Data Mining techniques to predict sewer condition

Guilherme Carvalho
guilherme.carvalho@ist.utl.pt

Instituto Superior Técnico, Lisboa, Portugal

June 2015

Abstract

The proper management of wastewater pipes is an important issue in modern society, with serious financial and sanitary implications. The ability to adequately prioritise maintenance inspections on these pipes may consequently significantly increase the quality of life of the affected populations. In this article, techniques to achieve this are studied. Using Data Mining procedures the goal is to be able to predict which pipes in a network are more likely to be close to failure. Predictions are done using random forests, forests of conditional inference trees, logistic regression and Naive Bayes. To have a more global view of the problem, as well as to be able to obtain better predictions, variable selection techniques are also studied and applied. Since it was found that only very rarely are pipes found to be in poor state, the examination of the problem extends to class balancing methodologies. Results show that there is no clear winning algorithm, although both ensembling and class balancing techniques manage to boost the performance of the tested algorithms.

Keywords: Classification, Wastewater pipes, Random Forest, Class imbalance

1 Introduction

1.1 Setting

Efficient management of water drainage infrastructures is a crucial part of an utility's duties. As these age they deteriorate becoming more liable to failure. Incapacity to detect malfunctions in sewer networks may incur in serious financial loss, decrease in the quality of life and even health hazards in the affected areas. Currently, the usual approach by many utilities to this problem is to perform routine maintenance using mostly basic data and expert knowledge to decide how to prioritise inspections. One of the maintenance procedures utilities use consists in conducting Closed Circuit Television (CCTV) inspections in the zone of interest. CCTV captures images along the pipes, which is extremely useful because most pipes have small diameters (under 1,20 meter) and man entry is not possible. These images will be later analysed by a qualified technician who will grade the observed pipes in terms of the severity, type and location of their damage according to some standard classification system. It is thus important to optimise decision making processes, a challenge more feasible as digital data becomes available and thanks to good practises of recording failures and system characteristics researchers are supplied with more data and tools to study the underlying causes and model the possible problems.

The overall goal of this work is to study ways in which statistical insight could help the decision making process. A database of pipes, their condition and characteristics relative to Gwinett County, Georgia, USA, is used as the base to construct prediction models. This data as well as the indispensable expert knowledge about wastewater systems was kindly provided by the Laboratório Nacional de Engenharia Civil (LNEC).

1.2 Literary review

The problem of predicting failures in sewer systems has been subject of extensive research. A brief review on some of the work done in these areas is provided next:

Salman (2010) uses both multinomial and binary logistic regression to predict sewer pipes' probability of failure. This is then combined with an estimate of the consequences of failure of each pipe to obtain what is called a risk ranking. Ana (2009) investigates forecasting both at group and pipe level, using cohort survival and semi-Markov models for the former and binary logistic regression, multiple discriminant analysis and probabilistic neural networks for the latter. Ens (2012) also studies the suitability of logistic regression for predicting the condition of sewer trunks, reaching however unsatisfactory results.

Harvey and McBean (2014) use the Random Forest algorithm to predict the probability of individual pipes being in poor condition, combining it with an analysis of the ROC curve to decide how to translate these probabilities into classifications. The Random Forest approach is also followed by Santos (2013) .

1.3 Article structure

This article is organised in five sections. The current serves as an introduction to the topic at hand. Section 2 outlines some of the models that will be used, starting with the basic elements of the Random Forest, decision trees, leading up to the Random Forest itself before analysing conditional trees, a similar yet distinct technique to their decision counterparts. Section 3 concerns variable selection. Here different techniques and their advantages and disadvantages are under the scope. Methods to deal with data imbalance in Data Mining problems are then reviewed in section 4. The article finishes with section 5 where the results and conclusions are presented.

2 Models

2.1 Decision trees

Decision trees are non-linear predictive models widely used for their performance and interpretability. They recursively partition the feature subspace by performing tests on the training set and dividing the data based on these. Decision trees can be used both for classification and regression tasks. They can be formalised as directed acyclic graphs that start at a vertex (vertices will from now on be called nodes, as it's common with decision tree terminology), called root and branch out until its terminal nodes called leaves. The intermediary nodes are referred as internals nodes. Each internal node has one incoming edge and at least two outgoing edges, the root has no incoming edge and the leaves have no outgoing ones. The decision tree can be seen as a sequential series of ”questions” about the data, each node being a question and the outgoing edges the possible answers. In a Data Mining context, given an observation identified by a set of features, these ”questions” are formalised as tests regarding the features, usually decision trees use only one feature at a time. These tests aim at separating different data points with respect to the response variable. After constructing a decision tree based on the training data, given a new observation one can predict its response value by travelling down the tree until reaching a leaf node. Here the predicted value can be taken as an average of the training points in this leaf in a regression context or as the majority vote in classification.

The challenge is then to construct the best decision tree possible from a given training set. Important components of this process are the kind of test being conducted at each node and when to stop growing nodes. Several algorithms have been proposed, like ID3 and C4.5 by Quinlan (1986), used for classification. Here, however, the focus will be on the classification and regression trees (CART) algorithm.

Introduced by Breiman, CART is a popular decision tree methodology and often used as the basic element of random forests. As the name suggests, unlike algorithms like ID3 and C4.5, CART can be used both for classification and regression. Starting at the root node, CART first looks at all the features in the data set and tests all possible binary splits. For continuous variables only ordinal splits, that is, splits dividing the points as greater or lower than a give threshold, are considered possible. After considering each feature and corresponding set of possible splits, CART chooses the pair (feature and split) that maximises the impurity reduction. There is more than way to capture this notion, usually CART uses the **Gini Gain**. To introduce this concept the **Gini diversity Index**, herein referred simply as Gini Index, is presented first, as defined by Raileanu and Stoffel (2000).

Definition 2.1. Gini Index Given a node t with points belonging to one of k classes, c_1, c_2,...c_k, its Gini Index is

$$G(t) = \sum_{i=1}^{k} \sum_{j=1, j \neq i}^{k} p(c_i|t)p(c_j|t) \qquad (1)$$

Where $p(c_i|t)$ is the probability of an element of t belonging to class i.

Definition 2.2. Gini gain
For a given node t and a split s diving it in children nodes t_L and t_R of proportions p_L and p_R respectively, the corresponding Gini gain is computed as:

$$\Delta G(s,t) = G(t) - p_L G(t_L) - p_R G(t_R) \qquad (2)$$

The larger the Gini gain, the larger the decrease in the class impurity after the partition, the better the feature that originated the split. Thus this criterion looks to maximise Equation (2) .

CART trees can be grown until there is no more impurity in a node, that will then become a leaf, or by establishing a minimum number of examples threshold, so that any node that has fewer cases than that becomes terminal. A popular practice to avoid overfitting is *pruning*, which consists on reviewing a built tree and turning subtrees into leaves if they fall under certain criteria.

2.2 Random Forests

In the context of a classification task, a common methodology is to search for the procedure that fits the data the most. Different procedures may have different performances, and even the same procedure may have varying results if its parameters or training features are changed. A popular method to achieve better performances is to combine several predictors in order to create a single one that surpasses any of its parts. This is called an ensemble. Perhaps the most widely known ensemble are random forests.

Developed by Breiman (2001), random forests are an ensemble of decision trees. The basic idea is to grow many trees based on the training set and do classification on a new example by feeding it to all the trees and take the majority vote. Notice how this can easily be turned into a ”probabilistic” result by taking, instead of the majority vote, the fraction of trees that voted for each alternative. There is, however, a bit more to Breiman's Random Forest than simply growing many trees (which will be considered to be CART trees). Firstly, bagging is employed, meaning each tree will not be grown on the original data set but on a bootstrap sample of it[1]. This helps building a more robust algorithm, as instead of using one random sample the bootstrap procedure mimics what would be like to draw many more from the underlying distribution. Bagging also creates an useful side effect in that for each tree there will be some points in the training set that aren't utilised to construct it. These points will be called out-of-bag for the tree in question and can be useful in the estimation of the random forest's error. The second major consideration is in the way the CART trees are grown. For each bootstrap sample, an unpruned CART tree is built, with one important difference from the procedure described earlier; at each node, instead of looking for the best split among all the variables, only a random subset of these is selected. This adds a second source of randomness to the forest. It enables the selection of variables that would

[1] A bootstrap sample from a set is simply a sample of the same size drawn with repetition from said set.

otherwise be systematically discarded but may have relevant interaction with others. While in practice there are a number of parameters to set in a random forest, namely the stopping criteria for the trees, the number of variables to be tried at each node and the number of trees grown, it is usually found that the forests are considerably robust and results tend to not deviate from the ones obtained with most implementations default values.

2.3 Conditional Inference Trees

While the overfitting problem can be solved by pruning, as seen before, there is yet another issue with the algorithms for recursive partition; the bias towards variables with many possible splits. To tackle this complication Hothorn *et al.* (2006) propose a new way to build trees, based on statistical properties of the variables, creating **conditional inference trees**. These deal with both the overfitting and the variable selection problems. Building a conditional inference tree is a procedure with two major steps;

1. **Step 1** Test the null hypothesis of independence between each of the covariates and the response variable. Stop if this cannot be rejected. Otherwise, pick the variable whose association with the response variable is the strongest.

2. **Step 2** Once having chosen a covariate, determine the best binary split and divide the data points according to it. Iterate steps 1 and 2 for both members of this split.

Notice how in this algorithm the choosing and splitting of the covariates are separate processes, avoiding the bias problem. There is also a statistical criteria to stop the algorithm, preventing overfitting without resorting to pruning.

2.3.1 Forest of conditional trees

Like with usual decision trees, it is possible to build forests using conditional inference trees, herein called conditional forest for simplicity. There are however a couple of notable differences when building the forests. First, the aggregation scheme does not count the votes of each tree, instead aggregating the observations from each leaf and from there computing a single predictor. Secondly, in the version used in this study and as recommended by Strobl *et al.* (2007) to obtain unbiased forests, the bootstrap samples are in fact just subsamples of the original with 0.632 of its size, that is, the expected number of different observations in a bootstrap sample. The reason to avoid samples with replacement is that even samples that are drawn from a variable that is independent from the response may show slight variation from this hypothesis. This effect is accentuated for variables that have more classes as the relative frequencies will tend to be lower and thus more volatile when resampled.

3 Variable selection

3.1 Introduction

Variable selection is an important part of any data analysis problem. It offers several benefits like: sparing the costs of monitoring unnecessary variables, making data visualisation easier, improving the algorithm's performance both at predictive and computational level by discarding irrelevant variables, helping tackle the curse of dimensionality when there are many features and few observations, making the model simpler thus less likely to overfit and being a source of information about the underlying causes of the phenomena being studied, for a more in depth analysis, Reunanen (2003) and Guyon and Elisseeff (2003) can be consulted. Variable selection methods can be divided in three groups, **filters**, **wrappers** and **embedded** selection, the first two of which will be briefly described in the following.

3.2 Filter methods

Filter methods aim at measuring the features relevance towards the response variable. These methods are used in the pre-processing stage and as such have no access to the model's information. This however can be an advantage as it makes these methods more generic since they're independent of model or their parameters. Another point in favour of this type of variable selection is that it tends to be faster than its counterparts, especially wrapper methods. It is important to bear in mind when sorting the covariates that there is no knowledge of any possible relationship between them. Examples of filter variable selection methods are Fisher's Score or simply the correlation coefficient between the predictor variable and the covariates. The most recent and popular filter method, however, is probably the mutual information criterion which is examined in more detail next.

3.2.1 Mutual Information

Mutual information measures the information shared by two variables. Put differently, mutual information can be seen as the decrease in uncertainty in one variable assuming knowledge of the other. In order to capture the concept of uncertainty one can use the notion of entropy.

Definition 3.1. Entropy Let X be a discrete random variable, its (Shannon's) entropy is defined as:

$$H(X) = -\sum_{x\in\mathcal{X}} p(x)log(p(x)) \tag{3}$$

Where p its probability mass function with support $\mathcal{X}$. In case X is a continuous random variable with probability density function f with support $\mathcal{X}$ its differential entropy is defined as:

$$H(X) = -\int_{x\in\mathcal{X}} f(x)log(f(x))dx \tag{4}$$

The base of the logarithms is usually 2 or e. Similar definitions could be given for the concepts of conditional and joint entropy by taking the respective mass/density functions.

While differential entropy seems a natural extension of the discrete case to continuous random variables it must be treated with caution as both notions do not share all properties. As an example, while Shannon's entropy is non-negative, the same is not necessarily true about the differential version. Approximations of these quantities must also be done with care, for example, taking a continuous variable, X, and dividing its values into bins of size Δ we obtain a discrete variable, X^{Δ}. It is not true, however, that $H(X^{\Delta}) \xrightarrow{\Delta\to 0} H(X)$, in fact, $H(X^{\Delta}) + log(\Delta) \xrightarrow{\Delta\to 0} H(X)$.

The mutual information (MI) between two variables is closely related to entropy, and is defined below, for the case where X and Y are discrete random variables

Definition 3.2. Mutual Information I
For discrete random variables X and Y with (joint) probability mass function, relative to the subscripted random variable(s), p, their mutual information can be defined as:

$$MI(X,Y) = \sum p_{X,Y}(x,y)log(\frac{p_{X,Y}(x,y)}{p_X(x)p_Y(y)}) \tag{5}$$

An extension of this notion for continuous and mixed random variables can be found in Cover and Thomas (2006) .

Definition 3.3. Mutual Information II

$$MI(X,Y) = \sup_{\mathcal{P}\ \mathcal{Q}} MI([X]_{\mathcal{P}}, [Y]_{\mathcal{Q}}) \tag{6}$$

Where $\mathcal{P}$ and $\mathcal{Q}$ are finite partitions of the random variables ranges and the quantisation of a variable X by a partition $\mathcal{P}$, $[X]_{\mathcal{P}}$, is a discrete random variable such that $P([X]_{\mathcal{P}} = i) = P(X \in P_i)$, where P_i is the $i-th$ component of the partition P.

Mutual information has several interesting properties, in particular it is useful to notice that

$$MI(X,Y) \geq 0 \tag{7}$$

$$MI(X,Y) = H(X) + H(Y) - H(X,Y) \tag{8}$$

$$MI(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) \tag{9}$$

A common practical problem associated with the usage of mutual information is that the variables' distributions are seldom known, meaning one can only estimate this quantity. As such, a popular method when dealing with discrete variables is to take their frequencies as the probabilities and start by estimating the entropy. This can be extended to deal with continuous variables, by first creating bins thus discretizing the data and then proceeding as for discrete variables (essentially using a quantisation of the variable of interest). The effect of taking the discrete entropy instead of the differential entropy in $H(X)$ is offset by doing the same in $H(X|Y)$ (see (8) above). Consequentially, the empirical estimator of $H(X)$ is

$$H_{emp}(X) = -\sum_{x=1}^{k} \frac{n_x}{N} log \frac{n_x}{N} \tag{10}$$

Here n_x denotes de number of sample points in bin x and N the total number of sample points. This estimation can be extended to the joint entropy of X and Y using in that case n_{xy}, the number of samples in both bins x and y. Once having a way to estimate the entropy it suffices to use (7) to estimate the mutual information.

3.2.2 Minimum Redundancy Maximum Relevance

Typically, when using mutual information to select features, these are ranked according to this criteria and the top m selected, for the investigator's choice of m. A potential problem, though, is that two features may be highly informative in regards to the response class but very similar between them, so it would be reasonable to only choose one. To account for this effect Ding and Peng (2003) suggest the Minimum Redundancy Maximum Relevance (mRMR) criteria, which bases its feature selection not only on the relevance towards the response variable but also in the similarity with the already chosen features. Formally this method tries to maximise the mutual information with the labels class, while keeping the pairwise mutual information between the selected features as low as possible. In the version used in this work, this is done by choosing at the $m-th$ step a feature X_k such that:

$$k : \arg\max_{i \notin S_{m-1}} \frac{MI(X_i, Y)}{\sum_{j \in S_{m-1}} MI(X_j, X_i)} \tag{11}$$

where S_m is the set of indexes of the first m selected features and Y the the response variable.

3.3 Wrapper methods

Wrapper methods of variable selection are dependent of the learning algorithm being employed, using it to choose the best subset of variables. An optimal (although prone to overffiting) approach would consist in evaluating all possible combinations of variables and determine which has the best results for the performance metric being used. This, however, is too computationally expensive in practice (Amaldi and Kann (1993) proved it to be NP hard), so other techniques are favoured, in particular forward and backwards search. The forward search starts from an empty set of variables and at each iteration finds the variable whose addition to the model will result in the largest improvement, the backwards case is analogous but starts with the set of all variables and proceeds by removing them one by one, again looking to maximise the performance criterion. While not an exhaustive search, Guyon and Elisseeff (2003) suggest this doesn't necessarily entail a loss in predictive power, and may even help prevent overfitting according to Reunanen (2003). Another device used to avoid favouring models that are too complex is to add a penalty for each feature used, Kohavi and John (1997), for example, add a 0.1% penalty for every feature when using a five-fold cross-validation scheme for forward search variable selection.

4 The imbalance problem

4.1 Introduction

Class imbalance is a frequent problem in classification tasks. This happens when the distribution of the response variable is highly skewed, that is, one or more classes are very rare compared to others. This is a common occurrence in fraud detection, medical diagnoses, ad-clicking patterns and many other settings. While this problem may be present in multi-class tasks most studies focus on the two class context, as will be done here. Also, following common terminology, the majority class will called the negative examples while the minority one will be dubbed as the positive examples. The problem with having extremely disproportionate classes is that many classification algorithms are not prepared to deal with this scenario, using general performance metrics such as the accuracy to guide them in the learning process. The issue here is that in an imbalance situation, a measure like accuracy is usually maximised by the trivial solution of classifying all points in the majority class. The use of such metrics is not only a problem in the learning stage of the algorithm but also when trying to determine it's usefulness. Notice that in most class imbalance problems the positive class is the one of interest, and that should be reflected in the performance measure. Since the problem examined in this work suffers from considerable class imbalance some measures were taken in an attempt to both maximise the different algorithms performance and create intelligible performance metrics.

4.2 Performance metrics

As mentioned before, some conventional metrics do not translate the predictive power of an algorithm well in the imbalance context. With this in mind measures known to be robust relative to this problem will be used. Important examples are metrics like sensitivity, the probability of an example being classified in the positive class if it belongs to it, and the specificity, the analogous for the negative cases. Notice that that the trivial classifying methodology discussed before would have a perfect specificity of one, but a sensitivity of zero as no examples are considered positive. It is thus desirable to use these quantities in tandem in the search for the best model. Another idea is to capitalise on the fact that the classifiers examined can actually estimate probabilities instead of just outputting blunt classification. A metric that takes advantage of this is the cumulative gain. This measure plots the number of examined examples (ordered by the probability of being positive) versus the number of truly positive examples. Point (x, y) in this curve means that the $x\%$ of examples more likely to be positive contain $y\%$ of all actually positive examples. The area under the cumulative gain curve is going to be the major model assessment measure throughout.

4.3 Tackling the data imbalance

Once having settled on a metric that is not influenced by the class imbalance, it is important to take the necessary steps to make the models as robust as possible to it. Methodologies to face the problems of class imbalance have been focus of intense study in the machine learning community. Pozzolo *et al.* (2013) divide these into four groups: sampling, ensemble, distance-based and hybrid.

- **Sampling** techniques are the most traditional to deal with class imbalance. In their most basic form there are two alternatives, oversampling and undersampling. Oversampling consists in drawing examples from the positive class, with repetition, until both classes have equal frequency. Undersampling takes a different approach, focusing instead on the majority class and discarding instances until it reaches the size of the minority one. While both these cases have been found useful they carry some negative effects too. Undersampling may overlook important information by not using all the examples while oversampling can increase the computational burden significantly. Another problem with oversampling is that it has been found to bias the classifier towards the positive examples that it replicates. SMOTE , Chawla *et al.* (2002), is a popular oversampling method that instead of merely replicating existing examples from the minority class interpolates them to create new, synthetic ones.

- **Ensemble** methods couple attempts to balance the data with the classifier being used, examples are *EasyEnsemble* and *BalanceEnsemble*. Both developed by Liu *et al.* (2006), the former iteratively creates balanced subsets by undersampling and trains several weak classifiers with AdaBoost, combining all for the final result, while the latter takes a similar approach but with a boosting methodology, removing correctly classified instances of the majority class at each iteration.

- **Distance-based** methodology use the position of the examples to decide which may be noise or borderline to their class. For instance, the *Condensed Nearest Neighbour* rule finds a subset of the training data that correctly classifies all examples in the whole training set (with the 1-nearest-neighbour classifier) and uses it to train. The aim is to discard negative instances that are away from the border. Alternatively, one can try to eliminate the examples in the border region to have a clearer division between the classes. To this end *Tomek links* are defined to be the pairs of examples in different classes that are closer to each other than to any other instance. Majority points that are Tomek links are then removed.

- **Hybrid** approach consists in the combination of methods belonging to the other techniques.

5 Results

5.1 Data overview

In this section the previously studied approaches are tested on a real-case dataset. This dataset consists on a collection of 35692 observations over 32264 wastewater pipes from Gwinnett County, Georgia, USA. Several characteristics relative to the pipes have been recorded as was their working condition. The idea is then to apply Data Mining techniques to be able to predict whether a pipe is in good state from its previously known characteristics. The pipe data used to model this problem, prior to variable selection, consists of the following variables; **Length**, **Age**, **Diameter**, **Slope**, **Material**, **Upstream Depth** (US_Depth), **Downstream Depth** (DS_Depth), **Basin** and **Zone**. The binary response variable will take the value of 0 for working pipes and 1 otherwise, pipes with label 1 will be called critical. Only 2226 of the 35692 observations resulted in a critical evaluation, underlining the (fortunate) rarity of these. Next the variable selection techniques discussed are used to determine a set of covariates for the problem.

5.2 Variable selection

5.2.1 Mutual Information

The following results concern the mutual information criteria for variable selection. The results were obtained using the methods described before, the mutual information was divided by the entropy of the response variable, thus becoming bounded between 0 and 1, to give a better intuition of the features' usefulness. For this effect there was also another variable added to the analysis, "Random", drawn from a standard normal distribution. This served as a benchmark to compare the potential explanatory variables against.

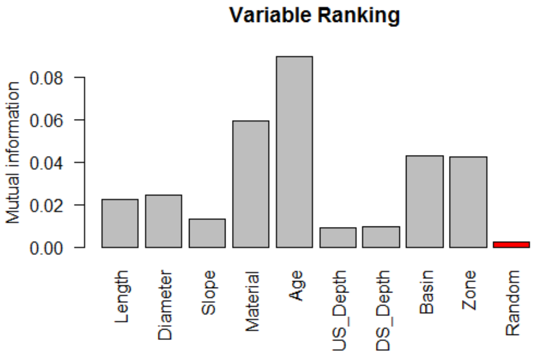

Figure 1: Mutual information results

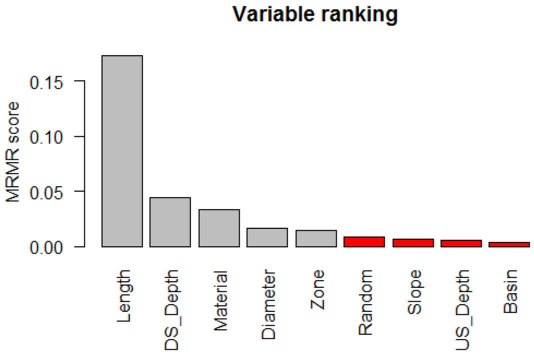

Figure 2: mRMR:Variables selected after "Age"

The graphs above detail the mutual information and mRMR gains of the variables. Figure 1 shows that ”Age” and ”Material” are the most important variables using mutual information, followed by the the locality-based ”Zone” and ”Basin”. The mRMR approach, however, suggests that while the ”Age” may be the most valuable feature, after having chosen it, the importance of the ”Material” and location-based variables becomes less significant. This may imply these variables share considerable information.

Considering the stepwise search methodology, Table 1 describes the results obtained for the backwards version of the algorithm. The classifier used were random forests and the performance metric the area under the cumulative gains curve, simply called ”Score” henceforth.

Table 1: All variables score: 0.5086

Variable\Iteration	1	2	3	4	5	6	7
Age	0.4223	0.4154	0.4104	0.3450	0.3217	0.2575	0.1665
Slope	0.5084	0.5046					
Diameter	0.5060	0.5011	0.4950	0.4780			
Material	0.5047	0.5040	0.4963				
DS_Depth	0.5093	0.5030	0.4903	0.4775	0.4583		
US_Depth	0.5074	0.5040	0.4949	0.4752	0.4507	0.4067	
Length	0.4807	0.4801	0.4664	0.4483	0.4186	0.3587	0.1177
Zone	0.5106						
Basin	0.5077	0.4759	0.4658	0.4449	0.4216	0.3920	0.3404

Table 1’s entry (i, j) corresponds to the score obtained from the random forest in the $j - th$ iteration after removing variable i. A blank space means this variable has already been removed and so didn’t make it to this iteration. The importance of the ”Age” variable is underlined by the heavy drop in scoring power when it is removed. On the other hand its clear that ”Basin” and ”Zone”, two geographical variables, may not be both necessary but once one is removed the other becomes considerably important.

Taking into account both variable selection procedures, the variables chosen were **”Age”**, **”Diameter”**, **”Material”**, **”DS_Depth”**, **”Length”** and **”Basin”**. Notice this corresponds to removing ”US_Depth” in the third iteration of Table 1, for a score of 0.4949. This is not the best score possible in this table, but due to the overfitting nature of the wrapper methodology and variance in the data, a model with less variables was favoured. The reason to pick ”US_Depth” to be discarded instead of ”Material” were the good results the latter showed in Figures 1 and 2 and the fact that both depth variables seem somehow similar, as the results gotten from removing one do not differ by much from the corresponding scores after removing the other, in Table 1.

5.3 Models results

The results of the models studied are presented next. The training methodology was a simple $80\%\backslash 20\%$ division of the data in train and test sets, respectively. Besides the algorithms studied in section 2, two others were used, Firth regression (a bias corrected version of the logistic regression) and Naive Bayes. One thing to have in mind, though, is that a random forest was used as part of the variable selection which may introduce some bias towards this classifier. Data imbalance techniques were also considered, namely undersampling, oversampling and a different approach, here called *BalanceM*. The latter, similar to *EasyEnsemble* and the not described Balanced Random Forest (see Chen *et al.* (2004)), partitions the negative examples of the data set in several pieces of the same cardinality as the positive set and uses each to train, coupled with all the minority observations. It then aggregates all the predicted probabilities using the average. Finally a benchmark measure is provided, in the form of the ”Age Ranking”. Unsurprisingly this corresponds to the results obtained following the popular methodology of simply ordering the pipes in terms of probability of being critical from oldest (more probable) to youngest (less probable).

	Original data	Undersampling	Oversampling	*BalanceM*
Age Ranking	0.4240	-	-	-
Random Forest	0.5001	0.5011	0.4743	0.5073
Conditional Forest	0.4959	0.5025	0.4657	0.5125
Firth Regression	0.4969	0.4958	0.4960	0.4969
Naive Bayes	0.4661	0.4670	0.4656	0.4662

Table 2: Score table for the different algorithms

As Table 2 summarises, there are great benefits to using the algorithmic approach compared with the more standard age ranking. Looking at the results using the original data, the random forest had the best score, followed by Firth's regression and the conditional forest. Naive Bayes is clearly behind the other alternatives but still improves on the age ranking. The results of resorting to class balancing techniques are mixed. Oversampling deteriorates scores for all algorithms but undersampling boosts the performance metric for both forest methodologies and Naive Bayes, although the latter is still far from being competitive. The best scores, however, come from the *BalanceM* methodology. This leads to the best results both for random and conditional forests. The latter edges out the former, but it should be said that in practical terms it may be significantly more expensive computationally. Figures 3 and 4 provide a visual intuition of these scores.

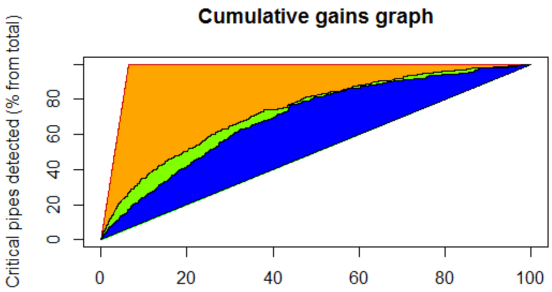

Figure 3: Random Forest *BalanceM*

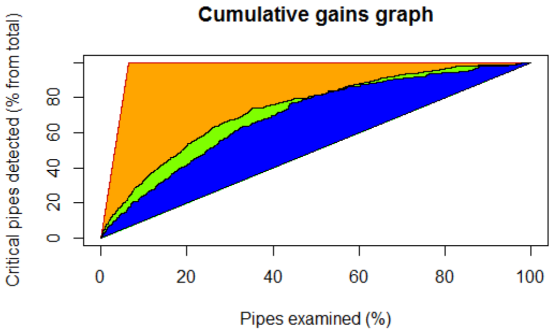

Figure 4: Conditional Forest *BalanceM*

Three curves can be identified in the graphs. The one delimiting the orange region corresponds to a perfect classifier, unfeasible in pratice. The one above the green area is the performance of the algorithm in question. Lastly the curve defining the blue portion is relative to the age ranking. It can be seen in both figures that not only are there substancial gains from the procedures but they tend to happen in the early stages of the curve. This is desirable since in real cases only the pipes most likely to be critical are examined.

A final option explored was the use of ensembles of these classifiers. Notice that the forests are already ensembles and *BalanceM* is one also. Another use of this technique is to take the probabilities from the algorithms in Table 2 and use them as extra features. The two questions to answer now are what probabilities to add to the features and what algorithm to use them on. After extensive testing, that will not be reproduced here, the best score came from the Firth regression, using the previously chosen variables as well as Firth regression and random forest predictions. This resulted in a score of 0.5267 with the corresponding graph below:

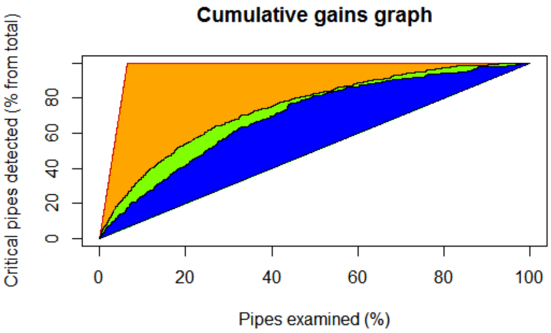

Figure 5: Ensemble cumulative gains graph

As it can be seen, the ensemble approach is vastly superior to the traditional one. In particular, after the examination of the 20% most likely to be critical pipes, the algorithmic methodology seem to have detected about half the critical pipes against 40% in the age ranking.

5.4 Conclusion

In this article the problem of predicting the condition of wastewater pipes was analysed. Data Mining techniques were used in an attempt to improve current practices. Filter and wrapper forms of variable selection helped single out the most relevant features and data balancing methodologies proved to be an advantage given the nature of the problem. In the end, however, a stacked ensemble showed to be a better classifier, for the test set used, than balanced alternatives. Further studies should be conducted, though, before taking any strong conclusions about which is stronger in general.

Acknowledgements

The author would like to thank Professor Conceição Amado and Doctor Rita Brito for their availability and precious advices.

References

Amaldi, E. and Kann, V. (1993). The complexity and approximability of finding maximum feasible subsystems of linear relations. *Theoretical Computer Science* **147**, 181–210.

Ana, E. (2009). Sewer asset management - sewer structural deterioration modeling and multicriteria decision making in sewer rehabilitation projects prioritization. *Ph.D. thesis, Vrije Universiteit Brussel.*.

Breiman, L. (2001, October). Random forests. *Mach. Learn.* **45**(1), 5–32.

Chawla, N. V., Bowyer, K. W., Hall, L. O. and Kegelmeyer, W. P. (2002, June). Smote: Synthetic minority over-sampling technique. *J. Artif. Int. Res.* **16**(1), 321–357.

Chen, C., Liaw, A. and Breiman, L. (2004). Using Random Forest to Learn Imbalanced Data. Technical report, Department of Statistics, University of Berkeley.

Cover, T. M. and Thomas, J. A. (2006). *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)* (2nd ed.). Wiley-Interscience.

Ding, C. and Peng, H. (2003). Minimum redundancy feature selection from microarray gene expression data. In *J Bioinform Comput Biol*, pp. 523–529.

Ens, A. (2012). A framework for deterioration modelling development in infrastructure asset management. *Master's. thesis, Department of Civil Engineering University of Toronto.*.

Guyon, I. and Elisseeff, A. (2003, March). An introduction to variable and feature selection. *J. Mach. Learn. Res.* **3**, 1157–1182.

Harvey, R. R. and McBean, E. A. (2014). Predicting the structural condition of individual sanitary sewer pipes with random forests. *Canadian Journal of Civil Engineering* **41**(4), 294–303.

Hothorn, T., Hornik, K. and Zeileis, A. (2006). Unbiased recursive partitioning: A conditional inference framework. *JOURNAL OF COMPUTATIONAL AND GRAPHICAL STATISTICS* **15**(3), 651–674.

Kohavi, R. and John, G. H. (1997). Wrappers for feature subset selection. *ARTIFICIAL INTELLIGENCE* **97**(1), 273–324.

Liu, X., Wu, J. and Zhou, Z. (2006). Exploratory under-sampling for class-imbalance learning. In *Proc. Int,l Conf. Data Mining*, pp. 965–969.

Pozzolo, A. D., Caelen, O., Waterschoot, S. and Bontempi, G. (2013). Racing for unbalanced methods selection. In *IDEAL*, Volume 8206 of *Lecture Notes in Computer Science*, pp. 24–31. Springer.

Quinlan, J. R. (1986, March). Induction of decision trees. *Mach. Learn.* **1**(1), 81–106.

Raileanu, L. E. and Stoffel, K. (2000). Theoretical comparison between the gini index and information gain criteria. *Annals of Mathematics and Artificial Intelligence* **41**, 77–93.

Reunanen, J. (2003, March). Overfitting in making comparisons between variable selection methods. *J. Mach. Learn. Res.* **3**, 1371–1382.

Salman, B. (2010). Infrastructure management and deterioration risk assessment of wastewater collection systems. *Ph.D. thesis, Department of*

Civil and Environmental Engineering, University of Cincinnati, Ohiol..

Santos, P. (2013). Decision support tools for urban drainagesystem management. *Master's. thesis, Instituto Superior Técnico*.

Strobl, C., Boulesteix, A.-L., Zeileis, A. and Hothorn, T. (2007). Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics*.