About automatic learning for advanced sensing, monitoring and control of electric power systems

Louis Wehenkel, Mevludin Glavic, Pierre Geurts, Damien Ernst
Department of Electrical Engineering and Computer Science
University of Liège - Sart-Tilman B28 - B-4000 Liège
{L.Wehenkel,P.Geurts,dernst}@ulg.ac.be , glavic@montefiore.ulg.ac.be

Abstract - The paper considers the possible uses of automatic learning for improving power system performance by software methodologies. Automatic learning per se is first reviewed and recent developements of the field are highlighted. Then the authors' views of its main actual or potential applications related to power system operation and control are described, and in each application present status and needs for further developments are discussed.

Keywords - Automatic learning, sensing, monitoring, control, electric power systems

1 INTRODUCTION

In the field of power systems automatic learning was first proposed by Tom DyLiacco in the late sixties, in the specific context of on-line security assessment [1]. Since then, automatic learning has been applied by the academic community to many other power system problems, including load forecasting, equipment monitoring, expansion planning, and automatic control. While electric load forecasting has become a standard application of automatic learning, in the field of decision making in operation and control of power systems, real-life applications have been scarce, in spite of a very significant and successful research effort since the mid eighties.

Having been involved in research in power system monitoring and control on the one hand, and automatic learning and data mining on the other hand, and much with the application of the latter to the former, our main objective in this paper is to provide our view on promising applications of automatic learning in the context of advanced sensing, monitoring and control of electric power systems, and to suggest areas for further development, as well as guidelines to take better advantage of the available methods in pratice.

To fix ideas, we start the paper with a quick review of what automatic learning and data mining are all about, introducing the main learning problems, protocols and terminology and reviewing the main results of research in the field while providing some pointers to the relevant literature. The reader already familiar with automatic learning, be it at an intuitive level, can skip this section.

The body of the paper is composed of several independent sections reviewing different types of applications that we deem relevant for the future, although they are currently at very different levels of maturation. Each one of these sections has its own discussion and conclusions. Currently, this is a working paper without much references to existing work in the field. At a later stage, we intend to complete the survey by a more systematic review of the litterature in the field.

2 AUTOMATIC LEARNING PER SE

Generally speaking, automatic learning aims at exploiting data gathered from observations (or simulations) of a system (or an environment), in order to build models explaining the behavior of the system and/or decision rules to interact in an appropriate way with it.

In what follows, we first describe the three main automatic learning problems, then we review different protocols, and we provide a short discussion of the relation of automatic learning to other fields.

2.1 *Types of automatic learning problems*

To introduce the three main types of automatic learning problems (supervised, reinforcement, unsupervised), we will use the probabilistic/statistical formalization and terminology. We refer the interested reader to more general textbooks for further information about automatic learning theory, its relation to other disciplines, and the precise description of the algorithms that we only mention [2, 3, 4, 5, 6].

2.1.1 *Supervised learning problem*

Given a sample $\{(x^i, y^i)\}_{i=1}^N$ of input-output pairs, a supervised learning algorithm aims at automatically building a model $\hat{y}(x)$ to compute approximations of outputs as a function of inputs. Belong to this category methods like decision trees, neural networks, linear regression etc.

The standard probabilistic formalization of supervised learning considers $x \in X$ and $y \in Y$ as two random variables drawn from some probability distribution $P_{X,Y}$ defined over $X \times Y$, a loss function ℓ defined over $Y \times Y$, and a hypothesis space $\mathcal{H} \subset Y^X$ of input-output functions, and measures the inaccuracy (or average loss) of a model $f \in \mathcal{H}$ by

$$L(f) = \int_{X \times Y} \ell(y, f(x)) dP_{X,Y}.$$

Denoting by $(X \times Y)^*$ the set $\bigcup_{N=1}^{\infty} (X \times Y)^N$ of all finite size samples, a (deterministic) supervised learning algorithm A can thus formally be stated as a mapping

$$A : (X \times Y)^* \rightarrow \mathcal{H}$$

from $(X \times Y)^*$ into the hypothesis space $\mathcal{H}$. For any sample $ls \in (X \times Y)^*$ we will hence denote by $A(ls)$ the model returned by the algorithm A. Assuming that samples $ls^N = \{(x^i, y^i)\}_{i=1}^N$ are drawn according to some sampling distribution $P_{(X,Y)^N}$, the sampling process and algorithm induce a probability distribution over the hypothesis space and hence a probability distribution over inaccuracies $L(A(ls^N))$. Let us denote by

$$\overline{L}_A^N = \int_{(X,Y)^N} L(A(ls^N)) dP_{(X,Y)^N}$$

the expected average loss of A for fixed sample size N, by

$$L_{\mathcal{H}}^* = \inf_{f \in \mathcal{H}} L(f)$$

the lowest reachable average loss in $\mathcal{H}$, and by

$$L^* = \inf_{\mathcal{H} \subset Y^X} L_{\mathcal{H}}^*$$

the lowest possible average loss.

Besides defining general conditions (on X, Y, $P_{X,Y}$, $P_{(X,Y)^N}$, ℓ, $\mathcal{H}$, A etc.) under which the above introduced quantities indeed exist, the objective of statistical learning theory is essentially to study whether or in what sense $\overline{L}_A^N$ and $L(A(ls^N))$ converge to $L_{\mathcal{H}}^*$ [7].[1]

On the other hand, the design of supervised learning algorithms essentially aims at constructing sequences of hypothesis spaces $\mathcal{H}_n$ and learning algorithms A_n with good convergence properties and such that $L_{\mathcal{H}_n}^* \to L^*$. In particular, much of the research in supervised learning has focused on the design of algorithms which scale well in terms of computational requirements with the sample size and with the dimensionality of the input and output spaces X and Y, and which use "large" hypothesis spaces able to model complex non-linear input-output relations. From this research two broad classes of algorithms have emerged during the last fifteen years, based respectively on kernels [8, 9] and on ensembles of trees [10, 11].

2.1.2 Reinforcement learning problem

Given a sample of trajectories of a system

$$\{(x_0^i, d_0^i, r_0^i, x_1^i, \dots x_{h_i-1}^i, d_{h_i-1}^i, r_{h_i-1}^i, x_{h_i}^i)\}_{i=1}^N,$$

reinforcement learning aims at deriving an approximation of an optimal decision strategy $\hat{d}^*(x,t)$ maximizing system performance in terms of a cumulated performance index over a certain horizon h, defined by

$$R = \sum_{t=0}^{h-1} \gamma^t r_t,$$

where $\gamma \in (0, 1]$ is a discount factor. In this framework, x_t denotes the state of a dynamic system at time t, d_t is the control decision applied at time t, and r_t is an instantaneous reward signal [12, 13].

From a theoretical point of view, reinforcement learning can be formalized within the stochastic dynamic programming framework. In particular, supposing that the system obeys to a time invariant dynamics

$$x_{t+1} = f(x_t, d_t, w_t),$$

where w_t is a memoryless and time invariant random process and obtains a bounded time invariant reward signal

$$r_t = r(x_t, d_t, w_t),$$

over an infinite horizon ($h \to \infty$), one can show that the two following equations define an optimal decision strategy

$$Q(x, d) = E\{r(x, d, w) + \gamma \max_{d'} Q(f(x, d, w), d')\},$$

$$d^*(x) = \arg\max_d Q(x, d).$$

Reinforcement learning can thus be tackled by developing algorithms to solve these equations (or their time-variant and finite horizon counterparts) approximately when the sole information available about the system dynamics and reward function are provided by a sample of system trajectories. The theoretical questions that have been studied in this context concern the statement of conditions on the sampling process and on the learning algorithm ensuring convergence to an optimal policy in asymptotic conditions (i.e., when $N \to \infty$).

Recent work in the field has allowed to take full advantage from state-of-the art supervised learning algorithms by defining appropriate frameworks to plug these algorithms in the reinforcement learning protocol. In particular, model based reinforcement learning methods use the sample to build approximations of the system dynamics and reward function and dynamic programming methods to derive from them an approximation of the optimal decision strategy. On the other hand, the Q-learning framework uses supervised learning in order to construct from the sample an approximation of the Q-function and derive from it the decision policy. While the first generation of Q-learning methods used parametric approximation techniques together with on-line gradient descent [14], the recently proposed fitted Q iteration method allows to fully exploit any parametric or non-parametric batch mode supervised learning algorithm in this context [15].

Notice that even when the system dynamics and reward functions are known (or can be simulated), the reinforcement learning framework may still be used as an alternative to direct optimization (e.g. dynamic programming or model predictive control), by extracting decision policies from samples generated automatically by Monte-Carlo simulation. In this context, the advantages of reinforcement learning are its capability to exploit efficiently large samples and cope with high-dimensional non-linear and stochastic problems.

[1] Notice that while originally, statistical learning theory was developed in the late seventies and eighties under the classical assumption of i.i.d. sampling according to the distribution $P_{X,Y}$, i.e. under the assumption that $P_{(X,Y)^N} = P_{X,Y}^N$, more recent work aims at weakening the assumptions to cases where the samples are not independently distributed anymore [3].

2.1.3 Unsupervised learning problems

Given a sample of observations $\{y^i\}_{i=1}^N$ obtained from a certain sampling distribution P_Y over a space Y, the objective of unsupervised learning is essentially to determine an approximation of the sampling distribution. In the most interesting case, Y is a product space $Y_1 \times \cdots \times Y_n$ defined by n discrete or continuous random variables, and the main objective of unsupervised learning is to identify the relations among these latter (independance relations, colinearity relations) as well as the parameters of their distributions.

Earlier work in this field concerned clustering, principal component analysis and hidden Markov models. More recent research topics, still very active today, concern independent component analysis as well as the very rich field of graphical probabilistic models, such as Bayesian belief networks.

Independent component analysis aims at explaining the observed variables y_i as linear combinations

$$y_i = \sum \beta_{i,j} x_j,$$

where the x_j are independent source variables.

Bayesian networks model the joint distribution of the random variables as a product of conditional distributions

$$P(y_1, \ldots, y_n) = \prod_{i=1}^{n} P(y_i | Pa(y_i)),$$

where $Pa(y_i)$ denotes for each variable a subset of so-called *parent* variables [16, 17]. The parent-child relation is encoded in the form of a directed acyclic graph, which explicitly identifies conditional independence relationships among subsets of variables. Unsupervised learning of Bayesian networks aims at identifying from a sample of observations the structure of the parent-child relationship and for each variable the parameters defining the conditional probability distribution $P(y_i|Pa(y_i))$ [18]. A more sophisticated version of this problem, currently subject of active research, consists of introducing so-called hidden variables into the model and defining the probability model over the observed variables as a marginalization of the following form [19]

$$P(y_1, \ldots, y_n) = \sum_x P(y_1, \ldots, y_n, x_1, \ldots x_m),$$

where the sum extends over all configurations of the m hidden variables x_i. Notice that a particular case of this type of model is the so-called hidden Markov model where the joint distribution $P(x, y) = P(y_1, \ldots, y_n, x_1, \ldots x_n)$ of observed and hidden variables factorizes as follows

$$P(x, y) = P(x_1) P(y_1|x_1) \prod_{i=2}^{n} P(x_i|x_{i-1}) P(y_i|x_i).$$

In this particular case, the identification of the structure of the model reduces to the determination of the number of states (i.e. the number of possible values of the variables x_i) and efficient learning algorithms for this have already been developed several decades ago [20].

2.2 Review of different learning protocols

In the above description of the different automatic learning problems, we have assumed that the learning algorithm uses a whole batch of samples to build its model. In this subsection we review adaptations of these algorithms needed to cope with practical conditions when it is not possible (or not desirable) to assume that all the samples are available (or should be collected) beforehand.

2.2.1 Batch mode vs on-line mode learning

In many practical applications samples are provided one by one and it is useful to consider so-called on-line learning algorithms which essentially generate a sequence of models in the following way

$$m_i = A(m_{i-1}, z^i)$$

where m_0 is an initial model, and z^i stands for input-output pairs (x^i, y^i) in supervised learning, for system transitions $(x_t^i, d_t^i, r_t^i, x_{t+1}^i)$ (or longer trajectories) in reinforcement learning, and for observation vectors y^i in unsupervised learning.

A typical example of this situation concerns a learning agent interacting with a system and collecting continuously information about the system behavior subject to the decisions taken by the agent. Ideally, such an agent should be able to adapt its decision policy at each time step in constant time, and with bounded memory requirements, as soon as a new observation becomes available. Furthermore, if the system is not stationary, the agent should also be able to forget obsolete information collected in remote past so as to adapt its learning on the most recently acquired observations.

Typically, the computational constraints of on-line learning imply the use of simple parametric models by the learning agent. However, the investigation of appropriate tradeoffs between these computational requirements and the flexibility of the used hypothesis spaces deserves further research, so as does the formalization of adaptive learning strategies.

2.2.2 Passive vs active learning

In the above description we have also assumed that the learning algorithm can not influence the sampling process and is purely passive. However, in many practical (e.g. on-line) situations it is possible and interesting to influence the sampling process so as to speed up learning and reduce time and cost implied.

Active learning is a quite rich research field aiming at the design of algorithms which are able to interact with the sampling mechanism in order to influence the information gathering process and thereby speed up learning [21]. This area is strongly related to optimal experiment design [22] and dual control methods [23].

2.3 Discussion

As it may be clear from the previous overview, automatic learning tackles essentially classical modeling problems of statistics. However, while classical statistics has much more focused on the analytical study of parameter identification, assuming that the functional forms of distributions are already known, automatic learning has much more focused on the design of data driven algorithms, which are generally not exploiting any strong parametric assumptions and hence can in principle cope with a larger class of modeling problems [24].

In automatic learning many algorithms have been originally designed in a heuristic way and were initially studied only empirically, by applying them to synthetic or real-life datasets and comparing their results with those of other methods. The developments in computer hardware, the availability of large databases and the good empirical performances of these algorithms made them become more and more popular in practice. During the last twenty years, statisticians and theoretical computer scientists became more strongly interested in this field and they drove significant theoretical research allowing to better understand the behavior of these algorithms, and even improve their design thanks to this new insight [3, 4, 7, 10].

In practice, many different types of methods exist today which are able to cope with millions of samples and/or millions of dimensions.

Further work is focusing on developing tailored algorithms well suited to handle specific classes of practical problems, like time-series forecasting, image and text classification for instance, where the input (and/or the outputs) have specific properties [25, 26].

2.4 A note on automatic learning vs data mining

Data mining aims at extracting relevant and useful information from large bodies of data [27, 28]. As such, it is one of the main application fields of all automatic learning algorithms. Data mining focuses typically on applications where a field expert uses various algorithms together with his domain knowledge to extract information from very large bodies of data. In addition to theoretical accuracy of automatic learning methods it is thus also concerned with interpretability, scalability and validation of results through interaction with the field expert.

In the last years, data mining has been one of the main drivers for research in automatic learning.

3 SECURITY ASSESSMENT STUDIES

The application of automatic learning to power system security assessment aims at extracting decision rules allowing to identify the main weak points of the system, to quickly assess its security, and if necessary to choose appropriate actions in order to reduce the risk of insecurity. In this context, the datasets are generally not obtained from real-life measurements, rather they are generated automatically by Monte-Carlo simulations using existing security assessment tools [29, 30].

3.1 Methodology

The methodolgy consists essentially of three steps:

1. *Database generation.*

   The goal of this step is to screen a representative set of operating scenarios in order to gather detailed information about the capability of the studied system to face disturbances. Each security scenario is specified by three components: an operating point specification; a disturbance (or contingency) specification; a description of the static and dynamic modeling assumption.

   For a given security study, the database generation consists of two successive steps. The first step aims at specifying the range of conditions that will be screened (in the form of a set of independent parameters and the probability distributions that will be used for sampling them) and the type of information that will be extracted from each scenario (in the form of a set of attributes describing the system pre-fault conditions and its post-fault dynamic behavior). The second step consists of sampling a given number of scenarios and carrying out the time-domain simulations and extracting the selected variables and storing them into the database. This latter purely automatic step can take advantage of a grid-computing infrastructure to speed up the database generation process.

   Typically, the independent parameters that are screened are composed of two types of parameters: primary parameters of which the study aims at evaluating the effect on the security of the system (e.g. load level, generation dispatch, topological conditions etc.); secondary parameters which reflect uncertainties with respect to which the outcomes of the study should be robust (e.g. external system conditions, detailed load distribution, uncertain dynamic models of load and protection systems).

   As concerns the range of attributes extracted from each simulation, they depend also on the particular target of the study. For example, in a preventive security assessment study, where the goal is to define safe operating limits expressed in terms of parameters that are meaningful to the operator, the attributes will cover on the one hand pre-fault variables such as powerflows and injections, topological conditions and voltages, and on the other hand a security margin measuring how acceptable the post-fault behavior is. On the other hand, in the context of emergency control, where the goal is to determine triggering rules for emergency control expressed in terms of post-fault measurements, the attributes will also provide detailed information about the post-fault behavior of the system.

2. *Application of automatic learning.*

   The quality of the information that can be extracted from a database strongly depends on the number and representativity of the scenarios it contains. Thus, the first step of data analysis consists in validating the database information by analysing the distributions of attributes and number of scenarios of different types found in the database. At this stage, the kind of tools that are useful are mostly unsupervised methods and graphical visualization tools such as histograms and scatter plots. If the database is not sufficiently representative this analysis should lead to recommendations allowing to modify the database specification.

The second step of analysis consists of using supervised learning methods in order to extract from the database decision rules allowing to determine the security level of the system as a function of pre-fault or post-fault attributes. To this end, a subsample of scenarios is chosen and among the attributes stored in the database a subset is defined as candidate attributes (input variables) and an output variable is selected among the computed security indicators (margins, classes as appropriate). Once these are defined, different supervised learning algorithms may be applied to the corresponding dataset, and their accuracy is estimated by cross-validation on an independent test sample. Eventually, this analysis allows to identify among the candidate attributes those that really influence the security level of the system and to build decision rules using them as inputs. It allows also to assess the learned rules by comparing them with those rules previously used in operation.

Notice that at this stage, the analysis generally starts with decision or regression tree induction, since these latter are able to quickly identify the most salient and informative attributes among a large number of candidate ones, thus allowing one to reduce the dimensionality of the problem and provide more easily interpretable information. However, since tree induction is often suboptimal from the accuracy point of view, it is also very useful to apply more sophisticated techniques such as neural networks, kernel based methods or ensemble methods, so as to have a more precise idea of the residual error due to the influence of external system conditions, detailed load distribution and dynamic models which can not be taken into account in the decision rules since they are not available in the context where the rules are going to be used.

3. *Validation, exploitation and maintenance of extracted information.*

   At this step the goal is to decide whether the decision rules extracted during the study should indeed be exploited in operation (or used to change the settings of emergency control devices). Beyond the effect on security, it is also necessary to evaluate the potential effect of the new rules in terms of induced costs, and it is often required to translate the rules into a directly exploitable form for decision making.

   Finally, maintenance of the extracted decision rules is necessary when the system conditions change significantly with respect to the range of conditions screened during the previous study. Depending on the focus (or broadness) of the study, maintenance may be necessary at more or less frequent intervals. Notice however that while an initial study is generally rather time consuming, the maintenance of the decision rules is typically much more incremental and fast to carry out.

3.2 Status

This approach was first proposed by Tom DyLiacco in the late sixties, in order to develop fast enough on-line methods for preventive dynamic security assessment. Research in this field was carried out mainly during the eighties and early nineties, leading to a mature and wide ranging methodology presently used by several (although not many) system operators for planning and operational planning studies.

In particular, a joint project between RTE (French system operator) and National Grid (English system operator) called ASSESS, has led to the development of a software platform combining scenario specification, sampling and simulation, with data mining and reporting tools specifically targeting these kind of studies. This tool is presently used both for operation planning and system expansion planning by several European TSOs.

The above described methodology is a sound and highly scalable approach for carrying out security assessment studies in complex and uncertain environments, such as electric power systems.

3.3 Further work

Many TSOs have developped in the past Monte-Carlo simulation tools used for expansion planning under uncertainties. While these tools are typically only extracting synthetic information (such as estimates of expectations and variances of costs and reliability indices), they could be upgraded by combining them with data mining tools in order to extract more refined information about conditions leading to high costs or low reliabliity, and thereby help engineers to take better advantage of their system.

The described approach could allow to assess the effect of uncertainties due to limited amount of information available for decision making on the security/economy tradeoff. It thus would provide a systematic means to assess off-line how to decompose security assessment and control over a large interconnection into well suited subproblems, and to identify which information to exchange among the corresponding decision making entities so as to ensure reliable control. More generally, it could provide a systematic approach to assess the robustness of the power system dynamic behavior in unusual conditions and how this robustness is affected by various parameters under control of the designer.

Today, the methodology is used mainly in off-line planning kind of studies, typically several weeks or months ahead of time. However, it could as well be used in day-ahead studies or even on-line to support operator decision making. Nevertheless, the effective use of the methodology requires a shift of paradigm with respect to traditional deterministic tools, and this needs significant education efforts among power system engineers.

Also, in security assessment studies the application of batch-mode reinforcement learning could be of value in order to design decision policies from Monte-Carlo simulations, when there exists no alternative way to determine the optimal decisions.

Finally, the systematic use of this methodology is possible only if significant investment is made in terms of software tools and computational infrastructure.

4 AUTOMATIC CONTROL SYSTEM DESIGN

Since security assessment studies essentially aim at providing decision aids to human operators its results must be interpretable and compatible with the information available in a control center. Given the large amount of available inputs and of possible decisions, the main goal of security assessment studies is to reduce complexity by identifying a subset of relevant inputs and decisions.

On the other hand, the design of an automatic control device has typically different requirements, related to limited (often local) data acquisition and strong time constraints. Thus in a typical automatic control system design application, the number of available measurements is much smaller, the control signal is already defined, and the problem to be tackled is already well circumscribed.

4.1 Methodology

Reinforcement learning application to the design of the control policy of an automatic control device is essentially composed of three steps.

1. *Formulation of the optimal control problem.*

   The problem formulation essentially aims at defining a pseudo-state and reward signal that can be computed from available measurements. Typically, these measurements do not provide direct observability of the full system state, and it is better to use as pseudo-state an information vector computed from present and past measurements and past control signals.

   The reward signal on the other hand should reflect the control objective and penalyze undesired situations (e.g. violation of stability or safety constraints).

2. *Off-line gathering of data and initial learning.*

   Generally, when a new device is put in operation, the first stage of designing the control policy should be based on simulated scenarios. If an existing controller is already working on the system, and the objective is to redesign its control policy, past measurements related to this controller could also be used at this stage. In either case, batch-mode reinforcement learning can then be applied to samples of trajectories in off-line mode, until the performance of the controller is sufficiently good.

   Just like in the security assessment studies this off-line tuning needs a good experiment design and a careful validation of the resulting controller, and systematic comparisons of alternative designs.

3. *On-line learning and control.*

   Once the control agent is plugged in the system, it uses its policy in order to control the system in closed-loop fashion. If the system conditions change, the controller becomes suboptimal and eventually needs to be retuned. This can either be done off-line or on-line depending on the amount of computing power that can be made available to the control agent. In both cases the learning agent can exploit real measurements collected from the system measurements during the time the control agent has been in operation.

4.2 Status

Up to now, work on reinforcement learning application to power system automatic control has been carried out exclusively in the academic context, based on simulations with small sized systems in well defined conditions. The applications considered concerned the damping control by TCSC devices and under-frequency load-shedding agents [31].

The main progress in the last years came from the research in reinforcement learning itself, with the design of new algorithms able to extract more efficiently information from system trajectories. In power systems, some recent studies aimed at assessing the advantage of reinforcement learning based control with respect to model predictive control and other more classical deterministic control methods [32, 33].

4.3 Further work

Further significant amount of work is required in order to highlight the intrinsic advantages of reinforcement learning methods, which stem from their capability to handle stochastic conditions and to adapt automatically their control policy to changing system conditions, and to convince the power systems engineering community of the usefulness of this approach to help designing for increased robustness and optimality the numerous old and new automatic control devices that interact through the power system.

Within this context it is important to notice the fact that the learning controllers will operate in a highly distributed multi-agent context, and that the theory of multi-agent reinforcement learning is presently only starting to be developed.

5 APPLICATION TO FORECASTING

5.1 Methodology

Forecasting essentially aims at predicting the value of some quantity at some future instant, given present and past measurements of this quantity and some exogenous variables which may affect the behavior.

From the viewpoint of automatic learning, forecasting is thus basically a supervised learning problem, and supervised learning methods may be viewed as alternative solutions to be compared or combined with classical time-series forecasting techniques.

5.2 Status

System load forecasting has been one of the most successful applications of supervised and unsupervised learning to electric power systems. More recently market price forecasting and wind forecasting have been investigated along the same lines.

5.3 Further possibilities

Within the context of power system monitoring and control, short term and local load and wheather forecasting tools could be very useful in order to enhance decision making.

It would therefore be worth to analyse the potential usefulness of automatic learning in this context, where it would not be practical to use a lot of human expertise to design a forecasting model for each individual load or geographical area.

6 EXTERNAL EQUIVALENT MODELS

6.1 Suggestion

Dynamic security assessment as well as system design studies carried out by a system operator, rely on the quality of the models that are used to represent subsystems which are not directly under his control and whose internal state is not monitored by him, such as distribution subsystems and interconnected neighboring transmission systems.

The operators of the external systems can collect information about interface variables and they could also enrich these measurements by providing measurements corresponding to a richer set of simulated conditions. Using such datasets, it would in principle be possible to construct, by supervised learning, synthetic input-output models relating the dynamics of input signals to those of outputs. The same could be done to improve models used to represent large industrial plants in system studies.

6.2 Further work

To our knowledge not much, if any, work has been carried out in the direction of designing external equivalent models by automatic learning. Nevertheless, we believe that the need for increased quality equivalents is strongly felt and that their availability would be an interesting alternative to the use of centralized (and more and more complex to operate and maintain) wide area data acqusition systems.

Further research work should allow to assess the possibility of accurately representing the dynamics of a large transmission system seen from outside using automatically learned black-box models.

7 DESIGNING SOFT SENSORS

7.1 Principle

A soft sensor is an algorithm computing an estimate of some internal variable of a system which can not be directly measured nor computed from available measurements and models because of limited data or computing ressources.

A soft sensor can be designed from detailed numerical simulations of a system, by recording the simulated internal and external variables and applying supervised learning in order to compute an approximation of the conditional expectation of the internal variable given the external measurements. In other circumstances, they can be designed using real system measurements obtained off-line.

Soft sensors can be useful in real-time monitoring and control applications when full flegded model based state estimation is not feasible either for computational reasons or because no good models exist.

7.2 Example application

The idea of soft sensors has been applied to the design of a rotor angle and speed estimator from synchronized phasor measurements, using neural networks in supervised learning mode [34].

Within this context, it seems plausible that one can design by automatic learning a soft-sensor using only on local measurements in order to predict when a power plant is in the process of loosing synchronism. Such a device could then be used in order to determine closed loop local control devices able to stabilize the power plant.

Similar applications could be imagined for voltage collapse prediction and control as well as for the identification and damping of slow inter-area modes.

8 APPLICATION TO MONITORING

8.1 Suggestion

Monitoring applications are multitudinous in power systems operation and control. Intrinsically, monitoring aims at combining information from low level real-time measurements in order to compute a high level indicator of system health, related to the proximity of the current state of the system to stability limits, to the direction in which the current trend is driving the system, or simply to identify whether the system has entered an abnormal condition.

These monitoring problems may directly be formulated as automatic learning problems, supervised or unsupervised ones. We thus believe that automatic learning methods could be useful in order to synthesize automatically system monitoring algorithms from measurements or from simulations.

9 CONCLUSION

In the first part of this paper we have reviewed state-of-the-art automatic learning problems, protocols and algorithms with the objective of highlighting their application potentials in the context of advanced sensing, monitoring and control of electric power systems.

In the second part of the paper we have tried to explain how automatic learning can be applied to various broad classes of practical problems, related to security assessment, automatic control, forecasting, equivalencing, soft sensing, and monitoring.

We believe that the potential of application of automatic learning to power systems is huge, and given the growing difficulties to manage complexity within this context, we hope that this paper can contribute to foster further research and in particular more serious and widespread attempts for real-life applications.

ACKNOWLEDGMENTS

Damien Ernst and Pierre Geurts acknowledge the support of the Belgian FNRS (Fonds National de la Recherche Scientifique) where they are post-doctoral researchers.

REFERENCES

[1] T. E. DyLiacco, "Control of power systems via the multi-level concept," Ph.D. dissertation, Case Western Reserve University, Systems Research Center, 1968.

[2] S. Russel and P. Norvig, *Artificial Intelligence: a Modern Approach*. Prentice Hall, 1994.

[3] M. Vidyasagar, *A Theory of Learning and Generalization: with Applications to Neural Networks and Control Systems*. Springer, 1997.

[4] V. Vapnik, *Statistical Learning Theory*. Wiley, New York, 1998.

[5] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer, 2001.

[6] R. Duda and P. Hart, *Pattern Classification*, 2nd ed. John Wiley & Sons, Inc., 2001.

[7] T. Poggio, R. Rifkin, S. Mukherjee, and P. Niyogi, "General conditions for predictivity in learning theory," *Nature*, vol. 428, pp. 419–422, 2004.

[8] B. Scholkopf, C. Burges, and A. Smola, *Advances in Kernel Methods: Support Vector Learning*. MIT Press, Cambridge, MA, 1999.

[9] C. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*. MIT Press, Cambridge, MA, 2000.

[10] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[11] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, pp. 1 – 39, 2006, (to appear).

[12] D. Bertsekas and J. Tsitsiklis, *Neuro-Dynamic Programming*. Belmont, MA: Athena Scientific, 1996.

[13] R. Sutton and A. Barto, *Reinforcement Learning. An Introduction*. MIT Press, 1998.

[14] C. Watkins, "Learning from Delayed Rewards," Ph.D. dissertation, Cambridge University, England, 1989.

[15] D. Ernst, P. Geurts, and L. Wehenkel, "Tree-based batch mode reinforcement learning," *Journal of Machine Learning Research*, vol. 6, pp. 503–556, April 2005.

[16] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*. San Mateo: Morgan Kaufmann, 1988.

[17] R. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter, *Probabilistic Networks and Expert Systems*. New York: Springer, 1999.

[18] V. Auvray and L. Wehenkel, "On the construction of the inclusion boundary neighbourhood for markov equivalence classes of bayesian network structures," in *Proceedings of Uncertainty in Artificial Intelligence*, 2002, pp. 26–35.

[19] V. Auvray, P. Geurts, and L. Wehenkel, "A semi-algebraic description of discrete naive bayes models with two hidden classes," in *Proceedings of the 9th International Symposium on Artificial Intelligence and Mathematics*, jan 2006, (to appear).

[20] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[21] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," *Journal of Artificial Intelligence Research*, vol. 4, pp. 129–145, 1996.

[22] V. Fedorov, *Theory of Optimal Experiments*. Academic Press, 1972.

[23] B. Wittenmark, "Adaptive dual control methods: an overview," in *Proceedings of the 5th IFAC Symposium on Adaptive Systems in Control and Signal Processing*, Budapest, Hungary, 1995, pp. 67–72.

[24] L. Breiman, "Statistical modeling: the two cultures," *Statistical Science*, vol. 16, no. 3, pp. 199–231, 2001.

[25] R. Marée, P. Geurts, J. Piater, and L. Wehenkel, "Random subwindows for robust image classification," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, CVPR 2005*, vol. 1, 2005, pp. 34–40.

[26] P. Geurts and L. Wehenkel, "Segment and combine approach for non-parametric time-series classification," in *Proceedings of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, October 2005.

[27] C. Olaru and L. Wehenkel, "Data mining," *IEEE Computer Applications in Power*, vol. 12, no. 3, pp. 19–25, July 1999.

[28] G. Saporta, "Data mining and official statistics," in *Proceedings of the Quinta Conferenza Nationale di Statistica, ISTAT*, 2000, pp. 1–4.

[29] L. Wehenkel, *Machine Learning Approaches to Power System Security Assessment*. University of Liège - Coll. Faculté des Sciences appliquées, 1994.

[30] ——, *Automatic Learning Techniques in Power Systems*. Kluwer Academic, 1998.

[31] D. Ernst and L. Wehenkel, "FACTS devices controlled by means of reinforcement learning algorithms," in *Proceedings of the 14th Power Systems Computation Conference, PSCC02*, Sevilla, Spain, June 2002.

[32] L. Wehenkel, M. Glavic, and D. Ernst, "New developments in the application of automatic learning to power system control," in *Proceedings of the 15th Power Systems Computation Conference, PSCC05*, 2005.

[33] D. Ernst, M. Glavic, F. Capitanescu, and L. Wehenkel, "Model predictive control and reinforcement learning as two complementary frameworks," in *Proceedings of the 13th IFAC Workshop on Control Applications of Optimisation*, 2006, (to appear).

[34] A. Del Angel, M. Glavic, and L. Wehenkel, "Using artificial neural networks to estimate rotor angles and speeds from phasor measurements," in *Proceedings of Intelligent Systems Applications to Power Systems, ISAP03*, 2003.