

# Learning the Preferences of News Readers with SVM and Lasso Ranking

Elena Hensinger, Ilias Flaounas, Nello Cristianini

► **To cite this version:**

Elena Hensinger, Ilias Flaounas, Nello Cristianini. Learning the Preferences of News Readers with SVM and Lasso Ranking. Harris Papadopoulos; Andreas S. Andreou; Max Bramer. 6th IFIP WG 12.5 International Conference on Artificial Intelligence Applications and Innovations (AIAI), Oct 2010, Larnaca, Cyprus. Springer, IFIP Advances in Information and Communication Technology, AICT-339, pp.179-186, 2010, Artificial Intelligence Applications and Innovations. <10.1007/978-3-642-16239-8\_25>. <hal-01060666>

**HAL Id: hal-01060666**

**<https://hal.inria.fr/hal-01060666>**

Submitted on 17 Nov 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# Learning the Preferences of News Readers with SVM and Lasso Ranking

Elena Hensinger, Ilias Flaounas, and Nello Cristianini

Intelligent Systems Laboratory, University of Bristol, UK  
{elena.hensinger, ilias.flounas, nello.cristianini}@bristol.ac.uk

**Abstract.** We attack the task of predicting which news-stories are more appealing to a given audience by comparing ‘most popular stories’, gathered from various online news outlets, over a period of seven months, with stories that did not become popular despite appearing on the same page at the same time. We cast this as a learning-to-rank task, and train two different learning algorithms to reproduce the preferences of the readers, within each of the outlets. The first method is based on Support Vector Machines, the second on the Lasso. By just using words as features, SVM ranking can reach significant accuracy in correctly predicting the preference of readers for a given pair of articles. Furthermore, by exploiting the sparsity of the solutions found by the Lasso, we can also generate lists of keywords that are expected to trigger the attention of the outlets’ readers.

**Key words:** Learning to Rank, News Content Analysis, User Preferences, Support Vector Machines, Lasso

## 1 Introduction

We are interested in understanding and predicting which news articles are most intriguing for readers of a given news outlet. This problem is of importance to both media analysts and news editors, as well as to engineers designing recommendation systems.

“What makes stories newsworthy?” has been the classical question in media studies [1]. A related question is “What makes stories popular?”, and it can be addressed today due to the online availability of news media and the capability to capture preferences of readers by their clicks on news items. There are many possible answers to this ‘Popularity question’: it does not only depend on the textual content of the article, but it is also influenced by its position on the web page [2], on the images or videos associated with it and on its timing and according novelty [3].

The study of ranking user preferences has been recently approached by researchers in data mining, text processing and web analysis. Examples and applications in the area include the study of popularity of micro-blogging [3], advertisement selection [4] and search engine log analysis [5].

In our study, we focus on news media data with the goal to learn and predict preferences of news readers. We work on the content of news articles that readers will use in order to make their reading choices: the title and short description – the text snippet that is presented to readers before they click on a given link – from RSS (Real Simple Syndication) feeds of 10 news outlets with online presence, including broadcast media such as ‘CBS’ and newspapers such as ‘LA Times’. We used two feeds per outlet: a) the ‘Main’ feed which contains the main stories of the day, as selected by the outlets’ editors and b) the ‘Most popular’ feed formed by the readers’ click choices. Can this information be sufficient to predict the interests and preferences of readers? To our knowledge, this question has not been addressed in literature.

We model the task as a problem of ordinal learning, or ‘Learning to Rank’: given a set of ordered data pairs, we learn a utility function that can be used to score every article, in a way that the scores can be used to rank them appropriately. The methods we use to infer linear utility functions are based on quadratic programming: Ranking SVMs (Support Vector Machines), as introduced by Joachims [5], and Lasso [6] that we applied for the ranking task. The motivation for the latter choice is to obtain sparse solutions, and hence extract the keywords that are most likely to predict article popularity and to trigger readers’ attention.

In Sect. 2 we describe the two algorithms and present their relative merits in terms of generalisation performance and feature extraction ability. While Ranking SVM and Lasso originally solve different tasks, we show how Lasso can be applied to learn pairwise ranking.

In Sect. 3 we discuss the application of the two methods to news media datasets and report our results. For most outlets, the SVM-based ranking method can make predictions that are significantly stronger than random guessing. Under the posed experimental framework the Lasso-based ranking method performs less efficiently but still making significantly valid predictions and it has its actual strength in getting sparse solutions. Therefore we use the utility function inferred by the SVM to rank articles and the Lasso utility function to detect the strongest keywords that make articles become popular.

## 2 Task and Techniques

In this section we first present how the task of learning pairwise preference relationships can be transformed into a classification problem, secondly the ranking SVM and Lasso methods we used to solve it, and last their empirical comparison on synthetic data.

### 2.1 Pairwise Ranking as Classification

We aim to learn the ranking of items  $x \in \mathbb{R}^n$ , based on partial information about the preference relationship between pairs  $(x_i, x_j)$ . We denote that item  $x_i$

is preferred to  $x_j$  by  $x_i \succ x_j$ . By defining a linear utility function  $u : \mathbb{R}^n \rightarrow \mathbb{R}$  of the form  $\langle w, x \rangle$ , we can express this relationship as:

$$x_i \succ x_j \iff u(x_i) > u(x_j) \iff \langle w, (x_i - x_j) \rangle > 0 \quad (1)$$

where  $u(x_i)$  is the utility value of  $x_i$ . The task of learning the relationship between items  $x_i$  and  $x_j$  can now be expressed as a binary classification problem on the vector of their *difference*, which we will denote as  $s_k \in S$ , with  $S$  being the set of pair differences. We assign class label  $y_k = +1$ , if  $\langle w, s_k \rangle \geq 0$ , and  $y_k = -1$  otherwise.

## 2.2 Ranking SVM

SVMs is a well studied method for binary classification [7, 8]. The SVM formulation has been extended to learn pairwise preference relationships based on difference vectors of form  $s_k$  through the Ranking SVM [5]. Slack variables  $\xi_k$  allow to deal with non-linearly separable data, and the entire task for  $l$  training items of  $S$  is expressed as a quadratic optimisation problem:

$$\begin{aligned} & \text{minimise}_{\xi, w} && \langle w, w \rangle + C \sum_{k=1}^l \xi_k && (2) \\ & \text{subject to} && y_k \langle w, s_k \rangle \geq 1 - \xi_k, \\ & && \xi_k \geq 0 \quad \forall k = 1, \dots, l \end{aligned}$$

The solution is a weight vector  $w$  that can not only compute the preference relationship between two items  $x_i$  and  $x_j$ , but can also give the rank, *i.e.* the utility value, of item  $x_i$  since  $u(x_i) = \langle w, x_i \rangle$ .

## 2.3 Lasso for Ranking

The second method we used is a special form of linear regression, the Lasso (Least Absolute Shrinkage and Selection Operator)[6]. In our setting rather than searching for a fit to real values we try to classify the pairwise differences  $s_k$  to their labels  $y_k$ . In order to produce a sparse solution, the regression problem is constrained on the  $L_1$ -norm, denoted as  $\|\cdot\|_1$ , of the computed weight vector  $w$  through a parameter  $t \in \mathbb{R}$ . The according quadratic optimisation problem is:

$$\begin{aligned} & \text{minimise}_w && \sum_{k=1}^l (\langle w, s_k \rangle - y_k)^2 && (3) \\ & \text{subject to} && \|w\|_1 \leq t \end{aligned}$$

The solution is a weight vector  $w$  which, as in the case of SVMs, is used for calculating the utility value of item  $x_i$  as  $u(x_i) = \langle w, x_i \rangle$ .

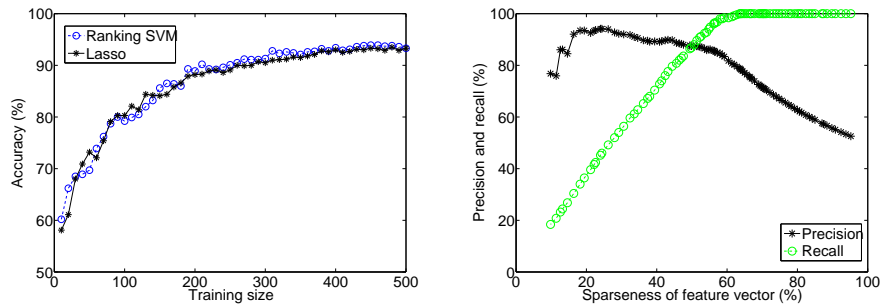
## 2.4 Empirical comparison of Ranking SVM and Lasso

We compared Ranking SVM and Lasso in an experiment on small scale synthetic data to explore their potential on: a) correctly predicting class labels for the

pairwise ranking task and b) handling sparse data and especially their ability to correctly identify relevant features, that is excluding features with zero value.

We synthetically created data items of 50 features with uniformly distributed values in  $(0, 1)$ . A weight value from the real interval  $[0, 10]$  was randomly assigned to each feature, and 50% of features were randomly set to zero to simulate sparse data. We created difference vector pairs  $s_k$  and assigned class label  $+1$  or  $-1$ , depending on the utility values of  $x_i$  and  $x_j$ . In all our computations we trained on 100 different data sets, ranging from 10 to 500 pairs of items, and we tested on 100 item pairs, averaging over 10 different data initialisations. We set parameters  $C$  in Ranking SVM and  $t$  in Lasso to their optimal values, as found for the described experimental setup.

To evaluate the two methods on preference prediction performance we used accuracy, defined as the percentage of correctly classified pairs. The resulting learning curves are illustrated in Fig. 1A. Both approaches improve their performance with increasing amount of training data, reaching an asymptotic bound on their performance. While Ranking SVM scores better in most cases, ‘Lasso Ranking’ can keep up successfully in this setting of predicting pairwise preferences.



**Fig. 1.** A. Learning curves for Ranking SVM and Lasso. B. Lasso’s ability to identify zero-valued features.

To evaluate performance on identifying zero valued features ( $ZF$ ) we used precision and recall measures, defined as:

$$\text{precision} = \# \text{ of correctly identified } ZF / \# \text{ total retrieved } ZF \quad (4)$$

$$\text{recall} = \# \text{ of correctly identified } ZF / \# \text{ total true } ZF \text{ in features} \quad (5)$$

Figure 1B presents precision and recall on detecting zero features using Lasso as a function of different levels of sparseness of the solution  $w$  for the 500 training pairs case. In our experimental setup, Ranking SVM produced dense solutions, even with the relaxation of counting features with absolute values less than 0.0005 as zero-valued.

Based on these results, we have an indication that Lasso can perform comparably with Ranking SVM in terms of prediction accuracy on pairwise difference

data. Furthermore, it can identify relevant features in data, allowing to retrieve a sparse and meaningful solution.

### 3 Experimental Results on News-Outlets Data

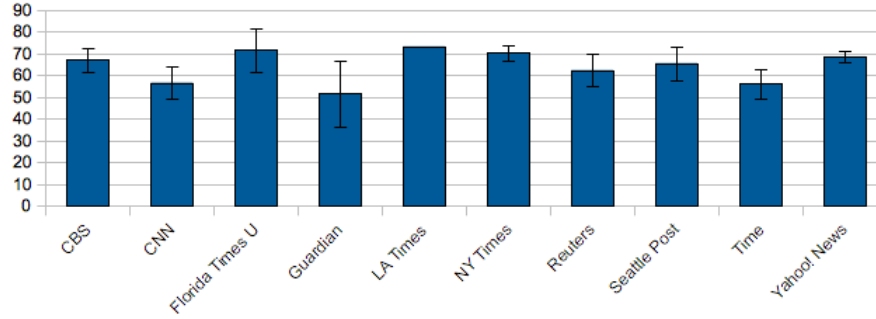
We tested Ranking SVM and Lasso on real world data from 10 news-media outlets monitored from June 1st to December 31st, 2009. Since there are no publicly available datasets that provide annotated articles to use for learning readers' preferences we created our own system to gather this information from outlets that offer their content online [9].

Our system processes the content of media outlets that offer their content as RSS feeds, an easily machine readable format. Feeds contain for each article only limited information: a title, a summary and publication date. Most outlets serve multiple different RSS feeds of which we are interested in two specific ones, namely the 'Main' feed, and the 'Most Popular Stories' feed. The first one contains articles that comprise the leading stories of the outlet as chosen by their editors and roughly correspond to the front page of a newspaper. The second contains articles that readers found most interesting – these articles are not necessarily present in the 'Main' feed. We define feeds as 'Most Popular' if they carry the 'Most clicked' articles. For the outlets which miss such feeds, we adopted the 'Most emailed stories' or the 'Most discussed stories'. The article titles and summaries were gathered on a daily basis and processed with typical text mining methods such as stemming [10], stop words removal, indexing and TF-IDF (bag-of-words) representation of articles [11].

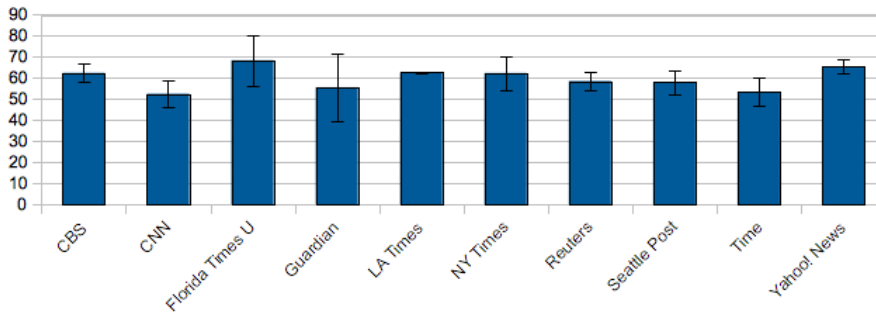
For each day of the period of study and for each outlet separately, we created pairs of articles from all combinations of articles that a) appeared in both 'Main' and 'Most Popular' feeds, and b) articles that appeared only in 'Main' feed and not in the 'Most Popular' one. The pairwise preference relationship is given by the fact that some articles of the 'Main' feed were preferred by the outlet's readers, while others were not. In total we formed 405,487 pairs – on average 5942 pairs per outlet per month. For 'Florida Times-Union' we managed to collect data for five months and for 'LA Times' for three months only.

We train both Ranking SVM and Lasso on one month per outlet and test on the following month's data. We have seven months of data resulting in six training/testing setups per outlet. For this set of experiments we used the  $SVM^{Rank}$  implementation of Ranking SVM [5]. We measured classification accuracy for three different values of  $C$  parameter, namely 1, 10 and 100, and we report the best results per outlet in Fig. 2.

In the case of Lasso, we are interested to discover the most interesting keywords that characterise popular articles, thus we adjusted the  $t$  parameter to achieve a sparse solution of roughly 100 words per outlet per month. Average month-to-month prediction results of readers' preferences are reported in Fig. 3. In both cases, we consider classifiers to have significant performance if they achieve an average accuracy over 50% for the 95% confidence intervals. Using SVMs, we get significant performance for seven outlets, reaching up to 73.14%



**Fig. 2.** Average month-to-month prediction accuracies per outlet using Ranking SVMs. Error bars are set to 95% confidence interval. Notice that for ‘Guardian’ we have limited amount of articles per day, and for ‘LA Times’ only three months of data resulting in a small confidence interval.



**Fig. 3.** Average month-to-month prediction accuracies per outlet using Lasso. Error bars are set to 95% confidence interval.

for ‘LA Times’. For ‘Guardian’ we had the least available data – only 482 pairs on average per month rather than thousands as in the rest of the outlets – and this explains the low performance. For ‘CNN’ and ‘Time’ we managed to get only marginally significant results, and we assume the diversity of their audience and topic coverage to be the reason. As expected, we achieved on average 5.78% less accuracy using Lasso for the seven outlets for which we had significant results. However, it is the same outlets that failed to give significant results for SVMs which also fail for Lasso.

We used the SVM classifiers that predict article popularity with significant performance to detect the most popular stories per month for each outlet. To identify the most promising keywords that can trigger the audience’s attention, we applied Lasso. As an example, we present the titles of the top-3 most popular articles per outlet and the top-10 keywords for December 2009 in Table 1. The stories we found reflect stories that were in the focus of media for that period of time. The stemmed keywords are similarly sensible and understandable. The

**Table 1.** Titles of most popular articles per outlet as ranked using Ranking SVMs and keywords per outlet as identified using Lasso, for December 2009.

Outlet	Titles of Top-3 Articles	Top-10 Keywords
CBS	Sources: Elin Done with Tiger - Tiger Woods Slapped with Ticket for Crash - Tiger Woods: I let my Family Down	defend, murphi, tiger, lake, mom, flight, bengal, man, goldman, unredact
Florida Times-Union	Pizza delivery woman killed on Westside - A family's search for justice, 15 years later - Rants & Raves: Napolitano unqualified	wade, middleburg, chest, send, charlton, petti, kill, augustin, shoot, levar
LA Times	Pacquiao to fight Mayweather in March - Bone marrow transplant 'gets rid of' sickle cell anemia - Disney toys get Pixar animation guru's touch	insul, immigr, disnei, futur, california, cash, studi, radiat, lo, selznick
NY Times	Poor Children Likelier to Get Antipsychotics - Surf's Up, Way Up, and Competitors Let Out a Big Mahalo - Grandma's Gifts Need Extra Reindeer	studi, visual, art, monei, gift, drug, journal, counterterror, water, korean
Reuters	Dubai says not responsible for Dubai World debt - Boeing Dreamliner touches down after first flight - Iran's Ahmadinejad mocks Obama, "TV series" nuke talks	secur, health, shot, landmark, ahmadinejad, exxon, oilfield, wood, jihad, test
Seattle Post	Hospital: Actress Brittany Murphy dies at age 32 - Actor Charlie Sheen arrested in Colorado - Charlie Sheen accused of using weapon in Aspen	patrolman, gainesvil, breakfast, girl, insur, salvat, hoax, ohio, sheen, utah
Yahoo! News	Yemen is growing front in al-Qaida battle - Report: US helped Yemen's strike against al-Qaida - AP source: Al-Qaida operative killed by US missile	care, storm, famili, white, sheen, slai, brittani, qaida, medicar, volcano

different findings between outlets reflect a) differences of editors' choices on which stories to cover, and b) differences of what readers of different media outlets preferred to read.

## 4 Conclusions

In this paper, we presented the feasibility of applying machine learning algorithms for the task of modelling readers' preferences on stories gathered from real world news-media outlets. We formulated the problem as a Learning to Rank task on preference-ordered pairs of items and applied two different methods, each with its own strengths. Ranking SVM allowed us to calculate a ranking of all articles and to retrieve the most popular articles per outlet. Lasso was applied to identify the keywords that relate to popularity of articles.

To rank articles we used only limited textual information, their titles and summaries, and we managed to achieve more than 70% accuracy in several cases. This is quite impressive given the limited data available and all the other factors that affect readers' choices, such as the placement of the article in the layout of the web-page, the presence of accompanying photos or videos, the potential sentiment content in the article, or the exact timestamp. Also the segmentation of the audience is very coarse: the readers of a given outlet are bundled into the same set, as we do not have more refined information about them (*e.g.* their location). That is we managed to quantify the contribution of content to the choices of readers, isolating this from other factors. Even more, in this research we focused only on the main page of each outlet – stories that the editors expect to be interesting and popular. We expect to be able to achieve a



higher performance if we don't restrict the data to the first page only, but use the entire content of an outlet as a negative set of non-popular stories.

Future plans include research on higher level features to describe articles, such as topic or sentiment to improve performance in learning, predicting, and understanding what affects readers' preferences.

**Acknowledgements.** The authors would like to thank Mark Schmidt for his implementation of Lasso LARS algorithm. We used 'QPC – Quadratic Programming in C' Matlab Interface in the empirical comparing of SVM and Lasso section. I. Flaounas is supported by the A. S. Onassis Public Benefit Foundation; N. Cristianini is supported by a Royal Society Wolfson Merit Award. This research was partly supported by the "University of Bristol Bridging the Gaps Cross-Disciplinary Feasibility Account" (EP/H024786/1). Group activities are supported by PASCAL2 Network of Excellence.

## References

1. Gans, H.J.: Deciding What's News: A Study of CBS Evening News, NBC Nightly News, Newsweek, and Time. 25th anniversary edition edn. Northwestern University Press (2004)
2. Wu, F., Huberman, B.A.: Popularity, novelty and attention. In: Proceedings 9th ACM Conference on Electronic Commerce (EC-2008). (2008) 240–245
3. Szabó, G., Huberman, B.A.: Predicting the popularity of online content. CoRR **abs/0811.0405** (2008)
4. Ghose, A., Yang, S.: An empirical analysis of search engine advertising: Sponsored search in electronic markets. *Management Science* **55**(10) (2009) 1605–1622
5. Joachims, T.: Optimizing search engines using clickthrough data. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), ACM (2002) 133–142
6. Tibshirani, R.: Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* **58**(1) (1996) 267–288
7. Boser, B., Guyon, I., Vapnik, V.: A training algorithm for optimal margin classifiers. In: Proceedings of the 5th Conference on Computational Learning Theory (COLT). (1992) 144–152
8. Cristianini, N., Shawe-Taylor, J.: An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press (2000)
9. Flaounas, I.N., Turchi, M., Bie, T.D., Cristianini, N.: Inference and validation of networks. In: European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD), Springer LNCS 5781 (2009) 344–358
10. Porter, M.: An algorithm for suffix stripping. *Program* **14** (1980) 130–137
11. Liu, B.: Web Data Mining, Exploring Hyperlinks, Contents, and Usage Data. Springer (2007)