Comparison of Bayesian classifiers to detect pollution

D. GACQUER, F. DELMOTTE, V. DELCROIX, S. PIECHOWIAK

LAMIH
Université de Valenciennes et du Hainaut-Cambrésis
Le Mont Houy
59313 CEDEX 9
{david.gacquer,francois.delmotte}@univ-valenciennes.fr

Abstract

This paper addresses the pollution detection problem by using a camera and analysing the pictures. A camera is used to record visual scenes around complex plants. Then several signals are computed to describe the pictures. Our aim is to detect among the various clouds if there are polluting smokes. We assume in this paper that the signals are useful to classify the clouds and that we do not need other data. In this paper two types of classifiers are studied: Bayesian belief networks and a k-nearest neighbour classifier as a reference.

Keywords: Bayesian network, classification.

1 Introduction

Researches presented in this paper are based on collaboration with the Aloatec Company, based at Calais, France. For heavy plants, such as cookeries, steelworks, and so on, pollution may appear everywhere. It is thus impossible to use chemical sensors. This paper concerns hazardous smokes that can be detected by camera. Indeed, the solution is based on the use of a single sensor, a camera, used far away from the relevant plant. Then, by analysing the pictures, we must define a system that will detect if there are dangerous smokes or not.

In this paper we assume that we can use a set of useful signals, such as the density, the shape, or the colour of the detected clouds. So we have to define a classifier that will associate to a set of pictures a degree of pollution, from 0 to 3 (high danger).

In this paper we will study two types of classifiers. The first type is a Bayesian classifier [3] which is very good when the data used in the learning step are numerous. The second classifier is a k nearest neighbour classifier [5, 10], more robust when data are parsimonious.

Section 2 presents the two types of classifiers, and section 3 details our problem of detection of smokes. Section 4 presents the results obtained by experimentations on real data sets.

2 Presentation of the two types of classifiers

2.1 Principle of Bayesian approaches

A Bayesian network is an acyclical graphical model representing the links between variables: the nodes represent the variables, and each arc between two nodes represents a link. They are weighted by conditional probabilities, to traduce the fact that links can be strong or weak. Bayesian networks are expert system weighted by probabilities, and are used for diagnostic and classification problems [1, 2, 3, 8, 9].

A Bayesian network consists in both the structure of the graph and the conditional probabilities. Moreover, we must know the probability a priori of the variables without any parents (the roots of the graph). Then the Bayes rule enables us to compute the probability of each successive node:

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)} \tag{1}$$

In the detection of pollution example, we use a naive Bayesian classifier and we also try to find a more complex structure for the network by using the library of Murphy [6] for the learning step.

For the naive Bayesian network, the degree of pollution is simply given by:

$$P(pollution|x) = \frac{P(x|pollution) \times P(pollution)}{P(x)} \tag{2}$$

with $x \in \mathbb{R}^p$ the vector of the p measured signals assumed independent, and $pollution$ the level of pollution, a discrete variable ranging from 0 to 3, and

$$P(x) = \sum_i P(x|pollution_i) \times P(pollution_i) \tag{3}$$

Note that in every approaches described above, we always use the class node as root node. Note also that to use most of the learning algorithms for Bayesian networks, all the variables have to be discrete. We use the information theory exposed in [7] to obtain discrete values from the original continuous data. The values are split into several intervals which are merged when there is no information loss.

2.2 Structure learning for the Bayesian network

In the previous section, we use a naïve network to classify the pollution clouds. Naïve Bayes classifiers are known to produce good results when the correct structure isn't given by an expert. To determine whether it is possible to achieve better performances using a different structure, we also use Murphy's Bayes Net Toolbox (BNT) [6] and Philippe Leray's Structure Learning Package (SLP) [4] for Matlab. Used simultaneously, those two Matlab libraries provide an environment to learn the network's structure directly from data. Different learning algorithms are implemented in those libraries, such as Greedy Search, K2 algorithm or Tree Augmented Naïve for example. The general principle of those algorithms is as follow: a starting graph (an empty graph or a fully connected graph for example) is used to initialize the problem. The algorithm then proceeds by steps, tries to add or remove connections between the nodes and computes the gap obtained using a score, to determine whether or not there must be a connection between particular nodes. We use the hist_c function from the SLP package to perform optimal discretization.

In this paper, we will use three Bayesian classifiers: a naïve one, and a derived form of the latter, called Tree Augmented Naïve (TAN), in which we allow the existence of links between feature nodes, assuming that there may be dependences between the attributes. We will also supply results obtained with the Greedy search, which does not have constraints on the structure. Other algorithms provide other networks, but the error rates are equivalent.

2.3 K nearest neighbours classifier

Bayesian classifiers rely on a correct estimation of the conditional probabilities. But in our detection problem, conditional probabilities regarding the dangerous smokes are not well estimated.

This is why the k-nearest neighbour classifier is studied [5, 10]. It is based on a direct estimation of the conditional probabilities. Its principle is as follows: an object is classified according to the class of its nearest neighbours. So this algorithm is based on distances between an object to classify, and the other objects available in the database, and for which we know the correct class. A rule of majority is used among the k nearest neighbours, k being a parameter to optimize.

With the classical classifier, problems may arise if there is no majority among the k nearest neighbours. We improve the algorithm and solve this problem by using a weighted sum. Indeed, the level of pollution/danger associated to a cloud is not a discrete variable (from 0 to 3), but a continuous one. So the next relation can be used:

$$pollution = \sqrt[\beta]{\frac{\sum_{i=1}^{n} \frac{1}{d_i} \left(pollution(i)\right)^{\beta}}{\sum_{i=1}^{n} \frac{1}{d_i}}} \qquad (4)$$

with d_i the distance between the cloud to classify, and the i^th^ closer in the data base, and $pollution(i)$ being its level of pollution. β is a parameter to optimize used to weight more or less high levels of pollution. Of course if k = 1, we get exactly the classical classifier.

Since changes of scaling for the coordinates used to compute distances have a strong influence on the result, all data are normalized with such an approach (we withdraw their mean value, and divide by their standard deviation).

3 Detection of dangerous smokes

The two previous types of classifiers were used on real data from an industrial plant of the north of France. The Aloatec Company has developed a real time component that studies the pictures and provides a set of signals when a cloud is believed to be dangerous. In this paper we do not tackle this component.

Once we receive a set of signals, we enter it in our classifier to compute the corresponding pollution level using inference. We assume that one set of signals concerns only one object, one cloud to classify. The real time component discards small clouds, and provides data only for the main cloud.

The real time component detects the beginning and the end of a cloud, and records the whole evolution of all the signals. In this paper we do not deal with a dynamical classifier. All signals are replaced by their mean value computed on the window defined by the beginning and the ending of a cloud. The classifier must provide a decision about the degree of pollution a little bit after the end of the emission. This introduces delays with the beginning of a cloud, but usually these delays are short, since the typical duration of an event to classify is shorter than 5 minutes.

4 Experiments

To lead our experiments, the Aloatec Company provided us with a database containing about 3000 objects recorded during 3 months of activity. We define by the word object a smoke reject observed by the camera. For our experiments, this database has been split into two parts, containing the same number of objects : the first one for the learning step, and the second for the tests. When hazardous smoke is detected by the camera, several pictures are recorded during the event, and the image processing system developed by Aloatec computes several features such as the density, the surface or the duration of the emission. Each object is then defined by a set of attributes (12 in our case) and by its level of gravity, given by an expert. Further details concerning the calculation of the different features can't be given in this paper due to a non disclosure agreement with the Aloatec Company. Our objective consists in computing the estimated gravity level given the observed attributes. Then we compare the estimated level with the one given by the expert. We note that in the database given by Aloatec, there is a high proportion of low pollutions (level 0) for only a small amount of level 3 pollutions. That is why, in general, the estimation of low pollution will be more accurate.

We use a confusion matrix, to compare the classifier's decision and the level given by the expert. On each matrix, we compute two criteria: the error rate, which describes the differences between classifier and expert decisions, and the efficiency, which traduces the fact that false alarms and non detections greatly penalize the classifier's performance. The definition of those two criteria is given below:

$$Error = \frac{\sum_{i=0}^{3}\sum_{\substack{j=0 \\ i\neq j}}^{3} M_{ij}}{\sum_{i=0}^{3}\sum_{j=0}^{3} M_{ij}}, \quad Efficiency = \frac{0.3\left(M_{13}+M_{31}\right)+0.8\left(M_{23}+M_{32}\right)+M_{33}}{\sum_{i=0}^{3} M_{i3}+\sum_{j=0}^{2} M_{3j}} \tag{5}$$

The next tables provide the confusion matrices as well as the two criteria for the different classifiers. For the k nearest neighbour, results are provided for k and β giving the lowest error rates (k=8,β=1). A better efficiency can be obtained for other values. These algorithms have been implemented using the Matlab BNT library from Murphy and the additional Structure Learning Package from P. Leray.

Table 1. Confusion matrix for the naïve Bayesian classifier

	Level 0	Level 1	Level 2	Level 3
Level 0	1005	38	1	1
Level 1	30	152	27	2
Level 2	1	33	26	12
Level 3	0	5	9	108

Error: : 10.72 %
Efficiency : 93.01 %

Table 2. Confusion matrix for the k-nearest neighbour classifier

	Level 0	Level 1	Level 2	Level 3
Level 0	1018	29	1	0
Level 1	27	174	10	0
Level 2	3	41	20	8
Level 3	0	2	16	100

Error : 9.45 %
Efficiency : 95.08 %

Table 3. Confusion matrix for the greedy search based Bayesian classifier

	Level 0	Level 1	Level 2	Level 3
Level 0	997	41	0	7
Level 1	21	171	16	3
Level 2	0	39	18	15
Level 3	1	4	5	108

Error: 10.52%
Efficiency : 88.18 %

Table 4. Confusion matrix for the TAN classifier

	Level 0	Level 1	Level 2	Level 3
Level 0	998	24	1	22
Level 1	24	145	18	24
Level 2	3	26	11	32
Level 3	0	5	4	109

Error : 12.66 %
Efficiency : 74.75 %

These matrices confirm that the k nearest neighbour classifier is the best one in case of low amount of data, here for the level 3 degree of pollution. We can see that the results obtained using the greedy search algorithm are not better than a naïve network in terms of efficiency. However, differences can be observed concerning the classification of the highest pollution (level 3 degree) but we totally lack of any explanation about the links between the variables. The TAN classifier provides a poor efficiency due to bad classification for high degree pollution. We can be surprised that the two structure learning algorithms (the greedy search and the TAN algorithms) do not provide efficiency at least equal to the naive Bayesian classifier. Indeed the naive graph is a possible choice for the learning algorithms, and this latter should at least find it. This is a well known problem [6], due to the fact that data are missing and the learning step does not test every possible graph.

5 Conclusion

In this paper we have compared different types of classifiers on an example of detection of pollution by means of cameras. Generally, results obtained are better than those provided by Aloa-Detect, the current classifier from the Aloatec Company, which is based on a set of rules. The k nearest neighbour classifier has proven to be more reliable when we lack of data concerning clouds of high degree pollution.

In the future we will define non linear models. In this paper the signals used to classify the clouds are not time varying: we take their mean value on a window. It is obvious that the next step is to define dynamical classifiers, and we will have to work deeper with the Aloatec Company, because the classifier and their real time component are intimately linked.

6 References

[1] G. F. Cooper. *Expert Systems based on belief networks*. Current Research Directions, 1988.

[2] D. Heckerman. A tutorial on learning with Bayesian networks. *In Learning in Graphical Models*, M. Jordan, ed. MIT Press, Cambridge, MA, 1999.

[3] V. Jensen Finn. *An introduction to Bayesian networks*. UCL Press, 1999.

[4] Philippe Leray, Olivier François (2004). BNT Structure Learning Package : documentation and experiments. Technical Report. Laboratoire PSI - INSA Rouen – FRE CNRS 2645.

[5] Machine Learning, T.M. Mitchell, Mac Graw Hill, 1997

[6] Murphy K (2001). The BayesNet Toolbox for Matlab, *Computing Science and statistics : Proceeding of Interface, 33.* http://www.ai.mit.edu/~Murphyk/Software/BNT/bnt.html .

[7] O. Colot, C. Olivier, P. C. & El Matouat A. (1994). Information criteria and abrupt changes in probability laws. Signal Processing VII : Theorie and Applications.

[8] Roberto G. Cowell, A. Philip Dawid, Stephen L. Lauritzen, and David J. Spigelhalter. *Probabilistic Networks and Expert Systems. Springer*, 1999.

[9] D. J. Spiegelhalter, S. L. Lauritzen, A. P. Dawid and R. G. Cowwel. *Bayesian analysis in expert systems*. Statistical Science, 8: 219-1247, 1993.

[10] A. Storkey. *Learning from Data : Nearest Neighbour Methods*. School of Informatics. University of Edimburgh.