Constructing Virtual Sensors using Probabilistic Reasoning

Pablo H. Ibargüengoytia and Alberto Reyes

Instituto de Investigaciones Eléctricas
Av. Reforma 113, Palmira
Cuernavaca, Mor., 62490, México
{pibar,areyes}@iie.org.mx

Abstract. Modern control systems and other monitoring systems require the acquisition of values of most of the parameters involved in the process. Examples of processes are industrial procedures or medical treatments or financial forecasts. However, sometimes some parameters are inaccessible through the use of traditional instrumentation. One example is the blades temperature in a gas turbine during operation. Other parameters require costly instrumentation difficult to install, operate and calibrate. For example, the contaminant emissions of power plant chimney. One solution of this problem is the use of analytical estimation of the parameter using complex differential equations. However, these models sometimes are very difficult to obtain and to maintain according the changes in the processes. Other solution is to borrow an instrument and measure a data set with the value of the difficult variable and its related variables at all the operation range. Then, use an automatic learning algorithm that allows inferring the difficult measure, given the related variables. This paper presents the use of Bayesian networks that represents the probabilistic relations of all the variables in a process, in the design of a virtual sensor. Experiments are presented with the temperature sensors of a gas turbine.

1 Introduction

Computers are invading all kinds of human activities given the decreasing costs of software and hardware. Every time, more processes are controlled and monitored automatically by computers. More algorithms have been developed for the efficient and useful treatment of the data acquired. Examples of this include algorithms for automatic learning, intelligent control, all kind of diagnosis and planning. In all these cases, while better is the information and more reliable are the readings of variables, a better performance can be obtained. However, sometimes, the full range of all the information is difficult to obtain. In some cases, the variables to measure can be in inaccessible locations. In other cases, some variables require expensive and complex pieces of instrumentation. For example, a fuel viscosity sensor is expensive and has to be cleaned perfectly every short periods of time, given the nature of the object measured: a dense flow of raw oil[5]. As another example, the control emission monitoring system (CEMs) are expensive equipment that has to be installed at the top of chimney in power plants, but also require calibration every short periods of time. Literature reports the use of predictive emission monitoring system (PEMs) as a common solution for contaminant emissions sensors.

One common solution for the problem of difficult readings is the estimation of these parameters, given other related parameters in the same process. In this research project, this estimation is referred as **virtual sensors**.

One approach used in traditional chemical processes is analytical. This includes the development of complex differential equations that relate the virtual variable with other easier to read parameters [2]. However, this is sometimes very difficult to obtain and the complexity tends to increase when several variables are considered. Also, any small change in the process may represent huge changes in the analytical models that relate these variables. Additionally, analytical models require the participation of high experimented experts of the process. This also, is difficult to find.

Computational intelligence methods have been used in this estimation. For example, neural networks, fuzzy logic and genetic algorithms are used in the estimation of one variable after a training phase. Sometimes, a combination of these methods are utilized in specific environments [3]. However, these combinations result in unique prototypes that are difficult to apply in similar problems.

Another approach consists in the use of artificial intelligence techniques for the development of virtual sensors. If the estimation is based on probabilistic relations of the variables, then Bayesian networks mechanism can be used. This mechanism includes robust and efficient automatic learning algorithms that provide the models, given real data from the process. This approach requires the acquisition of all the variables while the process is operating at full range. This means that, obtaining and calibrating the costly instrument for a few days, it is possible to create a data base with all the information. Later, a probabilistic model can be built using one of the learning algorithms. Finally, estimation of the virtual sensor can be made through probability propagation, once that the related values have been read.

This paper proposes and demonstrates the use of probabilistic reasoning, i.e., Bayesian networks, in the creation of virtual sensors. The main contribution is the procedure to create virtual sensors using real data from the process. The models are obtained off-line and the virtual sensors are utilized on-line. Additionally, Bayesian networks can be used for the design of several virtual sensors given the same data set. This is, inference in Bayesian networks produces posterior probability distributions of all the variables that were no possible to read. Also, this mechanism works even in the absence of some of the related variables. A prototype was constructed and tested with temperature sensors of a gas turbine in a power plant.

This paper is organized as follows: we start describing the Bayesian network mechanism used traditionally in uncertainty management. Section 3 explains the application domain where this virtual sensor wes first developed. Then, we present our approach in the construction of probabilistic models that will be used to create the virtual sensors. Then, section 5 discusses the experiments reported of the temperature of gas turbine virtual sensor. We finally conclude giving new directions of this work.

2 Introduction to Bayesian networks

A Bayesian network (BN) is a graphical representation of dependencies and independencies of random variables for probabilistic reasoning in intelligent systems [4]. Fig. 1 depicts an example of a simplified BN representation of 5 variables, and their relationships. In a BN, each node represents a discrete random variable and each arc a probabilistic dependency. The variable at the end of a link is dependent on the variable at its origin. Thus, the following considerations were taken in the construction of the BN of Fig. 1. The temperature t is caused by the flow of gas $f1$ and the flow of air $f2$ during the combustion. The flow of gas is caused by the gas fuel pressure supply ps and the real fuel valve position pr. Also, this position is caused by the position demand of fuel valve dp. The flow of air $f2$ is caused by the real inlet guide vane position pa and this is caused by the position demand da. This network can be taken as representing the joint probability distribution of the variables t, $f1$,..., da as:

$$P(t, f1, f2, ps, pr, pa, dp, da) = P(t \mid f1, f2)P(f1 \mid ps, pr)P(f2 \mid pa) \\ P(pa \mid da)P(pr \mid dp)P(ps)P(dp)P(da) \quad (1)$$

Equation 1 is obtained by applying the chain rule and using the dependency information represented in the network.

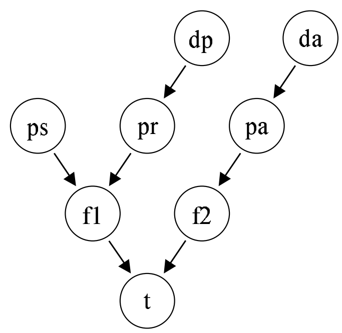

Fig. 1. Example of a polytree representing the causal relation between variables of the gas turbine.

The topology of a BN gives direct information about the dependency relationships between the variables involved. In particular, it represents which variables are conditionally independent given another variable. By definition, X is conditionally independent of Y, given Z, if:

$$P(X \mid Y, Z) = P(X \mid Z) \tag{2}$$

This is represented graphically by node Z "separating" X from Y in the network. In general, Z will be a subset of nodes from the network that if removed will make the subsets of nodes X and Y disconnected. For example, in the BN of Fig. 1, $\{t\}$ is conditionally independent of $\{ps, pr\}$ given $\{f1\}$. To completely specify a BN, the conditional probability of each node given its parents, and the prior probability of the root nodes, are required. That is the terms in equation 1 for the example.

Given a knowledge base represented as a probabilistic network, it can be used to reason about the consequences of specific input data, by what is called *probabilistic reasoning*. This consists in assigning a value to the input variables, and propagating their effect through the network to update the probability of the hypothesis variables. The updating of the certainty measures is consistent with probability theory, based on the application of Bayesian calculus and the dependencies represented in the network. For example, in the BN in Fig. 1, if $f1$ and $f2$ are measured and t is unknown, their effect can be propagated to obtain the posterior probability of t given $f1$ and $f2$.

The probabilistic model in Fig. 1 has a polytree structure, i.e., for any two nodes in the network, there is at most one chain between these two nodes. For singly connected networks, such as trees or polytrees, there is an efficient algorithm for probability propagation [4]. It consists on propagating the effects of the known variables through the links, and combining them in each unknown variable. This can be done by local operations and a message passing mechanism, in a time which is linearly proportional to the diameter of the network. The more complete Bayesian network representation is multiply connected network. For this kind of networks, there are alternative techniques for probability propagation, such as clustering, conditioning, and stochastic simulation [4].

Bayesian networks can be used to represent the dependency relations between the measurements, and obtain their posterior probabilities given the evidence of other measured variables. The next section presents the use of Bayesian networks in the construction of the virtual sensor.

3 Application domain: gas turbines

The virtual sensor approach was evaluated by applying it to the estimation of one temperature sensor of the gas turbine at the *Gómez Palacio* power plant in México. This is an interesting application of these techniques for many reasons. For example, since an analytical or functional model of the temperatures of a turbine is difficult to obtain, it is a good candidate for probabilistic methods. Additionally, some of the temperatures of gas turbine are indeed very difficult to measure. For example, the temperature of the inner blades of the turbine.

Finally, the size of this problem makes it ideal for testing the development of the prototype. Figure 2 shows a simplified diagram of a gas turbine.

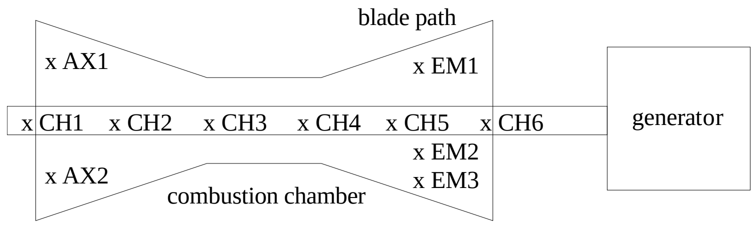

Fig. 2. Simplified schematic diagram of a gas turbine.

The combustion chamber receives air and gas in a specific proportion to produce high pressure gases at high temperature. These gases produce the rotation that moves the generator. Thus, the temperature is considered the most important parameter in the operation of the turbine since it performs more optimally at higher temperatures. However, a little increase in the temperature, over a permitted value, may cause severe damage. The distributed control system that governs the plant is continuously monitoring these signals in order to correct any deviation of the process. In the case of an illegal increase of a temperature parameter, the plant is stopped and taken to a safe state. Conversely, an error in a sensor's measure may cause an unnoticed increase of the temperature, or may result in an unnecessary shut down. The consequences of the former can be severe damage to the equipment and even human fatalities, and the latter could result in loss of time and fuel. Figure 2 shows the physical location of some of the temperature sensors used in the turbine. It shows six sensors across the beadings of the shaft ($CH1, CH2, \ldots, CH6$), three sensors on the turbine blades ($EM1$, $EM2$ and $EM3$), and two sensors of the temperature of the exciter air ($AX1$ and $AX2$). The experiments were carried out over a set of 21 sensors (though not all are shown in Fig. 2). These sensors can be grouped into the following sets of measurements:

- 6 beadings (CH1 - CH6),
- 7 disk cavities (CA1 - CA7),
- 1 cavities air cooling (AEF),
- 2 exciter air (AX1 - AX2),
- 3 blade paths (EM1 - EM3), and
- 2 lub oil (AL1 - AL2).

The instrumentation of the plant provides the readings of all the sensors every second. The data set utilized in the experiments corresponds to the temperature readings taken during approximately the first 15 minutes after the start of the combustion. That corresponds to the start up phase of the plant, where the thermodynamic conditions change considerably. Therefore, the data set consists of 21 variables and 870 instances of the readings.

4 Constructing the virtual sensor

The idea in the construction of a virtual sensor is to suppose that one of these sensors will be a virtual sensor. Then, estimation of the value can be made and later, a comparison of the real value. This allows evaluating the proposed approach.

The first step in the design of a virtual sensor is to provide a dependency or probabilistic model. A dependency model for the temperatures was obtained by utilizing an automatic learning program included in the Hugin package [1, 6]. Fig. 3 shows the network obtained with this data set.

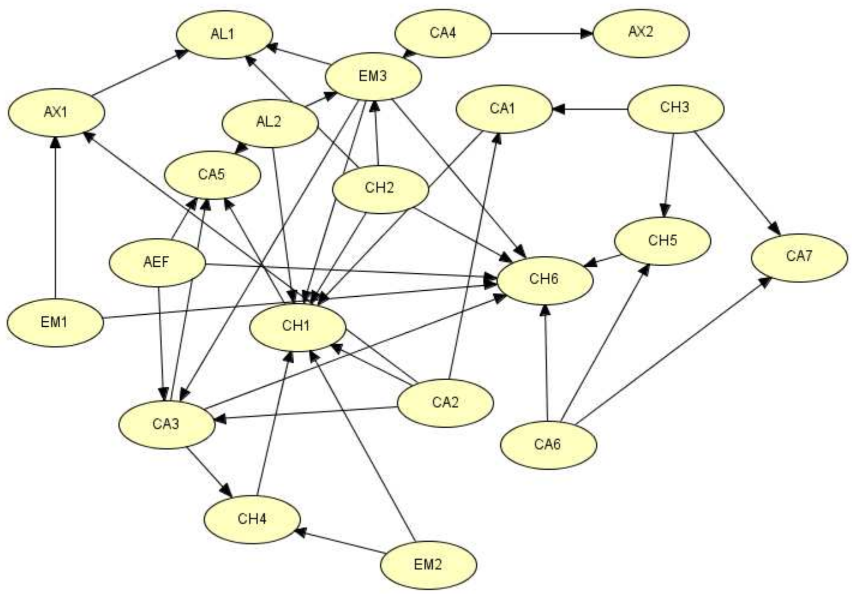

Fig. 3. Learned Bayesian network for the gas turbine temperature sensors.

In this project, discretization was necessary since all these variables are continuous valued. In this preliminary experiments, a very simple discretization was chosen. For most of the variables, the full range of values were divided in 5 intervals, except the turbine blades temperatures ($EM1$, $EM2$ and $EM3$) that were divided in 10 intervals.

Now, from the 870 readings of all the sensors, a partition of the data set was made: one partition for training the network (obtaining the initial conditional and prior probabilities), and the other partition for testing.

The testing procedure consists in the instantiation of all variables except $EM1$, and then propagate probabilities in the network, This propagation produces a posterior probability distribution of $EM1$ given all the related sensors. In the testing stage, this probability distribution is compared with the known value of $EM1$. Table 1 presents the final evaluation of the prototype.

Table 1. Results of selected experiments for the estimation of EM1.

AEF	$AX1$	$CA3$	$CA6$	$CH2$	$CH5$	$CH6$	$EM3$	$EM1$	$virtualEM1$	Probability
112	23.9	195.1	260.6	99.4	70.9	67.9	934.0	894.4	888-916	39
117.0	23.9	195.1	271.1	116.3	75.0	74.1	1007.8	974.4	973-1002	91
122.1	23.9	166.5	292.6	128.4	77.1	80.2	1054.4	1031.1	1002-1031	67
132.2	23.9	204.7	314.2	137.2	83.2	91.2	1002.5	993.3	973-1002	91
163.4	26.3	172.2	319.9	137.2	87.3	106.0	844.3	855.7	830-859	37
196.1	27.5	158.1	325.6	133.1	91.4	112.2	793.8	799.5	773-802	95
232.9	29.9	152.9	336.6	131.1	95.5	118.4	777.0	781.4	773-802	89
269.8	31.1	165.2	341.9	131.1	97.6	124.5	763.0	763.4	744-773	91
305.2	33.5	190.2	347.6	133.1	99.6	130.7	750.2	763.4	744-773	94
347.6	35.9	208.7	358.1	137.2	103.7	136.8	742.8	745.0	744-773	94

The first 8 columns show a selection of testing cases for the estimation of $EM1$. These 8 variables are sufficient for the estimation of $EM1$ since they represent the *Markov blanket* of $EM1$. A Markov blanket (MB) is defined as the set of variables that makes a variable independent from the others. In a Bayesian networks, the following three sets of neighbors are sufficient for forming a MB of a node: the set of direct predecessors, direct successors, and the direct predecessors of the successors (i.e. parents, children, and spouses) [4]. Independence in this case means that given the values of the MB of a node, the rest of the values are completely unnecessary for the propagation. Thus, the MB of $EM1$ according to Fig. 3 is formed by its children $AX1$ and $CH6$, plus the parents of its children (spouses): $CA2$, AEF, $CA3$, $CA6$, $CH2$, $CH5$ and $EM3$. Notice that $EM1$ has no parents. The 9th column shows the real value of $EM1$ in the data set. The next column shows the discretized values corresponding to the real value of $EM1$, and the last column shows the resulting probability of the corresponding discretized value after the propagation. Notice that in 6 of the examples, the probability is higher than 90%.

5 Discussion

The preliminary experiments described in Table 1 look promissory. Most of the examples have accuracy higher than 90%. In the rest of the cases, the posterior probability distribution was wider. However, the real value coincides with the interval of higher probability (37 % or 67 %). This is due principally to the lost

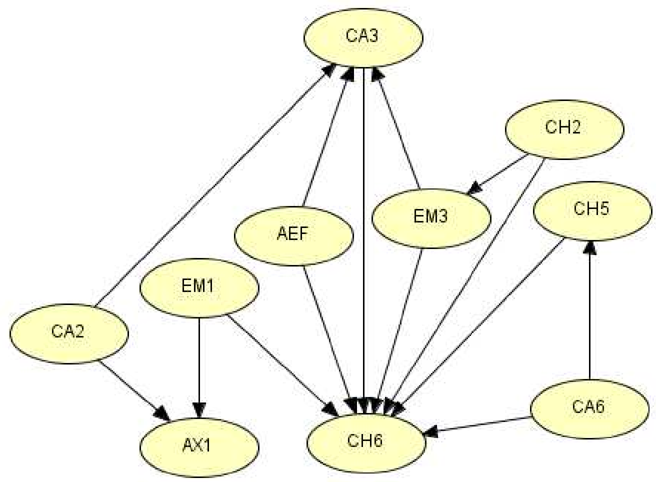

Fig. 4. Sub-network corresponding to the Markov blanket of node EM1.

of precision caused by the discretization process. Five intervals may be few for variables with large variation, but the computational cost is maintained low. If higher precision is required, better discretization approaches can be utilized, but the computational cost would increase. This cost is with respect to memory storage and the propagation time. In a Bayesian network, a conditional probability table is defined with all the values of a node, given all the combination of the values of its parents. Thus, a node with 5 possible values, and 8 parents with 5 values each, represents a table with 1,953,125 entries. In this example, node $CH6$ has 7 parents (see Fig. 4).

For a better performance, it is advisable to prove with different discretizations schemes or more intervals. With this, models result in more exact representation of the signals behavior in the range of operation of the process.

The solution of this problem is to keep the probabilistic model with the lowest interconnectivity possible. The problem is not the number of nodes but the number of arcs between them. While the Markov blanket of the virtual sensor is smaller, lower is the computational cost. The simplest case is to learn a tree structure, i.e., a network where all the nodes can have at most one parent. In this case, discretization can be carried out with larger number of intervals. Notice that the structure learned in this experiment, shown in Fig. 3 is very interconnected, so the number of intervals had to be maintained low. However, in the present days, dealing with models that utilize mega bytes of memory is a common task.

6 Conclusions and future work

This paper has shown the construction of virtual sensors using probabilistic reasoning. Borrowing a real sensor, readings can be collected while the process is being executed. Later, a model is learned automatically using the PC algorithm. Finally, the virtual sensor is executed on-line, estimating the corresponding value. This paper utilized real data from temperature sensors in a gas turbine of a power plant. With all the readings, one variable is supposed to be absent, so estimation and evaluation can be made. A prototype was built and tested with promising results.

Future work will be done in the construction of predictive emission monitoring system (PEMs) that will be used in most of the boilers of electric power plants from the Federal Commission of Electricity (CFE) in Mexico. Also, this sensor can be also be installed in other king of boilers in the chemical industry, including petroleum.

Viscosity of fuel virtual sensor is also being studied and designed and it is expected to be installed in many of the ducts of gas in Mexico.

Acknowledgments

Thanks to the anonymous referees for their comments which improved this article. This research is supported by a grant from IIE.

References

1. S. K. Andersen, K. G. Olesen, F. V. Jensen, and F. Jensen. Hugin a shell for building bayesian belief universes for expert systems. In *Proc. Eleventh Joint Conference on Artificial Intelligence, IJCAI*, pages 1080–1085, Detroit, Michigan, U.S.A., 20-25 August 1989.
2. Dorsey A.W. and Lee J.H. Building inferential prediction models of batch processing using subspace identification. *J. Proc. Control*, 13:397–406, 2003.
3. Hanai et. al. Analysis of initial conditions for polymerization reaction using fuzzy neural network and genetic algorith. *Com. Chem. Eng.*, 27:1011–1019, 2003.
4. J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference.* Morgan Kaufmann, San Francisco, CA., 1988.
5. Aplein Ingenieros S.A. Medida en línea de viscosidad en blending de fuel oil. Technical report, Aplein Ingenieros S.A., 2006.
6. H. Steck, R. Hofmann, and V. Tresp. Concept for the pronel learning algorithm. Technical report, Siemens AG, Munich, 1999.