sustainability

MDPI

Supplementary Materials:

S.1. Reviews on the State-of-the-Art Machine Learning Techniques

Decision Tree (DT) recursively splits the data into branches according to a preset criterion, e.g., on the information gain, to maximize the prediction accuracy [1]. It results in a top-down tree-like structure, in which each non-leaf corresponds to a test on an attribute of cases, each branch corresponds to an outcome of the test, and each leaf node corresponds to a class prediction. DT is one of the most commonly used and well-known techniques for classification and prediction because it handles complex problems in a rapid and effective way. In addition, as a symbolic classifier, DT provides an understandable representation that is easy to interpret by imposing logical rules of classification. The most widely used classification models for DT are the entropy-based algorithms, such as the chi-squared automatic interactive detector (CHAID), classification and regression trees (CART), interactive dichotomizer version 3 (ID3), and C4.5. These produce DTs that are different from one another in the following ways: the number of splits that are allowed at each level of the DT; how those splits are chosen when the DT is built; and how DT growth is limited to prevent overfitting [2]. Among these, the widely used C4.5 model is chosen for the DT in this study.

Naïve Bayes (NB) is a fairly simple probabilistic classification algorithm that uses strong independence assumptions regarding various features [3]. NB assumes that the true distribution of data is a convex combination of distinct distributions, where each feature of the data is conditionally independent. From the training data, NB tries to learn the weights of the combination and the feature marginal within each distribution [4]. For classification, NB conducts the classification process by forecasting the probabilities of a specified instance belonging to each class. It first calculates the probability of the unclassified data belonging to each class, and then classifies the data into the class with the highest probability. NB allows classification models to be built efficiently, so it is found in many cases that NB rivals or outperforms more sophisticated classifiers [5].

Neural network (NN) is based on the nervous system of an organism, such as a neuron, and emulates the accumulation of knowledge in the biological central nervous system [6]. Contrary to conventional computational techniques, NN is able to solve nonlinear and ill-defined problems based on parallel composition [7, 8]. Because of this unique learning capability, NN has been popular and has achieved good performances in different applications [9, 10]. The types of NN are categorized into single-layer NN and multilayer NN. Multilayer NN is preferable and the multilayer feed-forward neural network (FFNN) with the back-propagation learning algorithm is the most popular multilayer NN [6]. However, NN has two main drawbacks: its learning processes are time-consuming and it has a tendency to become stuck in local minima [11]. Radial Basis Function Network (RBFN) has been identified recently as a potential alternative approach because it offers some advantages, such as robustness to noisy data. In addition, RBFN has been shown to be superior over other NN approaches in the following aspects: RBFN models nonlinear data effectively; it can be trained in one stage, rather than using an iterative process, as in MLP; and it can learn the given application quickly [2]. Therefore, RBFN is used as the NN in this paper.

Support Vector Machine (SVM) is based on the structural risk minimization principle from computational learning theory. The support vector can be used to create a hyperplane between two classes [12]. That is, for classification, SVM finds an optimal separating hyperplane that correctly classifies data points as well as possible and separates the points of two classes as much as possible, by minimizing the risk of misclassifying the training samples and unseen test samples. SVM applications that use kernels became popular for many reasons: SVM often concentrates on convex problems; it allows many linear algebra techniques to be used in a nonlinear way; it has shown robustness in many application domains; and it spends fewer resources and half the time of NN [13].

Lastly, as the cutting-edge classification technique, this study adopts a deep learning architecture, which refers to a collection of artificial neural networks that contain many layers of information processing units in a hierarchical manner [14]. While the shallow architectures in modern machine learning algorithms, e.g., SVM, have the serious problem that they are inefficient in terms of the number of computational units, this problem can be resolved by representing a highly varying function compactly through many nonlinearities, via a deep architecture [15]. Among the various deep learning architectures, Deep Belief Network (DBN) is used widely in the various applications, e.g., image processing and speech recognition, where the input data can be represented as fixed feature sets. DBN has multilayers, each of which is composed of a visible layer and one or more hidden layers. The visible layer of the DBN takes the feature sets as its input data, and it delivers the input data to the hidden layers, which are built by stacking one or more restricted Boltzmann machines (RBMs). RBM is a generative stochastic artificial neural network that has only two layers: the input layer and output layer. Parameters in the DBN can be fine-tuned by using the backpropagation algorithm after the greedy layer-wise pre-training of all RBMs in the DBN. These two stages make DBN one of the effective deep-learning-based techniques [16].

References

1. Oztekin, A.; Kong, Z. Y. J.; Delen, D., Development of a structural equation modeling-based decision tree methodology for the analysis of lung transplantations. *Decision Support Systems* **2011,** 51, (1), 155-166.
2. Suh, J. H.; Park, C. H.; Jeon, S. H., Applying text and data mining techniques to forecasting the trend of petitions filed to e-People. *Expert Systems with Applications* **2010,** 37, (10), 7255-7268.
3. Yang, Y. M.; Slattery, S.; Ghani, R., A study of approaches to hypertext categorization. *J Intell Inf Syst* **2002,** 18, (2-3), 219-241.
4. Roy, B. V.; Yan, X., Manipulation Robustness of Collaborative Filtering. *Management Science* **2010,** 56, (11), 1911-1929.
5. Witten, I. H.; Frank, E.; Hall, M. A., Data Mining: Practical Machine Learning Tools and Techniques: Practical Machine Learning Tools and Techniques. Elsevier Science: CA, USA, 2011.
6. Lee, Y. Y.; Kim, J. T.; Yun, G. Y., The neural network predictive model for heat island intensity in Seoul. *Energy and Buildings* **2016,** 110, 353-361.
7. Tkáč, M.; Verner, R., Artificial neural networks in business: Two decades of research. *Applied Soft Computing* **2016,** 38, 788-804.
8. Guzella, T. S.; Caminhas, W. M., A review of machine learning approaches to Spam filtering. *Expert Systems with Applications* **2009,** 36, (7), 10206-10222.
9. Tolle, K. M.; Chen, H.; Chow, H.-H., Estimating drug/plasma concentration levels by applying neural networks to pharmacokinetic data sets. *Decision Support Systems* **2000,** 30, (2), 139-151.
10. Kim, Y. H.; Lewis, F. L., Optimal design of CMAC neural-network controller for robot manipulators. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* **2000,** 30, (1), 22-31.
11. Wei, C.-P.; Yang, C. C.; Lin, C.-M., A Latent Semantic Indexing-based approach to multilingual document clustering. *Decision Support Systems* **2008,** 45, (3), 606-620.
12. Farquad, M. A. H.; Ravi, V.; Raju, S. B., Churn prediction using comprehensible support vector machine: An analytical CRM application. *Applied Soft Computing* **2014,** 19, 31-40.
13. Pinto, T.; Sousa, T. M.; Praça, I.; Vale, Z.; Morais, H., Support Vector Machines for decision support in electricity markets′ strategic bidding. *Neurocomputing* **2016,** 172, 438-445.
14. Abdel-Zaher, A. M.; Eldeib, A. M., Breast cancer classification using deep belief networks. *Expert Systems with Applications* **2016,** 46, 139-144.
15. Bengio, Y.; Lamblin, P.; Popovici, D.; Larochelle, H., Greedy layer-wise training of deep networks. *Advances in neural information processing systems* **2007,** 19, 153.

16. Bengio, Y., Learning Deep Architectures for AI. *Foundations and Trends® in Machine Learning* **2009,** 2, (1), 1-127.