UDC 004.8

COMPARISON OF NEURAL AND BAYESIAN NETWORKS FOR REAL-TIME DATA CLASSIFICATION

N.H. Abroyan

National Polytechnic University of Armenia

Neural and Bayesian networks have been successfully used in different classification tasks during the last several decades. Then during the last several years, the interest towards deep neural networks have been hugely increased and they have started to be used in vast majority of fields including image, speech, signal processing. Currently field researchers and specialists try to apply neural networks in almost every sphere and system, including systems that deal with real-time data. Eventually neural networks became more popular in industry than Bayesian networks. However, there are some concerns and unanswered questions about this type of usage of neural networks. Especially neural networks are being misused very often in classification tasks, and field specialists do not consider the fact that Bayesian networks could be better solution with better performance and accuracy for a specific problem. In addition, there is a need to consider some factors before choosing the network type, such as transparency of the algorithm, theoretical justification, missing values in data, restriction of being only supervised approach, network building and training time, adaptiveness in case of real-time data. In this work, we present differences of neural networks and Bayesian networks, more specifically for classification tasks for real-time data and carry out theoretical and practical comparison between them. Afterwards, we provide some ideas on which approach is preferable in case of real-time data classification.

Keywords: machine learning, classification, real-time data, neural networks, Bayesian networks.

Introduction. Classification is one of the most popular machine learning problems and there are many algorithms that try to solve it. This work concentrates mostly on classification of real-time data. By saying real-time data, we mean that its total correctness depends not only upon its logical correctness, but also upon the time in which it is used [1]. A lot of examples of such data are in the financial sphere. As an example of such data could be stock prices that rapidly change over time. Another example, which is considered as a popular classification task, could be financial (e.g. credit card) transaction fraud detection system. That is, all orders in financial sphere (e.g. stock exchange markets, banks) are processed and tested for not being fraud. Another real-time data classification problem would be deciding of credit approval, i.e. whether a person should be allowed to take a credit from a bank or not. In all above cases, correctness of data and approval decision strongly depend on time. Thus, the machine learning model should be adaptive to changes. Otherwise, the trained model, some time later could be irrelevant to the newly received data. In this work, we also work with real-time data on the example of the financial transaction fraud detection problem.

Although, some approaches to the usage of neural networks have been proposed for solving the classification task with high accuracy for financial transaction fraud detection problem [2-4], there are some questions and problems that have not been answered yet in case of usage of neural networks. In the frames of this paper we are going to concentrate on those questions and problems, try to compare the two approaches and provide some answers.

Neural Networks. Neural networks are powerful techniques for representing complex relationships between inputs and outputs [5]. They are inspired by the neural structure of the brain, and for certain tasks, they can contain many layers and nodes [6]. During the last several years, deep learning revolutionized and started a new era of machine learning and artificial intelligence [4]. Thanks to the usage of convolutional, recurrent neural networks, restricted Boltzmann machines and some other technologies machine learning researchers and specialists achieved notable results in visual, voice and natural language processing tasks [4]. In our previous research we showed that it is possible to achieve best results in financial transaction fraud detection problems by using recurrent neural networks [4]. However, there are some questions and problems in case of neural networks usage such as the lack of theoretical solid justification, the restriction to be a supervised approach, adaptiveness to newly received data. We will discuss these problems later in this paper.

Bayesian Networks. Bayesian networks are directed acyclic graphs that represent a set of variables (i.e. nodes) and their probabilistic conditional dependencies (encoded in its arcs) [7]. Nodes can represent any kind of variable: a measured parameter, a latent variable or a hypothesis. There are efficient algorithms that perform inference and learning in Bayesian networks [8, 9]. If there is an arc from node A to another node B, A is called a parent of B, and B is a child of A. The intuitive meaning of an arc from node A to node B is that A has a direct influence on B. The set of parent nodes of a node x_i is denoted by *parents*(x_i). The joint probability distribution of the node variables can be written as the product of the local distributions of each node and its parents as [7]:

$$P(x_1, \dots, x_n) = \prod_{i=1}^{n} P(x_i | parents(x_i)).$$

Each node has conditional probability table, that quantifies the effect of the parent nodes. A simple example of a Bayesian network is presented in Fig. 1.

This example presents a Bayesian network for a high school student, who wants to apply for a university. For that he/she needs to pass a SAT test (*SAT* node has *pass* and *no pass* possible states) and present a good recommendation letter (*Letter* node has *good* and *bad* possible states) from one of his high school professors. We can assume, that as SAT is a standard test with a standard level of difficulty, its results depend only on the intelligence of the student (or on the preparation level). A high school Professor may give a good recommendation letter in case if that student received a good grade in the final graduation test. And as that exam is not standardized and specific to that particular high school, its difficulty can vary. Thus, the grade depends on both the difficulty of the test and the student's intelligence (or on the preparation level).

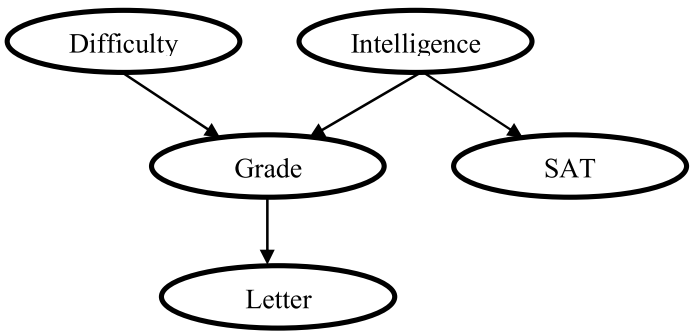

Fig. 1. An example of Bayesian network

Bayesian technology has become popular and well-established, as demonstrated by numerous companies. Some spheres where Bayesian network classifiers have been successfully applied are the following: computing and robotics, medicine and health care, economy, finance and banking, environmental science [10].

The structure of Bayesian networks can be defined either by problem field experts or learned from the data [8]. In case of big datasets with many parameters, it can take a long time to learn the structure. For that reason, there are some Bayesian networks with a predefined structure, which speed up or eliminate the structure learning process.

Naive Bayes. Naive Bayes is one of the Bayesian networks with a predefined structure. In a naive Bayes classifier, each feature variable has the class variable as its only parent. This means that the structure is fixed, and the only task involved in learning is to estimate the parameters [8]. Here, it is assumed that all the features are conditionally independent given the value of the class or more formally $p(\bigcap_{i=1}^{n} x_i \,|y) = \prod_{i=1}^{n} p(x_i|y)$, where x_i *is* the i-th feature, y is the class variable and n is the feature count [10]. An example of naive Bayes is presented in Fig. 2 (a). There are also some extensions of naive Bayes and the most popular one is tree augmented naive Bayes. Here each feature variable can have maximum one feature variable as parent (besides the class variable) [8]. Here structure learning is needed to be done. However, the only thing that needs to be done is finding the optimal single links between the features. An example of tree augmented naive Bayes is presented in Fig. 2 (b).

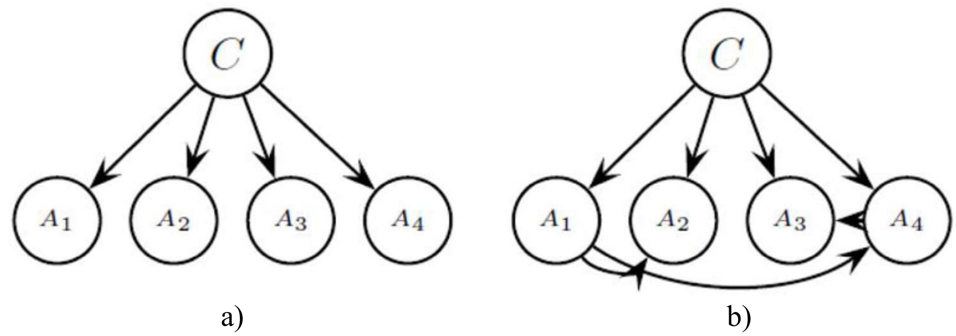

Fig. 2. Examples of naive Bayes (a) and tree augmented naive Bayes (b)

Comparison. Although both neural and Bayesian networks solve similar classification problems and can be presented as graphs, there are significant differences between them, which we have presented below.

Neural networks are discriminative algorithms, Bayesian networks are generative algorithms [5]. In case of classification tasks, discriminative algorithms try to find decision boundaries between classes (i.e. h(x) function) and learn $p(y|x)$ directly, where y represents the class, and x represents features. A new training example is classified in one of classes according to

$$y \epsilon \begin{cases} class\ 1,\ h(x) \geq T\ , \\ class\ 2,\ h(x) < T \end{cases}$$

formula, where T is the threshold value. In contrast, generative algorithms try to build models for each class based on their features, and they learn $p(x|y)$ and $p(y)$. In this case there is one more additional step for computing $p(y|x)$, which is done with Bayes theorem [8, 9]:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}.$$

Here $p(y)$ is the prior probability, which can affect the result, and there are some approaches how to choose it [8, 9].

Neural networks lack theoretical solid justification, nodes (neurons), and edges do not have any meaning individually. A network receives inputs, performs computations and gives an output. Very often neural networks are called black box approach [5]. Up to now, one cannot explain why the network produced that result. Although neural networks have been successfully applied to a vast number of problems [11], these theoretical gaps often prevent the usage of neural networks in some fields. The reason is that the field specialists almost always want to know not only the accuracy of the algorithms, but also an explanation about to prediction. In case of Bayesian networks, all the nodes and edges have a very concrete meaning. Nodes correspond to dataset feature and edge relationships between them [8]. This approach gives an opportunity to understand the way of working of networks and explain why network predicted that result. In case of classification real-time data (e.g. financial transaction fraud detection) Bayesian networks have advantage over neural networks.

The lack of theory and proofs in case of neural networks includes not only neural network training and prediction, but also data preprocessing such as dealing with missing values. There are some approaches to filling the missing values in dataset, however they are also empirical and may or may not work well in practice [5]. In case of Bayesian networks, there are theoretically proved algorithms (such as expectation maximization), which impute the missing values, and that guarantees the most optimal result [8, 9]. These algorithms also perform well in practice.

Another difference is the restriction of neural networks to be a supervised approach [11]. Supervised learning formally can be defined as follows: let $X = (x_1, ..., x_n)$ be a set of n examples, where $x_i \in X$ for all $i \in [n] := \{1, ..., n\}$. The goal is to learn a mapping from x to y, given a training set made of pairs (x_i, y_i) [6]. However, in systems dealing with real-time data, no one could guarantee that all the data will be labeled and y_i will be available for all the examples. Thus, it would be nice to be able to learn from both labeled and unlabeled data, because it is not always feasible to hire experts to label the data. In case of neural networks, all the unlabeled data should be removed from the dataset, because it is not possible to include them in the training process. However, in case of Bayesian networks, the class variable can be monitored as an ordinary feature and the missing data imputing algorithms can be used (such as expectation maximization) [8, 9]. That eventually allows to learn both from labeled and unlabeled data. Thus, for this property, Bayesian networks are more preferable in case of real-time data classification than neural networks.

Neural network structures are defined by researchers/developers by doing experimenting and tuning. There is no theoretically proved rule, which allows to build the most effective neural network for a certain problem [5]. After carrying out experiments and choosing a neural network's structure, the network should be trained in order to learn weights. This can take quite a long time, depending on the training set size, feature count and hardware capabilities. In case of Bayesian networks, structure can be defined by experts or can be learned from data [8]. There are several algorithms that perform Bayesian network structure learning [8, 9]. After building a Bayesian network, there is no need for parameter (weight) learning, because they are automatically learnt. However, in case of predictions, a neural network will do faster than a Bayesian network. Besides, weights and output in neural networks are concrete real numbers, but in Bayesian networks they are probability distributions, which present the idea of uncertainty [8].

The last point is about adaptiveness. In case of dealing with real time data, this is a very important point. Once a neural network is trained, it will be impossible to update the weights without retrain, which will consume a lot of time. Although there are some proposed methods in literature [12], which try to do a workaround, they have not completely proved to be good. In case of Bayesian networks, weights can be updated on the fly without any extra efforts [8]. This gives a model an opportunity to adapt to the continuously received real-time data.

Experiments. After some theoretical comparison, in this section some experimental results are presented. We implement naive Bayes and tree augmented naive Bayes. After doing experiments we compare the results with those of our previous work [4]. We implement Bayesian networks in R with the help of bnlearn package [13]. As an experimental dataset, the "German Credit Data" dataset has been used as previously [14]. For comparing our experiments with the already known results, we are going to measure our model performance with F_1 score [5]:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}.$$

Thirty experiments were performed for classifier with data shuffling. The results are shown in Table 1.

Table 1

Experimental results of F1 score

Experiments	**F_1 scores**		**Experiments**	**F_1 scores**	
№	**Naive Bayes**	**TA Naive Bayes**	**№**	**Naive Bayes**	**TA Naive Bayes**
1	0.79	0.75	16	0.79	0.75
2	0.818	0.772	17	0.794	0.753
3	0.799	0.757	18	0.808	0.764
4	0.83	0.782	19	0.797	0.755
5	0.819	0.773	20	0.798	0.756
6	0.813	0.769	21	0.83	0.789
7	0.807	0.764	22	0.812	0.767
8	0.83	0.791	23	0.795	0.754
9	0.83	0.782	24	0.79	0.75
10	0.827	0.779	25	0.79	0.791
11	0.798	0.756	26	0.808	0.765
12	0.83	0.784	27	0.816	0.771
13	0.825	0.778	28	0.818	0.772
14	0.815	0.791	29	0.82	0.774
15	0.805	0.762	30	0.82	0.774

Experiments show, that the best F_1 score for naive Bayes is up to 0.83 and for the tree augmented naive Bayes is up to 0.79. The comparison of the best and the worst results with already known results of neural networks is presented in Table 2.

Table 2

Results comparison of naive Bayes, TA naive Bayes and neural networks

	Naive Bayes	**TA Naive Bayes**	**Neural networks**
Best result	0.83	0.79	0.92
Worst result	0.79	0.75	0.88

As we can see from the results, despite the advantages over neural networks, in practice, Bayesian networks did not perform as well as neural networks for this specific problem. Currently it could be a tradeoff and a matter of choice which approach to use in the real-time data classification task. Our future work will be devoted to raising the accuracy of Bayesian networks by combining them with discriminative algorithms, such as neural networks.

Conclusion. We presented a comparison and showed the differences of neural and Bayesian networks for the real-time data classification task. Bayesian networks have a strong theoretical basis and are considered trustworthy. While neural networks lack theoretical proofs and are more empiric, a black-box approach. However, our experiments showed that neural networks give better results for real-time data classification. Thus, it is a matter of choice which approach to choose for the task – Bayesian networks with moderate accuracy, but theoretically justified, or neural networks with better accuracy, but considered as a black-box. Our future research will concentrate on finding ways to combine and build a hybrid model from discriminative and generative algorithms, which will improve the accuracy and better performance.

References

1. **Abroyan N.**, **Hakobyan R.** A review of the usage of machine learning // Proceedings of NPUA: Information technologies, Electronics, Radio engineering.- Yerevan, Armenia, 2016.- № 1. - P. 46-54.
2. **Wiese B.**, **Omlin C.** Credit card transactions, fraud detection, and machine learning: modelling time with LSTM recurrent neural networks// Innovations in Neural Information Paradigms and Applications. - 2009. - P. 235-272.
3. **Fu K.**, **Cheng D.**, **Tu Y.**, **Zhang L.** Credit card fraud detection using convolutional neural networks // Proceedings of 23rd International Conference, ICONIP. - Kyoto, Japan, 2016. - Part III.- P. 483-490.
4. **Abroyan N.** Convolutional and Recurrent Neural Networks for Real-time Data Classification // Proceedings of the 7th International Conference on Innovative Computing Technology (INTECH 2017). - Luton, UK, 2017. - P. 42-45.

5. **Boneh D.**, **Ng A.** CS229: machine learning course materials. – Stanford University [http://cs229.stanford.edu/syllabus.html]. – 2016.
6. **Russell S.**, **Norvig P.** Artificial Intelligence: A Modern Approach. - London, UK, Prentice Hall, 2003.
7. **Correa M.**, **Bielza C.**, **Pamies-Teixeira J.** Comparison of Bayesian networks and artificial neural networks // Expert Systems with Applications. - 2009.- Vol. 36, No. 3. - P. 7270–7279.
8. **Jensen F.**, **Nielsen T.** Bayesian Networks and Decision Graphs. - New York, USA, Springer, 2007.
9. **Koller D.**, **Friedman N.** Probabilistic Graphical Models. - Cambridge, Massachusetts, The MIT Press, 2009.
10. **Martinez A.** New models and algorithms semi-naive Bayesian classification focused on AODE paradigm: PhD Thesis / Computing Systems Department, University of Castilla La-Mancha. - 2012.
11. **Goodfellow I.**, **Bengio Y.**, **Courville A.** Deep Learning. - Cambridge, Massachusetts, The MIT Press, 2016.
12. **Sahoo D.**, **Pham Q.**, **Lu J.**, **Hoi S.** Online Deep Learning: Learning Deep Neural Networks on the Fly. – Singapore, arXiv [https://arxiv.org/abs/1711.03705]. - 2017.
13. **Scutari M.**, Learning Bayesian Networks with the bnlearn R // Journal of Statistical Software. - 2010. - Vol. 35. - P. 1-22.
14. **Dheeru D.**, **Karra T.** Efi UCI machine learning repository. - Irvine, California, University of California, School of Information and Computer Sciences [http://archive.ics.uci.edu/ml]. - 2017.

Received on 03.09.2018.
Accepted for publication on 17.01.2019.

ՆԵՅՐՈՆԱՅԻՆ ԵՎ ԲԱՅԵՍՅԱՆ ՑԱՆՑԵՐԻ ՀԱՄԵՄԱՏՈՒՄԸ ԻՐԱԿԱՆ ԺԱՄԱՆԱԿԻ ՏՎՅԱԼՆԵՐԻ ԴԱՍԱԿԱՐԳՄԱՆ ԴԵՊՔՈՒՄ

Ն.Հ. Աբրոյան

Վերջին մի քանի տասնամյակների ընթացքում նեյրոնային ցանցերը և բայեսյան ցանցերը հաջողությամբ կիրառվել են դասակարգման տարբեր խնդիրներում: Այնուհետև վերջին մի քանի տարիներին հետաքրքրությունը նեյրոնային ցանցերի նկատմամբ մեծապես աճեց, և դրանք սկսեցին կիրառվել ամենատարբեր ոլորտներում, այդ թվում՝ նաև պատկերների, ձայնի և ազդանշանների մշակման դեպքում: Ներկայումս ոլորտի հետազոտողները և մասնագետները նեյրոնային ցանցերը փորձում են կիրառել գրեթե ամեն տեսակ համակարգերում՝ ներառյալ այն համակարգերում, որոնք գործ ունեն իրական ժամանակի տվյալների հետ: Ի վերջո, արդյունաբերությունում նեյրոնային ցանցերը դարձան ավելի հայտնի, քան բայեսյան ցանցերը: Սակայն նեյրոնային ցանցերի նման օգտագործման վերաբերյալ կան մի շարք մտահոգություններ: Մասնավորապես, նեյրոնային ցանցերը հաճախ ոչ ճիշտ են օգտագործվում դասակարգման խնդիրների

համար, և ոլորտի մասնագետները հաշվի չեն առնում այն փաստը, որ բայեսյան ցանցերը, կոնկրետ խնդրի դեպքում, կարող են լինել ավելի լավ լուծում՝ ավելի մեծ արագագործությամբ և բարձր ճշտությամբ: Ավելին, մինչ ցանցի տեսակի ընտրելն անհրաժեշտ է հաշվի առնել մի շարք հանգամանքներ, ինչպիսիք են՝ ալգորիթմի թափանցիկությունը, տեսական հիմնավորումը, տվյալներում արժեքների բացակայությունը, միայն վերահսկվող ուսուցումով սահմանափակումը, ցանցի կառուցման և ուսուցման ժամանակը, իրական ժամանակի տվյալների դեպքում՝ հարմարվողականությունը: Ներկայացվում են նեյրոնային և բայեսյան ցանցերի տարբերությունները, հատկապես իրական ժամանակի տվյալների դասակարգման խնդրի դեպքում, և կատարվում է դրանց տեսական և գործնական համեմատություն: Առաջարկվում են մի քանի գաղափարներ՝ իրական ժամանակի տվյալները դասակարգելիս ավելի գերադասելի մոտեցման վերաբերյալ:

Առանցքային բառեր. մեքենայական ուսուցում, դասակարգում, իրական ժամանակի տվյալներ, նեյրոնային ցանցեր, բայեսյան ցանցեր:

СРАВНЕНИЕ НЕЙРОННЫХ И БАЙЕСОВСКИХ СЕТЕЙ ДЛЯ КЛАССИФИКАЦИИ ДАННЫХ РЕАЛЬНОГО ВРЕМЕНИ

Н.О. Аброян

На протяжении последних нескольких десятилетий нейронные и байесовские сети параллельно использовались в различных задачах классификации. В последние годы интерес к нейронным сетям значительно возрос, и они начали использоваться в подавляющем большинстве областей, таких как обработка изображений, речи, сигналов. В настоящее время исследователи и специалисты пытаются применять нейронные сети практически в каждой системе, включая системы, которые обрабатывают данные реального времени. В конце концов нейронные сети стали более популярными в индустрии, чем байесовские сети. Тем не менее есть некоторые опасения по поводу обоснованности использования нейронных сетей при обработке данных реального времени. Перед выбором типа сети необходимо учитывать некоторые факторы, такие как прозрачность алгоритма, теоретическое обоснование, отсутствие некоторых значений в данных, ограничение только контролируемого подхода, время построения и обучения сети, адаптивность в случае данных реального времени. К сожалению, специалисты часто злоупотребляют использованием нейронных сетей в задачах классификации, не учитывая тот факт, что применение байесовских сетей может быть более хорошим решением с лучшей производительностью и точностью для конкретного случая. В данной работе приведены различия нейронных и байесовских сетей, особенно для задач классификации данных реального времени, и сделаны теоретическое и практическое сравнения между ними. Представлены некоторые идеи о том, какой подход предпочтительнее в случае классификации данных реального времени.

Ключевые слова: машинное обучение, классификация, данные реального времени, нейронные сети, байесовские сети.