Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 169 (2020) 851–859

Procedia Computer Science

www.elsevier.com/locate/procedia

Postproceedings of the 9th Annual International Conference on Biologically Inspired Cognitive Architectures, BICA 2018 (Ninth Annual Meeting of the BICA Society)

Sensing the Web for Induction of Association Rules and their Composition through Ensemble Techniques

Agnese Augello[a], Ignazio Infantino[a], Giovanni Pilato[a], Filippo Vella[a]

[a] *ICAR-CNR, Institute for High Performance Computing and Networking, National Research Council of Italy Palermo, Italy*

Abstract

Starting from geophysical data collected from heterogeneous sources, such as meteorological stations and information gathered from the web, we seek unknown connections between the sampled values through the extraction of association rules. These rules imply the co-occurrence of two or more symbols in the same representation, and the rule confidence may vary according to the collected data. We propose, starting from traditional algorithms such as FP-Growth and Apriori, the creation of complex association rules through boosting of simpler ones. The composition enables the creation of rules that are robust and let emerge a larger number of interesting rules.

Keywords: Association Rules; Web Sensing; Emergency; Big Data; Boosting; Ensemble

1. Introduction

The diffusion of sensor networks for environmental monitoring and the development of knowledge discovery techniques in databases has allowed scientists and engineers to focus more and more on data-driven discovery while modeling their domains of interest. Nowadays, the Big Data paradigm allows the use of commodity hardware to process huge amounts of data that are collected in large areas and during a large time span and extract from them unknown connections.

Alongside with this approach, the weather risk assessment and prevention can be achieved through the extraction of association rules linking two or more different types of weather data. An example of such rule is: “if there is heavy

* Corresponding author. Filippo Vella Tel.: +39-091-803-1058
E-mail address: name.surname@icar.cnr.it

10.1016/j.procs.2020.02.152

rain, there is also a high river level". Another example may regard the specification of which level of measurement for a weather variable determines an emergency situation, such as "if there is heavy rain and there is a high river level, there is an emergency".

Association rules algorithms have been used in this context in several works. For example in [16] to assess and correlate weather conditions and their effects on wind turbine (WT) failures, while in [15] they have been exploited to discover hidden rules in time series climate data.

In [4] and [5], we exploited pattern mining techniques to find co-occurrence relationships of a relevant situation such as risky events.

In these works single techniques are used to obtain association rules. Instead, the novelty of our work is putting together rules from multiple algorithms by means of a standard boosting approach to let the generation of stronger association rules. The motivation lies in the consideration that through a boosting approach, which is a general method to build classifiers combining less accurate classifiers, it is possible to improve the performances of single classifiers. Therefore we expect that also the performances of the association rules generation can be improved, obtaining a new set of rules, from rules obtained with two methods.

The two sets of rules are extracted separately, by applying two seminal algorithms for association rules extraction. Specifically, we used the Parallel FP-Growth (PFP) algorithm [12], that is a parallel version of the well known FP-Growth [10] and the Apriori algorithm [2].

As first source of association rules we used an Apache Mahout™implementation of the Parallel FP-Growth (PFP) through the Hadoop framework. Hadoop is an open source environment initially inspired by Google's MapReduce framework that has become the *de facto* standard for storing, processing and analyzing Big Data.

As second source of association rules we used an R implementation of the Apriori algorithm by [2]. Apriori algorithm generates different itemsets with a level-wise approach. It extracts patterns with increasing cardinality. It usually repeatedly scan the database to count the support of each pattern. FP-Growth is generally faster since it scans the database avoiding to generate all the candidates. To reduce the cost of database, FP-growth adopts projected database to reduce the set of explored set in a depth-first search manner.

The whole set of extracted rules is the input to the AdaBoost algorithm, first proposed by [7]. Beyond using the sets of rules separately, we also combined them to evaluate whether the boosting on the enlarged can produce stronger classifiers. The sets of the rules generated by the two methods have a non-zero intersection since some rule is extracted by both methods. In the same way, the intersection does not coincide with either of the two sets. That is some rule extracted by Apriori is not extracted by FP-Growth and some rule that is extracted by FP-Growth is not extracted by Apriori. Experimentally we had this evidence and the limited resources, in time and space, allocated to the single methods create these disparities between the extracted rule sets.

We tested the technique on freely available measurements by Servizio Idrogeologico Regionale della Toscana (SIR), gathered by heterogeneous sensor networks. Furthermore, we collected emergency information on the monitored region that were indexed by search engine in on-line newspapers and weblogs from 2012 to 2014. We have compared these values with the emergency detection in the same region along the same years, with promising results.

The paper is organized in the following way: in section 2 we present a survey on frequent pattern mining; in section 3 an overview of boosting is presented; in section 4 we show how we put multiple rules together, and in section 5 data for this task are shown in detail. Finally, section 6 shows the results for the chosen dataset.

2. Frequent pattern mining

Frequent pattern mining is the task aimed at finding relationships among the items in a database D. The problem was firstly introduced in [1], given a set of items $I = \{a_1, a_2, ..., a_n\}$ a collection of transactions $T_1 \dots T_N$ is stored. A transaction t_j is a set of items of I, $t_j \subseteq I$. A well known example is composed of market baskets: each item a_i corresponds to an item available in a superstore, a transaction is the set of items bought by a client. Representing the transaction as a vector, with dimension equal to the number of items, a binary value represents whether the item is present or not in the transaction. The pattern mining determines the patterns P that are present in at least a fraction of the transactions. An interesting pattern, for the market example, is whether two or more items are frequently bought together. The aforementioned approach has successfully been applied to several other applications in the context of data mining since then.

A set $P \subseteq T_i$ is called an l-sized *itemset* if the number of items it contains is l. It has a support $supp\,(P_i) = \frac{|P_i(t)|}{|D|}$ that is the ratio of transactions in D containing P.

The itemset P will be deemed *frequent* if its support is equal to, or greater than, a given threshold minimal support. An association rule is the implication $X \Rightarrow Y$, where X and Y are subset of P that do not intersect.

An evaluation on the validity of each rule can be performed using several quality measurements, among which we find the support and the confidence of rule, where, among which we find:

- the support of a rule $X \Rightarrow Y$, that is the support of $X \cup Y$ and states the frequencies of occurring patterns;
- the confidence of a rule $X \Rightarrow Y$, indicated as $conf\,(X \Rightarrow Y)$, that is defined as the ratio $\frac{supp(X \cup Y)}{supp(X)}$, states the strength of implication.

Beyond the traditional measures, we also considered the Rule Power factor, that has been proposed recently and is evaluated as $conf \cdot supp$. According to the authors, the Rule Power factor produces significant results even when other measures fail[11] and is inspired by the three principles of interesting measures proposed by Piatetsky-Shapiro [14]. Given a minimal support s_{MIN} and minimal confidence c_{MIN} by users or experts, $X \Rightarrow Y$ is considered as a valid rule if both $supp\,(X \Rightarrow Y) \geq s_{MIN}$ and $conf\,(X \Rightarrow Y) \geq c_{MIN}$.

The techniques that detect association rules in a database are usually composed by a task that spots the frequent itemsets, that are all the itemsets with a support higher than a threshold, and a second task that discovers the association rules between the found itemsets.

2.1. PFP: The FP-Growth Algorithm in a parallelized environment

MapReduce is a framework for processing parallelizable problems across datasets and has been presented in [6]. The procedure uses a large number of inter-connected computer systems, called *worker nodes*, which take advantage of locality of data to reduce transmission distances. It is often used with commodity hardware.

While a master node manages the exclusive use of a redundant copy of input data and coordinates error recovery, each worker processes a tiny part of a group of actions defining a complex task, identified by a `key`, during the **Map** stage. After an intermediate **Shuffle** stage, where it is ensured that all data belonging to one key is kept on the same worker node, during the **Reduce** stage each worker processes each group of output data, per key, in parallel, and in turn produces a collection of values in the same domain.

The FP-Growth Algorithm, a *divide et impera* algorithm that extracts frequent patterns by pattern fragment growth proposed by [10], has been adapted to the MapReduce framework by [12]. Apache Mahout™implements it with some slight optimization.

Given a transaction database D, the three MapReduce phases used to parallelize FP-Growth can be outlined as follows:

1. **Sharding**: D is divided into several parts, called *shards*, stored on P different computers.
2. **Parallel Counting**: The support values of all items that appear in D is counted, one shard per mapper. This step implicitly discovers the items vocabulary I, which is usually unknown for a huge D. The result is stored in a frequency list.
3. **Grouping Items**: Dividing all the $|I|$ items on the frequency list into Q groups. The list of groups is called group list (*G-list*), where each group is given a unique group identifier (gid).
4. **Parallel FP-Growth**: During the map stage, transactions are rearranged using the groups defined in the previous step: when all mapper instances have finished their work, for each group-id, the MapReduce infrastructure automatically gathers every group-dependent transaction into a shard. Each reducer builds a local FP-tree and recursively grows its conditional FP-trees, returning discovered patterns.
5. **Aggregating**: The results generated in Step 4 are coalesced into the final FP-Tree.

2.2. Apriori

While FP-Growth is one of several algorithms using optimized data structures to store candidate itemsets or to compress the dataset, the Apriori algorithm seeks to reduce the number of potential candidates without counting their support values by using *support-based pruning*: it can be shown that if an itemset is frequent, then all of its subset must also be frequent; conversely, if an itemset in infrequent, then all of its supersets must be infrequent.

In Apriori, every item is initially considered a candidate 1-itemset; every candidate whose support is lower than the requested support threshold is discarded. In the next iteration candidate 2-itemsets are generated combining 1-itemsets only with others having higher support. This process is repeated until no k-itemsets can be formed from $(k-1)$-itemsets.

Candidate generation and support counting are probably the most computationally intensive parts of the algorithm: while too many unnecessary candidates are to be avoided, it must also be ensured that the candidate set is complete and without duplicates. Moreover, comparing itemsets and transactions may result in bad performance when the numbers of transactions and candidate itemsets are large, so alternative strategies, such the use of a hash trees, may be used.

After itemsets have been generated, rules can be extracted. The Apriori algorithm uses a level-wise approach for generating association rules. First of all, all the high-confidence rules that have only one item in the rule consequent are extracted, and those are used to generate new candidate rules by merging the consequents of any two chosen rules. For more information, see [17].

3. Construction of a robust classifier through Boosting

The word *boosting* refers to a general method of classifier production that combines less accurate classifiers to form more accurate ones. Boosting assumes the availability of a so-called "weak learning algorithm" which, given labeled training examples, produces several basic classifiers. The goal of boosting is to improve its performance while treating it as a "black box", which can be called repeatedly, but whose innards cannot be observed or manipulated: it is only assumed that the error rates of those classifiers are at least lower than a classifier whose every prediction is a random guess.

We have chosen to improve the performance of our association rules using the AdaBoost meta-algorithm, first proposed by Freund and Schapire in [8]. Our AdaBoost implementation loosely follows [19]. A further example of the profitable employment of the boosting technique is presented in [18].

AdaBoost takes as input a set of training examples $(x_1, y_1), \ldots, (x_m, y_m)$ where each x_i is an instance from X,and each y_i is the associated label or class: in this work $y_i = 0$ for negative examples, $y_i = 1$ otherwise. We repeat the weak classifier training process exactly T times.

At each iteration $t = 1, \ldots, T$ a base classifier $h_t : X \in 0.1$ is extracted. We attempt to choose a weak hypothesis with low weighted error $\epsilon_t = \sum_i w_i \left|h_j(x_i) - y_i\right|$.

Each weak hypothesis is thus assumed to be slightly better than random guessing by some small amount, as measured by its error. Once the base classifier h_t has been received, a parameter α_t measuring the importance that is assigned to h_t is calculated. As $\alpha_t > 0 \iff \epsilon_t < 1$, the more accurate the base classifier h_t is, the more importance we assign to it.

The weight distribution is then updated as to give more prominence to hard-to-classify examples:

$$w_{t+1}, i = w_t \beta_t^{1-e_i}$$

where $e_i = 0$ if example x_i has been correctly classified, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.

The final robust classifier is:

$$H(x) = \begin{cases} 1 \text{ if } \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t; \\ 0 \text{ otherwise.} \end{cases}$$

where $a_t = \log \frac{1}{\beta_t}$.

4. Boosting of association rules

On a theoretical level, association and classification rules approach the problem of record space exploration from two different points of view: in the words of [13] "association rules aim to find all rules above the given thresholds involving overlapping subsets of records, whereas decision trees find regions in space where most records belong to the same class".

Association rule extraction algorithms produce, on average, many rules since they do not repeatedly partition record space in smaller subsets and they search for sets of items that are present together in the samples of the dataset. This characteristic means that association rules can be very granular, and their extraction algorithms are generally slower. Anyway, a balance between granularity and performance may be found imposing support and confidence thresholds on itemsets. It turns out that association rules can be used as classifiers if a discretization of the attribute space is performed and the established bins can serve as feature sets. Association rule mining can be thus applied to find patterns of the form $< featuresets >\Rightarrow ClassLabels$, ranking rules first by confidence, and then by support. A similar approach is shown in [20], while we follow a similar steps, we diverge in several aspects:

- as we use a Big Data approach, we can leverage much more computing resources than those available to them, so we do not limit the order of generated rules in advance, but rather perform boosting on the whole set of generated rules. This allows a "black box" approach to the association rule extraction algorithm, that can be replaced without altering other parts of our system;
- a correct weak classification occurs if both or neither of antecedent and consequent are present;
- following Viola-Jones' approach in [19], we penalize the weight only if the error rate of a given weak classifier is lower than 50%: by contrast, Yoon and Lee use fixed update value of β for confidence values making their approach not adaptive

5. Information representation

5.1. Dataset features

The Tuscanian datasets used in this work has been made available by Servizio Idrogeologico Regionale della Toscana (SIR) [1]. Their sensor and surveillance network, spanning the entire surface of Tuscany, can provide both real-time and historic samples from hydrometric, pluviometric, thermometric, hygrometric, freatimetric and mareographic sensors, allowing a general characterization of hydroclimatic phenomena. Generally, stations in a sensor network are placed in a way that ensures optimum coverage of a given region: different restrictions due to the domain of interest and regulations already in force when considering the placement need to be taken into account, so any two given network may have very different topologies. Given a station, relevant neighbors belonging to the other networks must be found. In this work, we group values using concentric circles having radiuses $r_1 = 25km, r_2 = 50km, r_3 = 75km$ centered on basin stations, as they constitute the sparsest network among those managed by SIR.

An outline of the data transformation steps we perform follows:

1. *Per-network grouping*: As every station stores a small subset of data, each station is polled by a central facility at regular intervals. SIR provides a single file for each station in a given network. For our convenience, a single table is created for gathering data coming from all the stations in the same network;
2. *Discretizations*: Each sensor measure is replaced with a vector representation with seven bins corresponding to seven range values. This quantized representation is needed since the rule extraction algorithms extract connections among recurring symbols.
   For rain values our discretization mirrors the 7 classes SIR uses to classify rain values according the daily quantity of rain, with these thresholds - C1: <1 mm; C2: from 1 to 10 mm; C3: from 10.1 a 20 mm; C4: from 20.1 a 30 mm; C5: from 30.1 a 40 mm; C6: from 40.1 a 50 mm; C7: ≥50 mm. The values of the variation of the basins levels and the phreatic measures are discretized evaluating the standard deviation of the values during a period of time (e.g. three days) and classifying the deviation according seven bins, in analogy with the rain classes. Other quantization techniques could be employed to reduce the quantization error according the bin selection frequency. Furthermore, a second source of background knowledge could be used to operate a supervised binning.

[1] http://www.sir.toscana.it/

3. *Basket arrangement and emergency binding*: The output of the discretization process must be converted to a transactional format for use with the association rules extraction algorithm. The input file for Apache Mahout will be a boolean matrix with as many rows as the number of the samples as much columns as there are discretized bins for every measure we take into account. Given a row vector r_1, there will be a `TRUE` or `1` value in (r_1, B_k) if the discretized value is k, and `FALSE` or `0` otherwise.
   Exactly $ab = 21$ columns are required to store the other kind of measures. An emergency flag is set if for a given date the basin station was near enough to dangerous phenomena. The bins with label B are referred to the quantized value of the basins; the values of P_i are referred to the rain value (in italian *Pioggia*) coupled the the R_i values according the quantized distance from the basin. The composition of the values for the samples of the meteorological stations and the emergency information create a $t-$uple as the one shown below.
4. *Inverse mapping*: Apache Mahout™requires transactions items to be expressed using integer keys, so we map the column names in the basket arrangement made in the previous step, keeping trace of the mappings to the original items to properly present results in the output study phase.

5.2. Emergency information

While sensor networks provide numerical values, quantitative figures with a certain degree of reliability, they do not convey information about emergencies in itself, but we need this information in order to train a classifier that can identify situation that are interesting for potential emergential situations.

We assume when an emergency occurs traces of it can be found in the World Wide Web by means of online newspaper articles or even posts on a personal blog: relevant content may hopefully contain a word in the set A of words describing the phenomenon, and another one in the geocoded set B of Tuscanian cities. The set A is formed by key words about hydrogeological emergencies such as : *esondazione*, *violento temporale*, *diluvio*, *allagamento*, *inondazione*, *rovinosa tempesta*, *violento acquazzone*. The set B is formed by the names of the cities in the Tuscany region such as : *Firenze*, *Pisa*, *Livorno*, *Grosseto*, *Lucca*, *Siena*, *Massa*, *Carrara*, *Pistoia*.

We have therefore queried the Bing™search engine, through its Search API, using keywords in the set given by the Cartesian product of the set A for the set B, $C = A \times B$.

In the end, we used the subset of search results that employed so-called "pretty" URLs, in particular those bearing day, month and year information separated by forward slashes, as they are less likely to inadvertently get altered after publication. This subset has been finally filtered by visual check h to remove spurious and incomplete data. We are unable to presume that some of the remaining information has not been altered by the content authors, either willingly or because of an error: moreover, this caveat is to be considered for every automatically collected data where a second, trusted source is unavailable.

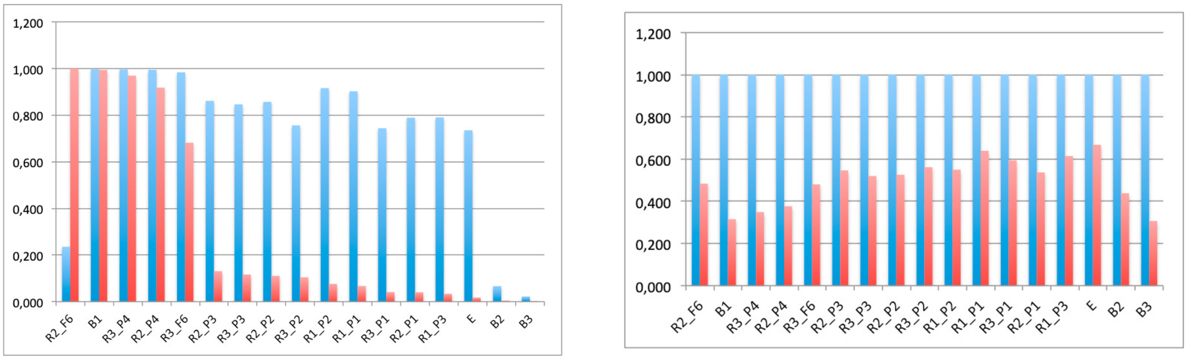
The emergency flag is then set to `TRUE` for every basin station being at most 75 km far from each interesting location that has been found. We retrieved 58907 results with 25 real emergencies, 57 false emergencies, 5 doubtful results and 22 unique places identified.

6. Experimental setup and results

6.1. Data from SIR Tuscany for hydrological phenomena

A subset of SIR basin levels, rain measures and phreatic zone data for the years 2012-2014 has been used. A number of software tools have been developed specifically to extract and aggregate data provided by SIR, and parse Apache Mahout™output. After the creation of the basket connecting all the basin level station with the nearest rain of phreatic values, the data have been divided in two subsets: a **training** subset, containing a 60% of the items in the original set, to be used as PFP input for association rules extraction and a **test** subset, containing the remaining 40%, over which the extracted rules have been tested. Candidates for both sets are chosen using a random sampling.

The PFP algorithm generated two hundreds thirty four rules out of the training data subset. We decided to discard a small subset of rules having a confidence ratio inferior to 25% because we deemed them irrelevant. We also removed rules with a missing value as consequent.



(a) Precision

(b) Recall

Fig. 1: Comparison of boosted classifiers(in blue) vs. average value of association rules with the same consequent(in red)

The extracted rules have been evaluated considering their performance over the test set. A **True Positive** (TP) classification takes place when both the antecedent and the consequent of the rule are satisfied, while in a **True Negative** (TN) one neither of them is. A **False Positive** (FP) classification satisfies the antecedent, but not the consequent: for **False Negatives**(FN), the reverse applies. The accuracy of a rule is thus defined as:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision and recall are instead defined as:

$$prec = \frac{TP}{TP + FP} \qquad rec = \frac{TP}{TP + FN}$$

The harmonic mean of precision and recall is called F_1-score:

$$F_1 = 2 \times \frac{prec \times rec}{prec + rec}$$

We have partitioned the ruleset into twenty-nine subsets, one per each extracted symbol. We run the AdaBoost algorithm for five iterations over each subset. We then compared their performance criteria to the averages over each subset.

The test have been performed to validate the proposed technique. A first issue is the evaluation of the performance improvement with boosting technique. We test the method, with the rules from a single rule set, to assess if the method is effective. Secondly we test the extraction of rules from different rule sets.

We run five iterations of the AdaBoost algorithm over the rule set generated by the FP-growth algorithm. The result of the generated rules are shown in the following figures.

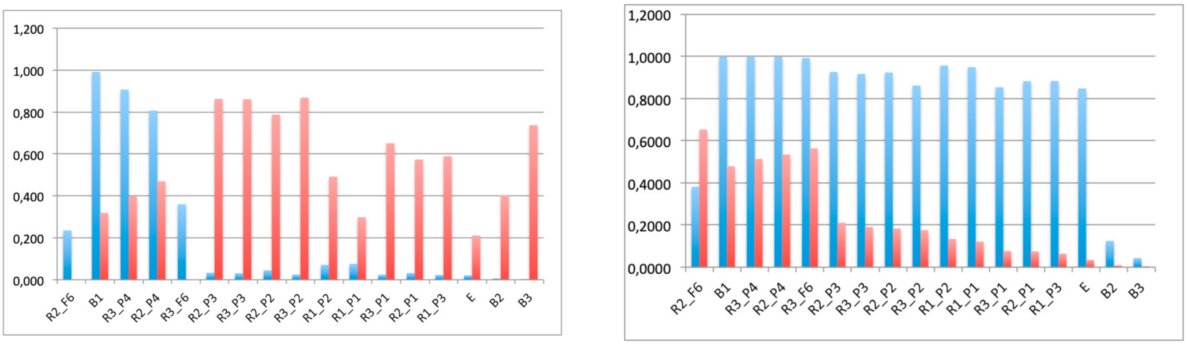
On the other side the original classifiers show a better precision, in average, than the boosted classifier. Typically simpler classifiers are very focused on a particular antecedent-consequent couple and may actually have high precision values. The figure 1a shows that although in some case the boosted rules have higher precision or in general equivalent, in some case ,the precision of the composed rule is sensibly worse in terms of precision.

In general, we seek classifiers having either a very high precision or a very high recall. We can see that boosted classifiers have maximum recall, as they leverage all the best simpler classifiers (Figure 1b).

Our tests show that, in general, using a linear combination of two or more classifiers yields an accuracy that can be higher but also lower(Figure 2a). An interesting result is that the majority of these stronger classifiers combine two association rules.

The general trend of the F_1-score, being the harmonic mean of precision and recall, closely resembles the lesser of the two. In general the performance of the boosted rules outperforms or equals the original, simpler rules. (Figure 2b).

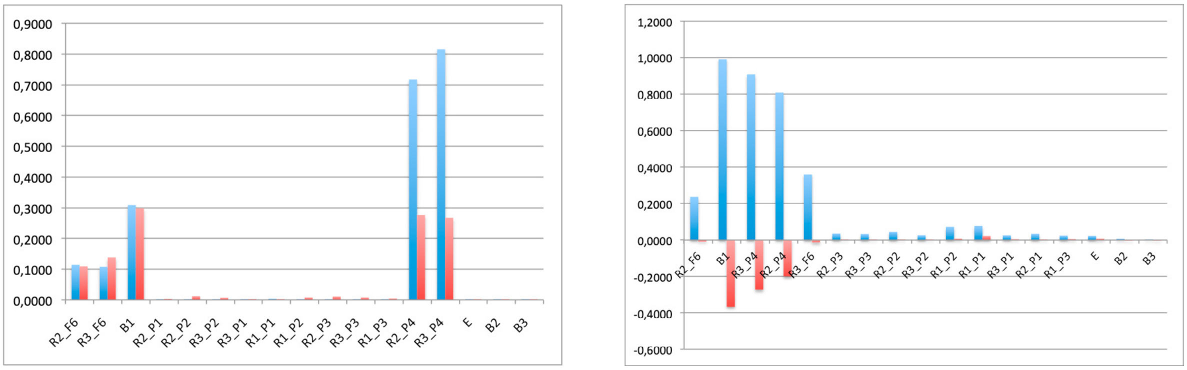
Also the value of the Rule Power Factor has been considered for the obtained rules. In most cases the performance is equal while for symbols such as "R2_P4" and "R3_P4" the value of the boosted rule is sensibly higher (Figure 3a).



(a) Accuracy

(b) F_1

Fig. 2: Comparison accuracy and F1 of boosted classifiers(in blue) vs. average value of association rules with the same consequent(in red)



(a) Rule Power Factor

(b) *Confirmed confidence*

Fig. 3: Comparison Rule Power Factor and Confirmed Confidence of boosted classifiers(in blue) vs. average value of association rules with the same consequent(in red)

As our approach is data-based, evaluating the usefulness of a rule can be difficult: as Geng and Hamilton presented in [9], "there is no widespread agreement on a formal definition of interestingness" of discovered patterns. They use several specific criteria for interestingness. High *coverage* implies an high support; *peculiar* patterns are generated from outliers, and may be "unknown to the user, hence interesting".

A further metric we used is the *Confirmed Confidence* that is described in [3] by Berzal et al. The definition of *Confirmed Confidence* for a rule $X \Rightarrow Y$is given as

$$ConfirmedConfidence = \frac{TP - FP}{TP + FP + TN + FN} \quad (1)$$

The values of the confirmed Confidence are given in the figure 3b. The values can be positive of negative according the number of true positive and false positive. For the given example, the boosted rules appear to be well performing and the creation of new rule beneficial for the task of rule extraction.

These criteria have been proposed to evaluate single rules, but we can easily use them to guide our choice of interesting stronger classifiers by applying them to their weaker components. We have chosen to present some examples among those having interesting improvements in performance giving a detail on the creation of boosted association rules.

7. Conclusions

In this paper, a method to detect relationships between different measure types in a sensor network has been devised for analysis and emergency detection purposes. A set of association rules has been extracted using a subset of Tuscanian Open Data by SIR containing geophysical measures; both the Apriori and FP-Growth algorithms have been employed to assess their performance in this domain. Emergency information have been extracted through queries to the Bing™Search Engine, and each has been tested using the remaining, lower-cardinality subset.

Introducing some measure from Information Retrieval and Pattern Mining fields, an estimation of the classification power of each rule has been estimated; in an effort to enhance the system, stronger classifiers have been generated using the AdaBoost meta-algorithm. Having generated a strong classifier per symbol, we have grouped each weak classifier and took their average performance as reference values to evaluate possible improvements.

The outcome of the use of boosting in this scenario have been mixed, with good performance when classifying for heterogeneous symbols, while unable to build a stronger classifier for every desired output. As the effectiveness of geophysical models is often dependent both on the amount of available data and on the place of their acquisition, it may be desirable to have more Tuscanian Open Data sources to improve pattern mining and to repeat the whole process in different locations, using different neighborhood shapes.

References

[1] R. Agrawal, T. Imielinski, and Arun Swami. Database mining: a performance perspective. *Knowledge and Data Engineering, IEEE Transactions on*, 5(6):914–925, Dec 1993.

[2] Rakesh Agrawal, Ramakrishnan Srikant, et al. Fast algorithms for mining association rules.

[3] Fernando Berzal, Juan-Carlos Cubero, Nicolás Marín, Daniel Sánchez, Jose-María Serrano, and Amparo Vila. Association rule evaluation for classification purposes. *Actas del III Taller Nacional de Mineria de Datos y Aprendizaje*, pages 135–144, 2005.

[4] Emanuele Cipolla, Umberto Maniscalco, Riccardo Rizzo, Dario Stabile, and Filippo Vella. Analysis and visualization of meteorological emergencies. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–12, 2016.

[5] Emanuele Cipolla and Filippo Vella. Boosting of association rules for robust emergency detection. In *2015 11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pages 185–191. IEEE, 2015.

[6] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, January 2008.

[7] Yoav Freund, Robert Schapire, and N Abe. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.

[8] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, August 1997.

[9] Liqiang Geng and Howard J. Hamilton. Interestingness measures for data mining: A survey. *ACM Comput. Surv.*, 38(3), September 2006.

[10] Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. *SIGMOD Rec.*, 29(2):1–12, May 2000.

[11] Suresh Kumar, Nisheeth Joshi, et al. Rule power factor: a new interest measure in associative classification. *Procedia Computer Science*, 93:12–18, 2016.

[12] Haoyuan Li, Yi Wang, Dong Zhang, Ming Zhang, and Edward Y. Chang. Pfp: Parallel fp-growth for query recommendation. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys '08, pages 107–114, New York, NY, USA, 2008. ACM.

[13] Carlos Ordonez and Kai Zhao. Evaluating association rules and decision trees to predict multiple target attributes. *Intell. Data Anal.*, 15(2):173–192, 2011.

[14] Gregory Piatetsky-Shapiro. Discovery, analysis, and presentation of strong rules. *Knowledge discovery in databases*, pages 229–238, 1991.

[15] Rabiatul AA Rashid, Puteri NE Nohuddin, and Zuraini Zainol. Association rule mining using time series data for malaysia climate variability prediction. In *International Visual Informatics Conference*, pages 120–130. Springer, 2017.

[16] Maik Reder, Nurseda Y Yürüşen, and Julio J Melero. Data-driven learning framework for associating weather conditions and wind turbine failures. *Reliability Engineering & System Safety*, 169:554–569, 2018.

[17] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

[18] F. Vella and C.-H. Lee. Boosting of maximal figure of merit classifiers for automatic image annotation. In *Proceedings of IEEE International Conference on Image Processing*, 2007.

[19] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. pages 511–518, 2001.

[20] Yongwook Yoon and G. G. Lee. Text Categorization Based on Boosting Association Rules. In *Semantic Computing, 2008 IEEE International Conference on*, pages 136–143, 2008.