four techniques is best. On the Amazon-83713 dataset, $RR(IGOR)$ is clearly the best when $\xi \in [0.10, 0.70]$, while when $\xi \in [0.001, 0.10]$ the best performer is $RR(NC * IDF)$. This latter even marginally outperforms the full feature set when the size of the feature set is reduced by two orders of magnitude, and only marginally underperforms it when the reduction is by three orders of magnitude; we think this is a striking performance. However, the situation is different on the TripAdvisor-15763 dataset, where $RR(NC * IDF)$ is the best performer for $\xi \in [0.3, 0.6]$ but is beaten by $Var * IDF$ when $\xi \in [0.1, 0.3]$, a range in which (in stark contrast to Amazon-83713) $RR(IGOR)$ clearly underperforms the other three.

In order to get a clearer sense of the relative merits of these four techniques, in Table 2 we report the performance (for both datasets) of all the techniques discussed, averaged across the 50 reduction levels $\xi \in \{0.001, 0.01, 0.02, \ldots, 0.49, 0.50\}$ (we consider that the reduction levels $\xi \in \{0.51, \ldots, 0.99\}$ are scarcely interesting, since halving (or less) the dimensionality of one's feature space seems hardly worth the trouble of engaging in feature selection). The results reported show that $RR(IGOR)$ is the best performer (followed by $RR(NC * IDF)$), on both datasets, for our evaluation measure of choice (MAE^M).

5. CONCLUSIONS

In this paper we have proposed four novel feature selection techniques for ordinal regression, 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 regression 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.

In the future we plan to carry out further experiments in

[5] A spreadsheet with detailed figures on all the 7 × 100 experiments conducted can be downloaded at `http://patty.isti.cnr.it/~baccianella/SAC10.xls`.

	Baselines					Our techniques			
	Full feature set	Trivial	Var	$RR(Var)$	PRP	$Var * IDF$	$RR(Var * IDF)$	$RR(IGOR)$	$RR(NC * IDF)$
TripAdvisor-15763	0.696	1.200	1.200	1.255	0.739	0.726	0.739	**0.683**	0.723
Amazon-83713	0.812	1.200	1.066	1.225	1.565	0.813	0.802	**0.793**	0.798

Table 2: Performance of different feature selection functions as averaged across the reduction levels $\xi \in \{0.001, 0.01, 0.02, \ldots, 0.49, 0.50\}$. The best performer is indicated in boldface.

order to more clearly see which among the four proposed techniques is “the winner”.

Acknowledgements

We thank Nitin Jindal and Bing Liu for kindly sharing with us their “Amazon” dataset, Xiubo Geng for clarifying the details of his experiments, and Salvatore Ruggieri for an important pointer to the literature.

6. REFERENCES

[1] 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.

[2] 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)*, pages 461–472, Toulouse, FR, 2009.

[3] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/\~{}cjlin/libsvm`.

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

[5] 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)*, pages 38–45, Banff, CA, 2004.

[6] N. Jindal and B. Liu. Review spam detection. In *Proceedings of the 16th International Conference on the World Wide Web (WWW’07)*, pages 1189–1190, Banff, CA, 2007.

[7] 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)*, pages 121–129, New Brunswick, US, 1994.

[8] A. Miller. *Subset selection in regression.* Chapman and Hall, London, UK, second edition, 2002.

[9] 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.

[10] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett. New support vector algorithms. *Neural Computation*, 12(5):1207—1245, 2000.

[11] 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)*, pages 412–420, Nashville, US, 1997.