Selecting Features for Ordinal Text Classification*

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

We present four new feature selection methods for ordinal regression and test them against four different baselines on two large datasets of product reviews.

1. INTRODUCTION

In (text) classification, *feature selection* (FS) 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. While feature selection has been extensively investigated for standard classification, it has not for a related and important learning task, namely, *ordinal classification* (OC – aka *ordinal regression*). OC consists in estimating a *target function* $\Phi : D \rightarrow R$ mapping each object $d_i \in D$ into exactly one of an ordered sequence $R = \langle r_1 \prec \ldots \prec r_n \rangle$ of *ranks*, by means of a function $\hat{\Phi}$ called the *classifier*.

We here address the problem of feature selection for OC. We use a "filter" approach, in which a scoring function $Score$ is applied to each feature $t_k \in T$ in order to measure the predicted utility of t_k for the classification task (the higher the value of $Score$, the higher the predicted utility), after which only the $|S|$ top-scoring features are retained. We have designed four novel feature scoring functions for OC, and tested them on two datasets of product review data, using as baselines the only three feature scoring functions for OC previously proposed in the literature, i.e., the *probability redistribution procedure* (PRP) function proposed in [4], and the *minimum variance* (Var) and *round robin on minimum variance* ($RR(Var)$) functions proposed in [2].

2. FEATURE SELECTION FOR OC

Our first method, $Var{*}IDF$, is a variant of the Var method described in [2], and is meant to prevent features occurring very infrequently, such as *hapax legomena*, to be top-ranked, which would obviously be undesirable. It is defined as

$$Score(t_k) = -Var(t_k) * (IDF(t_k))^a \quad (1)$$

where IDF is the standard inverse document frequency 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. Given that $Var(t_k) = 0$ implies that $Score(t_k) = 0$, we smooth $Var(t_k)$ by adding to it a small value $\epsilon = 0.1$ prior to multiplying it by $IDF(t_k)$.

Our second method, $RR(Var{*}IDF)$, is a variant of $Var{*}IDF$ meant to prevent it from exclusively catering for a certain rank and disregarding the others. It consists in (i) provisionally "assigning" each feature t_k to the rank closest to the mean of its distribution across the ranks; (ii) sorting, for each rank r_j, the features provisionally assigned to r_j in decreasing order of their value of the $Score$ function of Equation 1; and (iii) enforcing a "round robin" policy in which the n ranks take turns, for $\frac{x}{n}$ rounds, in picking their favourite features from the top-most elements of their rank-specific orderings.

Our third method, $RR(IGOR)$, is based on the idea of viewing ordinal classification on $R = \langle r_1 \prec \ldots \prec r_n \rangle$ as the generation of $(n-1)$ binary classifiers $\ddot{\Phi}_j$, each of which is in charge of separating $R_j = \{r_1, \ldots, r_j\}$ from $\overline{R}_j = \{r_{j+1}, \ldots, r_n\}$, for $j = 1, \ldots, (n-1)$. For each feature t_k we thus compute $(n-1)$ different values of $IG(t_k, c_j)$ (the classic *information gain* function of binary classification), 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 $RR(Var{*}IDF)$, we (i) sort, for each of the $(n-1)$ binary classifiers $\ddot{\Phi}_j$, the features in decreasing order of $IG(t_k, c_j)$ value, and (ii) enforce a round-robin policy in which the $(n-1)$ classifiers $\ddot{\Phi}_j$ take turns, for $\frac{x}{n-1}$ rounds, in picking their favourite features from the top-most elements of their classifier-specific orderings.

Our fourth method, $RR(NC * IDF)$, directly optimizes the chosen error measure E. Let us define the *negative correlation* of t_k with r_j in the training set Tr as

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

where $\tilde{\Phi}_j$ is the "trivial" classifier that assigns all $d_i \in D$ to r_j and $E(\hat{\Phi}, d_i)$ represents the error that $\hat{\Phi}$ makes in classifying d_i. Let the *rank* $R(t_k)$ *associated to a feature* t_k be

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

*This is a short summary of a paper forthcoming in the Proceedings of the 25th ACM Symposium on Applied Computing (SAC'10), Sierre, CH, 2010.

Appears in the Proceedings of the 1st Italian Information Retrieval Workshop (IIR'10), January 27–28, 2010, Padova, Italy.
http://ims.dei.unipd.it/websites/iir10/index.html

Dataset	$\|Tr\|$	$\|Va\|$	$\|Te\|$	1	2	3	4	5
TripAdvisor-15763	10,508	3,941	5,255	3.9%	7.2%	9.4%	34.5%	45.0%
Amazon-83713	20,000	4,000	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".

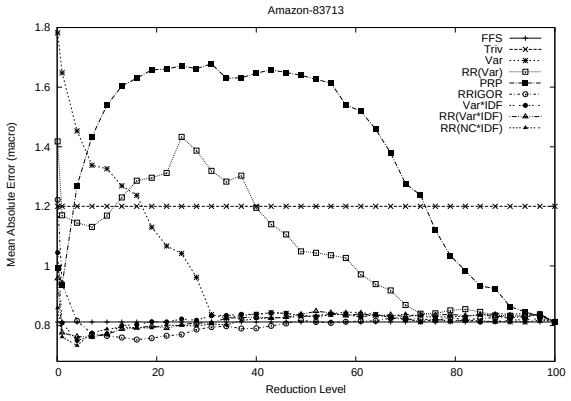

Figure 1: Results obtained on the Amazon-83713 dataset. Results are evaluated with MAE^M; lower values are better. "FFS" refers to the full feature set (i.e., no feature selection), while "Triv" refers to uniform assignment to the 3 Stars class.

We can now define

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

where the a parameter serves the same purpose as in Equation 1. Similarly to the 2nd and 3rd methods, to select the best x features we apply a round-robin policy in which each r_j is allowed to pick, among the features such that $R(t_k) = r_j$, the $\frac{x}{n}$ features with the best $Score$.

More details on these methods can be found in [3].

3. EXPERIMENTS

We have tested the proposed measures on two different datasets, TripAdvisor-15763 and Amazon-83713, whose characteristics are summarized in Table 1. Both datasets consist of product reviews scored on a scale of one to five "stars", and (as shown by Table 1) are highly imbalanced. See [3] for more details.

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

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

where Te_j denotes the set of test documents whose true rank is r_j and the "M" superscript indicates "macroaveraging".

We compare our methods with the three baselines mentioned at the end of Section 1 and with the "trivial baseline" that consists in scoring all test documents as 3 stars.

As a learning device we use *ϵ-support vector regression* (ϵ-SVR) [5] as implemented in the freely available LibSvm library. 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 a validation set Va extracted from the training set Tr, and then retrained the optimized classifier on the full training set Tr. For $RR(NC*IDF)$, $E(\hat{\Phi}, d_i)$ was taken to be $|\hat{\Phi}(d_i) - \Phi(d_i)|$.

The results of our tests are displayed in Figure 1, in which effectiveness is plotted as a function of the tested reduction level. For reasons of space only the Amazon-83713 results are displayed (see [3] for the TripAdvisor-15763 results, which are anyway fairly similar). The experiments show that

- the four novel techniques proposed here are dramatically superior to the four baselines;
- our four 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.
- for $\xi \in [0.01, 0.3]$ the proposed techniques even outperform the full feature set. This 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.

4. REFERENCES

[1] S. Baccianella, A. Esuli, and F. Sebastiani. Evaluation measures for ordinal text classification. In *Proceedings of the 9th IEEE Int'l Conference on Intelligent Systems Design and Applications (ISDA'09)*, pages 283–287, 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] 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.

[4] 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.

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