10

Machine learning techniques to examine large patient databases

Geert Meyfroidt, Deputy Head of Clinics [a,*], Fabian Güiza, PhD student [b], Jan Ramon, Postdoctoral Researcher [b], Maurice Bruynooghe, Professor, Head of Research Group Declarative Languages and Artificial Intelligence [b]

[a] *Department of Intensive Care Medicine, UZ Leuven - Campus Gasthuisberg, Catholic University of Leuven, Herestraat 49, 3000 Leuven, Belgium*
[b] *Department of Computer Sciences, Faculty of Engineering, Catholic University of Leuven, Leuven, Belgium*

Keywords:
intensive care unit
operating room
computerization
patient data management system
machine learning
data mining
decision tree learning
Bayesian networks
Support Vector Machines
Gaussian processes

Computerization in healthcare in general, and in the operating room (OR) and intensive care unit (ICU) in particular, is on the rise. This leads to large patient databases, with specific properties. Machine learning techniques are able to examine and to extract knowledge from large databases in an automatic way. Although the number of potential applications for these techniques in medicine is large, few medical doctors are familiar with their methodology, advantages and pitfalls. A general overview of machine learning techniques, with a more detailed discussion of some of these algorithms, is presented in this review.

> Where is the Life we have lost in living?
> Where is the wisdom we have lost in knowledge?
> Where is the knowledge we have lost in information?
>
> – T.S. Eliot's The Rock (1934)[1]

* Corresponding author. Tel.: +32 16 34 40 21; Fax: +32 16 34 40 15.
E-mail address: geert.meyfroidt@uzleuven.be (G. Meyfroidt).

Introduction

Intensive care units (ICUs) and operating rooms (ORs) are very data-rich environments. Monitors as well as therapeutic devices (such as mechanical ventilators, syringe- and infusion pumps for drug and fluid administration, or renal replacement therapy machines), generate large amounts of data on a continuous basis. Blood samples for laboratory analysis are drawn several times a day, and microbiology sampling occurs several times a week. Doctors and nurses write progress notes several times a day. Drug prescription and delivery is changed and charted more than once daily. In the nineties it has been determined that, on average, more than 236 variable categories were measured on each day for a standard ICU patient.[2] Until now, most of these data were available only on paper and thus difficult, if not impossible, to analyse. A Patient Data Management System (PDMS) is software that integrates these data from multiple sources. The number of ORs and ICUs implementing a PDMS is increasing worldwide.[3] A number of clinical studies demonstrate the potential benefits of these systems: a better quality of medical charting[4], higher ICU risk prediction scores[5], less administrative workload and more time available for patient care[6], and positive impact on health care practitioner satisfaction and nursing retention.[7] Nowadays most PDMS are equipped with a computerized physician order entry (CPOE) system, offering extra advantages regarding patient safety[8–10], decision support and protocolized care.[11,12] Apart from supporting care, a PDMS is also a digital archive. Such an archive is called a ‘relational’ database, when it integrates and stores all patient related data, is well organized, and readily accessible. The worldwide rise of relational databases creates a huge potential source of data from ICU and OR that could be used to investigate clinical, scientific or health care policy related problems. Particularly outcome research is a potential application where PDMS data could serve as a source.[13] The work of Ramon et al[14] highlights the challenges this specific application domain poses for data mining, and proposes some initial solutions.

Machine learning algorithms have been used in a variety of applications. They have been shown to be of special use in data mining scenarios involving large databases and where the domain is poorly understood and therefore difficult to model by humans.[15] These techniques are able to handle large amounts of data, to integrate data from different sources, and to incorporate background knowledge in the analysis.[16] Therefore they are probably the most valuable candidates for this type of research. In this era of fast, powerful and relatively cheap computers, (lack of) computer power should no longer be an obstacle.

Biomedical data from OR or ICU: characteristics and accuracy

There are four important barriers to the effective use of clinical or hospital databases for research purposes.[17,18]

First, there is the problem of confidentiality. Clearly, patient data should be protected against viewing from third parties, and the identity of the patient should be blinded. Regulations might be different in every country. For example, in the US, removing the patient identifiers from the database allows querying the data without it being a Health Insurance Portability and Accountability Act (HIPAA)[19] violation and without needing to have an audit trail of who accessed the information.[13]

Second, the amount of data is huge. The resolution with which these parameters are registered is high, usually in the order of minutes. At the university hospitals Leuven, the average amount of data stored per patient and per day in our PDMS is 5 Megabyte. Since all our 56 ICU beds are almost always occupied, we gather more than 100 Gigabyte of patient data every year! Moreover, the data have unusual characteristics: a relatively low number of patients are described by a large number of variables.

Third, proper organization of the data is an absolute prerequisite to research. When configuring a clinical database, attention should be paid to the structure. In a good relational database, every piece of data is ‘tagged’ as it is stored, like a library card catalogue. By way of those tags, the program can relate pieces of data to other types of data without searching the entire database.

The fourth barrier is related to the quality of the data. Data quality is not only difficult to obtain, but often hard to assess in retrospect. Apart from legibility, there are two aspects about the quality or accuracy of data: completeness and correctness.[20] Completeness refers to the proportion of observations that are actually recorded in the system; correctness refers to the proportion of observation in the system that are correct – as compared to the true situation of the patient or to a gold standard. On a higher level, Ward defines accuracy as “the ability of a set of data points, collected individually, to properly describe the clinical continuum during that time”.[13] The data in a clinical information system, gathered in an automatic way from medical devices and monitors in the ICU, are obviously more complete and perfectly legible, when compared to handwritten medical charts. Nevertheless, automatic data captation sometimes fails: temporary disconnections of the patients from their monitors or devices, as happens when they go on transport, or technical problems with devices, interfaces, or servers will result in temporary loss of data. Dealing with missing data in a time series could be the subject of a review in itself. In brief, several extrapolation methods such as deletion, mean substitution, mean of adjacent observations, and maximum likelihood estimation can be used. Automatically entered data have the advantage that they eliminate transcription errors. Nevertheless, they will always be ‘dumb’ data, since they have not undergone filtering on erroneous entries. It is well known that the monitors or devices generating the monitoring or therapeutic signals have little or no capacity to detect wrong data or artefacts. As has been demonstrated by several authors, most of the out of range data generated by monitors are wrong, either because of manipulation of the patient, or because they are truly false.[21] Validation of a clinical parameter by a human, before or after it is stored in the PDMS database, will add an appreciation to a value. This is a standard option of most PDMS. Some systems will only store these validated data in the database, discarding all unvalidated values. Discarding unvalidated values can lead to the loss of valuable information, as is shown in the example in Fig. 1, where episodes of paroxysmal atrial fibrillation occur between the validations. Looking at the validated values only, we would conclude that this patient had a normal heart rate all the time. The complete time series reveals episodes of tachycardia. Retrospective artefact removal from time series of clinical data is not the subject of this review. In the literature, many methods for filtering can be found, each with their applications and limitations. A recent study showed an unacceptably high inter-rater disagreement for retrospective artefact detection from a time series of clinical parameters. Therefore, automated artefact detection protocols should probably be based upon joint reference standards from multiple experts.[22] The higher sampling rate of automatic data captation could affect the conclusions a clinician draws from these data, and, as a consequence, alter the clinical picture of a patient. Many items in ICU prognostic scoring systems are based upon the extreme values of physiological parameters. Bosman et al[5] could demonstrate that these scores and the derived mortality predictions were significantly higher when they were based upon computerized data, versus a hand-written chart. Some data, such as the patient history, diagnosis, or the assessment of consciousness, have to be human or hand entered. The quality of human entered data varies, and probably depends on the time and devotion that ICU clinicians or nurses can spend to enter these data.

Machine learning: introduction and terminology

A PDMS database provides us with a large amount of data. Unstructured data will not allow us to generate new insights. A hierarchical division between the concepts data, information, knowledge and wisdom (DIKW) has been proposed to represent increasing levels of data organization and

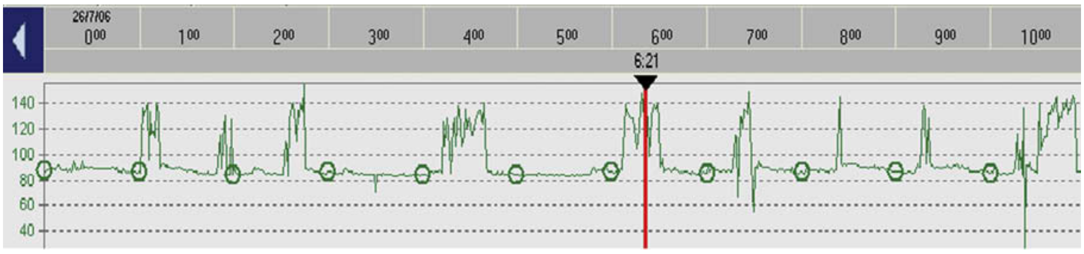


Fig. 1. This is a true screenshot taken from one of our patients after cardiac surgery. The green line represents the time series of the heart rate. The circles are the validated values. Because they occur between two validations, the episodes of tachycardia will be missed when only looking at validated data.

understanding.[23,24] In the DIKW hierarchy, data is the most basic level of unprocessed information, and comes in the form of raw observations and measurements. Information adds context to data, and is created by analysing relationships and connections between the data. Knowledge is created by using the information for action, as in a local practice or relationship that works. Knowledge is built by the learner through experience. Information is static, but knowledge is dynamic. Wisdom is the ultimate level of understanding. Wisdom is created through use of knowledge, through the communication of knowledge users, and through reflection. Data and information deal with the past. Knowledge deals with the present. Wisdom takes care of the future, as it takes implications and lagged effects into account. As most metaphors, this approach has limitations. The distinctions between data, information, knowledge, and wisdom are not very discrete, and their interrelations seem to be not always congruent.[25]

Knowledge Discovery from Databases (KDD)[26] is concerned with the automated extraction of valid, novel and potentially useful patterns from data. The KDD process is composed of several steps including: data preparation or preprocessing, search for patterns or data mining and knowledge evaluation or results analysis. A detailed description of the KDD process can be found in the work of Chapman et al.[27]

The search for these patterns of interest, also known as models, requires the use of data mining algorithms. Many of these stem from the fields of machine learning and statistics. Machine learning is an inherently multidisciplinary field, involving artificial intelligence, statistics, probability, computational complexity theory, control theory, information theory, philosophy, psychology, neurobiology and other fields. Machine learning is the subfield of artificial intelligence concerned with the development of algorithms that allows computer programs to learn from experience.[28] A computer program is said to *learn* from experience if its performance at certain tasks improves with experience. In other words, within the mathematical or computational boundaries of the selected method, the computer will search all possible hypotheses to determine the ones that best fit the observed data and any prior knowledge held by the learner. Statistics is of special importance within machine learning since statistical tests are used to evaluate the performance of individual models[29–31], to compare performance among different models[32], and in some cases as an integral part of the learning algorithm itself. Apart from that, performance of machine learning methods is often compared with more traditional statistical approaches in the medical domain, namely, logistic regression. Experience refers to the amount of data that is used for learning. Data is comprised of a set of examples. An algorithm that is allowed to learn on more examples, will thus gain more experience. An example can be described by different types of characteristics. In the PDMS setting, the patients are examples. They can be described by nominal or numerical characteristics called attributes (e.g. sex, birth date, diagnostic group,), and/or by time series of data (e.g. blood pressure, laboratory values …).

The output of the learning process, referred to as model, is automatically discovered knowledge that provides a description of the data. Models can be predictive if they predict the value of an attribute (referred to as the target attribute or target), by making use of the remaining attributes (referred to as predictive attributes). This is also called supervised learning. The target attribute is the focus of this process, and the data usually includes examples where the values of the target attribute have been observed. The goal is to build a model that can predict the value of the target attribute for new unseen examples. If the target value is nominal then the prediction task is known as *classification* and each possible value for the target variable is referred to as its class-label or class. For real-valued target attributes, the prediction task is known as *regression*. For instance, a classification task is the prediction of whether a patient survives his ICU stay or not. The class label in this case is ‘ICU survival’. The prediction of the length of stay of an ICU patient is a regression task.

Unsupervised learning refers to modeling with an unknown target variable. In that case, models are solely descriptive. The goal of the process is to build a model that describes interesting regularities in the data. Clustering[33] is an example of a descriptive data mining algorithm that is concerned with partitioning the examples in similar subgroups. For instance, we could analyze ICU data and find a subgroup of patients which are similar because they all received certain combinations of treatments and all experienced an increase in blood pressure. This knowledge is automatically discovered even though the model was not specifically built to predict blood pressure.

Current applications for data mining in medicine[34] include scientific purposes such as unraveling molecular mechanisms at the cellular level and support of clinical management, for example in cancer classification.[35] In intensive care, machine learning techniques have been used for surveillance of microbiological data[36], to detect changes in infection and antimicrobial resistance patterns[37], or for assessing morbidity after cardiac surgery.[38] Integrating machine learning methods with local logistic regression models performed better in predicting ICU-attributed mortality than traditional logistic regression models alone.[39] Depending on the type of knowledge of interest, Decision Trees (DT) and Random Forests (RF), Artificial Neural Networks (ANN), Bayesian Networks, and kernel methods such as Support Vector Machines (SVM) and Gaussian Processes (GP) are amongst the most frequently used algorithms. We will discuss them in more detail in the following section. Other approaches can be found in the work of Mitchell[15], Hastie[40], Witten[41] and Bishop.[42]

A glance at a few machine learning techniques

Decision trees and forests

In DT learning, the algorithm looks for the descriptive attribute that is most related to the target variable, divides the data set into subsets according to this attribute, and repeats the procedure on the subsets, until a termination criterion is fulfilled. The result is a tree-shaped model that identifies a small set of variables that together have high predictive power for the target variable.[43] To ease its readability the tree can be represented as a collection of if-then rules. Each rule is obtained by following the branches from the root node to a terminating node. The terminating node is called a leaf. An example is presented in Fig. 2.

Because of their ease of interpretability as well as their good performance, DT are among the most popular learning algorithms and have been successfully applied in a wide range of tasks and domains. In intensive care they have been used to classify pressure-volume curves in artificially ventilated patients suffering from Adult Respiratory Distress Syndrome (ARDS).[44] They have been compared to logistic regression in the task of classifying ICU patients with head injuries according to their outcome: good vs. poor Glasgow Coma Scores (GOS) and dead vs. alive.[45] Special emphasis was made on how the DT representation can lead to knowledge discovery and a better medical understanding of the variables that are more predictive for each subpopulation. DT has also been used to classify streams of physiological signals in neonatal ICU data in order to detect artifacts and thereby reduce the high number of false alarms.[46]

Apart from being easy to understand, DT has other advantages, such as being robust to labeling errors and noise. DT will still perform well, even if some examples are incorrectly labeled (assigned to the wrong class), or if the values of predictive attributes contain some noise or errors. On top of that, costs can be assigned to the attributes. For example in a (fictituous) prediction task for pneumonia, attributes might be: temperature, heart rate, respiratory rate, white blood cell count, CRP, results from sputum cultures, results from broncho-alveolar lavage cultures, chest X-ray, results from CAT-scan, lung biopsy results. A desired tree could assign different costs to these attributes (in terms of invasiveness of the test, time required for the test, possible side effects, financial costs, etc.). When building the DT, low-cost attributes are preferred for the test-nodes in the tree, and high-cost attributes are only used when the accuracy of the low-cost attributes is insufficient.

A RF is a model composed of an ensemble of DT. The procedure to build a DT is repeated a number of times on a slightly perturbed version of the original data set. To build a perturbed dataset, the algorithm randomly picks n times one case out of a dataset of n examples. The result is a dataset with some duplicates and some examples left out. The RF model contains the set of trees learned for each perturbed dataset and averages out their predictions to get a final prediction (Fig. 3). RF is a way of removing the influence that small random variations in the data set can have on a learned tree.[47]

RF consistently obtained better performances than DT in the ICU prediction tasks considered by Ramon et al[14] (Table 1). In this study, each RF consisted of a set of 33 DT and was built following the algorithm described by Vens et al.[48]

The boost in performance of forests over trees comes at the cost of less interpretability in the models, because a set of many trees is more difficult to comprehend than a single tree. Recent work[49]

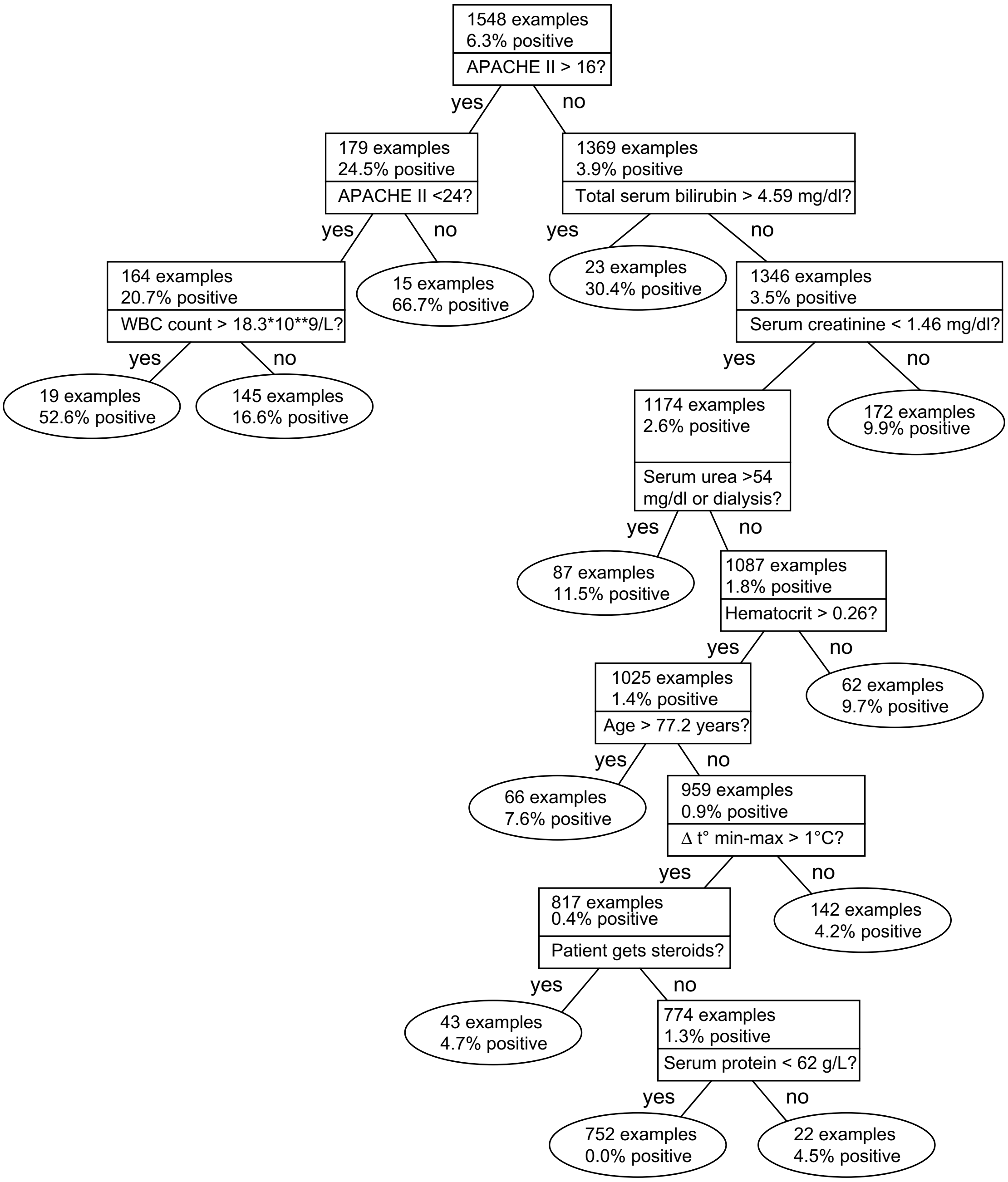


Fig. 2. Decision Tree to predict ICU mortality. We retrospectively analysed our database of 1548 patients from a previous clinical study on intensive insulin therapy in the surgical ICU[76], to examine the risk factors for mortality. The predictive attributes were taken only from the first day in the ICU. In this set of 1548 examples, 6.3% died while in the ICU. In other words, these patients were positively labelled, because the target variable 'Death while in the ICU' was present. The algorithm determined that the first attribute to test for was whether the APACHE II score of this patient was larger than 16. This divided the dataset into two subsets, with 179 examples where the APACHE II score was >16 and 1369 examples where the APACHE II score was ≤ 16. Of these 179 examples where the APACHE II score was >16, 44 examples (24.5%) were positively labelled. As we can see we have gained information, because now we know that if the APACHE II score is >16 then the probability of dying in the ICU increases from 6.3% to 24.5%. The algorithm then proceeds to find the most informative attribute for the subset of 179 examples where the APACHE II score was >16. The algorithm selected testing whether the APACHE II score is below 24. 20.7% of the 164 examples with an APACHE II < 24 and >16 were positively labelled, versus 66.7% of the 15 examples where APACHE II was ≥ 24 (and >16). Once again information has been gained since having an APACHE II score ≥ 24 increases the probability of dying in the ICU from 24.5% to 66.7%. The algorithm will continue until at a certain point a 'branch' (graphically represented by a square) becomes a 'leaf' (graphically represented by an oval). There are two reasons for the algorithm to stop. First, when no other statistically significant test can be found in these examples such that there will be information gain. For example, in the last leaf of this three, below the 'Serum protein < 62 g/L' test, none of the 752 examples are positive for the target variable. The second termination condition is when the number of examples in a subset is below a predefined minimum. For example, in the 66.7% leaf below the 'APACHE II < 24' test there are only 15 examples left.

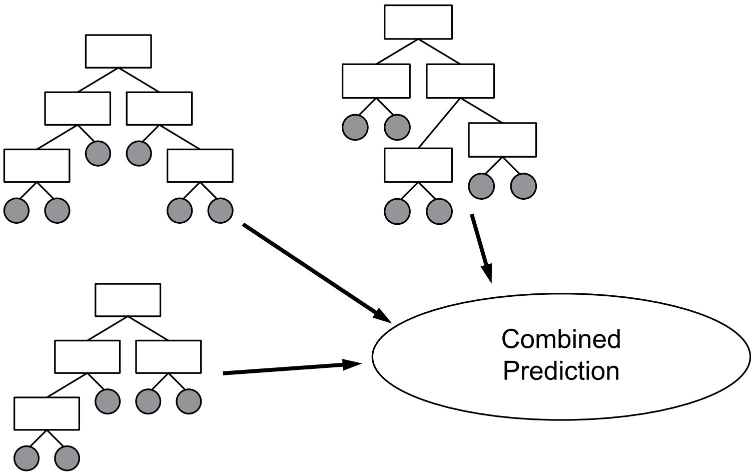


Fig. 3. Graphical representation of Random Forests. A forest consists of a set of Decision Trees, build on a perturbed version of the original dataset. The final prediction is obtained by averaging out the predictions of the set of trees.

however, promises to recover the understandability of the tree representation while still retaining the performance of the forest.

Artificial neural networks

Inspired by biological neural networks, artificial neural networks (ANNs) are a collection of simple processing units or nodes, interconnected to increase the computational power over any single unit. The weights of these interconnections are tuned during learning so that the output of the network is as close as possible to the desired training targets for a set of training input examples.

In a neural network, the input nodes are the observations or observed quantities that will be used for prediction. The network has one or several output nodes or predictions. Other nodes are calculated from the values of the input, and are then used to calculate the values for the output. These intermediate nodes are neither input nor output, but are calculated from the other nodes in the network. Because they only function within the network, they are called 'hidden' nodes. Although a network can be structured with more than two hidden layers, in practice this is rarely done.

The typical structure of a single node or neuron is depicted in Fig. 4.

Each input **x** is fed forward to each successive layer to determine the output **y**, forming a feed-forward network. During learning or training, the weights of the neurons are modified such that the output **y** is as close as possible to the target value **t**. The error, the difference between the obtained value **y** and the desired target value **t**, is used to adjust the weights of each neuron in the output layer. These errors are propagated backwards to previous layers, so that their weights can be adjusted. This is known as back propagation. It traverses the network in the opposite direction, and propagates the errors as weighted sums. The weights are updated iteratively according to a gradient descent method. Mathematical details can be found in the legend of Fig. 5.

Table 1
Performance of different machine learning algorithms when predicting ICU mortality using input data from the first 24 hours.

ICU mortality		aROC[a]/p-value[b]			
NbEx	Pos	DT	RF	NB	TAN
1548	6.3%	0.789/0.685	0.821/0.693	0.883/0.807	0.860/0.090

NbEx, Number of examples (= number of patients) from which the clinical information was used as input. Pos, Percentage of examples positive for the target variable, ICU mortality; DT, Decision tree learning; RF, Random forests; NB, Naïve Bayes; TAN, Tree-augmented naïve Bayes.

[a] Discrimination of the model is tested by the area under the receiver-operator characteristic curve (aROC).

[b] Calibration is tested by Hosmer-Lemeshow H test. The model is well calibrated according to this test if the p value > 0.050.

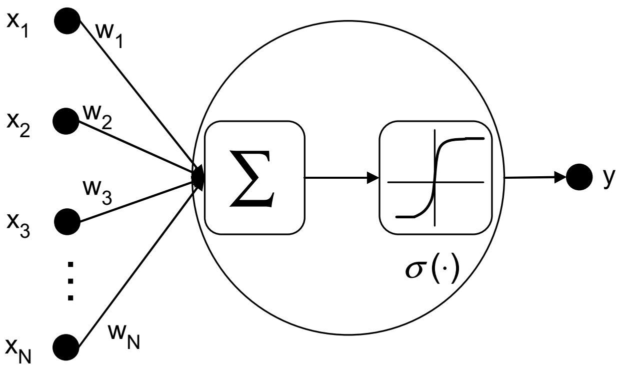


Fig. 4. A single hidden node or neuron in a network, which is a function of nodes (x) in a previous layer to predict the value of a node in the next layer. A weighted sum of the inputs is passed through a function σ(.) forcing the output of the neuron to be within a given interval. The output of a single unit is a thresholded linear combination of the inputs: $y = \sigma(x \cdot w) = \sigma(\sum_{n=1}^{N} w_n x_n)$. If σ is chosen to be a step-function such that the neuron's output can only take on the values +1 or −1, then the neuron is known as a perceptron. This type of neuron is typically used for scenarios with Boolean inputs. Another neuron of choice is the sigmoid unit, where σ is the sigmoid or logistic function. In this case, the output of the neuron can be any continuous value in the interval [0,1]. The weights can be seen as the components of a weight vector with the same dimensionality as the input space. This weight vector defines a decision surface in the input space. More complex and expressive (i.e. nonlinear) decision surfaces can be obtained when using a collection of units organized in a network configuration.

ANNs are known to be robust to errors in the training data and therefore they are well suited for learning from noisy examples, such as inputs from sensors like microphones and cameras. Because of their ability to represent highly nonlinear functions, ANNs often outperform other simpler methods. This malleability however often results in over-fitting. This can be solved by determining an adequate stopping iteration for the gradient descent in the back propagation algorithm.

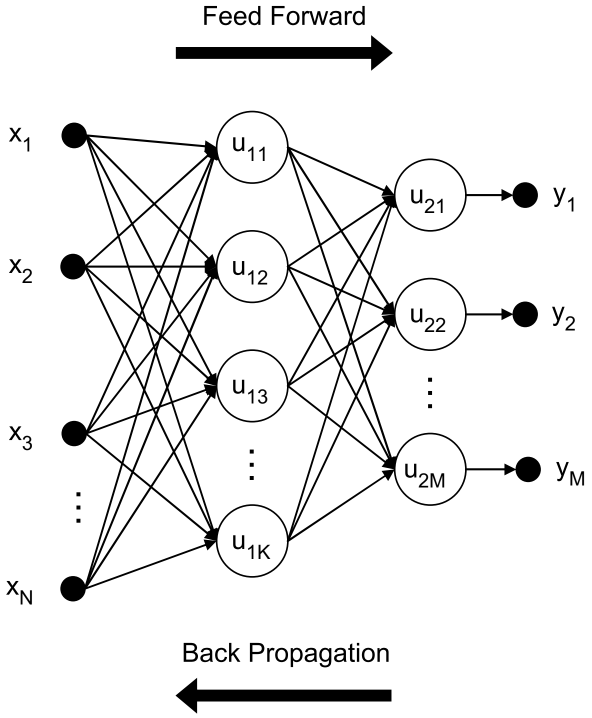


Fig. 5. Graphical representation of a neural network with N input units (x), K units (u) in a hidden layer, M units (u) in the output layer, corresponding to M outputs (y). For each training example **x** and target **t**: the input **x** is propagated forward through the network. The errors are propagated backward through the network. For each output unit u_{2m} the error term δ_m is calculated: $\delta_m = y_m(1 - y_m)(t_m - y_m)$. For each hidden unit u_{1k} (with output y_k), the error term δ_k is calculated: $\delta_k = y_k(1 - y_k) \sum_{m=1}^{M} w_{km}\delta_m$. Next, all weights w are adjusted according to: w = w + Δw. Where $\Delta w = \eta \delta x_i$. δ depends on whether the unit belongs to the output or to a hidden layer, x_i is the input value to the unit with weight w, η is the learning coefficient, a constant that regulates the magnitude of Δw, or how much the weights are allowed to change per iteration. This back propagation algorithm is used for training networks of sigmoid neurons, which have a continuous differentiable σ function, a requirement for the computation of the gradient of the error.

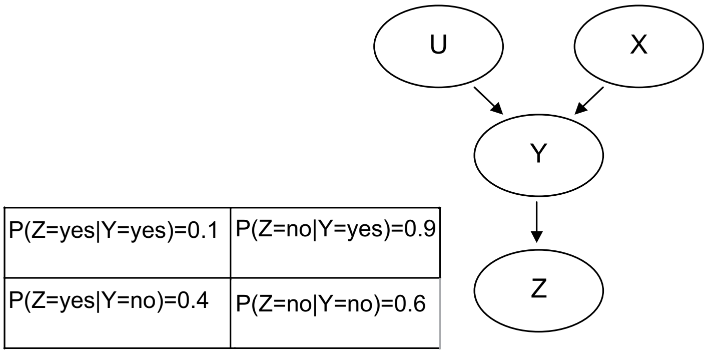
Some drawbacks in the use of ANNs include long training times when compared to other learning algorithms such as DT. Also, the number of neurons per layer that are necessary to effectively learn a desired function are not known in advance. In practice, several configurations have to be tried. But the major problem with ANN is that it lacks interpretability, because a set of weights is not as understandable as a collection of rules or a DT. This way, ANNs are true 'black box' models.

Regardless of these drawbacks, ANNs have been successfully applied in solving many learning tasks. A survey of applications can be found in.[50,51] Sargent[52] reviews a total of 28 studies on medical data sets in which ANNs are compared to standard statistical techniques such as logistic regression. ANNs were found to outperform regression in 10 cases, were outperformed in 4 and had a similar performance in the remaining 14 data sets. In intensive care, ANNs have been applied in a number of tasks. They have been used for survival prediction where their performance has been compared to that of logistic regression.[53] They were found to provide a higher percentage of ICU patients correctly classified as well as a smaller prediction error than logistic regression. Additionally ANNs, as opposed to logistic regression, do not require any assumptions regarding the underlying parameter distributions, nor about the interactions among the independent variables. Similar findings are reported in[54], where both logistic regression and neural network models had a similar performance in predicting death of patients with suspended sepsis in the emergency room. Nevertheless, there was a statistically significant difference in discrimination in favor of neural networks. Clermont et al[55] however report similar performances of both logistic regression and ANN for the task of hospital mortality prediction from data obtained from seven ICUs. Tong et al[56] developed an ANN to successfully classify a neonatal ICU population according to ventilation duration, a study that extends their previous success with the same technique and classification task in an adult ICU setting.

Bayesian networks

A Bayesian network is a probabilistic graphical model that specifies a joint probability distribution on a set of random variables. Bayesian networks have two essential components: a directed acyclic graph explicitly showing dependencies and independencies between variables, and a set of probability distribution tables. This is illustrated in Fig. 6.

There are two specific classes of Bayesian networks that are popular in the context of supervised learning: Naïve Bayesian networks (NB) and Tree-Augmented Naïve Bayesian networks (TAN). In NB there is a link from the target variable to each of the non-target variables. This means that a non-target variable is supposed to be independent of any other non-target variable, given the target variable (Fig. 7a). Because the independence assumptions for NB are often too strong, TAN allows taking into account certain extra dependencies between non-target variables by having links between them as is shown in Fig. 7b. It is very important to note that the presumed dependencies or independencies



P(Z=yes|Y=yes)=0.1	P(Z=no|Y=yes)=0.9
P(Z=yes|Y=no)=0.4	P(Z=no|Y=no)=0.6

Fig. 6. Probabilistic graphical model. The nodes in this graph represent the random variables and the arrows represent direct influences. This way, the graph contains information about dependencies and independencies between variables. Each variable in a Bayesian network has an associated conditional probability table (CPT) specifying exactly how that variable depends on its parents. The CPT depicted here specifies the probability of the presence of Z if we know that Y is present (upper left), the probability of the presence of Z in the absence of Y (lower left), the probability of the absence of Z in the presence of Y (upper right) and the probability of the absence of Z and Y (lower right).

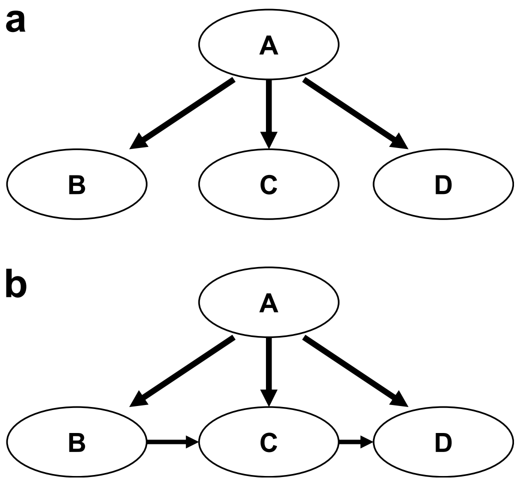


Fig. 7. (a) Naïve Bayesian Network. In NB, the non-target variables (B, C, D) are independent from each other given the target variable (A). (b) Tree-Augmented Naïve Bayesian Network. In TAN, arrows between non-target variables (B, C, D) indicate their dependencies.

learned by a Bayesian network do not necessarily have to make sense from an empirical, or in this case medical, point of view.

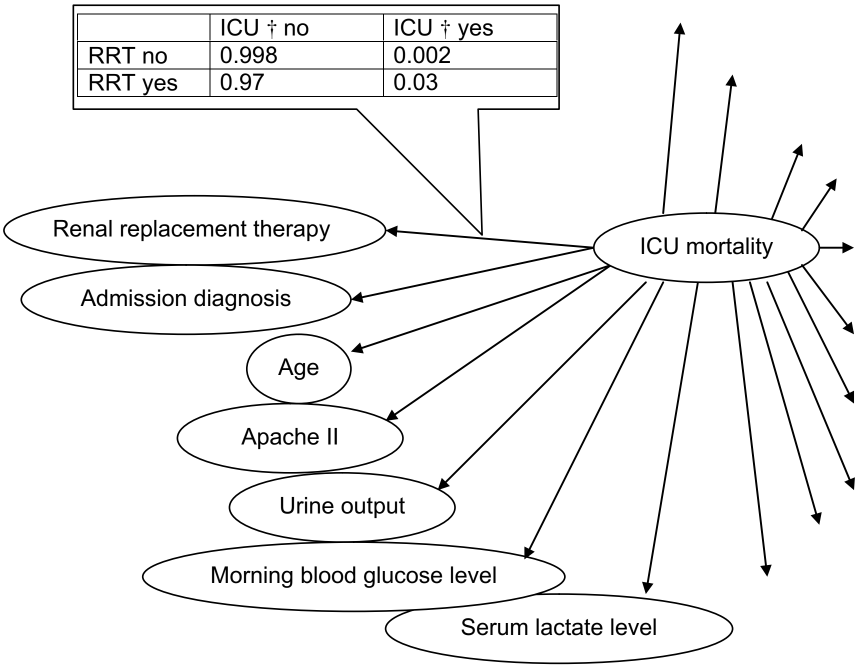
To automatically construct a NB or TAN several steps are typically followed. In a first step irrelevant variables or attributes are removed from the dataset. To determine which variables are relevant, a statistical test to determine the association between the variable and the target-variable is performed. In a next step the arrows of the graph are determined. For NB the arrows are fixed by definition, independent of the actual data. TAN basically makes use of the same arrows as NB, plus extra arrows that capture the most important dependencies between the non-target variables. Several algorithms can be used to derive these dependencies; the algorithm of Friedman[57] is just one example. In the final step, the conditional probability tables (CPT) for all variables in the graph are constructed using maximum likelihood estimation. In the CPT, the probability P(B| A) can be calculated as the number of examples in the database with attribute B that are positive for class label A divided by the total number of patients that are positive for class label A. This fraction (p/t) is the maximum likelihood estimate, and it is calculated in each entry of each CPT. Bayes' rule is applied to predict the outcome, which results from a product of different entries from the different CPT's. A survey of Bayesian Networks applied in health care can be found in the article of Lucas et al.[58]

Ramon et al[14] repeated the same experiment from Fig. 2, this time using NB and TAN. Part of the obtained Naïve Bayesian network is shown in Fig. 8. Table 1 shows the performances of the different machine learning algorithms in this outcome prediction task. In this paper, performances were assessed with 10-fold cross validation: the data were divided into 10 subsets of (approximately) equal size. Models were trained on the data 10 times, each time leaving out one of the subsets from training, using only the omitted subset to evaluate the performance. All methods performed well in this prediction task (aROC of more than 0.8, except DT) and none showed overfitting (Hosmer-Lemeshow statistic p-value $>$0.050).

Support Vector Machines (SVMs)

Support Vector Machines are machine learning techniques that have their origin in statistical learning theory. An excellent basic (non-mathematical) explanation on SVM is the paper by Noble.[59] A more mathematical presentation of SVM can be found in the work of Cristianini et al.[60]

An essential component of SVMs is the separating hyperplane. In a binary classification task (such as predicting ICU mortality or survival), the hyperplane is the geometrical division or separation between the two outputs. In a one-dimensional space, this is a single point, in a two-dimensional space a line, in a three-dimensional space a plane. We can extrapolate this procedure mathematically



	ICU † no	ICU † yes
RRT no	0.998	0.002
RRT yes	0.97	0.03

Fig. 8. Part of a Naïve Bayesian Network learned to predict a patient's probability of dying in the ICU.[14] Inputs were taken from the data collected on the first day in the ICU, in a dataset of 1548 patients with a mortality of 6.3%. 37 variables were found to correlate with mortality (of which only 7 are shown here). Also shown is the conditional probability table between one variable (did the patient undergo renal replacement therapy (RRT) on the first day in the ICU) and the target variable (ICU mortality). The table indicates for example, that a patient has a 99.8% probability of not dying in the ICU if he did not undergo RRT on the first day in the ICU.

to higher dimensions; the general term for a separator in such a high dimensional space is a hyperplane. The SVM algorithm will try to find the optimal hyperplane, called maximum margin hyperplane that offers the best classification. To increase the robustness of the classifier and allow for misclassification, soft margins can be set around the hyperplane. They determine the number of examples that are allowed to cross the hyperplane at a certain distance. A SVM is a kernel method that makes use of a kernel function. A kernel function will add a dimension to data, in order to obtain the most optimal classification. Any given dataset with consistent labels can be brought into a dimension where it can be linearly separated by a hyperplane. However, a too high dimensional space could lead to overfitting of the data. The optimal SVM is typically chosen trough trial and error, selecting the optimal kernel function by using cross validation. SVM can only handle binary classification problems. Multiclass classification can be obtained through the combination of multiple binary classifiers, but more sophisticated solutions for this problem also exist. Figure 9 is an illustration of the SVM procedure.

SVMs have been applied for classification in medical domains. When using SVM to predict the depth of infiltration in endometrial carcinoma based on transvaginal sonography[61], SVMs showed a better generalization behavior and a higher performance than logistic regression. Bazzani et al[62] used an SVM classifier to distinguish false signals from microcalcifications in digital mammograms. The SVM classifier performed slightly better than one implemented with an ANN, and had the advantage of being easier to train. Van Gestel et al[63] compared least squares SVMs with DT, NB and logistic regression for classification on 20 benchmark datasets. They report a significantly better performance of SVMs over the other methods for most of the datasets and no significantly worse performance on the remaining datasets.

In intensive care, SVMs have been investigated and tested to predict tacrolimus blood concentration in liver transplant patients.[64] They outperformed multivariate linear regression and they required significantly less inputs to achieve the same predictive performance. They have also been used to study drug dosage the ICU.[65] Hiissa et al[66] automatically classified nursing narratives. Giraldo et al[67] used Support Vector Machines to classify respiratory patterns of patients on weaning trials into those that will succeed or fail to sustain spontaneous breathing.

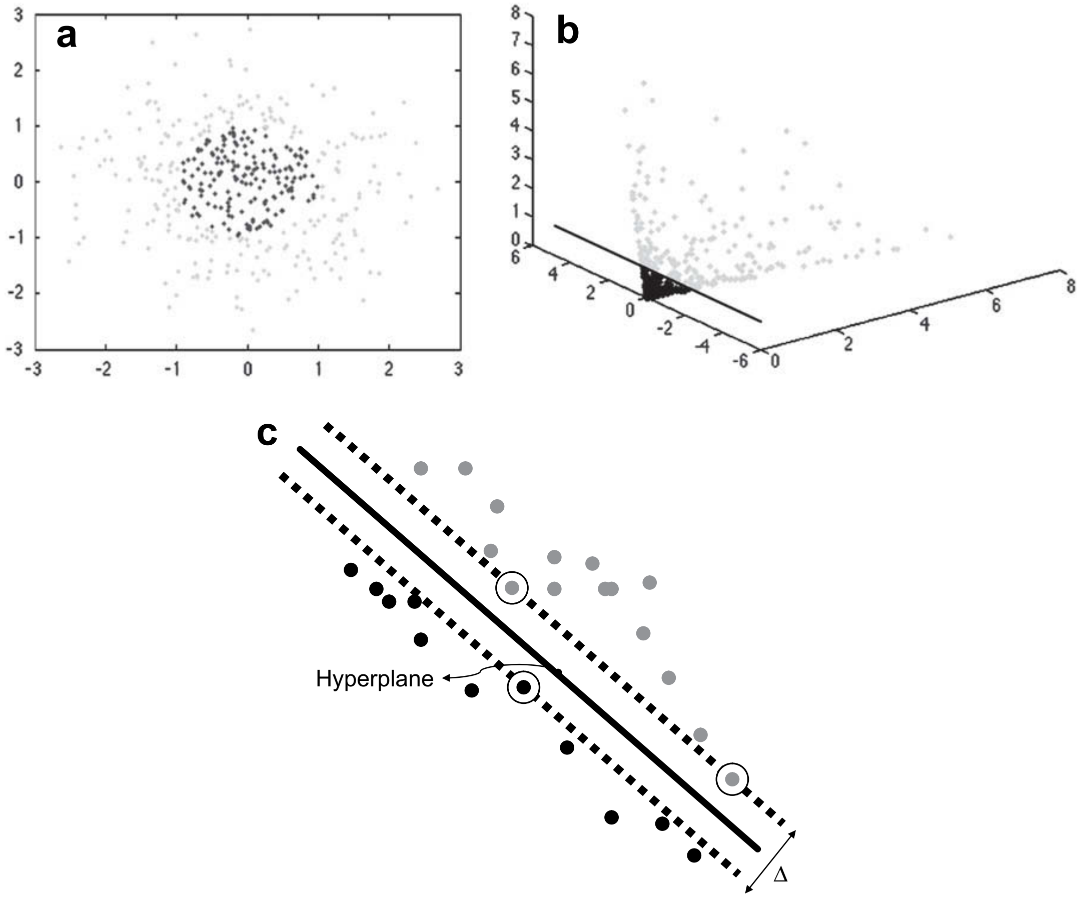


Fig. 9. (a–c). In Figure 9a, a two dimensional dataset with the attributes x_1 and x_2 is depicted. A SVM will be used to find the optimal hyperplane for classifying the data according to this attribute. Each example of the original dataset is a vector in a two-dimensional input space, and the mapping function $\phi(x)$ projects the two-dimensional data into a three-dimensional feature-space. The reason for doing so is that in the higher dimensional feature-space the data will possibly be easier to classify. The mapping in this case shows how the two classes in the dataset (denoted by the different shades of gray) can be easily separated with a linear model in feature-space (Figure 9b). Figure 9c shows the hyperplane found by the SVM algorithm that separates the 2 classes in feature-space. Note that the hyperplane in feature-space corresponds to a complex non-linear separation boundary in input-space. For example, to separate the two classes in the original space of Figure 9a an elliptical boundary would be necessary. This maximum margin hyperplane is unique and is the one that maximizes the distance ($\Delta/2$) between the hyper-plane and the closest examples of each class in the dataset. These closest examples (highlighted in the figure) are the only ones required to fully define the hyperplane, and they are called Support Vectors. The distance to the Support Vectors (and therefore to the hyperplane) is used to classify new previously unseen examples. During training of the model when the hyperplane is defined, and during testing when a new example is classified, only the distance in feature-space between examples is relevant. The actual value taken in feature-space $\phi(x)$ is not. The distance in feature-space can be written as a dot product between the mapped examples as $\phi(x)^T\phi(z)$ and can be defined as a function directly in the input-space $k(x,z) = \phi(x)^T\phi(z)$. This function is called the kernel function and it computes the distance between examples in input-space by first projecting them into the feature space. The fact that the mapping $\phi(x)$ does not need to be explicitly computed (it is implicitly defined by the kernel function) in order to determine the distance in feature-space, is known as the kernel trick. In Figure 9b the kernel function $k(x,z) = (x^Tz)^2$ implicitly defines the mapping $\phi(x) = (x_1, \sqrt{2}x_1x_2, x_2^2)$. The formulation is later extended such that the SVM model allows for misclassifications by means of the soft margin, and the kernel trick is reused for the regression scenario.

Gaussian processes

When making predictions, speaking loosely, we apply a certain function to the inputs to obtain an estimate of a certain output. In contrast to considering a single or a few optimal functions, Gaussian processes (GPs) give a prior probability to every possible function, with higher probabilities for the functions that are more likely. In other words, a GP is a distribution over functions and is a natural generalization of a Gaussian Probability Distribution. In analogy with a Gaussian Distribution which has a mean (a vector) and a covariance (a matrix), the GP over a function is specified by a mean function and a covariance function. A detailed description of GP can be found in Rasmussen & Williams.[68] GP can be used for regression (where the output is continuous) or classification tasks (where the output is discrete). Like SVMs, GPs are kernel methods. They allow for multi-dimensional inputs, have a small number of tunable parameters and result in full predictive distributions as opposed to the point predictions typical of other methods.

The way GP for regression works is explained in detail in Fig. 10.

In GP binary classification, a GP over a function $f(\mathrm{x})$ is defined just as in the regression case, but it is then transformed through a logistic function $\sigma(\bullet)$ so that its outputs lie in the [0,1] interval. This way they can be interpreted as probabilities:$\sigma(f(\mathrm{x}))$. The main advantage of using a GP Classifier over other kernel method classifiers is that it produces an output with a clear probabilistic interpretation.

Fig. 11 is an illustration of how this transformation process by the logistic function takes place.

GPs have been successfully used to model and forecast real dynamic systems because of their flexible modeling abilities and their high predictive performances. In his work, Rasmussen shows how

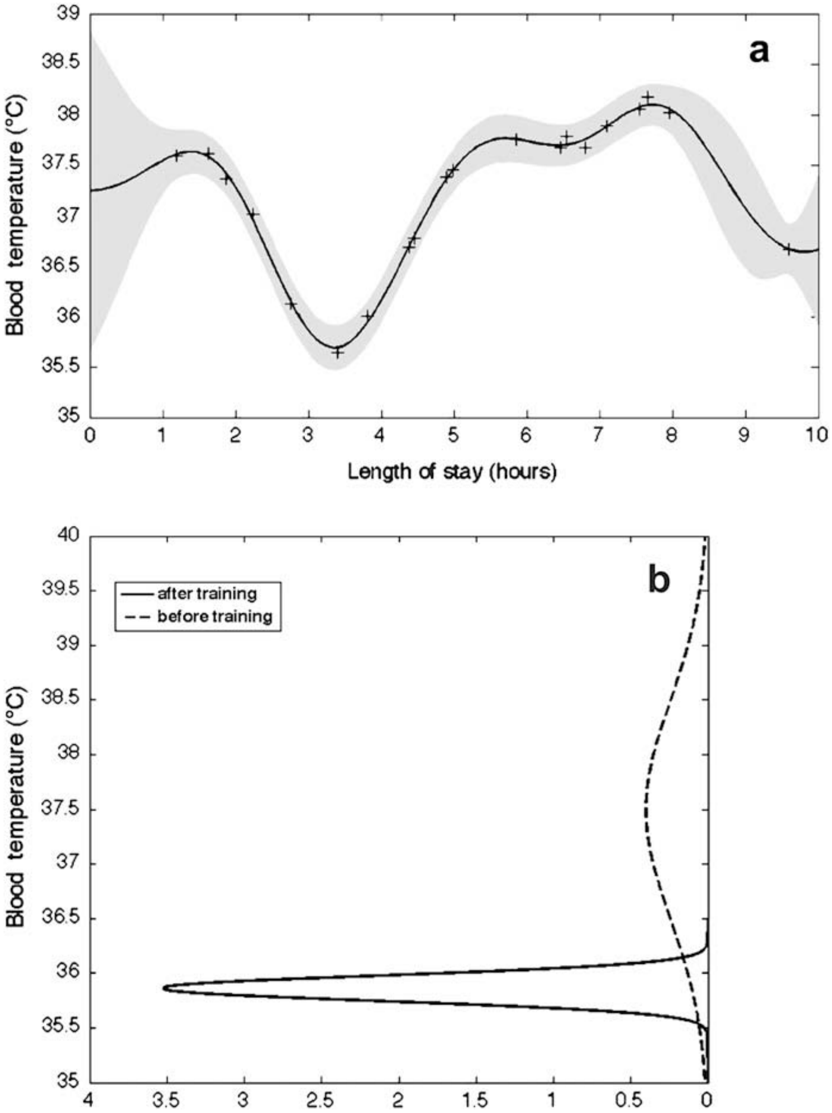


Fig. 10. (a) Gaussian Process to predict the blood temperature of a patient, learned from one-dimensional training data. The crosses correspond to the measured blood temperature values of a patient during his first hours of ICU stay. The bold line corresponds to the mean function of the GP (most likely given the data), and the shaded area corresponds to the 95% confidence region learned for the function distribution. It can be seen that the uncertainty of the prediction grows in regions where there are few training points. (b) Cut-section of the predicted distribution for hour 3 of ICU stay, which has a mean predicted value of 35.86 °C. Also shown (dashed line) is the predicted distribution before training, which has a mean predicted value of 37.5 °C (the mean value of the training data) and is very broad to reflect the uncertainty associated with this prediction. Once learning has occurred, the predictions become more certain because data has been seen in the vicinity of the test point, and the predictions must be consequent with these observations.

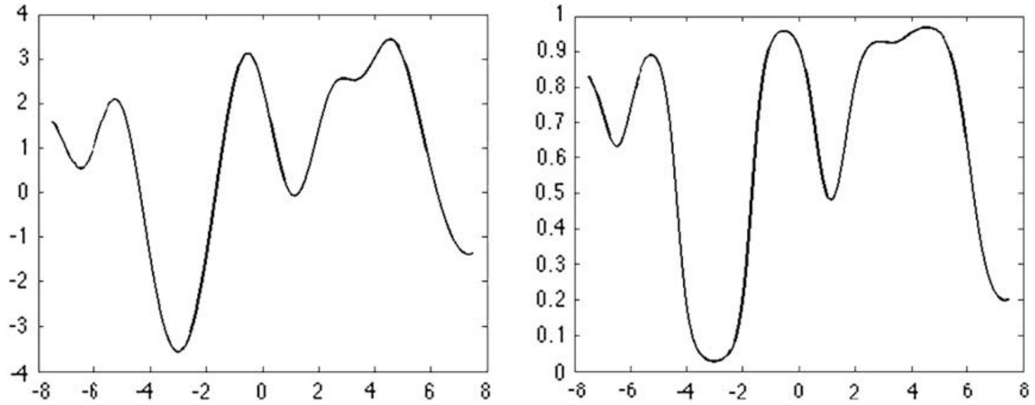


Fig. 11. When using GP for classification, the exemplary function $f(x)$ in the left figure is transformed through a logistic function to $\sigma(f(x))$ (right figure), in order to obtain outputs that lie in the [0,1] interval. The outputs of the logistic function can be interpreted as probabilities.

GPs consistently outperform more conventional methods such as ANNs in different regression tasks.[69] More in-depth analysis of the relationship between GPs and ANNs can be found in the work of Lilley[70] and MacKay.[71]

The application of GPs for regression has recently begun in the intensive care domain. In[72], the evolution of a patient's state during his stay in intensive care was predicted by means of specific patient characteristics. In[73] the patients' core temperature values were predicted several hours in advance.

GPs have been applied to the problem of neonatal seizure detection from electroencephalograph (EEG) signals, where they outperform other modeling methods currently in clinical use for EEG analysis.[74] In the context of intensive care, GPs have also been used to classify patients according to the time-frame in which they can be weaned from mechanical ventilation.[75] In agreement with previous studies, the ability to learn complex non-linear decision boundaries resulted in better performance than more traditional methods such as logistic regression.

Conclusion

A PDMS provides a unique view of the patient, integrating clinical observations, monitoring signals, laboratory values, and therapeutic information on mechanical ventilation, renal replacement therapy and drug administration. Moreover, this information is time-lagged, allowing assessment of the response to therapy. In the future, more and more intensive care units will have access to such large databases. They might contain hidden information for health care policy or benchmarking, and they could serve as sources for the discovery of new medical knowledge. Although many problems, challenges and pitfalls remain unresolved as of yet, there is a need to develop methods to analyse these data. The huge size of the database, and the varying data quality, remains a challenge.

In the field known as data mining, machine learning algorithms are being used routinely to discover valuable knowledge from large databases, such as financial transactions, but also medical records. They have been used in a variety of applications and have been shown to be of special use in data mining scenarios involving large databases with valuable implicit regularities that can be discovered automatically; when the domain is poorly understood and therefore difficult to model by humans; and in domains where dynamic models are needed to adapt to changing conditions.[15] Until now, no single machine learning technique proved to be superior to the others for different tasks. It might therefore be wise to try to run multiple algorithms whenever possible.

Because details of the different machine learning algorithms are not well known in the medical community, the purpose of this review was to give a basic overview of these techniques. Because they can handle large size data samples, and because they can integrate background knowledge into the analysis, they could enable ICU professionals to use their PDMS data for scientific, clinical, or health care policy purpose. This highly specialist branch requires collaboration between clinicians, database specialists, statisticians and computer scientists, who should engage in multidisciplinary teams to set out research projects in this promising domain.

Practice points

- The execution and programming of machine learning techniques is a specialized job for a computer scientist. Nevertheless, for an optimal result, a physician or clinical expert will only be able to guide and understand the research with sufficient basic knowledge of the algorithms.
- A multidisciplinary team of medical doctors, database specialists, statisticians, bench scientists and computer scientists is a prerequisite for using machine learning techniques for advanced patient database research.
- By identifying and formulating research questions at interest in advance, it will be possible to configure and setup the patient database in an optimal way.

Research agenda

- Minimal data quality standards for a patient database need to be developed.
- As the number of clinical databases with ICU and OR data is on the rise, we are only at the beginning of analyzing our clinical databases. The number of possible applications is high.
- As long as no single machine learning technique proved to be superior to the others for different tasks, it might be wise to try to run multiple algorithms whenever possible.

References

1. Elliot TS. *The rock*. Faber & Faber, 1934.
2. East TE. Computers in the ICU: panacea or plague? *Respiratory Care* 1992; **37:** 170–180.
3. Manjoney R. Clinical information systems market – an insider's view. *Journal of Critical Care* 2004; **19:** 215–220.
4. Roncati Zanier E, Ortolano F, Ghisoni L et al. Intracranial pressure monitoring in intensive care: clinical advantages of a computerized system over manual recording. *Critical Care (London, England)* 2007; **11:** R7. doi:10.1186/cc5155.
5. Bosman RJ, Oudemans-van Straaten HM & Zandstra DF. The use of intensive care information systems alters outcome prediction. *Intensive Care Medicine* 1998; **24**(9)**:** 953–958.
6. Bosman RJ, Rood E & Oudemans-van Straaten HM. Intensive care information system reduces documentation time of the nurses after cardiothoracic surgery. *Intensive Care Medicine* 2003; **29:** 83–90.

*7. Fraenkel DJ, Cowie M & Daley P. Quality benefits of an intensive care clinical information system. *Critical Care Medicine* 2003; **31:** 120–125.

8. Colpaert K, Claus B, Somers A et al. Impact of computerized physician order entry on medication prescription errors in the intensive care unit: a controlled cross-sectional trial. *Critical Care (London, England)* 2006; **10:** R21.
9. Shulman R, Singer M, Goldstone J et al. Medication errors: a prospective cohort study of hand-written and computerised physician order entry in the intensive care unit. *Critical Care (London, England)* 2005; **9:** R516–R521.
10. Available from: http://www.leapfroggroup.org/media/file/Leapfrog-Computer_Physician_Order_Entry_Fact_Sheet.pdf [accessed 15.12.07].
11. Morris AH. Decision support and safety of clinical environments. *Quality & Safety in Health Care* 2002; **11:** 69–75.
12. Berger MM, Revelly J-P, Wasserfallen J-B et al. Impact of a computerized information system on quality of nutritional support in the ICU. *Nutrition* 2006; **22:** 221–229.

*13. Ward NS. Using computers for intensive care unit research. *Respiratory Care* 2004; **49**(5)**:** 518–524.

*14. Ramon J, Fierens D, Güiza F et al. Mining data from intensive care patients. *Advanced Engineering Informatics* 2007; **21:** 243–256.

15. Mitchell T (ed.). *Machine learning*. McGraw Hill Publishing Company, 1997.

*16. Lucas P. Bayesian analysis, pattern analysis, and data mining in health care. *Current Opinion in Critical Care* 2004; **10:** 399–403.

*17. Rubenfeld GD, Angus DC, Pinsky MR et al. Outcomes research in critical care. *American Journal of Respiratory and Critical Care Medicine* 1999; **160:** 358–367.

*18. Ward NS. The accuracy of clinical information systems. *Journal of Critical Care* 2004; **19**(4)**:** 221–225.

19. Office of the Secretary, Health, and Human Services. HIPAA administrative simplification: standards for electronic health care claims attachments: proposed rule. *Federal Register* 2005; **70:** 55989–56025.
20. Hogan WR & Wagner MM. Accuracy of data in computer-based patient records. *Journal of the American Medical Informatics Association : JAMIA* 1997; **5:** 342–355.
21. Lawless ST. Crying wolf: false alarms in a pediatric intensive care unit. *Critical Care Medicine* 1994; **22:** 981–985.
22. Verduijn M, Peek N, de Keizer NF et al. Individual and joint expert judgments as reference standards in artifact detection. *Journal of the American Medical Informatics Association : JAMIA* 2008; **15**(2)**:** 227–234.

23. Cleveland H. Information as resource. *The Futurist* 1982**:** 34–39.
24. Ackoff RL. From data to wisdom. *Journal of Applied Systems Analysis* 1989; **16:** 3–9.
25. Hey J. The data, information, knowledge, wisdom chain: the metaphorical link. Available from: http://best.me.berkeley.edu/%7Ejhey03/files/reports/IS290_Finalpaper_HEY.pdf; 2004 [accessed August 2008].
26. Fayyad U, Piatetsky-Shapiro G & Smyth P. From data mining to knowledge discovery: an overview. In Fayyad U, Piatetsky-Shapiro G, Smyth P & Uthurusamy (eds.). *Advances in knowledge discovery and data mining*. The MIT Press, 1996, pp. 495–515.
27. Chapman P, Clinton J, Kerber R et al. *CRISP-DM 1.0: step-by-step data mining guide. Technical report*. CRISP-DM Consortium, 2000.
28. Langley P. *Elements of machine learning*. San Francisco: Morgan Kaufmann, 1996.
29. DeGroot MH. *Probability and statistics*. 2nd edn. Reading, Massachusetts: Addison-Wesley, 1986.
30. Casella G & Berger RL. *Statistical inference*. Pacific Grove. California: Wadsworth and Brooks/Cole, 1990.
31. Duda R & Hart P. *Pattern classification and scene analysis*. New York: John Wiley & Sons Inc, 1973.
32. Dietterich TG & Kong EB. *Proper statistical tests for comparing supervised classification learning algorithms (technical report)*. Corvallis, OR: Department of Computer Science, Oregon State University, 1996.
33. Michalski R & Clustering. In Shapiro S (ed.). *Encyclopedia or artificial intelligence*. New York: John Wiley & Sons Inc, 1987.
34. Lavrač N. Selected techniques for data mining in medicine. *Artificial Intelligence in Medicine* 1999; **16:** 3–23.
35. Tan AC & Gilbert D. Ensemble machine learning on gene expression data for cancer classification. *Applied Bioinformatics* 2003; **2**(3 Suppl)**:** S75–S83.
36. Moser SA, Jones WT & Brosette SE. Application of data mining to intensive care unit microbiologic data. *Emerging Infectious Diseases* 1999; **5:** 454–457.
37. Lucas PJ, de Bruijn NC, Schurink K & Hoepelman A. A probabilistic and decision-theoretic approach to the management of infectious disease at the ICU. *Artificial Intelligence in Medicine* 2000; **19:** 251–297.
38. Biagoli B, Scoletta S, Cevenini G et al. A multivariate Bayesian Model for assessing morbidity after coronary artery surgery. *Critical Care (London, England)* 2006; **10:** R94.
39. Abu-Hanna A & de Keizer N. Integrating classification trees with local logistic regression in intensive care prognosis. *Artificial Intelligence in Medicine* 2003; **29:** 5–23.
40. Hastie T, Tibshirani R & Friedman J. *The elements of statistical learning*. New York: Springer-Verlag, 2001.
41. Witten I & Frank E. *Data mining: practical machine learning tools and techniques*. 2nd edn. San Francisco: Morgan Kaufmann, 2005.
42. Bishop CM. *Pattern recognition and machine learning*. New York: Springer-Verlag, 2006.
43. Breiman L, Friedman J, Olshen R & Stone C. *Classification and regression trees*. California: Wadsworth and Brooks/Cole, 1984.
44. Ganzert S, Guttmann J, Kersting K et al. Analysis of respiratory pressure–volume curves in intensive care medicine using inductive machine learning. *Artificial Intelligence in Medicine* 2002; **26:** 69–86.
45. Andrews P, Sleeman D, McQuatt A et al. In: *Proceedings of the international conference on Artificial Intelligence in Medicine* 1999.
46. Tsien CL, Kohane IS & McIntosh N. Multiple signal integration by decision tree induction to detect artifacts in the neonatal intensive care unit. *Artificial Intelligence in Medicine* 2000; **19:** 189–202.
47. Breiman L. Random forests. *Journal of Machine Learning Research: JMLR* 2001; **45:** 5–32.
48. Vens C, Van Assche A, Blockeel H & Dzeroski S. First order random forests with complex aggregates. In Camacho R, King R & Srinivasan A (eds.). *ILP 2004: Proceedings of the 14th International Conference on Inductive Logic ProgrammingLecture Notes in Computer Science* 2004, pp. 323–340.
49. Van Assche A & Blockeel H. Seeing the forest through the trees: learning a comprehensible model from an ensemble. In Kok J, Koronacki J, Mantaras RJ, Mladenič D & Skowron A (eds.). *ECML 2007: Proceedings of the 18th European Conference on Machine LearningLecture Notes in Artificial Intelligence* 2007, pp. 418–429.
50. Rumelhart D, Widrow B & Lehr M. The basic ideas in neural networks. *Communications of the ACM* 1994; **37:** 87–92.
51. Braithwaite EA, Dripps J, Lyon AJ & Murray A. In Dybowski R & Gant V (eds.). *Clinical applications of artificial neural networks*. Cambridge University Press, 2001.
52. Sargent D. Comparison of artificial neural networks with other statistical approaches. *Cancer* 2001; **91:** 1636–1642.
53. Goss E & Ramchandani H. Survival prediction in the intensive care unit: a comparison of neural networks and binary logit regression. *Socio-Economic Planning Sciences* 1998; **32:** 189–198.
54. Jaimes F, Farbiarz J, Alvarez D & Martínes C. Comparison between logistic regression and neural networks to predict death in patients with suspected sepsis in the emergency room. *Critical Care (London, England)* 2005; **9:** R150–R156.
*55. Clermont G, Angus D, DiRusso S et al. Predicting hospital mortality for patients in the intensive care unit: a comparison of artificial neural networks with logistic regression models. *Critical Care Medicine* 2001; **29:** 291–296.
56. Tong Y, Frize M & Walker R. Extending ventilator duration estimations approach from adult to neonatal intensive care patients using artificial neural networks. *IEEE Transaction on Information Technology in Biomedicine* 2002; **6:** 188–191.
57. Friedman N, Geiger D & Goldszmidt M. Bayesian network classifiers. *Journal of Machine Learning Research : JMLR* 1997; **29:** 131–163.
*58. Lucas PJF, Gaag van der LC & Abu-Hanna A. Bayesian networks in biomedicine and health-care. *Artificial Intelligence in Medicine* 2004; **30:** 201–214.
*59. Noble WS. What is a support vector machine? *Nature Biotechnology* 2006; **24:** 1565–1567.
60. Cristianini N & Shawe-Taylor J. *An introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge: Cambridge University Press, 2000.
61. Pochet NLMM & Suykens JAK. *Ultrasound in Obstetrics and Gynecology* 2006; **27:** 607–608.
62. Bazzani A, Bevilacqua A, Bollini D et al. An SVM classifier to separate false signals from microcalcifications in digital mammogram. *Physics in Medicine and Biology* 2001; **46:** 1651–1663.
63. Van Gestel T, Suykens J, Baesens B et al. Benchmarking least squares support vector machine classifiers. *Journal of Machine Learning Research : JMLR* 2004; **54:** 5–32.
64. Van Looy S, Verplancke T, Benoit D et al. A novel approach for prediction of tacrolimus blood concentration in liver transplantation patients in the intensive care unit through support vector regression. *Critical Care (London, England)* 2007; **11:** R83.

65. Morik K, Brockhausen P & Thorsten J. Combining statistical learning with a knowledge-based approach – a case study in intensive care monitoring. In: *Proceedings of the16th International Conference on Machine Learning* 1999, pp. 268–277.
66. Hiissa M, Pahikkala T, Suominen H et al. *Towards automated classification of intensive care nursing narratives*. Finland: TUCS Technical Report, 2006.
67. Giraldo B, Garde A, Arizmendi C et al. Support vector machine classification applied on weaning trials patients. In: *Proceedings of the 28th IEEE EMBS Annual International Conference* 2006, pp. 5587–5590.
*68. Rasmussen CE & Williams C. *Gaussian processes for machine learning*. Cambridge, Massachusetts: MIT Press, 2006.
69. Rasmussen CE. Evaluation of Gaussian processes and other methods for non-linear regression. Ph.D. Thesis, Department of Computer Science, University of Toronto; 1996.
70. Lilley M & Frean M. Neural networks: a replacement for Gaussian processes? In: . *Intelligent Data Engineering and Automated Learning* 2005; vol. 3578. Springer, 2005, pp. 195–202.
71. MacKay D. Gaussian processes - a replacement for supervised neural networks? In: *Tutorial lecture notes for NIPS* 1997.
72. Güiza F, Ramon J & Blockeel H. Gaussian processes for prediction in intensive care. In Lawrence ND, Schwaighofer A & Quinonero J (eds.). *Proceedings of the Gaussian Processes in Practice Workshop* 2006, pp. 1–4.
73. Güiza F, Ramon J, Meyfroidt G et al. Predicting blood temperature using Gaussian processes. *Journal of Critical Care* 2006; **21:** 354–355.
74. Faul S, Gregorcic G, Boylan G et al. Gaussian process modeling of EEG for the detection of neonatal seizures. *IEEE Transactions on Biomedical Engineering* 2007; **54:** 2151–2162.
75. Güiza F, Van Loon K, Meyfroidt G et al. Time-series analysis techniques combined with Gaussian process classifiers for prediction of clinical stability after coronary bypass surgery. In Hierlemann A (ed.). *Proceedings of biomedical engineering* 2008, pp. 216–221.
76. Van den Berghe G, Wouters P, Weekers F et al. Intensive insulin therapy in critically ill patients. *The New England Journal of Medicine* 2001; **345:** 1359–1367.