ON BAYESIAN NEURAL NETWORKS

Igor Kononenko
University of Ljubljana, Faculty of electrical engineering & computer science,
Tržaška 25, SI-61001 Ljubljana, Slovenia
Phone: +386 61 1231121, Fax: +386 61 264990
e-mail: igor.kononenko@ninurta.fer.uni-lj.si

Keywords: Bayesian neural network, Hopfield's neural network, naive Bayesian classifier, continuous neural network, probability, entropy, machine learning, artificial intelligence, overview

Edited by: Matjaž Gams
Received: September 15, 1993 **Revised:** January 11, 1994 **Accepted:** April 12, 1994

In the paper the contribution of the work on Bayesian neural networks is discussed with respect to previous, current, and potential future research in machine learning. The discrete and the continuous Bayesian neural network model is compared with Hopfield's models. It is shown that the Bayesian neural network's equations are analogous to equations used to describe Hopfield's model. Two different models of the Bayesian neural network are compared with Hopfield's model, one based on Shannon's entropy (probability) and the other based on Good's plausibility (odds). A generalization of the naive-Bayesian classifier is described that enable the basic algorithm to detect the dependencies between neurons.

1 Introduction

Machine learning is a subfield of artificial intelligence (AI) research field (Michalski et al., 1983;1986). AI researchers are concerned with algorithms that would enable computers to behave intelligently. As intelligence is strongly related to learning, especially in recent years machine learning is becoming of central importance for AI research. Although opinions are not unique, machine learning can be defined as subsuming, besides symbolic learning approaches, also parts of pattern recognition and neural networks.

An artificial neural network is constructed from a set of artificial neurons connected with synapses. Artificial neurons are simple elements able to calculate the weighted sum of the contributions of other neurons. The output from a neuron is usually binary, with possible values 0 or 1. The features of artificial neural networks are: biological similarity, high scale parallelism, robustness with respect to missing and incorrect data and with respect to the damage of the network, locally available information, modest software requirements, and the ability of adaptation through learning.

The main problems when designing an artificial neural network are the selection of the appropriate topology, the selection of the learning rule, the convergence of learning, and the convergence of the execution. A major weakness is the inability of the explanation of the execution and the results.

The work on Bayesian neural networks (Kononenko, 1990a) tries to unify probabilistic, symbolic, and neural networks approaches to learning. Various results of the work on Bayesian neural networks were published in several journals and conference proceedings (Kononenko, 1989-1993; Kononenko & Bratko, 1991). Due to the "Jožef Stefan Golden Badge" reward for a distinguished Ph.D. dissertation in technical sciences in Slovenia I was asked by the editor of this paper to answer the following questions about my dissertation:

- How did the work come about?
- What was the key intellectual contribution of the dissertation?

- What was critically left out that is still an open issue?
- What could people build more easily as a result of this work?
- What part would I have done differently given what is known today?
- What current papers reflect the future of that work?

The next section introduces the Bayesian neural networks based on probability and some other contributions of the thesis (Kononenko, 1990a). In section 3 each of the subsections answers one of the above questions. The appendix describes the Bayesian neural network models based on probability ratio.

2 Bayesian neural networks

This section introduces work on Bayesian neural networks. The next subsection gives an overview of the thesis. Subsections 2.2 and 2.3 define discrete and continuous models of Bayesian neural networks based on probability and show the analogy with discrete and continuous Hopfield's models. In subsection 2.4 the information score of classifier's answers is defined. Subsection 2.5 defines the semi-naive Bayesian classifier.

2.1 Overview of the thesis

In (Kononenko, 1990a) it is shown that the "naive" Bayesian classifier can be implemented with an artificial neural network. A Bayesian classifier is called "naive" if it disregards the interdependences between attributes. Bayesian neural networks use for learning the basic Hebbian learning rule. This rule states that the weight of a synapse is increased if both connected neurons are active. The convergence of the execution of the multidirectional feedback Bayesian neural network is proved. The interpretation of the execution of one neuron is defined as the summation of information gains from other neurons.

A multidirectional feedback Bayesian neural network based on probability ratio is also defined and the convergence of the execution is proved. The execution of one neuron of such network is interpreted as the summation of the weights of evidence. The relation with systems for inductive learning of decision trees is given.

It is shown that the naive Bayesian formula can be transformed into the weighted sum which shows the analogy with the Hopfield's feedback neural network model. Analogously to the Hopfield's continuous model, both types of Bayesian neural networks are generalized to continuous neuron states together with the convergence proof.

For the comparison of the performance of different classifiers on different classification problems a method for estimating the information score of a classifier's answer was developed. This measure excludes the influence of prior probabilities of classes and allows the estimation of incomplete answers.

It was shown experimentally that the Bayesian neural network significantly outperforms Hopfield's model with respect to the classification accuracy and the information score while the complexities of the learning and the execution remain equal. In experiments with four medical diagnostic problems the Bayesian neural network slightly outperformed the naive Bayesian classifier as the iterations make the network less sensitive to noise and missing data. The naive Bayesian classifier and the network outperformed the diagnostic accuracy of physicians specialists.

To overcome the naivety of the Bayesian neural network which stems from the independence assumption, an algorithm was developed for learning "semi-naive" feedforward Bayesian neural networks. The idea is to optimize the tradeoff between the reliability of approximations of probabilities and the naivety with respect to the independence assumption. It was shown experimentally that the algorithm performs better than the naive Bayesian classifier.

2.2 Discrete Bayesian neural networks based on probability

Let objects in a given domain be described with a set of attributes, each having a fixed number of values. Let each value of an attribute be rep-

resented with one Boolean variable. Therefore, an object is described with a set of variables $V_i, i = 1..n$. Let the value of the V_j be unknown. If the independence of influences of other variables to the j-th variable is assumed the probability that $V_j = 1$ given the values of other variables can be computed with the following 'naive' Bayesian formula derived from the Bayesian rule (Good, 1950; Kononenko, 1989c) (for brevity conditions $V_i = 1$ will be written simply as V_i):

$$P(V_j|V_1,...,V_n) = P(V_j) \prod_{V_i=1,i\neq j} Q_{ji} \quad (1)$$

where

$$Q_{ji} = \frac{P(V_j|V_i)}{P(V_j)} = \frac{P(V_j\&V_i)}{P(V_j) \times P(V_i)} \quad (2)$$

The 'naive' Bayesian classifier proved to be very efficient in classification problems when compared to other classification methods and to human experts (Kononenko, 1993; 1993a). It can be naturally implemented by a neural network. In the *Bayesian neural network* (Kononenko, 1989a) every neuron in a network is connected with every other neuron via bi-directional connections, called synapses. The *learning phase* changes the weights associated with synapses according to the basic Hebbian learning rule (Hebb, 1949) so that each neuron can use (1) as a *combination function* to compute its *activation level.* Each neuron represents a single value of one attribute, i.e. an attribute with N_i values is represented with N_i neurons. Classes are represented as one additional attribute with one value for each class. Note that the network makes no difference between the attribute that represents classes and other attributes.

The condition $i \neq j$ in (1) is optional, depending on whether each neuron is connected to itself with a *feedback* connection or not. Here, for brevity, feedback connections are omitted. Their influence is studied in more details in (Kononenko, 1989a).

In (Kononenko, 1991b) the following interpretation of one neural network's iteration is proposed. The minus logarithm of (1) gives

$$-log_2P(V_j|V_1,...,V_n) = -log_2P(V_j) - \sum_{V_i=1,i\neq j} log_2Q_{ji} \quad (3)$$

"$-log_2P(Event)$" is interpreted as the *amount of information (in bits) necessary to find out that Event has happened* (or the entropy of the Event (Shannon & Weaver, 1949)). Therefore, (3) is interpreted as follows: the amount of information necessary to find out that j-th neuron is active, given the values of other neurons, is equal to that amount before knowing the values of other neurons minus the sum of information gains from active neurons for the same conclusion.

If $-log_2Q_{ji}$ is replaced with T_{ji}, $-log_2P(V_j)$ with I_j and the left-hand side of (3) with A_j ("A" stands for activation level), the classical weighted sum is obtained from (3) which is used as a combination function in Hopfield's (1982) model:

$$A_j = \sum_{V_i=1,i\neq j} T_{ji} + I_j = \sum_{i\neq j} V_iT_{ji} + I_j \quad (4)$$

In Hopfield's model T_{ji} are elements of the memory matrix obtained as the sum of outer products of training patterns (corrected so that states 0 are changed to -1) and I_j is a constant input to the j-th neuron. Note that the model with no feedback connections ($i \neq j$) corresponds to the memory matrix with zero-diagonal ($T_{ii} = 0$ for all i). Here V_i can be 0 or 1, like in the original Hopfield's model although the learning rule for Hopfield's model is designed for values 1 and -1. In Hopfield's model one memory element corresponds to the difference between a probability that the two connected neurons are in the same state minus the probability that they are in different states. One memory element of the Bayesian neural network corresponds to the probability that the two connected neurons are both active.

The proof of the convergence of the Hopfield's model is based on the idea of the energy of the model. The state of the model determines its energy which with iterations decreases until it reaches a minimum in a fixed point. The energy function used by Hopfield (1982) is:

$$E(V_1,...,V_n) = -\frac{1}{2}\sum_j\sum_{i\neq j} V_jV_iT_{ji} \quad (5)$$

Kosko (1988) uses the similar approach to prove the convergence of the execution phase of bi-directional associative memories. The proof of the convergence of the Bayesian neural network

(Kononenko, 1989a) assumes a similar function representing a measure of *similarity* between the current state of a network and the current activation levels of neurons. During the execution the similarity increases until a maximum is reached in a fixed point. The similarity function used is:

$$Sim(V_1,...,V_n) = \prod_{V_j=1} \frac{P(V_j|V_1,...,V_n)}{P(V_j)} \quad (6)$$

The logarithm of (6) shows the analogy with Hopfield's energy function. The following holds:

$$E(V_1,...,V_n) = -\frac{1}{2} log_2 Sim(V_1,...,V_n) \quad (7)$$

as

$$\begin{aligned} log_2 Sim(V_1,...,V_n) &= \sum_{V_j=1} log_2 \frac{P(V_j|V_1,...,V_n)}{P(V_j)} \\ &= \sum_j V_j log_2 \frac{P(V_j|V_1,...,V_n)}{P(V_j)} \\ &= \sum_j \left(V_j \times \sum_{i \neq j} V_i T_{ji} \right) \\ &= \sum_j \sum_{i \neq j} V_i V_j T_{ji} \end{aligned}$$

The major differences among the two functions are the same as for the differences among the combination functions. According to relation (7) it would be more appropriate to name (5) the *entropy* or the *information content* of the system. Namely the minus logarithm of (6) is interpreted according to the interpretation of (3) as the sum of information gains from all active neurons to the conclusion that other currently active neurons are in fact active. As every information gain appears in the sum twice, because of the symmetry, constant $\frac{1}{2}$ is added to (5). The fixed point is now interpreted as the state with (locally) minimal information content.

2.3 Continuous Bayesian neural network based on probability

In this subsection the discrete model of the Bayesian neural network is generalized to continuous states analogously to Hopfield's continuous model (1984). Instead of discrete states 0 and 1, the state of one neuron will now be represented by any value on the interval [0..1]. The state of a neuron will be proportional to the probability that the neuron is currently active.

Eq. (3) defines the information gain of i-th neuron to the conclusion that j-th neuron is active if the i-th neuron is active with probability 1. A more general definition of information gain is the product:

$$V_i \times log_2 Q_{ji} \quad (8)$$

where V_i represents the current state of i-th neuron (which stands for the probability that i-th neuron is active). (8) can be interpreted as the expected amount of information from i-th neuron to the conclusion that j-th neuron is active. The generalized eq. (3) contains the sum over all neurons (not only the active ones):

$$\begin{aligned} &-log_2 P(V_j|V_1,...,V_n) = \\ &-log_2 P(V_j) - \sum_{i \neq j} (V_i \times log_2 Q_{ji}) \end{aligned} \quad (9)$$

Applying the exponential function to (9) results in the combination function for the continuous model which is the generalization of (1):

$$P(V_j|V_1,...,V_n) = P(V_j) \prod_{i \neq j} {Q_{ji}}^{V_i} \quad (10)$$

Note that the correct generalization of the conditional probability $P(V_j|V_i = 1)$, given the uncertain evidence of $V_i = 1$, is $P(V_j|V_i = 1) \times V_i$ (Ihara, 1987) which leads to different definition than (10). However, (10) is a meaningful generalization if the influences of different neurons (although representing the same attribute) are regarded independently.

Hopfield (1984) defined the dynamics of his continuous model with the following equations:

$$C_j \frac{du_j}{dt} = \sum_{i \neq j} T_{ji} V_i - \frac{u_j}{R_j} + I_j = A_j - \frac{u_j}{R_j} \quad (11)$$

$$V_j = g_j(u_j) \quad (12)$$

where V_i is the output (state) of the i-th neuron, u_j is the input (activation level) of the j-th neuron and I_j, R_j and C_j are constants. "g_j" is the output function defining the input-output relation which is smooth and sigmoid with asymptotes 0 and 1. Eq. (11) states that the speed of

change of u_j is proportional to the difference between current u_j and the new one calculated with (4). Actually, Hopfield omits the condition $i \neq j$ as for all i in his model $T_{ii} = 0$.

Since (9) is analogous to (4), if same replacements are made as in (3) in order to obtain (4), the same dynamics described with (11) and (12) for Hopfield's model (Hopfield, 1984) can be used to define and prove the stability of the *continuous Bayesian neural network model.* The generalized similarity function (6) is therefore:

$$Sim(V_1, ..., V_n) = \prod_j \left(\frac{P(V_j|V_1, ..., V_n)}{P(V_j)} \right)^{V_j} \quad (13)$$

2.4 Information score

Let the correct class of a testing instance T_i be C and the prior probabilty of class C be denoted by $P(C)$. Let the probability, returned by a classifier, that a given testing object T_i belongs to class C be $P'(C)$. We define the *information score* $Inf(T_i)$ of classifier's answer as follows:

1. if $P'(C) > P(C)$ then

$$Inf(T_i) = -\log_2 P(C) + \log_2 P'(C) \quad [bits]$$

i.e., the amount of obtained information is the entire amount of information necessary to correctly classify an instance into class C minus the remainder of information necessary to correctly classify that instance.

2. if $P'(C) = P(C)$ then

$$Inf(T_i) = 0 \quad [bits]$$

i.e., the system didn't change the prior probability of the correct class therefore we didn't obtain any information.

3. if $P'(C) < P(C)$ then

$$Inf'(T_i) =$$

$$-\log_2(1 - P(C)) + \log_2(1 - P'(C)) \quad [bits]$$

i.e., the amount of information returned by the system is the entire amount of information necessary to decide that an instance *does not* belong to class C minus the remainder of the information necessary to make that decision. As this information is in fact wrong we define the information score of the system's answer in this case as negative:

$$Inf(T_i) = -Inf'(T_i) \quad [bits]$$

The *average information score* of an answer is calculated over all testing instances:

$$Inf = \frac{\sum_i^{\#testing\_instances} Inf(T_i)}{\#testing\_instances}$$

Note that this assumes that prior probabilities of classes are known or can be reliably approximated with relative frequencies from training instances. We define also *the relative information score* Inf_r as a normalization of the average information score of an answer with the expected necessary information to classify one instance (i.e. entropy):

$$Inf_r = \frac{Inf}{-\sum_C (P(C) \log_2 P(C))} \times 100\%$$

2.5 Semi-naive Bayesian classifier

Here we will limit our discussion on a feedforward Bayesian neural network that calculates the probability $P(C_j|V_1, ..., V_n)$ of class C_j of an object, described with values $V_1, ..., V_n$ of attributes. Note that for Bayesian neural networks, described in sections 2.2 and 2.3, the class attribute is just one of attributes that describe the object while here it is the target attribute.

When calculating the probability of class C_j in (1) the influence of attributes A_i and A_l is defined with:

$$\frac{P(C_j|V_i)}{P(C_j)} \times \frac{P(C_j|V_l)}{P(C_j)} \quad (14)$$

If, instead of assuming the independence of values V_i and V_l, the values are joint, the corrected influence is given with:

$$\frac{P(C_j|V_iV_l)}{P(C_j)} \quad (15)$$

For joining the two values two conditions should be satisfied: the values of (14) and (15) should be sufficiently different while the approximation of

$P(C_j|V_iV_l)$ with relative frequency should be sufficiently reliable. For the estimation of the reliability of the probability approximation the theorem of Chebyshev can be used. The theorem gives the lower bound on the probability, that relative frequency f of an event after n trials differs from the factual prior probability p for less than ε:

$$P(|f-p| \leq \varepsilon) > 1 - \frac{p(1-p)}{\varepsilon^2 n} \quad (16)$$

The lower bound is proportional to n and to ε^2. In our case we are interested in the reliability of the following approximation:

$$\hat{P}(C_j|V_iV_l)) = \frac{N_{C_jV_iV_l}}{N_{V_iV_l}} \quad (17)$$

Therefore the number of trials n in (16) is equal to $N_{V_iV_l}$, i.e. the number of training instances having values V_i and V_l of attributes A_i and A_l, respectively. As prior probability p is unknown, for approximation of p at the right-hand side of (16) the worst case can be assumed, i.e. $p = 0.5$.

It remains to determine the value of ε. As we are interested also if the values of (14) times $P(C_j)$ and (17) are significantly different we use ε that is proportional to the difference between the two values. The joint values will influence all classes $C_j, j = 1 \ldots m$. Therefore, ε will be the average difference over all classes:

$$\varepsilon = \sum_{j=1}^{m} P(C_j) \times \left| P(C_j|V_iV_l) - \frac{P(C_j|V_i)P(C_j|V_l)}{P(C_j)} \right| \quad (18)$$

It is necessary to determine the threshold for the probability (16) above which decides when it is useful to join two values of two attributes. Empirically (Kononenko, 1991c), a typical value of the threshold that gives satisfactory results is 0.5. Therefore, the rule for joining two values states: join two values if the probability is greater than 0.5 that the theoretically correct (unknown) influence of values V_i and V_l differs, in average over all classes, from the used (approximated) influence, for less than the difference between used influence and the influence of the two values without joining them:

$$1 - \frac{1}{4\varepsilon^2 N_{V_iV_l}} \geq 0.5 \quad (19)$$

The values can be iteratively joint so that more than two values can be joint together enabling the semi-naive Bayesian classifier to discover higher order dependencies.

3 Discussion

3.1 The origin of Bayesian neural networks

The naive Bayesian classifier (that assumes the conditional independence of attributes) is fast, incremental, has no problems with overfitting the training data, and can naturally deal with missing data. Despite its naivety, it achieves the impressive classification accuracy on many real world problems.

Our early experience with the naive Bayesian classifier was presented at the "International School for the Synthesis of Expert's Knowledge" Workshop (Kononenko et al., 1984). We compared the performance of the naive Bayesian classifier with Assistant, an inductive learning algorithm for generating decision trees. Although both systems achieved similar results we claimed that Assistant has an obvious advantage as the generated knowledge in the form of decision trees is transparent to human experts. However, at the workshop Professor Donald Michie pointed out that "if the same amount of effort had been devoted to the development of the naive Bayesian classifier as was used for the development of Assistant, the Bayesian approach would certainly outperform Assistant". Today it seems that his prediction was correct.

We performed a series of experiments in various medical diagnostic problems in order to develop medical diagnostic expert systems. Physicians were never realy satisfied with Assistant's decision trees although Assistant achieved excellent classification performance. They complained that decision trees contain too few attributes and therefore poorly describe the patients (Pirnat et al., 1989). On the other hand the naive Bayesian classifier uses all available attributes. It turned out, that a simple interpretation of its decisions as the sum of information gains for/against certain diagnoses is transparent to physicians and acceptable for everyday use of such diagnostic system (Kononenko, 1989b; 1990b; 1991b; 1993a).

In 1986 the PDP group, with the book (Rumelhart & McClelland, 1986), caused the beginning of the exponential growth of the research effort devoted to neural networks (Anderson & Rosenfeld, 1988). One of the most notable contributions are famous Hopfield's papers (Hopfield, 1982; 1984), that described the single layered feedback neural network architecture with interesting properties. It turned out that the naive Bayesian classifier can be implemented on the same neural network architecture as was used by Hopfield but with appropriately modified learning rule and combination function (Kononenko, 1989a). There is a strong analogy between Bayesian neural network and Hopfield's model (see sections 2.2 and 2.3). Discrete and continuous Bayesian neural networks were defined and empirically they significantly outperformed Hopfield's model with respect to classification accuracy (Kononenko, 1990a).

Although the naive Bayesian classifier is fast, incremental, has excellent performance on real life problems, and can explain its decisions as the sum of information gains, its naivety may result in poor performance in domains with strong dependencies among attributes. To avoid independence assumption we defined the "semi-naive" Bayesian classifier. The idea is to explicitly search for the dependencies between the values of different attributes and if such dependency is discovered the two values are joint (Kononenko, 1991c). The algorithm must solve the trade-off between the non-naivety and the reliability of probability approximations (see section 2.5).

A subproblem in comparison of different classifiers was that classification accuracy may be misleading especially for the classification problems with high variations in prior probabilities of different classes. This problem was particularly illuminated with experiments in two medical diagnostic problems. In the "breast cancer" problem classifiers typically achieved the classification accuracy of about 80% while in the "primary tumor" problem the classification accuracy was about 45%. The classification accuracy for breast cancer seems high while for primary tumor very poor. However, in breast cancer there are only 2 classes, and one has the prior probability equal to 80%! Therefore, a simple classifier that each object classifies into the majority class would also achieve a "high" classification accuracy. On the other hand, in primary tumor, there are 22 possible classes, and the majority class contains only 25% of cases. 45% of the classification accuracy is in fact a fairly good result in this problem!

Professor Michie helped us by suggesting entropy as the basis for the appropriate measure of classifier's performance. We developed the evaluation function called "information score" (Kononenko & Bratko, 1991), that can evaluate answers of classifiers in the form of a probabilistic distribution, appropriately considers differences in prior probabilities of classes, and has natural interpretation that stems from the information theory (see section 2.4).

3.2 The contribution of Bayesian neural networks

The key intellectual contribution is that the use of probability and information theory can naturally (and simply) solve several open issues in machine learning. Instead of using ad-hoc approaches or "fuzzy arithmetic" approaches (Kaufmann & Gupta, 1985) which are widely used in machine learning and neural networks, we used the *probability* (Good, 1950; 1964) as the basic tool for modeling, and the strongly related *information theory* (Shannon & Weaver,1949) as the basic tool for the interpretation.

We showed that the probability can be used to model neural networks by introducing the naive Bayesian formula as a combination function while preserving the basic Hebbian learning rule, which is one of the basic learning rules in neural networks that has also strong biological plausibility (Hebb, 1949). We also used the probability to detect the dependencies among attributes by considering the basic definition of the dependency and the reliability of probability approximations (see section 2.5).

The logarithm of the probability of an event can be interpreted as the information necessary to find out that the event has happened. The interpretation of the naive Bayesian formula directly

follows from this. It is simple, natural, and transparent to human users. Besides, it shows direct relationship of Bayesian neural networks with the Hopfiel's model and it shows also the analogy between the Hopfield's *energy* of the network's state and the *entropy* (or the *information content*) of the state (see section 2.2).

The definition of the "information score" (see section 2.4) of a classifiers answer naturally follows from the definition of the information. The information score has several advantages as was discussed in the previous subsection.

3.3 Open issues

Although the semi-naive Bayesian classifier partially solves the problem of naivety, there is still an open problem which seems to be the key issue of machine learning in general. Namely, for particular problems (e.q. parity problems of higher degrees) the discovering of dependencies between attributes may be either

1. unfeasible due to combinatorial explosion or
2. the discovered dependencies cannot be reliably estimated due to small number of training examples.

In such cases efficient heuristic algorithms are needed to discover the dependencies (first problem) or discover new attributes by deriving them from existing ones (second problem). For such new attributes it should be possible to reliably estimate probabilities from the given training set.

It is well known that the result of the learning strongly depends on the knowledge representation. If the attributes, used to describe objects, are primitive and low level, learning systems will not be able to extract regularities from data. On the other hand, if attributes are high level and informative, most of classification systems will achieve similar classification accuracy. In the former situation one of the two problems mentioned above should be solved. The development of efficient heuristic algorithms for solving these two problems seems to be currently the main research issue in machine learning.

3.4 Possible applications

The developed algorithms may be used either as a tool for analysing data or as a basis for an expert system shell. In fact, the majority of applications that were done in Ljubljana Artificial Intelligence Laboratories (Urbančič et al., 1991), that involved machine learning algorithms, used machine learning as an efficient tool for data analysis. On the other hand, general expert system shell based on Bayesian neural networks may be developed (Kononenko, 1991b) and several prototypes were already implemented (Ritoša, 1992; Grahor, 1992).

One promising wide area of potential applications is medical diagnosis (Kononenko, 1993a). Here the major requirement for any system for supporting medical diagnostic decisions is that decisions of the system must be transparent to physicians. The semi-naive Bayesian classifier seems to be the most appropriate for that purpose. Its decisions can be interpreted as the sum of information gains from different attributes (symptoms, laboratory tests, etc) for/against the diagnosis which is similar to the way physicians actually explain their decisions. We are currently developing one such application in the problem of the prediction of hip-bone break recovery (Kukar, 1993).

Other potential uses of the developed methods include the use of the information score as a simple, transparent, and unbiased evaluation criterion for estimating classifier's performance. Unfortunately, not many researchers have begun to use it in their experiments so far. A feedback Bayesian neural network can be used also as an auto-associative memory that is hopefully more efficient than the Hopfield's auto-associative memory. This, however, must yet be analysed.

3.5 Refinements using today's knowledge

After the dissertation was completed m-estimate of probabilities was developed and used independently by Cestnik (1990) and Smyth & Goodman (1990) to estimate the conditional probabilities in the naive Bayesian formula. Cestnik (1990) showed that m-estimate drastically improves the

classification performance of the naive Bayesian classifier in several real world problems.

The experiments described in the dissertation would be more attractive if m-estimate was used instead of the relative frequency. Besides, m-estimate should even improve the explanation ability of the naive and the semi-naive Bayesian classifier, as it eliminates the high fluctuations of probabilities when relatively small samples of instances are used for training.

3.6 Continuation of the work

The most important open issue described in the dissertation that was later completed is the problem of continuous data. The naive Bayesian classifier is designed to deal with discrete attributes while continuous attributes need to be discretized (its values grouped into intervals) in advance. The problem is how many intervals should one use. Too many intervals may be a too detailed split resulting in a small number of training instances corresponding to each interval. This causes the unreliable estimation of probabilities. On the other hand, too few intervals may result in the loss of the information content of a continuous attribute. Another problem with discretization is the loss of the order of values of the continuous attribute.

We developed a *fuzzy discretization* of continuous attributes that "softens" the bounds between neighbour intervals (Kononenko, 1991a; 1992a). Such discretization may use a lot of intervals without the loss of the reliability of probability approximations. It also implicitly keeps the information about the order of values.

Currently, the *multistrategy learning* is becoming the central research area. The idea is to combine several different learning strategies and/or apply several different learning algorithms on the same problem and then try to combine their results. One promising approach to combining the answers of different classifiers, that was used by Smyth et al. (1990), is (again) the naive Bayesian formula. We showed experimentally that the naive Bayesian combination of answers of different decision rules is acceptable and superior to several other combination methods (Kononenko & Kovačič, 1992; Kononenko, 1992b).

Another open issue, tackled with current research, was described in details in section 3.3. We developed the *successive naive Bayesian classifier* (Kononenko, 1993) and Langley (1993) developed the *recursive Bayes*. Both approaches try to overcome the independence assumption by several successive applications of the naive Bayesian classifier on intermediate results. Both approaches, however, are not satisfactory as the classification accuracy is not better than that of the naive Bayesian classifier and, besides, both systems loose the explanation ability of the simple naive Bayesian classifier. Therefore, currently the "semi-naive" Bayesian classifier seems the most promising.

Acknowledgements

I am indebted to many colleagues for their contribution to the present work. I would like to thank my mentor Prof. Ivan Bratko for many years of consultations, introduction, and guidance into the scientific work. Colleagues from the Artificial Intelligence Laboratory at the Faculty of Electrical Engineering and Computer Science and at the Jožef Stefan Institute have contributed to the dissertation with many suggestions and discussions. I am indebted to Bojan Cestnik for his cooperation in the development and implementation of ASSISTANT and for many informal discussions about machine learning. Informal discussions with colleagues Matevž Kovačič, Alen Varšek, Aram Karalič, Matjaž Gams and Nada Lavrač have often contributed to clearer notions and ideas. Collecting and assembling the experimental data would not be possible without the invaluable help of physicians specialists Dr. Matjaž Zwitter, Dr. Sergej Hojker and Dr. Vlado Pirnat from the University Medical Center in Ljubljana. Prof. Donald Michie has contributed with initial suggestions to the appearance of the article (Kononenko & Bratko, 1991) that defines and analyses the information score. This research was supported by Slovenian Ministry of Science and Technology. The reported work was done in the Artificial Intelligence Laboratory at the Faculty of Electrical Engineering and Computer Science in Ljubljana.

References

[1] Anderson J.A. & Rosenfeld E. (eds.) (1988) *Neurocomputing: Foundations of Research.* Cambridge-London: MIT Press.

[2] Cestnik, B. (1990) Estimating probabilities: A crucial task in machine learning, *Proc. European Conference on Artificial Intelligence*, Stockholm, August 1990, pp.147-149.

[3] Good I.J. (1950) *Probability and the weighing of evidence.* London: Charles Griffin.

[4] Good I.J. (1964) *The Estimation of Probabilities - An Essay on Modern Bayesian Methods*, Cambridge: The MIT Press.

[5] Grahor J. (1992) Simulation of continuous Bayesian neural networks and an expert system shell (in slovene), B.Sc. Thesis. University of Ljubljana, Faculty of electrical eng. & computer sc., Ljubljana, Slovenia.

[6] Hebb, D.O. (1949) *The Organization of Behavior.* New York: Wiley.

[7] Hopfield J.J. (1982) Neural networks and physical systems with emergent collective computational abilities. *Proc. of the National Academy of Sciences* 79:2554-2558.

[8] Hopfield J.J. (1984) Neurons with graded response have collective computational properties like those of two-state neurons. *Proc. of the National Academy of Sciences* 81:4586-4590.

[9] Ihara J. (1987) Extension of Conditional Probability and Measures of Belief and Disbelief in a Hypothesis Based on Uncertain Evidence. *IEEE trans. on pattern analysis and machine intelligence*, 9:561-568.

[10] Kaufmann A. & Gupta M. (1985) *Introduction to fuzzy arithmetic*, New York: Van Nostrand Reinhold.

[11] Kononenko, I. (1989a) Bayesian neural networks. *Biological Cybernetics*, 61:361-370.

[12] Kononenko, I. (1989b) Interpretation of neural networks decisions. *Proc. of IASTED Internat. Conf. on Expert Systems*, Zurich, Switzerland, June 26-28, pp. 224-227.

[13] Kononenko, I. (1989c) ID3, sequential Bayes, naive Bayes and Bayesian neural networks. *Proc. 4th European Working Session on Learning*, Montpellier, France, December 1989, pp.91-98.

[14] Kononenko I. (1989d) Neural networks and artificial intelligence, *Proc. 11th Internat. Conf. Computer at the university*, Cavtat, June 5-9 1989,pp 8.3.1-8.3.8. (also: ISSEK Workshop, Udine, Sept. 1989).

[15] Kononenko I. (1989e) Neural networks (in slovene), *Informatica*, 13:56-71.

[16] Kononenko I. (1990a) Bayesian neural networks (in slovene), Ph.D. Thesis, University of Ljubljana, Faculty of electrical engineering & computer science, Ljubljana, Slovenia.

[17] Kononenko I. (1990b) Comparison of inductive and naive Bayesian learning approaches to automatic knowledge acquisition. In: B. Wielinga et al. (eds.) *Current Trends in Knowledge Acquisition*, Amsterdam: IOS Press.

[18] Kononenko, I. (1991a) Feedforward Bayesian neural networks and continuous attributes, *Proc. Int. Joint Conf. on Neural Networks IJCNN-91*, Singapore, Nov. 1991, pp. 146-151.

[19] Kononenko I. (1991b) Bayesian neural network based expert system shell, *International Journal on Neural Networks*, 2:43-47.

[20] Kononenko I. (1991c) Semi-naive Bayesian classifier, *Proc. European working session on learning*, Porto, March 1991, pp.206-219.

[21] Kononenko I. (1992a) Naive Bayesian classifier and continuous attributes, *Informatica*, 16:1-8.

[22] Kononenko I. (1992b) Combining decisions of multiple rules, In: B.du Boulay & V.Sgurev (eds.) *Artificial Intelligence 5: methodology, systems, applications*, North Holland.

[23] Kononenko I. (1993) Successive naive Bayesian classifier, *Informatica*, 17:167-174.

[24] Kononenko I. (1993a) Inductive and Bayesian learning in medical diagnosis, *Applied Artificial Intelligence*, 7:317-337.

[25] Kononenko, I. & Bratko, I. (1991) Information Based Evaluation Criterion for Classifier's Performance. *Machine Learning Journal*, 6:67-80. (also: *Proc. ISSEK Workshop*, Udine, Sept. 1989).

[26] Kononenko, I., Bratko, I., Roškar, E. (1984) Experiments in automatic learning of medical diagnostic rules, Technical report, Faculty of electrical engineering & computer science, Ljubljana, Slovenia (presented at ISSEK Workshop, Bled, august, 1984).

[27] Kononenko I. & Kovačič M. (1992) Learning as optimization: Stochastic generation of multiple knowledge, In D.Sleeman & P.Edwards (eds.) *Machine learning: Proc. 9th Intern. Conf.*, Morgan Kaufmann, San Mateo, CA.

[28] Kosko, B. (1988) Bidirectional Associative Memories. *IEEE Trans. on Systems, Man, and Cybernetics*, 18:49-60.

[29] Kukar M. (1993) An application of machine learning in hip-bone break diagnosis (in slovene), B.Sc. Thesis. University of Ljubljana, Faculty of electrical eng. & computer sc., Ljubljana, Slovenia.

[30] Langley P. (1993) Induction of recursive Bayesian classifier, In. P.Brazdil (ed.) *Machine learning: Proc. European Conf.*, Springer-Verlag, pp. 153-164.

[31] Michalski, R.S., Carbonell, J.G. & Mitchell, T.M. (eds.) (1983/1986) *Machine Learning: An Artificial Intelligence Approach. Volume I* by Tioga Publ. Comp., 1983 and *Volume II* by Morgan Kaufmann Publ. Inc., 1986.

[32] Michie, D., A. Al Attar (1991) Use of Sequential Bayes with Class Probability Trees. In: J.E. Hayes-Michie, D.Michie & E. Tyugu (eds.) *Machine Intelligence 12*, Oxford: Oxford University Press.

[33] Pirnat V., Kononenko I., Janc T., Bratko I. (1989) Medical Estimation of Automatically Induced Decision Rules, *Proc. of 2nd Europ. Conf. on Artificial Intelligence in Medicine*, City University, London, August 29-31 1989, pp.24-36.

[34] Ritoša V. (1992) An expert system shell based on Bayesian neural networks (in Slovene), B.Sc. Thesis. University of Ljubljana, Faculty of electrical eng. & computer sc., Ljubljana, Slovenia.

[35] Rumelhart, D.E. & McClelland, J.L. (eds.) (1986a) *Parallel Distributed Processing, Vol. 1: Foundations.* Cambridge: MIT Press.

[36] Shannon & Weaver (1949) *The mathematical theory of communications.* Urbana: The University of Illinois Press.

[37] Smyth P. & Goodman R.M. (1990) Rule induction using information theory. In: G.Piarersky & W.Frawley (eds.) *Knowledge Discovery in Databases*, MIT Press.

[38] Smyth P., Goodman R.M., Higgins C. (1990) A hybrid Rule-based Bayesian Classifier, *Proc.European Conf. on Artificial Intelligence*, Stockholm, August, 1990, pp. 610-615.

[39] Urbančič T., Kononenko I., Križman V. (eds.) (1991) Review of applications by Ljubljana Artificial Intelligence Laboratories, Technical report IJS DP-6218, Jožef Stefan Institute, Ljubljana, Slovenia.

Appendix: Bayesian neural networks based on probability ratio

Discrete model

Here, we will briefly define the Bayesian neural network which is more appropriate for comparison with Hopfield's model as it assumes the symmetric interpretation of values of neurons. It is based on Good's (1950) *plausibility* as opposed to Shannon's *entropy* (Shannon & Weaver, 1949), or, from another point of view, it is based on *odds* defined as $\frac{P(X)}{P(\overline{X})}$ as opposed to *probability*.

The Bayesian neural network will implement the 'naive' Bayesian classifier based on odds. Each neuron will now represent one attribute, and one neuron will stand for a class. Each neuron has only two possible values (0 and 1) and therefore all attributes will have only two possible values and there will be only two possible classes. Of course, it will be possible to solve also problems with multivalued attributes and with more than two classes by appropriate coding (binarization of attributes and classes, i.e. one multivalued attribute will be represented with more binary attributes). Note that now the two values (0 and 1) are not interpreted anymore as inactive and active as each represents one

value of an attribute. Again, network makes no difference among attributes and classes.

As all attributes are binary, the value of i-th neuron (i.e. i-th attribute) will be represented with $V_i^{X_i}, X_i =$ 0 or 1. If the independence of influences of other neurons to the activation level of the current neuron is assumed, the analogous formula to (1) can be used:

$$\frac{P(V_j^1|V_1^{X_1},...,V_n^{X_n})}{P(V_j^0|V_1^{X_1},...,V_n^{X_n})} =$$

$$\frac{P(V_j^1)}{P(V_j^0)} \prod_{i \neq j} \frac{Z_{ji}(1,X_i)}{Z_{ji}(0,X_i)} \quad (20)$$

where

$$Z_{ji}(X,Y) = \frac{P(V_j^X, V_i^Y)}{P(V_j^X)P(V_i^Y)} \quad (21)$$

is the influence of i-th neuron on j-th neuron. Note that $P(V_j^0)$ can be calculated from $P(V_j^1)$ and that all $P(V_j^{X_j}, V_i^{X_i})$ can be calculated from $P(V_j^1, V_i^1), P(V_i^1)$ and $P(V_j^1)$. Therefore, if j-th neuron is to be able to calculate value of (21) then it needs the same information as in the Bayesian neural network defined in section 2.2 and *the same learning rule can be applied.*

In the execution phase the network again iterates. Each neuron calculates the quotient among the probability of state 1 and the probability of state 0 as defined in (20). Note that all neurons (not only the active ones as before) will influence the quotient. The threshold value for changing the neuron's state will now be the quotient among prior probabilities of state 1 and state 0 (the first factor at the right hand side of (20)).

The network iterates until there are no more changes in neuron's state (the network reaches a fixed point). In (Kononenko, 1989c) it is shown that this always happens in a finite number of iterations if the network works asynchronously (only one neuron changes its state at a time). The proof is analogous to that in (Kononenko, 1989a). The *measure of similarity* between current activation levels of neurons and their current states $X_i, i = 1..n$, in this case is:

$$Sim(V_1^{X_1},...,V_n^{X_n}) = \prod_j \frac{P(V_j^{X_j}|V_1^{X_1},...,V_n^{X_n})}{P(V_j^{X_j})} \quad (22)$$

The similarity is greater if the calculated probability of the current state of a neuron is greater than the prior probability of that state and vice versa. It is shown in (Kononenko, 1989c) that when one neuron changes its state the similarity increases. As there is a finite number of possible states of a network the similarity measure is bounded and therefore the network will always converge to a fixed point.

Continuous model

The *weight of evidence* from i-th neuron, in favor of the conclusion that j-th neuron is active if the i-th neuron is active with probability 1, is given with (Good, 1950; Michie & Al Attar, 1991):

$$log_2 \frac{Z_{ji}(1,1)}{Z_{ji}(0,1)}$$

The expected weight of evidence given that i-th neuron is active with probability V_i (which represents the current state of i-th neuron), is given with:

$$V_i \times log_2 \frac{Z_{ji}(1,1)}{Z_{ji}(0,1)} + (1 - V_i) \times log_2 \frac{Z_{ji}(1,0)}{Z_{ji}(0,0)} \quad (23)$$

Therefore, the generalized (20) is:

$$\frac{P(V_j^1|V_1,...,V_n)}{P(V_j^0|V_1,...,V_n)} =$$

$$\frac{P(V_j^1)}{P(V_j^0)} \prod_{i \neq j} \left[\left(\frac{Z_{ji}(1,1)}{Z_{ji}(0,1)} \right)^{V_i} \times \left(\frac{Z_{ji}(1,0)}{Z_{ji}(0,0)} \right)^{(1-V_i)} \right] \quad (24)$$

and the corresponding similarity measure:

$$Sim(V_1,...,V_n) =$$

$$\prod_j \left[\left(\frac{P(V_j^1|V_1,...,V_n)}{P(V_j^1)} \right)^{V_j} \times \left(\frac{P(V_j^0)}{P(V_j^0|V_1,...,V_n)} \right)^{(1-V_j)} \right] \quad (25)$$

Let $T_{ji}(X,Y)$ stand for $-log_2 Z_{ji}(X,Y)$. The minus logarithm of (24) does not correspond directly to (4) as it is of the form:

$$A_j = \sum_{i \neq j} \Big(V_i \times (T_{ji}(1,1) - T_{ji}(0,1)) +$$

$$(1 - V_i) \times (T_{ji}(1,0) - Tji(0,0)) \Big) + I_j \quad (26)$$

The equation that describes the dynamics of the continuous Bayesian neural network based on odds is obtained from (11) by replacing A_j with (26). As (26) differs from (4) the convergence proof is not so obvious as for the continuous Bayesian neural network based on probability.

From Hopfield's (1984) proof it can be shown that the sufficient condition for such a model to converge is given with the relation

$$\frac{dE_1}{dt} = \sum_j \left(\frac{dV_j}{dt} \times A_j \right) \quad (27)$$

where E_1 is a function of neural network's state and A_j the activation level of j-th neuron that appears in (11). Hopfield uses the following E_1 as the part of the energy function for the continuous model (again omitting the condition $i \neq j$):

$$E_1(V_1, ..., V_n) = -\frac{1}{2}\sum_j \sum_{i \neq j} V_j V_i T_{ji} + \sum_j I_j V_j \quad (28)$$

If, instead of A_j in (11), activation level defined with (26) is used and if, instead of (28), E_1 is defined using the minus logarithm of (25):

$$E_1(V_1, ..., V_n) = \sum_j I_j V_j +$$

$$\frac{1}{2}\sum_j \sum_{i \neq j} \Big(V_j V_i T_{ji}(1,1) + V_j(1 - V_i) T_{ji}(1,0) +$$

$$(1 - V_j) V_i T_{ji}(0,1) + (1 - V_j)(1 - V_i) T_{ji}(0,0) \Big) \quad (29)$$

then the relation (27) holds, namely:

$$\begin{aligned} \frac{dE_1}{dt} &= \sum_j \left(\frac{dE_1}{dV_j} \times \frac{dV_j}{dt} \right) \\ &= \frac{1}{2}\sum_j (A_j \times \frac{dV_j}{dt}) + \frac{1}{2}\sum_i (A_i \times \frac{dV_i}{dt}) \\ &= \sum_j (\frac{dV_j}{dt} \times A_j) \end{aligned} \quad (30)$$

as $T_{ji}(X, Y) = T_{ij}(Y, X)$.