RANK for spam detection
ECML - Discovery Challenge

Vadis Consulting – J.-F. Chevalier & P. Gramme

Vadis Consulting SA/NV., Allée de la recherche/ Researchdreeft 65
1070 Brussels (Anderlecht), Belgium
www.vadis.com
bjfc@vadis.com, bpgr@vadis.com

Abstract. This submission is aimed to benchmark of Vadis methodology in the context of spam detection. The work that has been done to provide these results can be separated in two different tasks: data preparation and modelisation .

Keywords: Very large scale problem, LARS, Variable recoding, Ridge regression, Features selection.

1 Introduction

Our approach can be summarized in four steps:

- Produce variables into a “single view”, containing one record per user ;
- Split the users into two segments, according to the type of content posted ;
- Bin variables and recode them according to the percentage of targets in each bin ;
- Apply LARS algorithm & backward cross-validation to perform linear regression on these recoded variables.

The first and the most time consuming task was to derive variables from the initial files. The un-homogeneity of the data led us to split it into two segments. Once we achieved this goal, we used our generic tool to build models. After analysis and fine tuning, we ended up with a model ready to be applied on the test set.

2 Data Preparation

The goal of data preparation is to build a number of variables describing each user. There is practically no limitation in the number of variables created since our modelisation tool RANK can cope with very large data set, with a very high number of columns.

Cleaning of text fields

The provided data included a number of fields consisting of text entered by the users: tags, description of web pages or articles, etc. Before computing information, we performed some cleaning of these fields:

- Put to lower case
- Remove special characters and count them

- Cut tags into words (using spaces, hyphens and punctuation as separators)
- Try to correct typos: each tag is compared to the 1000 most popular tags. If it “close” to any popular tag, it is replaced by that tag. Otherwise, it is left as is.
- Replace tag starting with a number by either “replace_of_year" or "replace_of_number”.

Measuring the information of text fields

The goal of this measure is to estimate the rarity of a document within a corpus. In this case, the document is the value of some text variable for some user or resource, and the corpus consists of all the values taken by this text variable for all users or resources.

The information of a text field is defined as the sum of the information of all its words. The information of a single word is the inverse of its log-frequency – i.e. divide the total number of words in the considered text field (across all users) by the total number of occurrences of the word, and take the logarithm of the quotient.

Variables describing tags

Several variables were produced describing the tags posted by a user. These variables concern the tag itself, not the resource pointed by the tag. They include among others

- The number of tags of a user which contain a given special character
- The total, average, minimum and maximum number of tags that the user posted per resource
- The total, average, minimum and maximum length of the tags posted by a user
- The total, average, minimum and maximum information (see above) of the tags posted by a user.

Manual aggregation of top words

The 1000 most frequent tags (after cleaning) were manually grouped into 10 categories. These categories were afterwards used for computing some variables:

- The main category used by the user
- The total number of categories used
- The number and proportion of the user’s tags in each category
- The set of all categories used at least once by the user (appended in order to form a string variable)

Variables describing resources

The resources (both URLs and BibTEX entries) pointed out by a user were described using the following variables:

- The number of resources bookmarked by the user
- The information of different fields describing the resource (url, url_hash, description and extended_description for bookmarks, and description for BibTEX entries). The per-user sum, average, minimum and maximum information is then computed for every of those fields.

3 Modelisation

Segmentation

Users can be divided into 3 segments: users with no BibTEX entry, users with no bookmark entry, and users with both BibTEX and bookmarks. The following table shows the proportion of spammers in each segment.

Has bookmark	Has BibTEX	Nb of users	% spammers
1	0	30386	95.9 %
0	1	682	3.8 %
1	1	647	14.2 %

The strong differences in the proportion of targets shown in this table suggests to separate users having BibTEX (and possibly bookmarks too) from users having no BibTEX. We thus performed two different models.

Since our modelisation tool, RANK, expect to predict the modality which is less represented, the prediction tasks were set up so as, for the BibTEX, predicting the spammers, and for the non-BibTEX, prediction of the non-spammers.

RANK

RANK is a predictive modeling tool designed by analysts for the analyst. As a result, it combines powerful techniques and modeling experience.

It is the first tool that automates many steps of the CRISP DM methodology (http://www.crisp-dm.org/) for building models.

RANK is built to allow an analyst to quickly build models on huge data sets, and have all elements to control the model choices and its quality, in order to focus his attention on the most important part of the modeling process: data quality, overfitting, stability and robustness. Using RANK, the analyst will get support for many modeling phases: audit, variable recoding, variable selection, robustness improvement, result analysis and industrialization.

Using ridge regression [2] on a linearized space, RANK combines the robustness of the linear models and the performance of a tidily controlled non-linear approach.

Variable recoding

Non linear Recoding

The recoding of variables is an extremely important and time-consuming step in the modeling process. Analysts know that the quality of a model can be heavily influenced by this phase. This is why RANK has been extensively developed on this step to ensure best model performance. RANK allows the user to specify which type of recoding he/she wants to test, and RANK will just do it, and select the best recoding scheme for each variable.

The types of recoding are the following:

- Nominal variables – Modalities will be converted to a numeric value that is related to its relation with the target density. This recoding is known under the name "weight of evidence recoding" [3]. Modalities can also be recoded using dummy variables.
- Numerical variable – There are two possibilities: simple normalization of the variables or binning of the variable using a proprietary algorithm ('intelligent quantiles') and then treated as nominal variables with order. The intelligent quantiles analyses the distribution of a variable in order to identify most relevant quantiles, identifying 'plateau' and jumps in the distribution. In this mode, jumps also produce dummy variables.

The recoding performed by RANK has two major effects:

- The first one is to get rid of the problem of non-normal distributions that should be a basic assumption when using regression models. The recoding will remove the dissymmetry and make the data more suitable for regression models.
- The second effect is that the recoding allows RANK to spot non-linear relationships of a variable with the target, thus improving the expression power of the model.

Modality grouping

RANK automatically analyzes all variables along their cardinality. If a nominal variable has many modalities, RANK will group them in a way that each grouped modality becomes significant. For example, if Zip code with 30.000 modalities is used, only the modalities that are significant will be left as they are. The others will be grouped in a default modality. The grouping will preserve the order relationship in a variable if any. For example, for an ordinal variable like 'number of sms sent', RANK will group only modalities that are adjacent, and will possibly create many grouped modalities

Missing Values

RANK treats missing values in a very careful way. Depending on the type of variable, RANK will recode missing values in a way such that its effect on the computed score is null. This ensures that the model focuses only on relevant information for the prediction.

Variable selection

LARS Forward

When the number of variables is high (> 500), a first variable selection made using the Least Angle Regression (LARS) [1]. LARS is an embedded technique which simultaneously estimates the parameters of a linear regression and selects the most relevant variables. It is only used here for variable selection as the regression coefficients will be re-estimated later on using a ridge regression. In RANK, a variant of LARS called, *LARS with Lasso modification* [1], is implemented. Interestingly, this method computes the parameters of the Lasso regression, *i.e.* a linear regression with an upper bound on the L1 norm of the vector of coefficients. Using the L1 norm enforces the sparseness of coefficients leading to effective variable shrinkage. Importantly, the LARS procedure returns *all* the Lasso solutions in a single run, *i.e.* the coefficients for any (positive) value of the upper bound. To do so, LARS operates iteratively. At each iteration, a new variable is selected and a step is taken in the direction equi-angular to the columns of the data matrix corresponding to the currently selected variables. Doing so allows one to progressively minimize the residual error of the model while spreading uniformly its variance over all the selected variables. The algorithm is iterated until the relative residual error of consecutive iterations falls below a user-defined threshold. Note that, even if LARS works in a forward fashion, it has the ability to take backward steps by removing variables becoming useless at some stage. The algorithm results in variables pre-selection that will, afterwards, be validated by the backward pruning.

This mode can be applied on data sets involving more than 200.000 variables.

Lift optimized Backward

The backward pruning in RANK can either start with all the variables or with the pre-selection returned by the LARS. In both cases, it iteratively eliminates variables when their removal does not influence the quality of the prediction more than a prescribed threshold.

Using cross-validation, it will end with a variables selection that maximizes the area under the lift curve.

Robust Regression

Ridge regression

The regression engine of RANK uses the so-called 'Ridge' regression [2]. This technology allows improving the robustness of the models as well as improving the usage of nominal variables with a lot of modalities, like zip codes.

Cross-Validation & Bootstrap

RANK extensively uses cross-validation technique when building a model: to assert which recoding is best, to select best variables, and to evaluate the ridge regression constant. This is extremely useful when the target density is very low, which is 90% the case in real life projects like churn prediction (0.7 % per month), cross- and up-selling (0.3% of our clients possess this product) or fraud detection (0.02% of all cases). Cross-validation and bootstrap is not only relevant for building a robust model, it is also important for the analyst to observe the volatility of the model quality.

Probability Estimation

The output of RANK is not just a score. It also gives for each record the best estimation of the response probability, based on the model score function and the *a priori* probability of the target in the data file.

4 Results

Selected variables

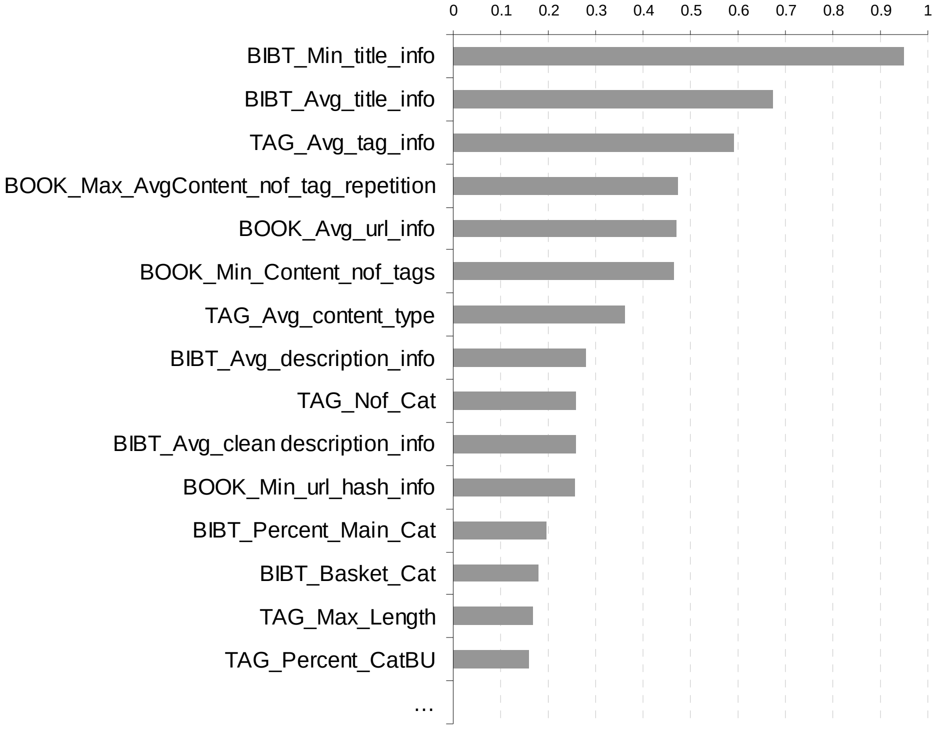
This section lists the top variables for each model. For some variables, it also contains graphics showing the relation between the variable distribution and the target.

We usually rank the variables according to their importance. The importance is the loss (in percentage) of lift quality that we observe if we remove the variable.

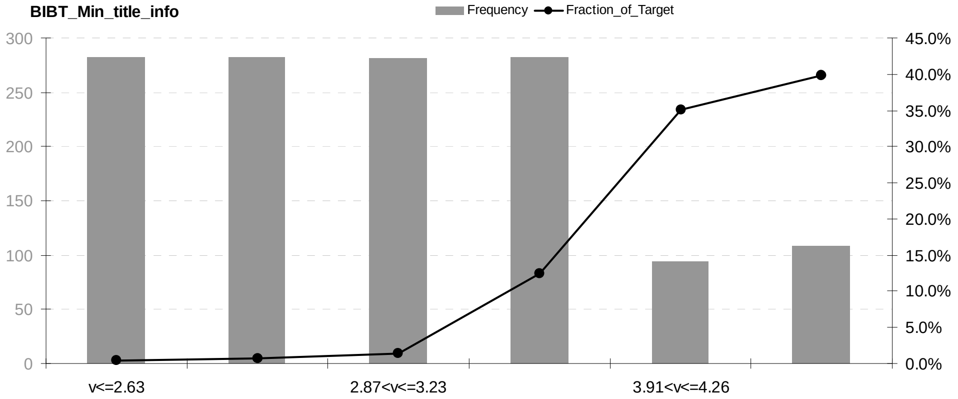
Model for BibTEX users

In this model, the target is spam user. We have 1,329 users and 118 target (8.88%). Our model is composed of 53 variables; most of them are measuring information of a text field.

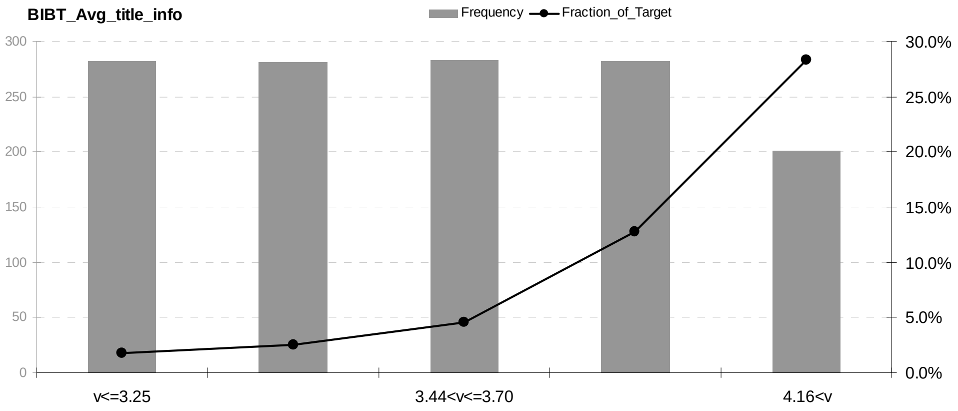
Fig. 1. Most important variables: Top 15 for BibTEXusers.



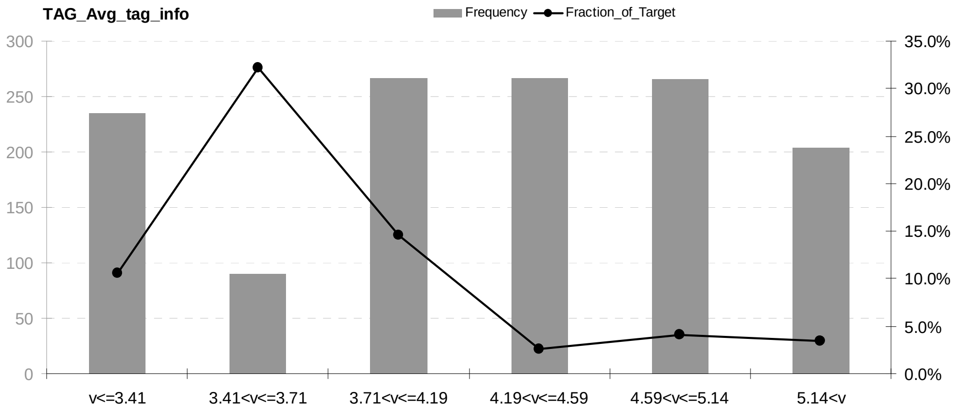
- **BIBT_Min_title_info**: minimum information contained among the titles of the BibTEX posted by the user.



- **BIBT_Avg_title_info**: average information contained among the titles of the BibTEX posted by the user.



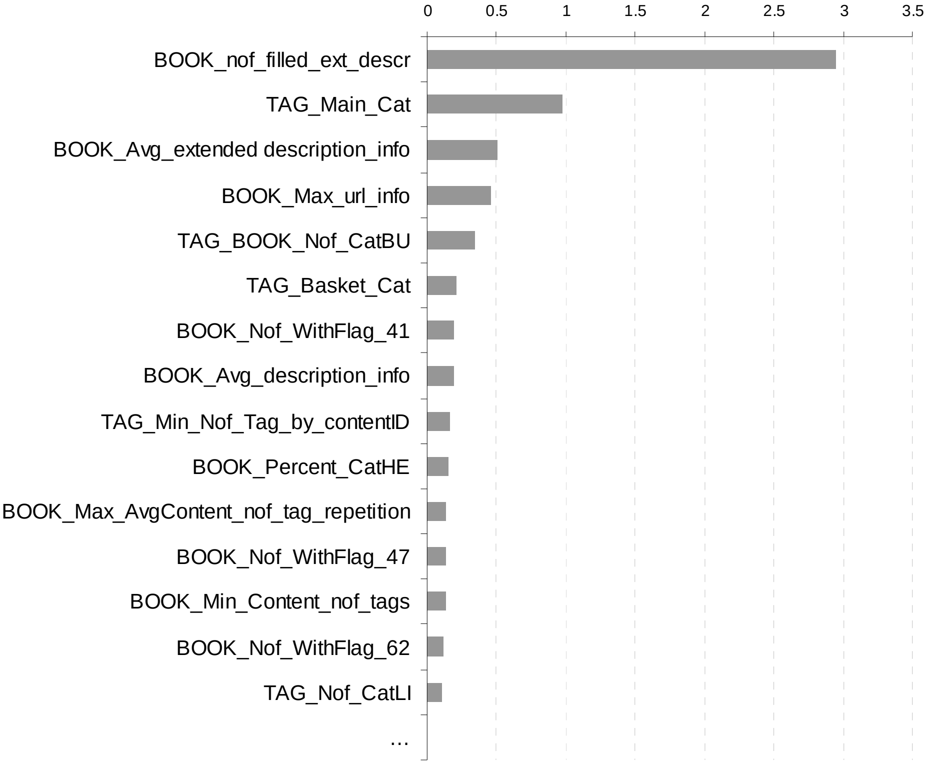
- **TAG_Avg_tag_info:** average information contained among all the tags posted by the user.



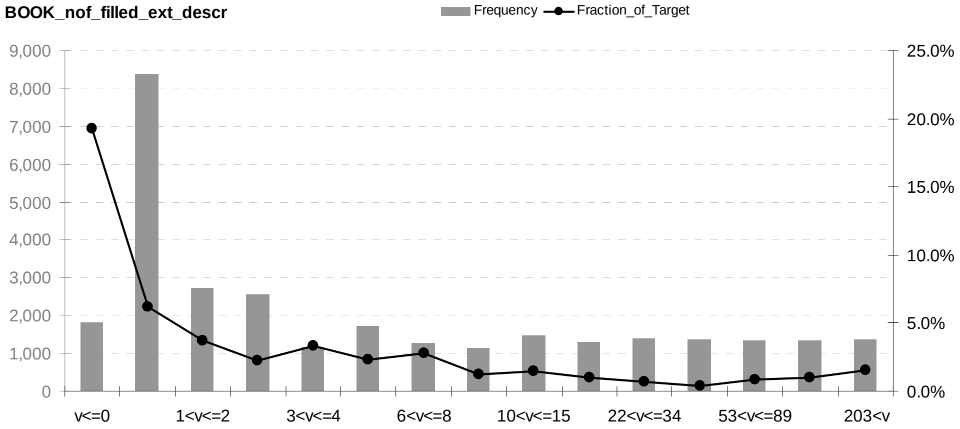
Model for non-BibTEX users

For this model, a target is a non spam user. We have 30,386 users with no BibTEX, 1,256 of them are not spammer (4.13%). In our model, we end up with 70 variables. The figure below shows the 15 most important variables.

Fig. 2. Most important variables: Top 15 for non-BibTEXusers.

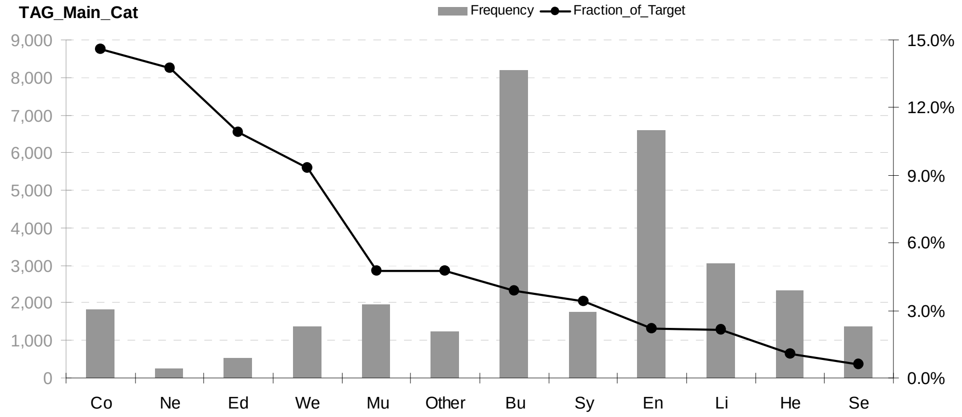


- **BOOK_Nof_Filled_ext_desc**. This first variable counts the number of filled extended_description.

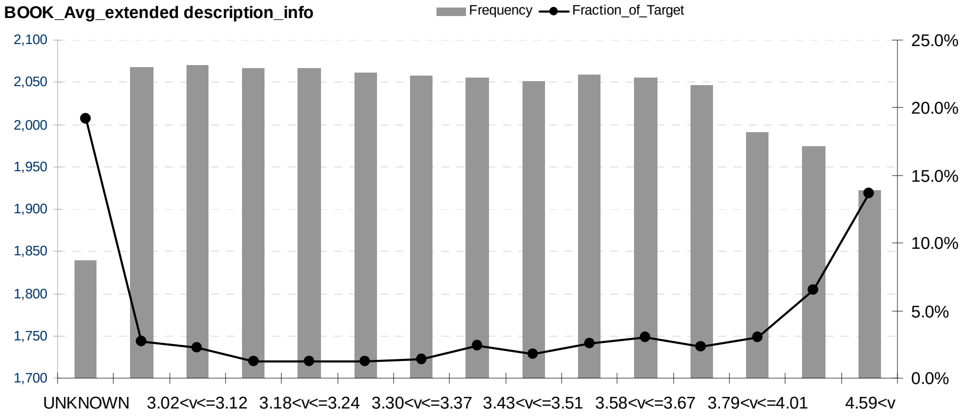


- **TAG_Main_Cat**. This variable shows the main category of the user tags. The most interesting categories are:

- Co = Computer (e.g.: software, program,...)
- Ne = News (e. g.: information, news,...)
- Ed = Education (e.g.: exercise, student,....)
- Li = Link word (e.g.: you, from,...)
- He = Health (e. g.: acne, treatment,...)
- Se = Sex (e. g.: lesbians, xxx,...)



- **BOOK_Avg_Extended_description_info**: This computes the average information in extended_description among all content ids of the user.

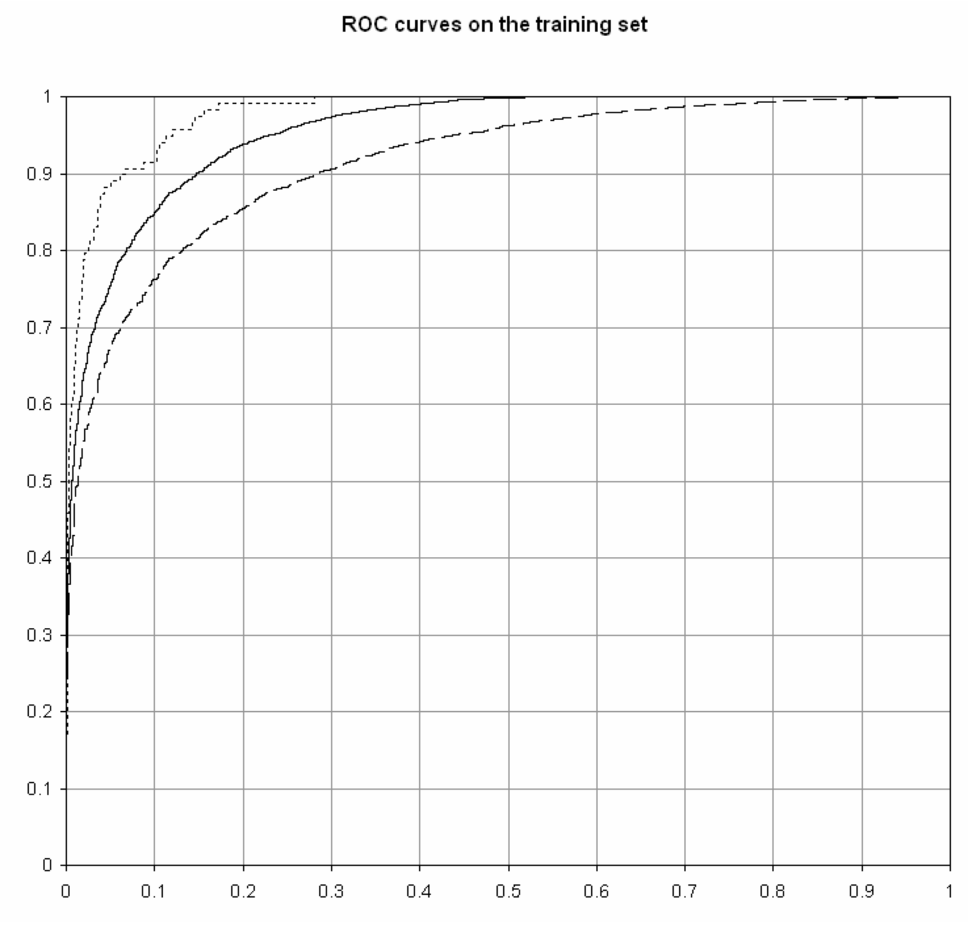


ROC curve

Training set

The following figure shows the ROC curves achieved on the training set for the two models (users with or without BibTEX), and for their combination. All these curves represent the performance for spam users prediction, hence the probability of the non-BibTEX model has been reverted. The area under the curve is 0.9792 for the BibTEX model, 0.9151 for the NoBibTex model, and 0.9556 for the global model.

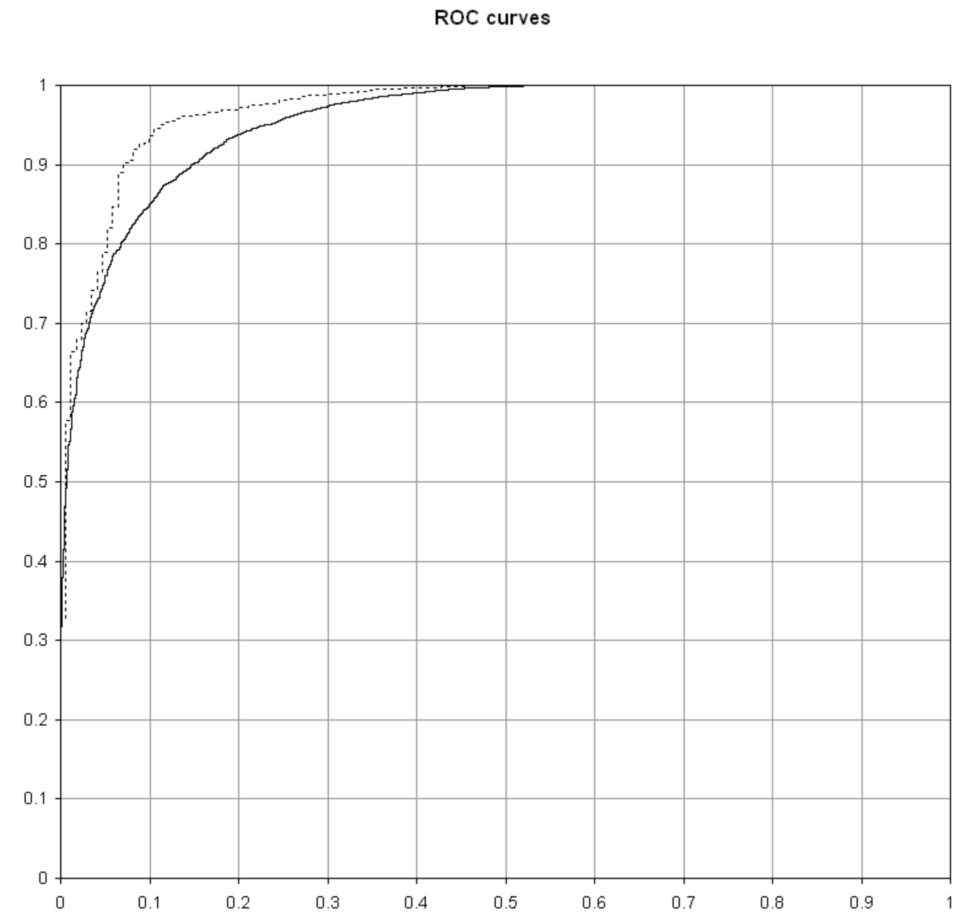
Fig. 3. ROC curve of the models on the build set. The dotted line represents the ROC curve of the BibTEX users model, the dashed line represents the ROC for the non-BibTEX users model, and the plain line shows the ROC we obtain when we combine the two models (global model).



Test set

Finally, the following pictures compares the performance of the model on the training and test sets. For the test set , the area achieved under the ROC is 0.9703. This is higher than the build! This is probably due to the fact that we have more BibTEX users in the test set.

Fig. 4. ROC curve of the global model. The plain line represents the ROC curve on the build set whereas the dotted line represents the ROC on the evaluation test set.



5 References

1. B. Efron, T. Hastie, I Johnstone and R. Tibshirani. Least Angle Regression, *The Annals of statistics* 2004, Vol 32, No 2, 407-499.
2. Hoerl, A. E. and Kennard, R. (1970). Ridge regression: biased estimation for nonorthogonal problems, *Technometrics* 12: 55-67.
3. Smith EP, Lipkovich I, Ye K. Weight of Evidence (WOE): Quantitative estimation of probability of impact. Blacksburg, VA: Virginia Tech, Department of Statistics; 2002.