Comparison of several classifiers for the detection of polluting smokes

David Gacquer, François Delmotte, Véronique Delcroix, Sylvain Piechowiak

LAMIH, University of valenciennes, France

1. Abstract

This paper presents several classifiers applied on a problem of detection of smokes by means of visual scenes analysis. 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 three types of classifiers are studied: two bayesian networks, a k-nearest neighbour classifier, and finally a linear model.

2. Introduction

Researches presented in this paper are based on a collaboration with the XXX company, based at XXX, XXX. For heavy plants, such as cookeries, steelworks, and so on, pollution may appear everywhere. It is thus impossible to use sensors locally. This paper concerns hazardous smokes that can be seen 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. Figure 1 shows an example.

Fig. 1. Example of visual scene to classify.

In this paper we assume that we can use a serie of useful signals, such as the density, the shape, or the colour of the detected clouds. Moreover we assume that an expert has access to a database of previously recorded clouds, and he has assessed their degree of pollution or danger.

So we have to define a classifier that will associate to a set of pictures a degree of pollution, by means of a learning step. The degree of pollution will vary from 0 (no danger) to 3 (theoretically the local population must leave the area).

There are many classifiers in the literature, from factorial analysis [1] to statistical one. In this paper we will study three types of classifiers. The first type is a bayesian classifier [3], [6], [16], which is very good when the data used in the learning step are numerous. However in our problem there are many data about no danger clouds, but only a few ones concern high danger clouds. Therefore the bayesian classifier may not be so good to detect highly polluting smokes. So the second classifier is a k nearest neighbour classifier [10, 12, 17]. Finally we will compare the results with a linear model [9].

Section 3 presents the three types of classifiers, and section 4 details our problem of detection of smokes. Section 5 presents the results obtained by experiments on real data sets.

3. Presentation of the three types of classifiers

3.1 Principle of Bayesian approaches

A bayesian network is an acyclical graphical model representing the links between variables: the nodes represent the variables, and each edge between two nodes represents a link. They are weighted by conditional probabilities, to traduce the fact that links can be strong or weak (see figure 2). Bayesian networks are expert system weighted by probabilities, and are used for diagnostic and classification problems [5, 6, 13, 14, 15, 16].

Figure 2 shows a classical example from Lauritzen and Spiegelhalter [15].

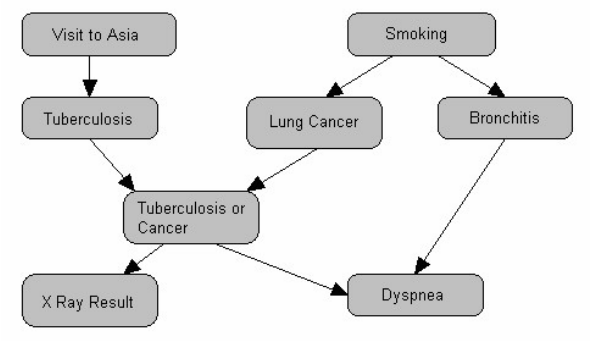

Fig. 2. Network about contracting an illness after a trip to Asia

A bayesian network consists in both the structure of the graph and the conditional probabilities. In the example of figure 2, we must estimate the probability that somebody gets the tuberculosis given that he went to Asia or that he is used to smoke. Moreover we must know the a priori probability of the variables without any parents (the roots of the graph). Then the bayes rule enables us to compute the probability of each successive nodes:

In the detection of pollution example, we have tested a naive bayesian network and we also tried to find a more complex structure for the network by using in the learning step the library of Murphy [11].

With a naive bayesian network, the main assumption is that every variables are independent from the others, except a single one, the root of the network which is in fact the class of the object to estimate. In our case the degree of pollution is simply given by:

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

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

$$P(x) = \sum_i P(x|pollution_i) \times P(pollution_i)$$

The bayesian network requires all variables to be discrete. So we had to discretize all our signals and we replaced them by 10 discrete variables uniformly spaced. This leads to a very high number of conditional probabilities to estimate.

3.2 Structure learning for the bayesian network

In the previous section, we used a naive network to classify the pollution clouds. Naive networks are known to produce good results when the correct structure is not given by an expert. To determine whether it would be possible to achieve better performances using a different structure, we decided to use Murphy's Bayes Net Toolbox (BNT) [11] and Philippe Leray's Structure Learning Package (SLP) [7] 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. A description of the different algorithms available in the SLP can be obtained in [8]. 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 initialise 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 in the optimal structure.

We will give in this paper the results obtained with the Greedy search. Other algorithms provide other networks, but the errors are comparable when the networks are used on the data to test.

3.3 K-Nearest neighbour 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 now studied [10, 12, 17]. It is based on a direct estimation of the conditional probabilities. Figure 3 gives its principle: 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, and k is a parameter to optimize

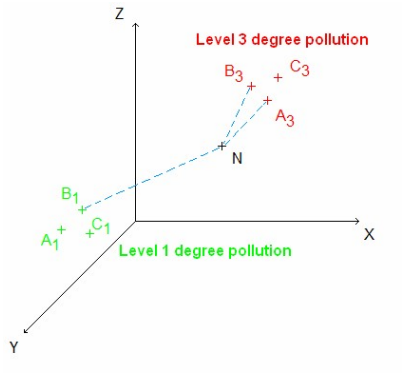

Fig. 3. Example of classification of the object N.

In this figure, we want to classify the object N in the 3-dimension space (X,Y,Z). We have 6 objects in the data base, 3 for the class 1, and 3 for the class 3. Here, it is obvious that N is much closer to the class 3 than to the class 1.

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 or 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:

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

with d_i the distance between the cloud to classify, and the ith 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 substract their mean value, and divide by their standard deviation).

3.4 A linear model

The two first classifiers have statistical roots, even if the theory leading to the k nearest neighbours was omitted.

With the bayesian classifier, all variables have to be discrete. With our version of the k nearest neighbours, the level of pollution is a continuous variable.

The next step is to try a model between all the signals representing the clouds, and their level of pollution or danger.

In this paper we have tried to define a linear model, with a relation of the type [9]:

$$Pollution = f\left(signal_1, signal_2, ..., signal_p\right)$$
$$= \alpha_1 \times signal_1 + \alpha_2 \times signal_2 + ... + \alpha_p \times signal_p$$

So we just have to estimate parameters a_i in order to get our linear model. These latter are given by a least

Once the parameters α_i are obtained, we can directly compute the level of pollution of a new cloud by means of .

3.4 Comparisons

Table 1 summarizes a few properties of the three previous classifiers:

Table 1. Properties of the three classifiers

	Model	Learning	Variables
Bayesian network	Non linear	Yes	Discrete
KNN	Non linear	No	Continues
Linear Model	Linear	Yes	Continues

It is obvious that the bayesian network and the K nearest neighbours classifier represent in fact a non linear model between the signals and the level of pollution. In the learning column, a yes represents the fact that through the learning step all the data in the data base are compressed in a few parameters: for the bayesian network there is the graph, and the weights of the graph. For the linear model, the scalars α_i represent all the data. A no, as with the K nearest neighbour, means that the data cannot be compressed and must be used directly, to

compute the distances. Thus a yes means a computation load much lower than with a no. The last column simply shows that with bayesian approaches, the level of pollution has to be a discrete variable.

4. Detection of dangerous smokes

The three previous types of classifiers were used on real data from an industrial plant of XXX, XXX. There are many clouds of low level of pollution, and very rare clouds of high degree of pollution.

The XXX compagny has developped 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, the classifier is used in order to know what is the corresponding degree of pollution, from 0 (no danger), to 3, a very dangerous smoke. We assume that one set of signals concerns only one object, one cloud to classify. The real time component discards small clouds, such as the ones at the bottom of figure 1, and provides data only for the main cloud.

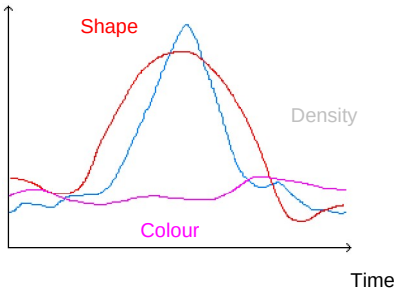

Fig. 4. Evolutions of a few signals.

The real time component detects the beginning and the end of a cloud, and records the whole evolution of all the signals, as given in figure 4. 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.

There are about 3000 clouds recorded in the data base, corresponding to 3 months of activity for a typical plant. Each cloud is described by about 20 signals, such as the colour of the background in the bottom left, the shape, the position of the sun, and so on….

Data from the database have been divided in two parts, one half for the learning step, and one half to compute the errors rates.

5. Comparison on the example

The main tool is the confusion matrix which gives the error rates between the expected level of pollution of a cloud, and the one given by the classifier.

Table 2. Confusion matrix

	Level 0	Level 1	Level 2	Level 3
Level 0	1012	34	2	0
Level 1	23	173	15	0
Level 2	3	40	20	9
Level 3	0	3	11	104

It is clear that the more the confusion matrix is closer to a diagonal matrix, the better results are. The worst results are the off diagonal terms in the lower left corner, or in the upper right one, corresponding either to a missed detection, or to a false alarm. Let us note M_{ij} the element [i, j] of the confusion matrix. Two criteria can bedefined.

The error describes the total error rate.

$$Efficiency = \frac{0.3(M_{13}+M_{31})+0.8(M_{23}+M_{32})+M_{33}}{\sum_{i=0}^{3} M_{i3} + \sum_{j=0}^{2} M_{3j}}$$

The efficiency describes only errors about the false alarms or the non detection, and has been chosen by the XXX company and its partners.

The next tables provides the confusion matrixes 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, $\beta = 1$). A better

efficiency can be obtained for other sets of values.

Table 3. Confusion matrix for the naive bayesian classifier

	Level 0	Level 1	Level 2	Level 3
Level 0	1009	39	0	0
Level 1	57	151	3	0
Level 2	7	53	12	0
Level 3	2	18	9	89

Error: : 12.97 %
Efficiency : 86.1 %

Table 4. Confusion matrix for the Greedy search structure based bayesian classifier

	Level 0	Level 1	Level 2	Level 3
Level 0	1028	15	0	2
Level 1	124	74	13	0
Level 2	14	32	23	3
Level 3	2	3	17	96

Error: : 15.56 %
Efficiency : 91.79 %

Table 5. Confusion matrix for the KNN 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 %
Efficency : 95.08 %

Table 6. Confusion matrix of the linear model

	Level 0	Level 1	Level 2	Level 3
Level 0	1017	29	2	0
Level 1	55	141	15	0
Level 2	5	36	25	6
Niveau 3	2	12	31	73

Error: 13.32 %
Efficency: 85.65 %

These matrices confirm that the k nearest neighbours classifier is the best one in case of low amount of data, here for the level 3 degree of pollution. The linear model does not provide the best model, but this is not surprising since the model can hardly be linear. But this approach shows that it is possible to define a model between the variables, and other non linear models will be defined laterly [4, or a fuzzy model]. We can see that the network obtained using the greedy search algorithm is not better than a naive network in terms of global error. However, differences can be observed concerning the classification of the highest pollution (level 3 degree). In terms of efficiency, the structure obtained using the greedy search could then achieve better performances. However we totally lack of any explanation about the links between the variables. We can be surprised that the greedy search algorithm does not provide errors rates at least equal to the naive bayesian classifier. Indeed the naive graph is a possible choice for the learning algorithm, and this latter should at least find it. This is a well known problem [10], due to the fact that data are missing and the learning step does not test every possible graph.

6. Conclusion

In this paper we have compared three different types of classifiers on an example of detection of pollution by means of cameras. All our classifiers improve the own classifier of the company, XXX, based on a set of rules. We have remarked that the k nearest neighbour classifier was the best, because we lack of data concerning clouds of high degree of pollution. Factorial methods such as principal component analysis will be used in the future to lower the number of signals to use in the detection step and to improve the overall performances

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 XXX company, because the classifier and their real time component are intimately linked.

Acknoledgement

Results presented here are based on a joint work with the company Aloatec and we want to thank their head, Philippe Bourrier.

7. References

[1] J. P. Benzécri. *Correspondence Analysis Handbook.* Marcel Dekker, 1992.

[2] G. F. Cooper. *Expert Systems based on belief networks.* Current Research Directions, 1988.
[3] G. F. Cooper. A diagnostic method that uses casual knowledge and linear programming in the application of bayes' formula. *Computer Methods and Programs in Biomedicine*, 1986.
[4] XXX
[5] D. Heckerman. A tutorial on learning with bayesian networks. *In Learning in Graphical Models,* M. Jordan, ed. MIT Press, Cambridge, MA, 1999.
[6] V. Jensen Finn. *An introduction to bayesian networks.* UCL Press, 1999.
[7] Philippe Leray, Olivier François (2004). BNT Structure Learning Package : documentation and experiments. Technical Report. Laboratoire PSI - INSA Rouen – FRE CNRS 2645.
[8] Philippe Leray, Olivier François (2004). Etude comparative d'algorithmes d'apprentissage de structure dans les réseaux bayésiens. Laboratoire Perception, Système, Information. FRE CNRS 2645.
[9] Ljung L., *Systems Identification: Theory for the user.* Prentice Hall, 1988.
[10] Machine Learning, T.M. Mithcell, Mac Graw Hill, 1997
[11] 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
.
[12] XXX
[13] Roberto G. Cowell, A. Philip Dawid, Stephen L. Lauritzen, and David J. Spigelhalter. *Probabilistic Networks and Expert Systems. Springer,* 1999.
[14] D. J. Spiegelhalter, S. L. Lauritzen, A. P. Dawid and R. G. Cowwel. *Bayesian analysis in expert systems.* Statistical Science, 8: 219-1247, 1993.
[15] D. J. Spiegelhalter, S. L. Lauritzen. Local computation with probabilities on graphical structures and their applications to expert systems. *Proceedings of the Royal Statistical Society,* Series B 50(2), 1988.
[16] D. J. Spiegelhalter, S. L. Lauritzen. Statistical Reasoning and learning in knowledgebases represented as causal networks. *Lecture notes in Medical Informatics,* 36: 105-112, 1988.
[17] A. Storkey. *Learning from Data : Nearest Neighbour Methods.* School of Informatics. University ok Edimburgh.