17th European Symposium on Computer Aided Process Engineesing – ESCAPE17
V. Plesu and P.S. Agachi (Editors)

Data-Driven Decision Support and its Applications in the Process Industries

Petr Stluka, Karel Mařík

Honeywell Prague Laboratory, Pod vodárenskou věží 4, 182 08 Prague 8, Czech Republic; stluka@htc.honeywell.cz; marik@htc.honeywell.cz

Abstract

Decision support systems represent specific form of control systems that help decision-makers to identify and solve problems, complete decision process tasks and make non-trivial decisions. In context of the process industries the decision support system (DSS) can help plant operators and engineers to deal with complex tasks like process monitoring, fault detection and diagnosis, data analysis or process optimization. The paper describes specific concept of a data-driven decision support system that leverages the principle of lazy learning, which builds predictive models locally in the nearest neighborhood around given point of interest. The methodology of memory-based regression, classification, novelty detection and optimization is described along with possible applications in the process industries.

Keywords: decision support, non-parametric statistics, lazy learning

1. Introduction

The general concept of decision support systems is defined very broadly. The architectural and functional complexity of DSS can range from relatively straightforward tools for answering simple queries, to much sophisticated systems that allow high-level modeling of what-if scenarios.
The most frequently referred types of DSS systems are rule-based, model-based and data-driven. [8] The major differences are given by the way how data and

knowledge are stored and processed. Given that efficient database and data warehousing technologies have become nowadays a commodity the interest naturally increases in data-driven DSS. [6]

The process industries represent a specific challenge for applications of data-driven systems. Firstly, in the process industries the target users are frequently process engineers and operators who usually need very fast advisory service. Secondly, many industrial processes are rather complex and behave according to underlying non-linear physics. Other challenges can be introduced by fast process dynamics and operation in multiple distinct modes. Thirdly, the process automation requires significant amounts of parameters to be measured with a short sampling interval. Given the advances in modern sensor technologies the industrial processes can be equipped with numerous measurement devices at affordable cost, and as a consequence, there easily can be huge amounts of data collected and stored in the plant historian databases. Finally, many of the measured process variables are highly correlated because of redundancy of measurement, interacting base level control loops, and physical phenomena like mass and energy conservation laws.

The most comprehensive development programs focused on bringing DSS tools to the process industries include the work done within the Abnormal Situation Management consortium [4] [7], and also the CHEM project [3] that was executed under the Fifth EU Framework Programme. Both these activities addresses similar problems including process monitoring, event detection and diagnosis, alarm management, and operator's advisory in general. Each program separately yielded a number of tools that can be combined together in various supervisory applications.

The aim of the paper is to describe the methodology and applications of an integrated data-driven decision support system that is being developed in Honeywell Prague Laboratory. The paper is divided into two parts. Key features and technology foundation of this specific DSS implementation are summarized in Section 2, which is followed by Section 3 that provides insights into applications in the process industries.

2. Methodology

The technical concept of the described decision support system is based on the methodologies known as non-parametric statistics and lazy learning. The key principle is that a predictive model is built on demand from relevant historical data, typically a small subset of entire history. The model is fitted to past data similar to the situation under study that usually corresponds to the current operating point, and is called a *query point*. The structure of the model is not specified a priori, but is instead determined from data. This approach does not estimate a global model but defers the processing of data until the prediction is explicitly requested. Important enabling infrastructure is the efficient underlying

database technology that makes possible to store relevant variables in dedicated tables – data marts – and access this data in an iterative fashion.

2.1. Similarity Search

Building multiple local models on the fly in the neighborhood of given query point requires the ability to find and retrieve nearest neighbors from historical database. This need makes the following concept of similarity search of fundamental importance for all other DSS components.
Assume a data set with m numerical variables $x = (x_1, x_2,\ldots, x_m)$. The neighborhood of a query point x^0 is defined by Euclidean distance d^2 as follows:

$$d^2 = \sum_{i=1}^{m} \left(\frac{x_i^0 - x_i^k}{h_i} \right) \leq 1 \tag{1}$$

where the vector $h = (h_1, h_2,\ldots, h_m)$ is composed of bandwidth parameters associated with individual variables x_i. Bandwidths define intervals around each query value x^0_i. Data points x^k that satisfy the above inequality lie inside the neighborhood whose shape is ellipsoidal.
Practical implementation of the search and retrieval of similar points is done in two steps. Capabilities of SQL database engine are used in the first step when a standard SELECT command is applied to historical data. Its WHERE clause is a conjunction of m inequality constraints formulated as:

$$(x^0_i - h_i) \leq x^0_i \leq (x^0_i + h_i) \tag{2}$$

This type of condition defines a cube-shaped neighborhood around given query point. All data points that satisfy conditions (2) are retrieved to memory and processed in the second step that applies Euclidean metric (1) to each of them.
Sometimes the final number of retrieved points is not sufficient for building of reliable local model. In such a case the neighborhood must be adapted – enlarged – until it contains a suitable number of points. This adaptation is done by multiplying bandwidths h_i by a constant greater than one, and consequent repetition of both above steps.
After completion of the search each data point is assigned a weight according to its squared distance d^2 to the query point. The weights $0 \leq w_k \leq 1$ are calculated by applying a specific kernel function – most frequently Gaussian or Epanechnikov – to the squared distance.
The historical data contains mostly numerical variables, but sometimes it is necessary to take into consideration also categorical variables like codes of individual operating modes, product grades, shifts, or days of the week. These categories can be effectively handled only if a specific similarity metric is provided typically by a domain expert. Otherwise the categories are considered as distinct cases that in fact partition historical data into several disjunct subsets.

2.2. Memory-based regression

Memory-based regression can be applied to a system on which a vector x of m independent (input) variables is used to predict the vector y of n dependent (output) variables. From the database point of view the time series of historical observations are stored in a table that has $(n + m)$ columns. For given query point x^0 the similarity search algorithm determines N of these historical points and retrieves them to the memory. Each of the data points (y^k, x^k), $k = 1,\ldots,N$ is assigned a weight, which expresses the relevance of the data point for prediction of output vector y^0 at a given query point x^0. The dependence of y on x is a general stochastic functional relationship $y^k = f(x^k)$, $k = 1,\ldots,N$, where $f(.)$ can be a parametric model – polynomial regression – whose parameters are to be estimated by Bayesian approach as described in [1].

2.3. Memory-based classification

Compared to memory-based regression each output variable y is now assumed categorical, taking on a finite set of values identified with sequence $\{c_1, \ldots, c_p\}$. p is the number of different values of y. The local model is fully defined by the probability vector $\theta = (\theta_1, \ldots, \theta_p)$ with positive entries $\theta_i > 0$, $i = 1, \ldots, p$, summing up to 1, where θ_i is the probability of y taking a particular value c_i. The vector θ is assumed to have a Dirichlet distribution. Bayesian approach for computation of probability density function is described in [5]

2.4. Novelty detection

Non-parametric approach to novelty detection can be based on the k-nearest neighbor algorithm. One of the currently tested approaches assumes that the vector h of default bandwidth parameters is determined by an automated procedure so as to reflect k-nearest patterns in the historical data. Consequently, this vector is iterated until the neighborhood around the query point x^0 contains exactly k neighbors. The difference between the two bandwidth vectors is used as indicator of novelty.

2.5. Data-driven optimization

Data-driven optimization can be applied to a system whose output variables y do depend on state variables x, and action variables u that can be manipulated by the system supervisor. In this case the query point x^0 corresponds to the current operating point, and the goal is to find such combination of actions u that maximizes certain objective function F in the neighborhood of x^0. The algorithm starts from ranking all historical actions according to objective function F. Consequently, the dependence of F on u and x is fitted by a local regression model. The best performing actions, called “best practices”, are

further perturbed utilizing the regression model for estimating F for the newly suggested actions. After pre-specified number of iterations the best found actions u^* are recommended to the supervisor. The algorithm is always restricted to the local neighborhood, which assures that the risk of suggesting rather bold, or practically infeasible actions is minimized.

3. Applications

The presented data-driven DSS has a wide range of applicability. The following list of applications gives an idea about possible uses.

- **Demand forecasting** is a type of application that can be efficiently solved by the memory-based regression algorithm. Demand forecasts are usually required for a longer time horizon, which means that the algorithm must be applied in batch to a sequence of future points in time. In practical implementation a new local model is built for each future point. This concept is referred as iterated one-step-ahead prediction [2] All influencing factors, which are used as inputs to the model – e.g. meteorological conditions – must be determined for the complete forecast horizon in advance. Description of such demand forecasting solution for power plants, heating plants, utilities, and distribution companies was provided in [1].
- **Property and performance prediction** are another typical applications of memory-based regression that can be seen as a flexible tool for inferential sensing. Specific examples are catalyst activity estimation, modeling of coke formation, or modeling of heat exchanger fouling. Iterating the predictions with regular step enables to monitor trends of these performance indicators, and alert when the speed of degradation is faster than expected. The value of the data-driven approach is in ability to infer the parameters' values for a broad range of conditions, taking into account all past fluctuations.
- **Event classification** and **fault diagnosis** are problems that can be addressed by the memory-based classification algorithm. The assumption is that historical data contains patterns of specific process states, typically abnormal situations, upsets, or faults, and that these patterns are coded in the database in terms of annotations – e.g. using a status column filled by categories "off-spec", "normal", "fault A" etc. Then the classifier is able to compute density functions for all such event locally in the neighborhood around the current operating point. This gives a possibility to foresee problems that will likely appear in near future.
- **Risk assessment and validation of operator's entries** is an example of possible use of the novelty detection algorithm. In terms of prevention of human errors, any set points being entered by the operator can be checked against historical data to identify if the process has ever been operated in the region defined by the new operating point. Given that all past control settings define a possibly multi-modal and complex distribution in multi-dimensional

space, the task of novelty detection is to evaluate how "close" or "far" the new entries are from frequently applied and safe settings.

- **Cautious optimization** is the way how the data-driven optimizer works. The assumption is that the historical actions that had been applied to the process can be ranked according to one or more key performance indicators (KPI). Examples of these KPIs are amount of energy and utilities used, occurrence of off-spec production, or the alarm rate observed after applying specific actions. Modeling of KPIs around the current operating point enables to drive multi-criteria optimization of the process, meaning that the control settings are adjusted in small steps leveraging past operating experience.

4. Conclusions

The paper presents key concepts and applications of a specific implementation of data-driven Decision Support System that takes benefit from combination of database technology with non-parametric statistics. Although the system can potentially work with complete process history, only a relatively small fraction of historical records is needed for fitting local models around the situation under study. The principle of building models on the fly allows both adjusting the model to the situation already met in the past, as well as continuous adaptation to new rends. Non-parametric modeling also allows to handle strongly non-linear behavior, which brings practical advantages compared to PCA and PLS based tools.

References

1. Z. Beran, K. Marik, P. Stluka, In: Proceedings of ESCAPE-16, Garmisch-Partenkirchen, 2006.
2. G. Bontempi, M. Birattari, H. Bersini, in Machine Learning: Proceedings of the 16th International Conference, San Francisco, Morgan Kaufmann, 1999.
3. S. Cauvin, B. Celse, CHEM: In: Proceedings of ESCAPE-14, Lisbon, 2004.
4. M. Elsass, J. F. Davis, D. Mylaraswamy, D. V. Reising, J. Josephson, An integrated decision support framework for managing and interpreting information in process diagnosis, www.asmconsortium.com
5. R. Kulhavy, NATO Science Series III: Computers & Systems Sciences, 190, IOS Press, Amsterdam.
6. R. Kulhavy, IEEE Control Systems Magazine, 23 (2003).
7. D. Morrison, W. Foslien, P. Jofriet, W. MacArthur, Early event detection white paper, www.asmconsortium.com
8. D. J. Power: A brief history of decision support systems, version 2.8, www.DSSResources.com