Data Dictionary Extraction for Robust Emergency Detection

Emanuele Cipolla and Filippo Vella

Abstract In this work we aim at generating association rules starting from meteorological measurements from a set of heterogeneous sensors displaced in a region. To create rules starting from the statistical distribution of the data we adaptively extract dictionaries of values. We use these dictionaries to reduce the data dimensionality and represent the values in a symbolic form. This representation is driven by the set of values in the training set and is suitable for the extraction of rules with traditional methods. Furthermore we adopt the boosting technique to build strong classifiers out of simpler association rules: their use shows promising results with respect to their accuracy a sensible increase in performance.

1 Introduction

Exploiting sensor networks for environmental monitoring enables the study of a wide variety of geophysical phenomena based on time-delayed measurements that can be furtherly processed to seek new, previously unknown connection among events. On the other hand, the development of knowledge discovery in databases has required scientists and engineers to focus more and more on data-driven discovery while modeling their domains of interest. More recently, the Big Data paradigm has been adopted to deal with huge quantities of data differing in range and representation, with a variable trustworthiness that the availability of powerful hardware commodities have made operable. Discretized time series are useful to reduce the cardinality of the set of symbols without a significant loss of information. A statistical approach, such as [1], may detect current anomalies in those data measuring entropy by means of mutual information or other suitable representations; discretized data may also be

E. Cipolla (✉) · F. Vella
Institute for High Performance Computing and Networking - ICAR,
National Research Council of Italy, Palermo, Italy
e-mail: cipolla@pa.icar.cnr.it

F. Vella
e-mail: vella@pa.icar.cnr.it

G. De Pietro et al. (eds.), *Intelligent Interactive Multimedia Systems and Services 2016*, Smart Innovation, Systems and Technologies 55,
DOI 10.1007/978-3-319-39345-2_3

used to extract rules based on frequent items. A first, straightforward, choice is to divide the range of all the values in a limited set of intervals and associate to each interval a different symbol. In this case the data levels are fixed and the thresholds for all the levels are generated by the previous knowledge about the experimental domain. A different choice is to create a set of symbols according the distribution of the data on a given set. The symbols are chosen according to the distribution of the data in the vector space. A reasonable choice is to use a clusterization process and extract the centroids from a given set of values (e.g. the training set). Each value of the set can therefore be replaced with a symbol that corresponds to the centroid associated to it. *k*-means is a well known algorithm for the clusterization of data.The values obtained with this method are compared with a set of centroids obtained with Vector Quantization algorithm evaluating advantages and shortcomings. In this work, we use a parallelized Parallel FP-Growth (PFP) [2], based on the seminal FP-Growth algorithm described in [3]. We used the Apache Mahout™ implementation of PFP to generate rules as it leverages the open source Hadoop environment that has become the *de facto* standard for storing, processing and analyzing Big Data. The whole set of rules found is the input to the AdaBoost algorithm, first proposed by [4]. Our main contribution to the topic lies in the use of pattern mining techniques to find co-occurrence relationships leading to risk situations, enhancing historic datasets gathered by sensor networks with emergency notifications commonly found online newspapers and weblogs; acknowledging that association rules can be heavily dependent on the training set, we strive to provide stronger classifiers built using boosting. We tested the proposed technique on a dataset of Tuscanian meteorological data ranging from 2000 to 2010, and we have compared these values with the emergency detection in the same region along the same years, with promising results.

2 Frequent Pattern Mining

Frequent pattern mining deals with finding relationships among the items in a database. This problem was originally proposed by Agrawal in [5]: given a database D with transactions $T_1 \ldots T_N$, determine all patterns P that are present in at least a fraction of the transactions. The set T_i of identifiers related to attributes having a boolean `TRUE` value is called *transaction*.

An example domain of interest is composed of market baskets: each attribute corresponds to an item available in a superstore, and the binary value represents whether or not it is present in the transaction: an interesting pattern is thus present if 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.

2.1 Association Rule Mining

Agrawal et al. [5] presented association rule mining as a way to identify strong rules using different measures of interestingness, such as high frequency and strong correlation.

Let $D = \{T_1, T_2, \ldots, T_n\}$ be a transaction database. A set $P \subseteq T_i$ is called an l-sized *itemset* if the number of items it contains is l, and has a support $supp\left(P_i\right) = \frac{|P_i(t)|}{|D|}$ that is the ratio of transactions in D containing X. X will be deemed *frequent* if its support is equal to, or greater than, a given threshold minimal support. An association rule R is the implication $X \implies Y$, where itemsets X and Y do not intersect. An evaluation on the validity of each rule can be performed using several quality measurements:

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

Given a minimal support s_{MIN} and minimal confidence c_{MIN} by users or experts, $X \implies Y$ is considered a valid rule if both $supp\,(X \implies Y) \geq s_{MIN}$ and $conf\,(X \implies Y) \geq c_{MIN}$.

2.2 PFP: The FP-Growth Algorithm in a Parallelized Environment

In 2008, Dean and Ghemawat [6] presented MapReduce, a framework for processing parallelizable problems across datasets using a large number of inter-connected computer systems, called *worker nodes*, taking advantage of locality of data in order to reduce transmission distances. The FP-Growth Algorithm, a divide et impera algorithm that extracts frequent patterns by pattern fragment growth proposed by Han in [3], has been ported to the MapReduce framework by Li et al. [2].

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 group-wise: 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.

3 Construction of a Robust Classifier Through Boosting

The word *boosting* refers to a general method of rule production that combines less accurate rules to form more accurate ones. A so-called "weak learning algorithm", given labeled training examples, produces several basic classifiers: the goal of boosting is to improve their global performance combining their calls, assuming that they fare better 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 [4]. 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 $h_t : X \in 0.1$ having low weighted error $\epsilon_t = \sum_i w_i \left| h_j(x_i) - y_i \right|$ is chosen. A parameter α_t, with $\alpha_t > 0 \iff \epsilon_t < 1$, is chosen, so that the more accurate the base classifier h_t is, the more importance we assign to it. To give prominence to hard-to-classify items, weights for the next iteration are defined as $w_{t+1}, i = w_t \beta_t^{1-e_i}$, 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 strong classifier is:

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

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

Association rule extraction algorithms produce, on average, many more rules than classification algorithms, because they do not repeatedly partition record space in smaller subsets—on the other hand, this means that association rules are much more 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 (as shown in Sect. 4) of the attribute space is performed, so that the established bins can serve as feature sets. Association rule mining can be

thus applied to find patterns of the form $< featuresets > \Longrightarrow ClassLabels$, ranking rules first by confidence, and then by support, as shown in Yoon and Lee's [7]. While we follow a similar approach, we diverge in several aspects:

- we perform boosting on the whole set of generated rules;
- a correct weak classification occurs if both or neither of antecedent and consequent are present;
- we penalize the weight only if the error rate of a given weak classifier is lower than 50 %.

4 Information Representation

The datasets used in this work has been made available by Servizio Idrogeologico Regionale della Toscana (SIR).[1] Their sensors 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 networks 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 $r_1 = 25\,\mathrm{km}, r_2 = 50\,\mathrm{km}, r_3 = 75\,\mathrm{km}$ 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 discretized value. This quantized representation is needed since the rule extraction algorithms extract connections among recurring symbols.
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. There will be a transaction row vector r per day per station. In each of them the column (B_k) will have a `TRUE` value if the discretized value is k, `FALSE` otherwise. An emergency flag is set if for a given date the basin station was near enough to dangerous phenomena.

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

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.

4.1 Discretization

The continuous nature of meteorological measurements is not suitable for association rules extraction: a reduced set of values is thus considered using a discretization process. We used both the *k*-means and Linde-Buzo-Gray algorithms to extract the discretization interval bounds.

k-means, first proposed by Lloyd in [8], takes as input the number *k* of clusters to generate and a set of observation vectors to cluster, returning exactly *k* centroids, initially chosen at random and converging to a stable position that minimizes the sum of the quadratic distance between the observations and the centroids.

A vector *v* belongs to cluster *i* if it is closer to centroid *i* than any other centroid; c_i is said to be the *dominating centroid* of *v*. Since vector quantization is a natural application for *k*-means, information theory terminology is often used: the centroid index or cluster index is also referred to as a *code* and the table mapping codes to centroids is often referred as a *codebook*, so *k*-means can be used to quantize vectors. Quantization aims at finding an encoding of vectors that minimizes the sum of the squared distances (SS) between each observation vector x_i^j and its dominating centroid c_j, called *distortion*:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 \tag{2}$$

k-means terminates either when the change in distortion is lower than a given threshold or a maximum number of iterations is reached.

Linde, Buzo and Gray algorithm, first proposed in [9], is another iterative algorithm which assures both proximity to centroids and distorsion minimization. The initial codebook is obtained by splitting into two vectors the average of the entire training sequence. At the end of the first iteration, the two codevectors are splitted into four and so on. The process is repeated until the desired number of codevectors is obtained.

A simple test, probably first proposed by R. Thorndike in 1953 and called *elbow method*, can be used to choose the right *k* with respect to the percentage of variance: if you plot the SS against the value of *k*, you will see that the error decreases as *k* gets larger; this is because when the number of clusters increases, they should be smaller, so distortion is also smaller. The idea of the elbow method is to choose the *k* at which the SSE decreases abruptly. This produces an "elbow effect" in the graph.

4.2 Emergency Information

While sensor networks provide quantitative figures with a certain degree of reliability, they do not convey any information about emergencies: we need this information in order to train a classifier that can identify potential unforeseen climaxes. We assume that if an emergency situation has been reported in the past, traces 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 Italian keywords 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*. As WWW pages cannot be easily dated, we used the subset of search results having day, month and year information in their URLs, usually found in weblogs and digital magazines, as they require specific expertise to get altered after publication. This subset has been filtered by visual check to remove spurious and incomplete data, but we are unable to exclude that some of the remaining information has not been altered by the content authors, either willingly or because of an error. The emergency flag is then set to `TRUE` for every basin station placed at a distance less than 75 Km from each interesting location that has been found.

5 Experimental Setup and Results

A subset of SIR basin levels, rain measures and phreatic zone data for the years 2000–2010 has been used. The Elbow test has been performed after having repeatedly run the k-means algorithm on each set of measures. We chose to use 3 bins for rain data, 4 bins for basin level data, and 6 bins for phreatic data. In the latter case, we actually have two elbow conditions, and the second one happens just before a slight increase in the sum of squares: we arbitrarily chose the number of bins that gives the absolute minimum sum of squares (see Fig. 1).

A similar profiling has been performed using Linde-Buzo-Gray quantization on our discrete data; the minimum value of distorsion is achieved using 8 codevectors (see Fig. 2).

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,

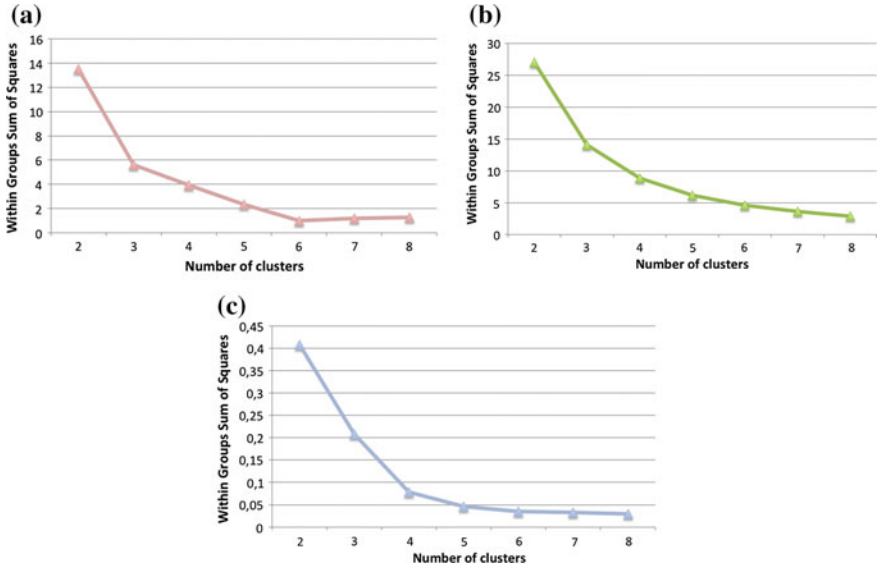

Fig. 1 Average distortion for *k*-means algorithm. **a** Phreatic levels. **b** Rain levels. **c** Basin levels

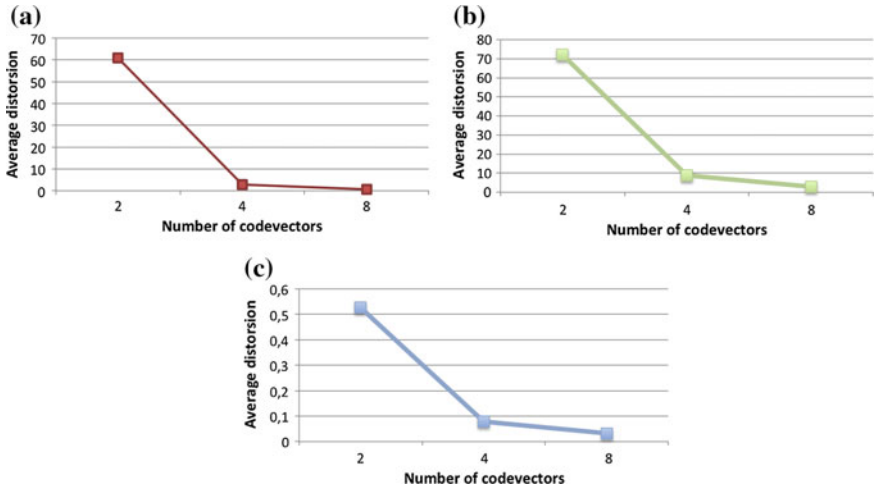

Fig. 2 Average distortion for LBG algorithm **a** Phreatic levels. **b** Rain levels. **c** Basin levels

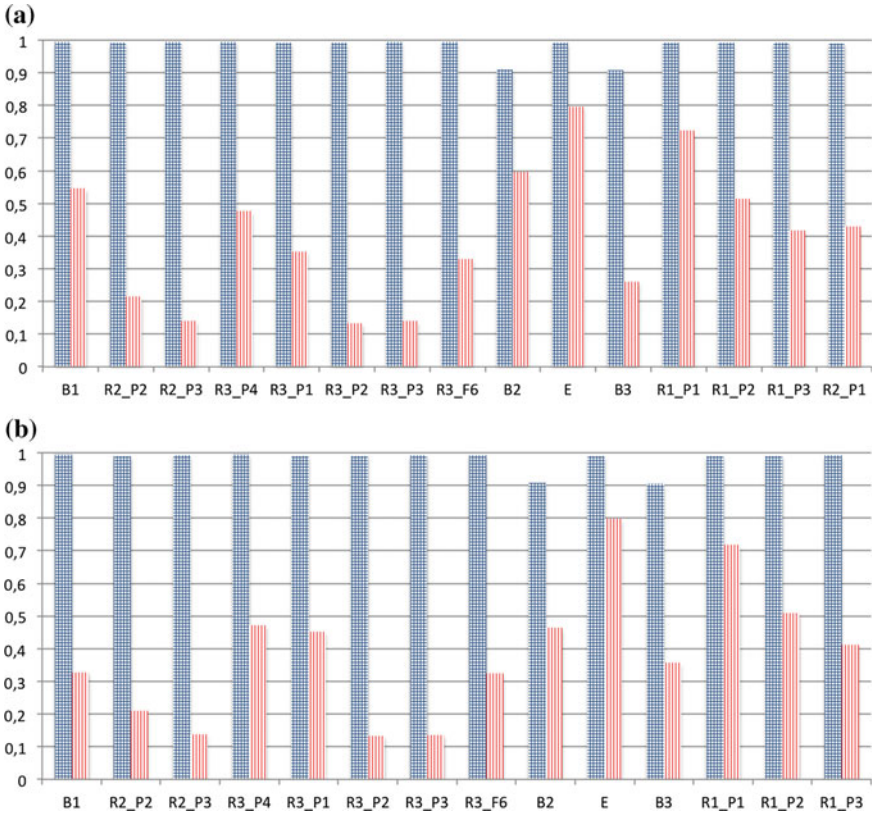

Fig. 3 Accuracy of boosted classifiers (*in blue*) versus accuracy values of association rules for the same consequent (*in red*). **a** LBG quantization. **b** *k*-means quantization (Color figure online)

having discarded a small subset of rules having either a confidence ratio inferior to 25 % or a missing value as consequent.

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 *Acc* is defined as $\frac{TP+TN}{TP+TN+FP+FN}$ and presented in Fig. 3.

Precision $prec = \frac{TP}{TP+FP}$ and Recall $rec = \frac{TP}{TP+FN}$ are shown in Figs. 4 and 5, respectively.

The harmonic mean of Precision and Recall, called F_1-score, is $F_1 = 2 \times \frac{prec \times rec}{prec+rec}$. It is shown in Fig. 6.

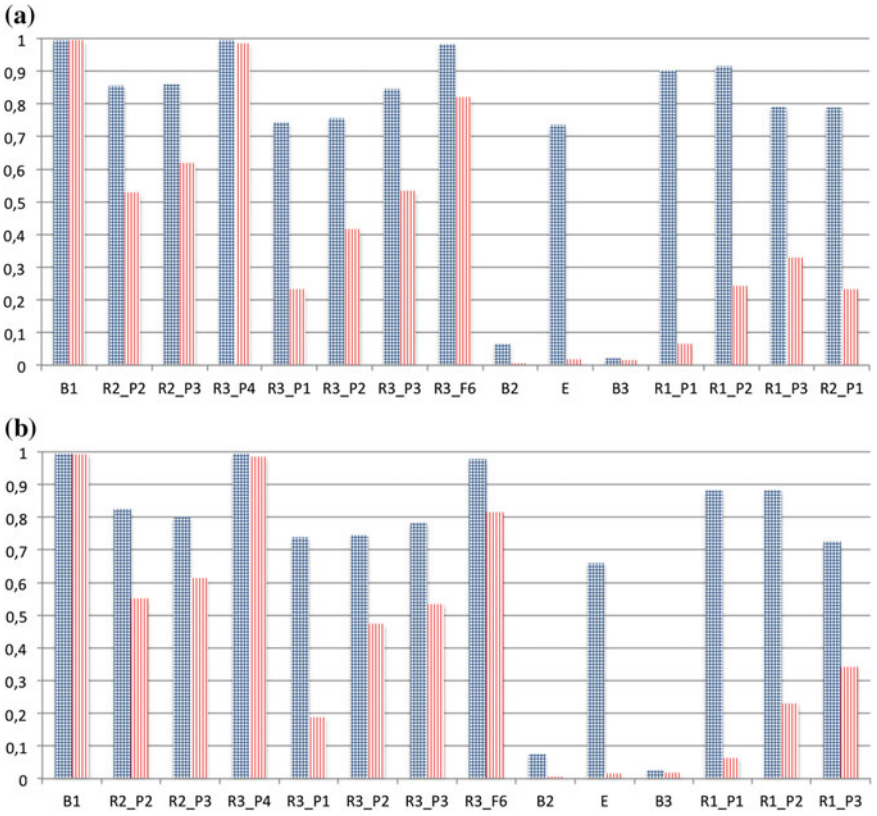

Fig. 4 Precision of boosted classifiers (*in blue*) versus average of precision values of association rules for the same consequent (*in red*). **a** LBG quantization. **b** *k*-means quantization (Color figure online)

Finally, we compared the Precision of those boosted classifiers yielded after using both techinques for vector quantizations on our source data with the Precision of similarly boosted classifiers generated from the SIR-inspired 7-bin discretization we presented in our previous work [10].

While the use of different quantization techniques has nearly halved the number of produced classifiers, those ones remaining are much more accurate. As you can see in Fig. 7, while LBG and *k*-means performance is nearly on par, with LBG being marginally better, they both outperform our old classifiers with fixed symbols. This is also true for the classifier for the emergency symbol E, whose Precision has grown by 68 %.

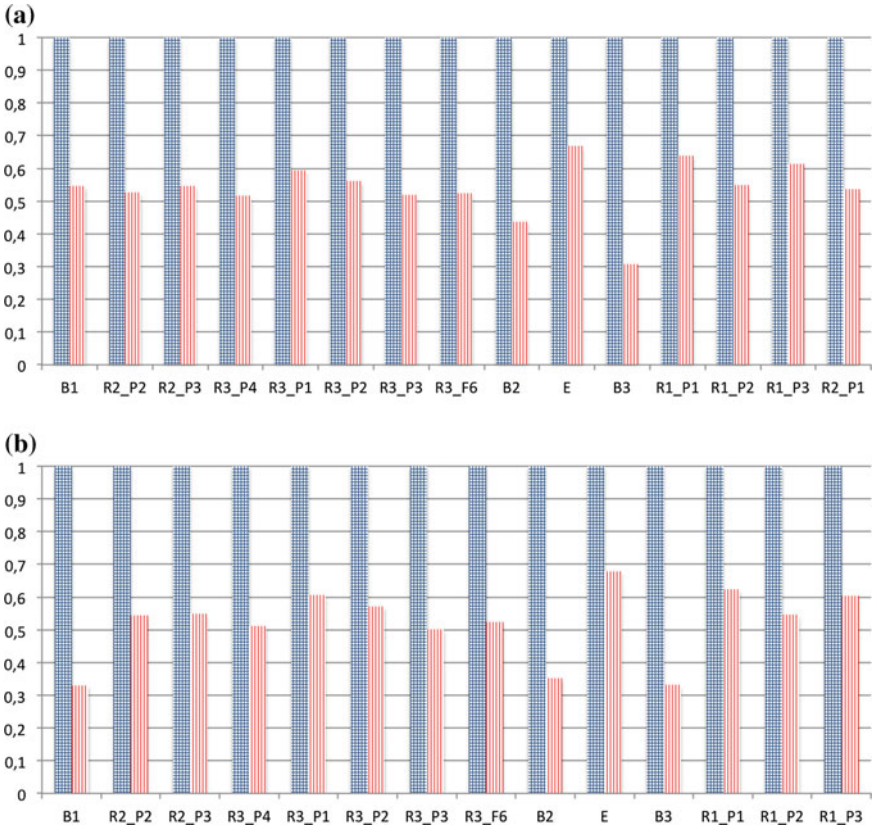

Fig. 5 Recall of boosted classifiers (*in blue*) versus average of Recall values of association rules for the same consequent (*in red*). **a** LBG quantization. **b** *k*-means quantization (Color figure online)

6 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 10 years of Tuscanian Open Data by SIR containing geophysical measures. Emergency information have been extracted through queries to the Bing™ Search Engine. Stronger classifiers have been generated using the AdaBoost meta-algorithm on association rules extracted by the PFP 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 use of vector quantization has significantly improved the accuracy of our boosted classifiers over the same dataset, especially that of our classifier for emergency situations.

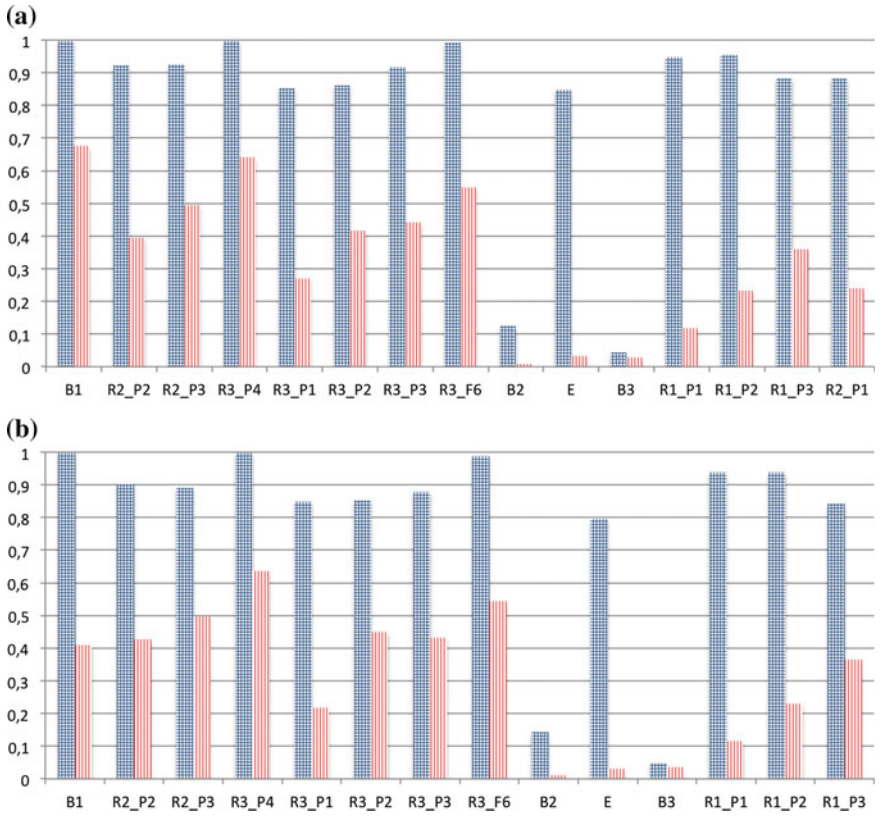

Fig. 6 F1-score of boosted classifiers (*in blue*) versus average of F1-score values of association rules for the same consequent (*in red*). **a** LBG quantization. **b** *k*-means quantization (Color figure online)

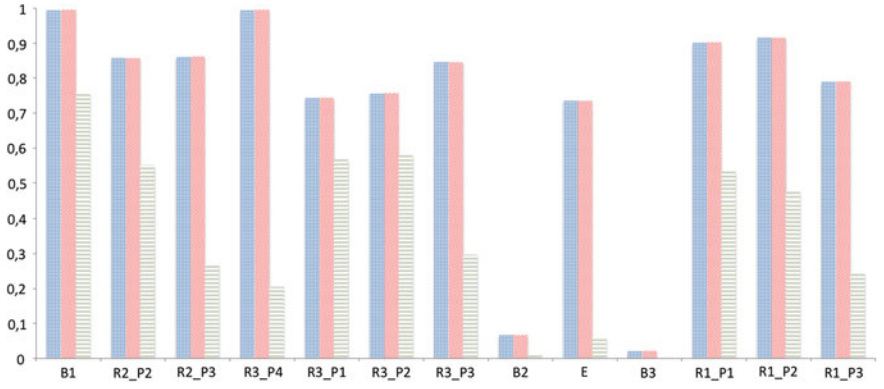

Fig. 7 Comparison of precision for *k*-means, LBG, 7-bin quantization

Acknowledgments This work has been partially funded by project SIGMA PONOI_00683 Sistema Integrato di sensori in ambiente cloud per la Gestione Multirischio Avanzata PON MIUR R&C 2007–2013.

References

1. Cipolla, E., Maniscalco, U., Rizzo, R., Stabile, D., Vella, F.: Analysis and visualization of meteorological emergencies. J. Ambient Intell. Hum. Comput. 1–12 (2016)
2. Li, H., Wang, Y., Zhang, D., Zhang, M., Chang, E.Y.: Pfp: parallel fp-growth for query recommendation. In: Proceedings of the 2008 ACM Conference on Recommender Systems, ser. RecSys '08, pp. 107–114. ACM, New York, NY, USA (2008)
3. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. SIGMOD Rec. **29**(2), 1–12 (2000)
4. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. **55**(1), 119–139 (1997)
5. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. SIGMOD Rec. **22**(2), 207–216 (1993)
6. Dean, J., Ghemawat, S.: Mapreduce: simplified data processing on large clusters. Commun. ACM **51**(1), 107–113 (2008)
7. Yoon, Y., Lee, G.G.: Text categorization based on boosting association rules. In: IEEE International Conference on Semantic Computing, vol. 2008, pp. 136–143 (2008)
8. Lloyd, S.: Least squares quantization in pcm. IEEE Trans. Inf. Theory **28**(2), 129–137 (1982)
9. Linde, Y., Buzo, A., Gray, R.M.: An algorithm for vector quantizer design. IEEE Trans. Commun. **28**, 84–95 (1980)
10. Cipolla, E., Vella, F., Boosting of association rules for robust emergency detection. In: 11th International Conference on Signal-Image Technology & Internet-Based Systems, SITIS 2015, pp. 185–191. Bangkok, Thailand, 23-27 Nov 2015