

Predicting Storms: Logistic Regression versus Random Forests for Unbalanced Data

Anne Ruiz-Gazen

Institut de Mathématiques de Toulouse and Gremaq, Université Toulouse I, France

Nathalie Villa

Institut de Mathématiques de Toulouse, Université Toulouse III, France

Abstract: The goal of this study is to compare two supervised classification methods on a crucial meteorological problem. The data consist of satellite measurements of cloud systems which are to be classified either in convective or non convective systems. Convective cloud systems correspond to lightning and detecting such systems is of main importance for thunderstorm monitoring and warning. Because the problem is highly unbalanced, we consider specific performance criteria and different strategies. This case study can be used in an advanced course of data mining in order to illustrate the use of logistic regression and random forests on a real data set with unbalanced classes.

Introduction

Predicting storms is a high challenge for the French meteorological group Météo France: often, storms are sudden phenomena that can cause high damage and lead to stop some human activities. In particular, airports have to be closed during a storm and planes have to be taken out of their way to land elsewhere: this needs to be planned in order to avoid airports' overbooking or security difficulties. To reach this goal, Météo France uses satellites that monitor the clouds that hover over France; these clouds are named "systems" and several measurements are known about them: some are temperatures, others are morphological measurements (volume, height, ...). Moreover, a "storm network", based on measurements made on earth, can be used to identify,

a posteriori, systems that have led to storms. Then, a training database is built: it contains both satellite measurements and information coming from the storm network for each considered system.

The goal of this paper is to present two statistical approaches in order to discriminate the convective systems (clouds that lead to a storm) from the non convective ones: a logistic regression is compared to a random forests algorithm. The main point of this study is that the data set is extremely unbalanced: convective systems represent only 4% of the whole database; therefore, the consequences of such unbalancing and the strategies developed to overcome it are discussed.

The rest of the paper is organized as follows: Section I describes the database. Then, Section II presents the methods and performance criteria that are suitable in this problem. Section III gives some details about the R programs that we used and Section IV presents the results of the study and discusses them.

I. Description of the data set

The working dataset was divided into two sets in a natural way: the first set corresponds to cloud systems observed between June and August 2004 while the second set corresponds to systems observed during the same months in 2005. So, the 11 803 observations made in 2004 are used to build a training sample and the 20 998 observations made in 2005 are used as a test sample. All the cloud systems are described by 41 numerical variables built from satellite measurements (see the Appendix for details) and by their nature (convective or not). Moreover, the values of the variables were recorded at 3 distinct moments: all the systems were observed during at least 30 minutes and a satellite observation made each 15 minutes. The convective systems variables were recorded 30 minutes and 15 minutes before the first recorded lightning and at the first lightning moment. A 30 minutes period also leading to three different moments was randomly sampled by Météo France in the trajectories of the non convective systems. A preprocessing program that will not be detailed further was applied to the data in order to clean highly correlated variables and outliers from the dataset.

The reader can find the observations made in 2004 in the file `train.csv` and the observations made in 2005 in the file `test.csv`. The first line of these files contains the names of the variable and the last column (“`cv`”) indicates the nature of the data: 1 is set for “convective” and 0 for “non convective”. The same data can also be found in the R file `ruizvilla.Rdata`.

For the rest of the paper, we will denote X the random vector of the $p = 41$ satellite measurements and $Y \in \{0,1\}$ the variable that has to be predicted from X (i.e. if the system is convective or not).

As most of the observed systems are non convective, the database is extremely unbalanced: in the training set, 4.06% of the systems are convective and in the test set, only 2.98% of the systems are convective. This very few number of convective systems leads to the problem of training a model from data having a very few number of observations in the class of interest.

II. Methods

The models that will be developed further lead to estimating the probability $P(Y = 1|X)$. The estimate of this conditional probability will be denoted by $\hat{P}(Y = 1|X)$. Usually, given these estimates, an observation is classified in class 1 if its estimated probability satisfies

$$\hat{P}(Y = 1|X) > 0.5. \quad (1)$$

A. Unbalanced data set

When the estimation of $P(Y = 1|X)$ is obtained from an unbalanced data set, the methodology described above could lead to poor performance results (see, for instance, Dupret and Koda (2001), Liu et al. (2006)). Several simple strategies are commonly proposed to overcome this difficulty.

The first approach consists in *re-balancing* the dataset. Liu et al. (2006) explore several re-balancing schemes both based on downsampling and on oversampling. Their experiments seem to show that downsampling is a better strategy. Thus, the training set has been re-balanced by keeping all the observations belonging to the minority class (convective systems) and by randomly sampling (without replacement) a given (smaller than the original) number of observations from the majority class (non convective systems). Moreover, as suggested by Dupret and Koda (2001), the optimal re-balancing strategy is not necessarily to downsample the minority class at the same level as the majority class. A previous mining of the dataset (see Section IV, A) leads to the choice of a downsampling rate equal to:

$$\# \text{ Convective} / \# \text{ Non Convective} = 0.2.$$

The sampled database that was used for training the methods can be found in the file `train_sample.csv`.

The second approach to treating the problem of unbalanced database is to infer the final decision from the choice of a *threshold* τ . In equation (1), the threshold is set to 0.5 but, as emphasized for example in Lemmens and Croux (2006), this choice is generally not optimal in the case of unbalanced datasets. Then, the distribution of the estimates $\hat{P}(Y = 1|X)$ is studied for all the observations of the training and test samples and the evolution of the performance criterion as a function

of the threshold is drawn in order to propose several solutions for the final decision function. This final decision can be adapted to the goal to achieve, making the balance between a good level of recognition for the convective systems and a low level of false alarms.

B. Description of the methods

We propose to investigate two classification methods for the discrimination of convective systems: the logistic regression and the more recent random forests method. These two approaches have been chosen for their complementary properties: logistic regression is a well-known and simple model based on a generalized linear model. On the contrary, the random forests technique is a non linear and non parametric model combining two recent learning methods: classification trees and the bagging algorithm. Among all the recent advances in statistical learning, classification trees have the advantage of being very easy to understand, an important feature for communication. Combined with a bagging scheme, the generalization ability of such a method has turned out to be very interesting. In the following, we present the two methods in more detail.

Logistic regression

“Logistic regression” is a classification parametric model: a linear model is built to estimate the logit of the probability that an observation belongs to a particular class. Using the notations introduced previously, the logistic regression estimates the probability $P(Y = 1|X)$ by using the following linear equation and the maximum likelihood estimation method:

$$\ln(P(Y = 1|X)/(1 - P(Y = 1|X))) = \alpha + \sum_{j=1}^p \beta_j X_j \quad (2)$$

with α and $\beta_j, j = 1, \dots, p$, the parameters to be estimated.

A “step by step” iterative algorithm allows us to compare a model based on p' of the p original variables to any of its sub-model (with one less variable) and to any of its top-model (with one more variable): non significant variables are dropped from the following model and relevant ones are added leading to a final model containing a minimal set of relevant variables.

This algorithm is performed by the function *stepAIC* from the R library “MASS” and leads to a logistic regression R object denoted by *rl.sbs.opt*. This object can be used for prediction by looking at *rl.sbs.opt\$coefficients* which gives the name of the variables used in the final model and their corresponding coefficient β . The *Intercep* is simply

the coefficient α of the model. Using the estimators and computing the model equation for a new system gives a predicted value S that can be transformed into a probability of being convective by:

$$\hat{P} = \exp(S)/(1 + \exp(S)).$$

Random Forests

The “Random Forests” method is a classification or regression tool that was developed recently (Breiman, 2001). It is a combination of tree predictors such that each tree is built independently of the others. The method is easy to understand and has proven its efficiency as a nonlinear tool. Let us describe it more precisely.

A classification tree is defined iteratively by a division criterion (node) obtained from one of the variables, x_j , which leads to the construction of two subsets in the training sample: one subset contains the observations i that satisfy the condition $x_j^i < T$ while the other subset contains the observations i that satisfy $x_j^i \geq T$ (T is a real number which is defined by the algorithm). The choices of the variable and of the threshold T are automatically built by the algorithm in order to minimize a heterogeneity criterion (the purpose is to obtain two subsets that are as homogeneous as possible in term of their values of Y). The growth of the tree is stopped when all the subsets obtained have homogeneous values of Y or when limits imposed by other stopping criteria (related for example to the number of nodes or parsimony considerations) have been reached. But, despite its intuitive form, a single classification tree is often not a very accurate classification function.

Thus, Random Forests improve upon this classification technique by aggregating several under-efficient classification trees using a “bagging” procedure: at each step of the algorithm, some observations and some variables are randomly chosen and a classification tree is built from this new data set. The final classification decision is obtained by a majority vote law on all the classification trees (and we can also infer an estimation of the probability of each class by calculating the proportion of each decision on all the classification trees). When the number of trees is large enough, the generalization ability of this algorithm is good.

This can be illustrated by the representation of the “out of bag” error which tends to stabilize to a low value: for each observation, the predicted values from the trees that were not built using this observation are calculated.

These predictions are said to be “out of bag” because they are based on classifiers (trees) for which the considered observation was not selected by the bagging scheme. Then, by a majority vote law, a classification “out of bag” is obtained and compared to the real class of the observation.

Figure 1 illustrates the convergence of the random forest algorithm: it depicts the evolution of the out-of-bag error as a function of the number of trees that was used (for the sample data train.sample). We can see that, up to about 300 trees, the error stabilizes which indicates that the obtained random forest has reached its optimal classification error.

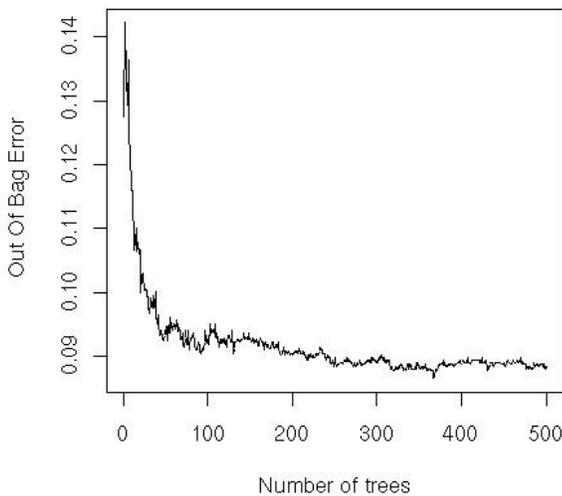


Figure 1. “Out of bag” error for the training of a random forest on the sample train.sample.

C. Performance criteria

For a given threshold τ (see section II A), the performance of a learning method can be measured by a confusion matrix which gives a cross-classification of the predicted class (Pred) by the true class (True). In this matrix, the number of *hits* is the number of observed convective systems that are predicted as convective, the number of *false alarms* is the number of observed non convective systems that are predicted as convective, the number of *misses* is the number of observed convective systems that are predicted as non convective and the number of *correct rejections* is the number of observed non convective systems that are predicted as non convective.

Usual performance criteria (see for instance, Hand, 1997) are the sensitivity (Se) which is the number of hits over the number of convective systems and the specificity (Sp) which is the number of correct rejections over the number of non convective systems. The well known ROC

(receiver operating characteristic) curve plots the sensitivity on the vertical axis against $(1 - \text{specificity})$ on the horizontal axis for a range of threshold values. But in this unbalanced problem, such criteria and curves are not of interest. To illustrate the problem, let us consider the following small example which is not very different from our data set (Table 1).

For this example, the sensitivity $Se=100\%$ and the specificity $Sp \approx 95\%$ are very high but these good performances are not really satisfactory from the point of view of Météo France. Actually, when looking at the 1000 detected convective systems, only 500 correspond to true convective systems while the other 500 correspond to false alarms; this high rate of false alarms is unacceptable practically as it can lead to closing airports for wrong reasons too often.

Table 1. First example.

True Pred \ Convective	Convective	Non convective	Total
Convective	# hits 500	# false alarms 500	1000
Non-convective	# misses 0	# correct rejections 9000	9000
Total	500	9500	10000

More generally, because of the unbalanced problem, the number of correct rejections is likely to be very high and so, all the usual performance criteria that take into account the number of correct rejections are likely to give overoptimistic results. So, in this paper, the focus will not be on the ROC curve but rather on the following two performance scores:

the **false alarm rate (FAR)**:

$$FAR = \frac{\text{\# false alarms}}{\text{\# hits} + \text{\# false alarms}}$$

We want a FAR as close to 0 as possible. Note that in the small example, $FAR = 50\%$ which is not a good result.

the **threat score (TS)**:

$$TS = \frac{\text{\# hits}}{\text{\# hits} + \text{\# false alarms} + \text{\# misses}}$$

We want a TS as close to 1 as possible. In the small example, the value of $TS=50\%$ which is also not very good.

These performance measures can be calculated for any threshold τ so that curves can be plotted as will be illustrated in section IV. Note that for a threshold near 0 and because the dataset is unbalanced, the number of hits and the number of misses are expected to be low compared with the number of false alarms and so the

FAR is expected to be near 1 and the TS near 0. When the threshold increases, we expect a decrease of the FAR while the TS will increase until a certain threshold value. For high threshold values, if the number of misses increases, the TS will decrease with the threshold, leading to a value near 0 when the threshold is near 1. These points will be illustrated in section IV.

III. Programs

All the programs have been written using the R software¹. Useful information about R programming can be found in R Development Core Team (2005). The programs are divided into several steps:

- a **pre-processing** step where a sampled database is built. This step is performed via the function `rebalance.R` which requires 2 inputs, `train`, which is the data frame of the original training database (file `train.csv`) and `trainr` which is the ratio (`# Convective / # Non Convective`) to be found after the sampling scheme. The output of this function is a data frame taking into account the ratio `trainr` by keeping all the convective systems found in `train` and by downsampling (without replacement) the non convective systems of `train`. An example of output data is given in the file `train_sample.csv` for a sampling ratio of 0.2.
- a **training step** where the training sampled databases are used to train a logistic regression and a random forest.

The *logistic regression* is performed with a step by step variable selection which leads to finding the most relevant variables for this model. The function used is `LogisReg.R` and uses the library “MASS” provided with R programs. It needs 3 inputs: `train.sample` is the sampled data frame, `train` is the original data frame from which `train.sample` was built and `test` is the test data frame (June to August 2005). It provides the trained model (`rl.sbs.opt`) and also the probability estimates $\hat{P}(Y = 1|X)$ for the databases `train.sample`, `train` and `test`, denoted respectively by, `prob.train.sample`, `prob.train` and `prob.test`.

The *random forest* is performed with the function `RF.R` which uses the library “randomForest”². This function has the same inputs and outputs as `RegLogis.R` but the selected model is named `forest.opt`. It also plots the out-

of-bag error of the method as a function of the number of trees trained: this plot can help to control whether the chosen number of trees is large enough to yield a stabilized error rate.

- a **post-processing step** where the performance parameters are calculated. The function `perform.R` requires 7 inputs: the probability estimates calculated during the training step (`prob.train.sample`, `prob.train` and `prob.test`), the value of the variable of interest (convective or not) for the three databases and a given value for the threshold, τ , which is named `T.opt`. Using the libraries “stats” and “ROCR”³, the function calculates the confusion matrix for the 3 databases together with the TS and FAR scores associated to the chosen threshold and the Area Under the Curve (AUC) ROC. Moreover, several graphics are built:
 - the estimated densities for the probability estimates, both for the convective systems and the non convective systems;
 - the histograms of the probability estimates both for convective and non convective systems for 3 sets of values (`[0,0.2]`, `[0.2,0.5]` and `[0.5,1]`);
 - graphs of TS and FAR as a function of the threshold together with graphs of TS versus FAR where the chosen threshold is emphasized as a dot;
 - the ROC curve.

A last function, `TsFarComp.R` creates graphics that compare the respective TS and FAR for two different models. The input parameters of this function are: two series of probability estimates (`prob.test.1` and `prob.test.2`), the respective values of the variable of interest (`conv.test1` and `conv.test2`) and a chosen threshold (`T.opt`). The outputs of this function are the graphics of TS versus FAR for both series of probability estimates where the chosen threshold is emphasized as a dot.

Finally, the programs `Logis.R` and `RandomForest.R` describe a complete procedure where all these functions are used. In `Logis.R` the case where no sampling is made is compared to the case where a sampling with rate 0.2 is pre-processed, for the logistic regression. In `RandomForest.R`, the probability estimates obtained for the re-sampled training set `train.sample` by the use of `forest.opt` are compared to the out-of-bag probability

1 Freely available at <http://www.r-project.org/>

2 L. Breiman and A. Cutler, available at <http://stat-www.berkeley.edu/users/breiman/RandomForests>

3 T. Sing, O. Sander, N. Beerenwinkel and T. Lengauer, available at <http://rocr.bioinf.mpi-sb.mpg.de>

estimates. These programs together with the R functions described above are provided to the reader with the data sets.

IV. Results and discussion

The previous programs have been used on the training and test databases. In this section, we present some of the results obtained in the study. In particular, we focus on the unbalanced problem and on the comparison between logistic regression and random forests according to different criteria.

A. Balancing the database

In this section, we consider the logistic regression method and propose to compare the results for different sampling rates in the non convective training database. First, let us consider the original training database containing all the recorded non convective systems from June to August 2004. A stepwise logistic regression is carried out on this training database as detailed in the previous section (program RegLogis) and predicted probabilities are calculated for the training and for the test databases. As already mentioned, once the predicted probabilities are calculated, the classification decision depends on the choice of a threshold τ which is crucial.

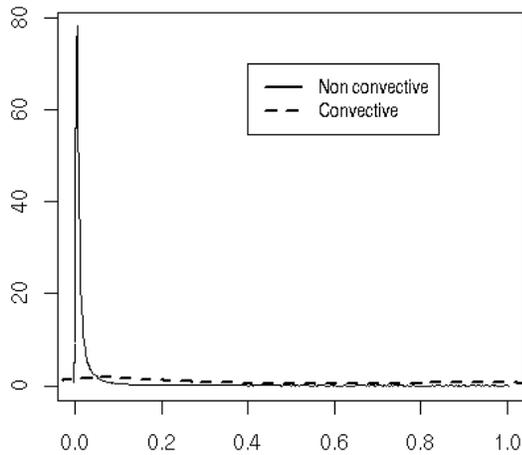


Figure 2. Densities of the predicted probabilities of the test set for logistic regression trained on train.

Figure 2 plots a kernel density estimate of the predicted probabilities for the non convective (solid line) and the convective (dotted line) cloud systems for the test database using the train set for training. The two densities look very different with a clear mode for the non convective systems near 0. This could appear as an encouraging result. However, results are not very good when looking at the FAR and TS.

Figure 3 plots the FAR as a function of TS for different threshold values from 0 to 1 for the training database (in green) and for the test database (in red). As expected (see section II), for low values of the threshold which correspond to high values of the FAR, the FAR decreases when the TS increases. On the contrary, for low values of the FAR (which are the interesting ones), the FAR increases when TS increases. As a consequence, the threshold associated with the extreme point on the right of the figure is of interest only if it corresponds to an admissible FAR. On Figure 3, such a particular threshold is associated with a FAR $\approx 20\%$ and a TS $\approx 35\%$ for the training database (which is not too bad). But it leads to a FAR $\approx 40\%$ and a TS $\approx 30\%$ for the test database. A FAR of 40% is clearly too high according to Météo France objectives but, unfortunately, in order to obtain a FAR of 20%, we would need to accept a lower TS ($\approx 20\%$).

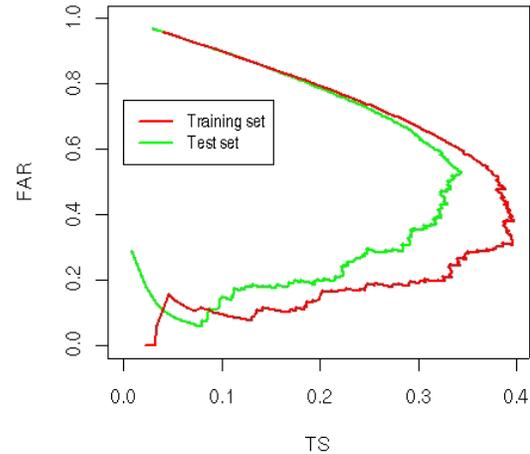


Figure 3. FAR as a function of TS for logistic regression for the training set (green) and the test set (red) using the training set train.

Let us now compare the previous results with the ones obtained when considering a sample of the non convective systems. Focus is on the test database only. The sampling rate is taken such that the number of non convective systems is equal to five times the number of convective systems (sampling ratio = 0.2).

When looking at Figure 4, the TS/FAR curves look very similar to those of Figure 3. When sampling, there is no loss in terms of FAR and TS but there is no benefit either. In other words, if a threshold $\tau_1 \in [0,1]$ is fixed when taking into account all the non convective systems in the training database, τ_1 corresponds to some FAR and TS; it is always possible to find a threshold τ_2 leading to the same FAR and TS for the test database when training on the sampled database. So, the main advantage of

sampling is that because the training data base is much smaller, the classification procedure is much faster (especially the stepwise logistic regression) for a similar performance.

Figure 5 gives the density of the predicted probabilities for the non convective (solid line) and the convective (dotted line) cloud systems for the test database using the train.sample set for training. Comparing Figures 2 and 5 is not informative because although they are quite different, they correspond to very similar performances in terms of the TS and the FAR.

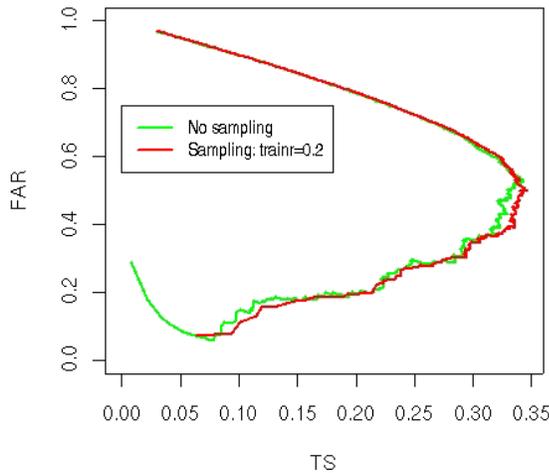


Figure 4. FAR as a function of TS for the test set for logistic regression trained with train (green) or train.sample (red) with 500 discretization points.

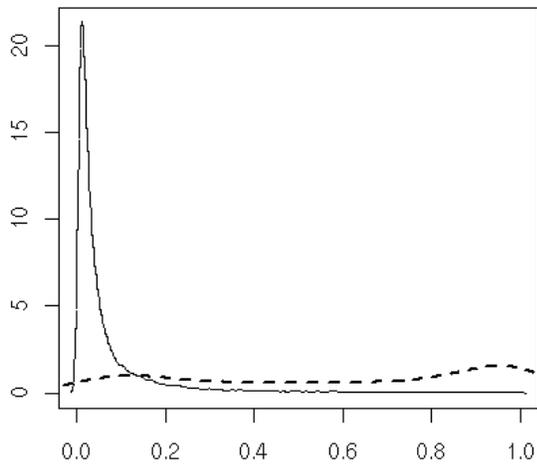


Figure 5. Densities of the probability estimates of the test set for a logistic regression trained on train.sample (train=0.2).

Note that further analyses show that the sampling ratio 0.2 seems to be a good compromise when considering computational speed and performance. Actually, the performance in terms of the FAR and the TS is almost

identical when considering larger sampling rates while the performance decreases a bit for more balanced training databases. As illustrated by Figure 6 which gives a comparison of FAR/TS curves for a sampling ratio of 0.2 and 1, the performance criteria are a little bit worse when the training database is completely rebalanced. Also note that, for the random forests algorithm, very similar results were obtained and thus will not be detailed further.

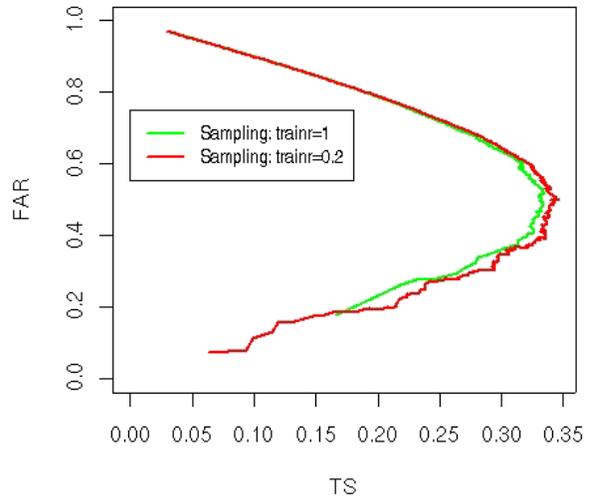


Figure 6. FAR as a function of TS for logistic regression on train.sample (red) and on a training sample with a trainr = 1 (re-balanced) with 500 discretization points.

B. Random forests versus logistic regression

Here, as explained in the previous section, we focus on the sampled database train.sample for which the sampling ratio (0.2) appears to be a good choice for both logistic regression and random forests. We now intend to compare the two methodologies and explain their advantages and drawbacks.

Let us first focus on the comparison of the densities of the probability estimates obtained by both models. Figures 7 and 9 illustrate this comparison for the training (train.sample) and the test sets. The first salient observation is the difference between a direct computation of the probability estimates for the random forest model and the out-of-bag computation for the training set: Figure 7 shows distinct densities, both for convective and non convective systems (black and green curves).

The probability estimates for the random forest method are given for “out-of-bag” values (green) and for direct estimation from the final model (black).

This phenomenon is better explained when looking at the respective ROC curves obtained for random forest from the probability estimates calculated from the whole model and those calculated by the out-of-bag scheme (Figure 8).

This figure shows that the ROC curve built from the direct calculation of the probability estimates (black) is almost perfect whereas the one obtained from the “out-of-bag” probability estimates is much worse. This illustrates well that random forests overfit the training set, giving an almost perfect prediction for its observations. As a consequence, the probability densities built directly from the training set (black in Figure 7) have to be interpreted with care: although they seem to look better than the logistic regression ones, they could be misleading about the respective performances of the two models.

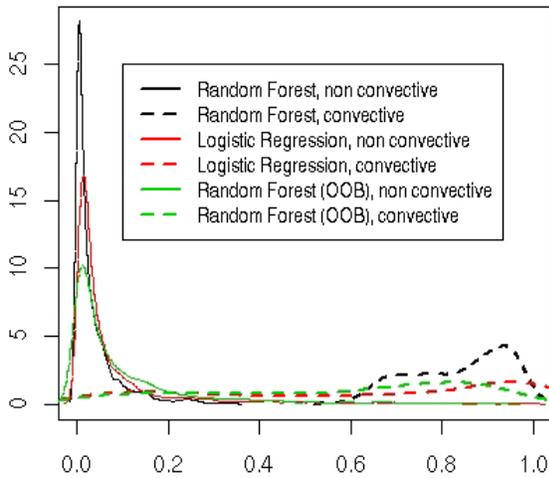


Figure 7. Densities of the probability estimates for both random forest and logistic regression on the training set.

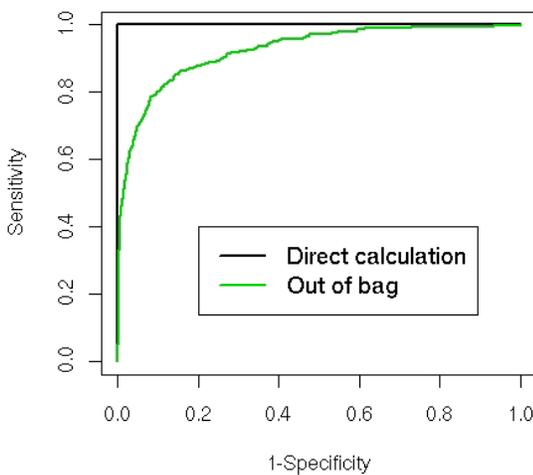


Figure 8. ROC curves for Random Forest for the training set built from the probability estimates directly calculated by the use of the chosen model (black) and from the out-of-bag

probability estimates (green).

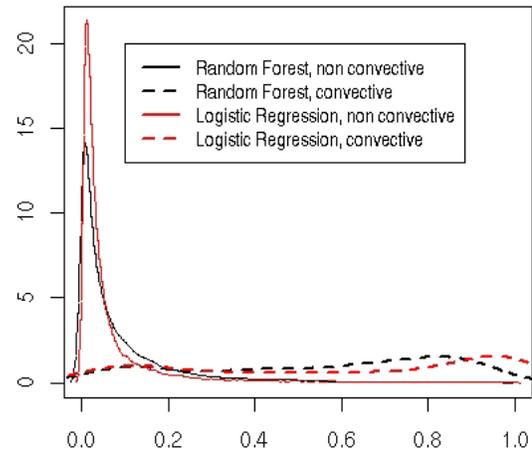


Figure 9. Densities of the probability estimates on the test set for random forest and logistic regression.

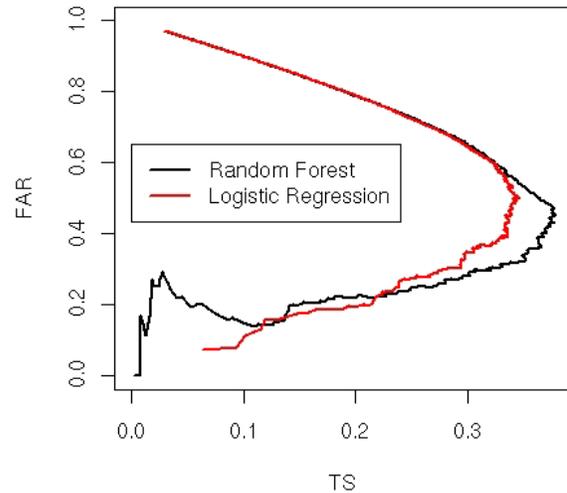


Figure 10. FAR as a function of TS for the test set (500 discretization points).

Moreover, looking at Figure 9, logistic regression seems to better discriminate the two families of probability estimates (for convective systems and non convective systems) than the random forests method for the test set. But, this conclusion should also be taken with care, due to possible scaling differences.

To provide a fairer comparison, the graph of FAR versus TS is given in Figure 10 for both random forests (black) and logistic regression (red). The first remark from this graphic is that the performances of both models are rather poor: for a FAR of 30%, which seems to be a maximum for Météo France, the optimal TS is about 25-30% only.

Focusing on the comparison between random forest and

logistic regression, matters are not as simple as the comparison of probability densities (Figure 9) might lead us to believe. Actually, the optimal model strongly depends on the objectives: for a low FAR (between 15-20%), the optimal model (that has the highest TS) is logistic regression. But if a higher FAR is allowed (20-30%), then random forests should be used for prediction.

These conclusions indicate that the two sets of densities can be simply re-scaled one from the other (approximately). Figure 11 explains in details what happens: in this figure, the density of the probability estimates for the logistic regression is estimated, as in Figure 7, but the densities for the random forest method have been re-scaled.

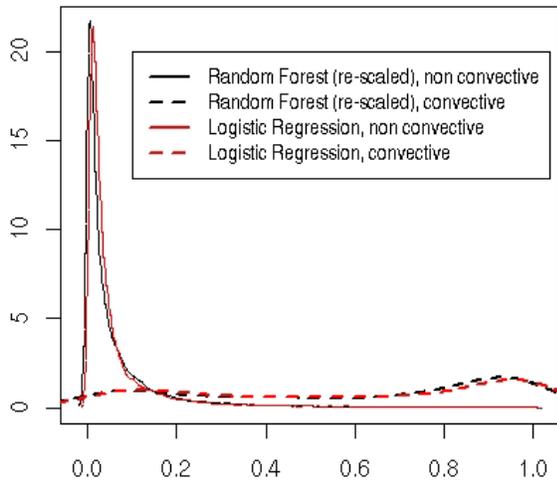


Figure 11. Rescaled probability densities for the test set (details about its definition are given in the text).

More precisely, these probability estimates have been transformed by the following function:

$$\varphi(x) = \begin{cases} \alpha_x x & \text{if } x \leq 0.5 \\ 1 - \beta_x (1 - x) & \text{if } x > 0.5 \end{cases}$$

where $\alpha_x = 2(1 - 0.6)x + 0.6$ and

$$\beta_x = 2(1 - 10^{-3})(1 - x) + 10^{-3} .$$

Clearly, this simple transformation does not affect the classification ability of the method (the ranking of the probabilities is not modified) and shows that random forests and logistic regression have a very close ability to discriminate between the two populations of clouds. Moreover, a single rank test (paired Kendall test) proves that the ranking of the two methods are strongly similar, with a p-value of this test is less than 10^{-15} .

V. Conclusion

Several conclusions can be reached from this study: the first one is that unbalanced data lead to specific problems. As was demonstrated in this paper, results obtained with these kind of data should be handled with care. Usual misclassification errors may be irrelevant in this situation: ROC curves can lead to overoptimistic interpretations while comparisons of predicted probability densities can lead to misunderstandings. In this study, where a low false alarm rate is one of the main objectives, curves of FAR versus TS are particularly interesting for interpretation and comparisons.

Another interesting aspect of this study is the surprisingly comparable results of the simple logistic regression compared to the popular and more recently developed random forest method. Hand (2006) already argued that the most recent classification methods may not improve the results over standard methods. He points out the fact that, especially in the case of mislabeling in the database, the gain made by recent and sophisticated statistical methods can be marginal. Both the way the database was built (by merging satellites sources and storm network records) and the poor performance results obtained with several classification methods tend to suggest mislabeling. As logistic regression is a faster method than random forests and because it leads to a simple interpretation of the explanatory variables, it can be of interest for meteorologists.

Finally, to better analyze the performances of the proposed methods, more information could be added to the database. For instance, a storm proximity index could indicate if false positive systems are closer to a storm than true positive ones.

Acknowledgments

We would like to thank Thibault Laurent and Houcine Zaghdoudi for cleaning the original database and helping us with the practical aspects of this study. We are also grateful to Yann Guillou and Frédéric Autones, engineers at Météo France, Toulouse, for giving us the opportunity to work on this project and for providing the databases.

Correspondence: ruiz@cict.fr

REFERENCES

- Breiman, L. 2001. *Random Forests*. Machine Learning, 45, 5-32.
- Dupret, G. and M. Koda. 2001. Bootstrap re-sampling for unbalanced data in supervised learning. *European Journal of Operational Research* 134(1): 141-156.
- Hand, D. 1997. *Construction and Assessment of Classification Rules*. Wiley.
- Hand, D. 2006. Classifier Technology and the Illusion of Progress. *Statistical Science*. 21(1), 1-14.
- Lemmens, A. and C. Croux. 2006. Bagging and boosting classification trees to predict churn. *Journal of Marketing Research* 134(1): 141-156.
- Liu, Y., N. Chawla, M. Harper, E. Shriberg, and A. Stolcke. 2006. A study in machine learning for imbalanced data for sentence boundary detection in speech. *Computer Speech and Language* 20(4): 468-494.
- R Development Core Team 2005. *R: a language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria.

Appendix: Definition of Variables

In this section, we describe the explanatory variables that were collected by satellite in order to explain storms. In the following, T will designate the instant of the storm (for convective systems) or the last sampled instant (for non convective systems). Δt will be the difference of time between two consecutive records (15 minutes).

Temperature variables

- $T_{sBT.0.0}$ is the threshold temperature at T recorded at the basis of the cloud tower.
- $toT_{moyBT.0.0}$ is the mean rate of variation of the mean temperature between T and $T - \Delta t$ at the basis of the cloud tower.
- $T_{min.0.30}$ is the minimal temperature at $T - 2 \Delta t$ recorded at the top of the cloud tower.
- $toT_{min.0.0}$, $toT_{min.0.15}$ and $toT_{min.0.30}$ are the minimal temperatures at T recorded at the top of the cloud tower ($toT_{min.0.0}$) or the mean rate of variation of the minimal temperature between T and $T - \Delta t$ ($toT_{min.0.15}$) or between $T - \Delta t$ and $T - 2 \Delta t$ ($toT_{min.0.30}$) recorded at the top of the tower of the cloud.
- $stT_{moyT_{min}BT.0.0}$ and $stT_{moyT_{min}BT.0.15}$ are the differences between the mean temperature at the basis of the cloud tower and the minimal temperature at the top of the cloud tower at T ($stT_{moyT_{min}BT.0.0}$) or at $T - \Delta t$ ($stT_{moyT_{min}BT.0.15}$).
- $stT_{moyT_{min}ST.0.0}$, $stT_{moyT_{min}ST.0.15}$ and $stT_{moyT_{min}ST.0.30}$ are the differences between mean temperature and the minimal temperature at the top of the cloud tower at T ($stT_{moyT_{min}ST.0.0}$), at $T - \Delta t$ ($stT_{moyT_{min}ST.0.15}$) or at $T - 2 \Delta t$ ($stT_{moyT_{min}ST.0.30}$).
- $stT_{sT_{moy}BT.0.0}$, $stT_{sT_{moy}BT.0.15}$ and $stT_{sT_{moy}BT.0.30}$ are the differences between the mean temperature and the threshold temperature at the basis of the cloud tower at T ($stT_{sT_{moy}BT.0.0}$), at $T - \Delta t$ ($stT_{sT_{moy}BT.0.15}$) or at $T - 2 \Delta t$ ($stT_{sT_{moy}BT.0.30}$).
- $stT_{sT_{moy}ST.0.0}$, $stT_{sT_{moy}ST.0.15}$ and $stT_{sT_{moy}ST.0.30}$ are the differences between the mean temperature at the top of the cloud tower and the threshold temperature at the basis of the cloud tower at T ($stT_{sT_{moy}ST.0.0}$), at $T - \Delta t$ ($stT_{sT_{moy}ST.0.15}$) or at $T - 2 \Delta t$ ($stT_{sT_{moy}ST.0.30}$).

Morphological variables

- $Q_{gp95BT.0.0}$, $Q_{gp95BT.0.15}$ and $Q_{gp95BT.0.30}$ are 95% quantiles of the gradients (degrees per kilometer) at the basis of the cloud tower at T , $T - \Delta t$ and $T - 2 \Delta t$.
- $Q_{gp95BT.0.0.15}$ and $Q_{gp95BT.0.15.30}$ are the change rates between $Q_{gp95BT.0.0}$ and $Q_{gp95BT.0.15}$ and between $Q_{gp95BT.0.15}$ and $Q_{gp95BT.0.30}$ (calculated at the preprocessing stage, from the original variable).

- Gsp95ST.0.0.15 and Gsp95ST.0.15.30 are the change rates between the 95% quantile of the gradients (degrees per kilometer) at the top of the cloud tower between T and $T - \Delta t$ and between $T - \Delta t$ and $T - 2 \Delta t$ (calculated, as a pre-processing, from the original variables).
- VtproT.0.0 is the volume of the cloud at T .
- VtproT.0.0.15 and VtproT.0.15.30 are the change rates of the volumes of the cloud between T and $T - \Delta t$ and between $T - \Delta t$ and $T - 2 \Delta t$ (calculated from the original variables as a pre-processing step).
- RdaBT.0.0, RdaBT.0.15 and RdaBT.0.30 are the “aspect ratios” (ratio between the largest axis and the smallest axis of the ellipse that models the cloud) at T , $T - \Delta t$ and $T - 2 \Delta t$.
- RdaBT.0.0.15 and RdaBT.0.15.30 are the change rates of the aspect ratio between T and $T - \Delta t$ and between $T - \Delta t$ and $T - 2 \Delta t$ (calculated, at the preprocessing stage, from the original variables).
- SBT.0.0 and SBT.0.30 are the surfaces of the basis of the cloud tower at T and $T - 2 \Delta t$.
- SBT.0.0.15 and SBT.0.15.30 are the change rates of the surfaces of the basis of the cloud tower between T and $T - \Delta t$ and between $T - \Delta t$ and $T - 2 \Delta t$ (calculated, at the preprocessing stage, from the original variables).
- SST.0.0, SST.0.15 and SST.0.30 are the surfaces of the top of the cloud tower at T , $T - \Delta t$ and $T - 2 \Delta t$.
- SST.0.0.15 and SST.0.15.30 are the change rates of the surfaces of the top of the cloud tower between T and $T - \Delta t$ and between $T - \Delta t$ and $T - 2 \Delta t$ (calculated, at the preprocessing stage, from the original variables).