Comparison of classifiers for different data in application of classification

Huirong Gu\textsuperscript{1, a, *, †}, Jiyuan Jiao\textsuperscript{2, b, *, †}

\textsuperscript{1}Department of Mathematical and Computational Sciences, University of Toronto Mississauga, Mississauga, Ontario, L5L 1C6, Canada
\textsuperscript{2}College of Art and Sciences, The Ohio State University, Columbus, Ohio 43210, USA

*Corresponding author's e-mail:
\textsuperscript{a}guzi12138@gmail.com, \textsuperscript{b}Jiao.149@buckeyemail.osu.edu

†These authors contributed equally.

Abstract. The classification task is very important in many application fields, such as image recognition, speech recognition, and text classification. Machine learning and deep learning methods are used as the classifiers in their specific classification tasks. Classical machine learning classifiers, including Random Forest, XGBoost, GMM, and SVM, and deep learning classifiers including CNN and LSTM are compared in this paper to show the different computing characteristics in their specific classification tasks. The comparison results show that the CNN-based classifier performs the best in its own classification, especially the image classification. The results illustrate that the complexity of the classification task may heavily influence the performance of the classifiers. The research in this paper has a reference significance for choosing the right classifier in applying the classification task.

1. Introduction

In our lives, the problems for classification are very important. A lot of data generates from our lives and influences our lives [1]. How to analyze and obtain some important information has a great significance [2-4]. Especially, the classification methods are the important tools for data mining [5]. As well known, the classification tasks are to assign an entity to an initially undefined class so that the individuals in a class are close to each other in a sense. The selection process for the defined classes to which a new entity is to be assigned is better known as identification or transfer [6,7]. Many machine learning and deep learning methods have been proposed in the field of classification [8-10].

Typically, machine learning can effectively deal with classification tasks. For example, Cristiano Premorbid et al. [11] proposed a flexible multi-module architecture for a Multi-Target Detection and Tracking System (MTDTS) complemented with a Bayesian object Classification layer based on finite Gaussian Mixture Models (GMM). GMM parameters are evaluated using an EM (Expectation Maximization) algorithm. As a result, finite-component models were generated from entity vectors extracted from the object classes during the training phase. The joint mixture Gaussian pdf is used to model each class, and using Bayesian approach to distinguish the categories of different objects (persons, re-trunks/posts, and cars) in an external semi-structured environment using laser range finder (LRF) data. Giles M. Foody et al. [12] used the Support Vector Machine (SVM) to train the satellite data, which is
very limited. And their result is very good. To improve the accuracy of land cover classification over a complex Mediterranean landscape with a large number of land cover categories and low inter-class separability, the Random Forest (RF) classifier was applied to spectral and mono- and multi-seasonal textural features extracted from Landsat TM imagery [13]. Recently, deep learning methods (DL) have been applied in the field of classification tasks, which has a great advantage. For example, Charles R. Qi et al. [14] designed a deep learning neural network (PointNet) that can-do classification tasks with geometric data formed by the point cloud. A traditional way of dealing with this data type is to convert such data to standard 3D voxel grids or image collections whereas renders data unnecessarily voluminous and causes issues. The key of Charles R. Qi and his group's work is that they used one symmetric function: max pooling. The network learns a set of optimization functions/criteria for identifying interesting or informative points in the point cloud and effectively encoding the reason for their selection. The network's last fully connected layers can aggregate these learnt optimal values into a global descriptor for the entire form (shape classification) or forecast per point labels (shape segmentation). Not only in classifying normal concrete objects but also in other areas that are more abstract, DL also has good performance. For instance, Heba Mohsen et al. [15] applied a deep neural network (DNN) to classify a dataset of 66 brain MRIs into 4 classes, e.g., normal, glioblastoma, sarcoma, and metastatic bronchogenic carcinoma tumours, and the performance was quite a good overall the performance measures.

In summary, many methods can effectively get a good result in the classification task in different fields. Yousef Rezaei Tabar et al. [16] used CNN to improve the classification performance of EEG motor imagery signals.

A review of the typical machine learning and deep learning methods for the classification task is very important if the processing data is very different.

The rest of this paper include an analysis of different methods for classification in Section 2, comparisons for different classification methods in the application of different data in Section 3, and the conclusion in Section 4.

2. Different Methods for Classification Task

In classification tasks, the classical machine learning methods and the deep learning methods are very useful. In this part, several typical machine learning and deep learning methods for classification tasks are given to analyze the computing characteristics of the above methods.

2.1. Machine Learning for Classification

2.1.1. Random Forest

The random forest model is proposed by Breiman [17] as a kind of classifier, consisting of a combination of tree classifiers where each classifier is generated using a random vector sampled independently from the input vector. Each tree casts a unit vote for the most popular class to classify an input vector.

Assume that a classifier \( m_n \) is said to be consistent if probability error,

\[
L(m_n) = P[m_n(X) \neq Y] \rightarrow 0
\]

Random Forest classifier is obtained by a majority vote among classification trees, such that

\[
\begin{align*}
m_{M,n}(X; \Theta_1, \ldots, \Theta_M, D_n) &= 1, \quad \text{if} \quad \frac{1}{M} \sum_{j=1}^{M} m_n(X; \Theta_j, D_n) > \frac{1}{2} \\
m_{M,n}(X; \Theta_1, \ldots, \Theta_M, D_n) &= 0, \quad \text{otherwise}
\end{align*}
\]

For each random tree classifier,

\[
\begin{align*}
m_n(X; \Theta_j, D_n) &= 1, \quad \text{if} \quad \sum_{i \in D_n^j(\theta_j)} 1_{X_i \in A_{Y_i=1}} > \sum_{i \in D_n^j(\theta_j)} 1_{X_i \in A_{Y_i=0}} \\
m_n(X; \Theta_j, D_n) &= 0, \quad \text{otherwise}
\end{align*}
\]
2.1.2. Extreme Gradient Boosting (XGBoost)

XGBoost is a boosting classifier with some acceleration methods. It first makes a combination of hundreds of tree models with low accuracy. It improves its accuracy by constant iterations of the model, which makes this classifier have high accuracy and low false positive probability [18]. Also, XGBoost can scale more than billions of data while only using much fewer resources [19]. The tree models are optimized by gradient boosting in XGBoost.

Let the output of a tree be

\[ f(x) = w_q(x_i) \quad (4) \]

where \( x \) is the input vector and \( w_q \) is the score of the corresponding leaf \( q \). The output of an ensemble of \( K \) trees will be

\[ y_i = \sum_{k=1}^{K} f_k(x_i) \quad (5) \]

The XGBoost algorithm tries to minimize the following objective function \( J \) at step \( t \):

\[ J(t) = \sum_{i=1}^{n} L\left(y_i, \hat{y}_i^{t-1} + f_t(x_i)\right) + \sum_{i=1}^{T} \Omega(f_i) \quad (6) \]

where the first term contains the train loss function \( L \) (e.g., mean squared error) between real class \( y \) and output \( \hat{y} \) for the \( n \) samples and the second term is the regularization term, which controls the complexity of the model and helps to avoid overfitting.

In XGBoost, the complexity is defined as:

\[ \Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} w_j^2 \quad (7) \]

where \( T \) is the number of leaves, \( \gamma \) is the pseudo-regularization hyper-parameter, depending on each dataset and \( \lambda \) is the L2 norm for leaf weights.

Using gradients for second order approximation of the loss function and finding the optimal weights \( w \), the optimal value of the objective function is:

\[ J(t) = -\frac{1}{2} \sum_{j=1}^{T} \left( \sum_{i \in I} g_i^2 \right) + \gamma T \quad (8) \]

Where \( g_i = \partial \hat{y}^{t-1}_i L(y, \hat{y}^{t-1}) \) and \( h_i = \partial^2 \hat{y}^{t-1}_i L(y, \hat{y}^{t-1}) \) are the gradient statistics on the loss function, and \( I \) is the set of leaves [20].

2.1.3. Gaussian Mixture Model (GMM)

A GMM model is a weighted combination of Gaussian probability density functions (pdf), and for multi-dimensional random vector \( x \), its distribution is defined as:

\[ p(x|\Theta) = \sum_{m=1}^{M} \alpha_m p(x|\Theta_m) \quad (9) \]

for \( M \) Gaussian pdfs \( p(x|\Theta_m) \), which \( \Theta_m = (\mu_m, \Sigma_m) \) denotes the parameter including mean vector \( \mu \) and covariance matrix \( \Sigma \) of the Gaussian distribution and \( \alpha_m \) denotes the weighted vector with \( \sum_{m=1}^{M} \alpha_m = 1 \). For a feature-vector \( \Omega \) in \( d \)-dimension, the matrix Gaussian probability distribution function for each \( i \)-th class that modelled by \( \Theta_i \), can be calculated as such

\[ p(\Omega|\Theta_i) = \sum_{m=1}^{M} \alpha_m i p(\Omega|\Theta_m) \quad (10) \]

where each probability distribution function component is given by
\[
p(\Omega|\theta^i) = \frac{1}{\sqrt{(2\pi)^d|\Sigma^i|}} \exp \left[ -\frac{1}{2}(\Omega - \mu^i)^T(\Sigma^i)^{-1}(\Omega - \mu^i) \right]
\]

(11)

If there is a missing part for the data, EM algorithm can estimate local maximum likelihood for parameters of an underlying distribution from the given data set. GMM parameters for each object’s class model, \( \Theta^i \), that maximize the joint likelihood among the pdf-components were estimated using this EM algorithm as:

\[
p(\Omega^N|\theta^i) = \prod_{j=1}^{N} p(\Omega_j|\theta^j)
\]

(12)

where \( \Omega^N = (\Omega_1, \Omega_2, ..., \Omega_N) \) is a set of N labelled feature-vectors [11].

2.1.4. Support Vector Machine (SVM)

SVM are supervised learning models with associated learning algorithms that analyse data for classification and regression analysis. The classification of two classes using SVM has better result than other methods of classification. Classification using SVM uses a hyperplane in high-dimensional feature space. Classification algorithm of SVM is based on kernel methods. SVM can be separated into two groups.

2.1.4.1 Linear SVM:

Linear SVM is the simplest one as its training samples linearly separable. The linear function is defined as:

\[
f(x) = w^T x + b
\]

(13)

For each training sample \( x_i \), if \( y_i = 1 \) then \( f(x_i) \geq 0 \), and if \( y_i = -1 \) then \( f(x_i) < 0 \). Therefore, training samples can be separated by hyperplane \( f(x) = w^T x + b = 0 \), where \( w \) is weighted vector that is normal to hyperplane, \( b \) is bias, and \( x_i \) is the data point.

2.1.4.2 Non-Linear SVM:

Most of the time, it is impossible to draw a straight line between two classes. Then a nonlinear SVM classifier can be used to find a nonlinear operator, which is used to map the input \( x \) into higher dimensional space \( H \). The nonlinear SVM classifier is defined as:

\[
f(x) = W^T \phi(x) + b
\]

(14)

In this case, different kernel functions are used to analyze the data for non-linearly separable data, such as Quadratic, higher order polynomial kernels. The output is still a linear combination of the training samples [21].

2.2. Deep Learning for Classification

2.2.1. CNN

The convolutional neural network (CNN) is a well-known deep learning method, which produces excellent results in the field of computer vision and pattern recognition [22], such as for visual recognition [23, 24], image retrieval [25], and scene annotation [26].

In the framework of CNN, the input layer can capture the input sequence, such as an image, where the sequence contains \( n \) entries which can be represented as a feature map of dimensionality \( d \times n \), if a \( d \)-dimensional dense vector can represent each entry. The convolution layer is used for representation learning from sliding \( w \)-grams. Especially, the \( w \)-gram \( x_{i-w+1}, \ldots, x_i \) using the convolution weights \( W \in R^{d \times w^d} ; p_j = \tanh(W \cdot c_i + b) \), where bias \( b \in R^d \). The max pooling layer processes all of the \( w \)-gram representations, such as \( x_j = \max(p_{1,j}, p_{2,j}, \ldots) (j = 1, \ldots, d) \). After that, the features of the input can be obtained, and the classifier would classify the features into different classes.
2.2.2. LSTM
The LSTM proposed by Hochreiter and Schmidhuber [27] attempts to circumvent the vanishing gradient problem by separating the memory and output representation and having each dimension of the current memory unit depending linearly on the memory unit of the previous time step. A popular modification of the LSTM uses three gates – input, forget, and output – to modulate how much of the current, the previous, and output representation should be included in the current time step. The long short-term memory (LSTM) units for utterance classification is widely used [28]. The LSTM model is especially used in the times sequence classification because the characteristic of LSTM is based on time sequence.

3. Comparison and Analysis of Different Classification Methods
In this paper, different classical machine learning methods and deep learning methods are compared based on different classification tasks. The comparison results of text classification tasks are shown in Table 1.

| Methods Name | Dataset Name | Classification Task Type | Results |
|--------------|--------------|--------------------------|---------|
| Random Forest | Amazon customers’ product-review data [29] | short text classification | 43.93% |
| XGBoost      | Indonesian news [30] | fake news classification | 92%     |
| SVM          | Amazon customers’ product-review data [29] | short text classification | 44.06% |
| CNN          | MR [31]      | movie reviews classification | 81.5%  |
| LSTM         | MR [31]      | movie reviews classification | 80.1%  |

Table 1: Different methods in different text classification tasks

In Table 1, the column "results" means the accuracy of the different classifiers. Although the specific classification tasks for different methods are very different, the tasks belong to the same type, "text classification".

Obviously, the accuracy of binary classification that "fake news classification" is much higher than the others, such as "short text classification". Suppose the classification tasks are close to each other, for example. In that case, the "movie reviews classification" is close to the "short text classification", however, the DL-based methods perform better than the ML-based methods.

The result shows that the easier the classification task is, the higher the accuracy would be. The more powerful the classifier's performance is, the higher the accuracy would also be.

The results of image classification can be found in Table 2.

| Methods Name | Dataset Name | Classification Task Type                  | Results |
|--------------|--------------|------------------------------------------|---------|
| Random Forest| Multi-temporal set of optical RedEye image [32] | Crops' color classification | 87.4% |
| XGBoost      | Synthetic-aperture radar images [33] | Classifying forests and other landscaped | 92.36% |
| GMM          | Single photon emission computed tomography images [34] | Diseases diagnose | 90.11% |
| SVM          | Multi-temporal set of optical RedEye image [32] | Crops' color classification | 88.1% |
| CNN          | MNIST [35]  | Numbers' classification                 | 99.28% |
| LSTM         | Hyperspectral images [36] | Classifying spatial features | 85.42% |

Table 2: Different methods in different image classification tasks
In Table 2, the results show that the accuracy of the classifiers is close to each other, which illustrates that the performance of different classifiers in the application of image classification has no significant difference. For CNN with numbers classification task, it has exceedingly well performance over all other methods. The results show that the CNN-based classifier has a better performance in the application of image classification. In addition, the complexity of the dataset may also influence the performance of the classifier. For example, the satellite images, pixels can be extremely complicated and hard to distinguish between them. Therefore, the results of satellite image classification are much worse than the results of the clearer image classification.

Furthermore, it is clear to see that the Overall performance of Deep Learning methods is higher than Machine Learning methods. A trend is that if the classification task is simpler, the accuracy is also higher compared with those complicated tasks with more classes to be classified.

4. Conclusion
This paper compared different classical machine learning methods and different classical deep learning methods during the process of different classification tasks. Especially, the results of text classification illustrated that the easier task is, the higher accuracy would be. And the higher performance of the method is, the higher accuracy would also be. However, the CNN-based method performs the best in the image classification task. This is because the CNN-based method has a higher performance when processing the images. This comparison of classifiers in classification tasks has a great significance for reference of selecting the best classifiers in classification tasks.

References
[1] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press, 1996.
[2] G. Piatetsky-Shapiro and W. J. Frawley. Knowledge Discovery in Databases. AAAI/MIT Press, 1991
[3] A. Silberschatz, M. Stonebraker, and J. D. Ullman. Database research: Achievements and opportunities into the 21st century. In Report of an NSF Workshop on the Future of Database Systems Research, May 1995.
[4] Runkler T A. Data mining[M]. Wiesbaden: Vieweg+ Teubner, 2010.
[5] Gorade S M, Deo A, Purohit P. A study of some data mining classification techniques[J]. International Research J. of Engineering and Technology (IRJET), 2017, 4.
[6] Chen Y, Lin Z, Zhao X, et al. Deep learning-based classification of hyperspectral data[J]. IEEE Journal of Selected topics in applied earth observations and remote sensing, 2014, 7(6): 2094-2107.
[7] Dagnetie P. A propos des différentes méthodes de classification numérique[J]. Revue de statistique appliquée, 1966, 14(3): 55-75.
[8] Cormack R M. A review of classification[J]. Journal of the Royal Statistical Society: Series A (General), 1971, 134(3): 321-353.
[9] P Kotsiantis S B, Zaharakis I, Pintelas P. Supervised machine learning: A review of classification techniques[J]. Emerging artificial intelligence applications in computer engineering, 2007, 160(1): 3-24.
[10] Fawaz H I, Forestier G, Weber J, et al. Deep learning for time series classification: a review[J]. Data Mining and Knowledge Discovery, 2019, 33(4): 917-963.
[11] Premebida C, Nunes U. A multi-target tracking and GMM-classifier for intelligent vehicles[C]/2006 IEEE Intelligent Transportation Systems Conference. IEEE, 2006: 313-318.
[12] Foody G M, Mathur A. Toward intelligent training of supervised image classifications: directing training data acquisition for SVM classification[J]. Remote Sensing of Environment, 2004, 93(1-2): 107-117.
[13] Rodriguez-Galiano V F, Chica-Olmo M, Abarca-Hernandez F, et al. Random Forest classification of Mediterranean land cover using multi-seasonal imagery and multi-seasonal texture[J]. Remote Sensing of Environment, 2012, 121: 93-107

[14] Qi C R, Su H, Mo K, et al. Pointnet: Deep learning on point sets for 3d classification and segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 652-660.

[15] Mohsen H, El-Dahshan E S A, El-Horbaty E S M, et al. Classification using deep learning neural networks for brain tumors[J]. Future Computing and Informatics Journal, 2018, 3(1): 68-71.

[16] Tabar Y R, Halici U. A novel deep learning approach for classification of EEG motor imagery signals[J]. Journal of neural engineering, 2016, 14(1): 016003.

[17] Breiman L. Random forests[J]. Machine learning, 2001, 45(1): 5-32.

[18] Chen, Zhuo, et al. "XGBoost classifier for DDoS attack detection and analysis in SDN-based cloud." 2018 IEEE international conference on big data and smart computing (bigcomp). IEEE, 2018.

[19] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016.

[20] Dimitrakopoulos, Georgios N., et al. "Pathway analysis using XGBoost classification in Biomedical Data." Proceedings of the 10th Hellenic Conference on Artificial Intelligence. 2018.

[21] Machhale, Ketan, et al. "MRI brain cancer classification using hybrid classifier (SVM-KNN)." 2015 International Conference on Industrial Instrumentation and Control (ICIC). IEEE, 2015.

[22] Schmidhuber J. Deep learning in neural networks: An overview[J]. Neural networks, 2015, 61: 85-117.

[23] Farabet C, Couprie C, Najman L, et al. Learning hierarchical features for scene labeling[J]. IEEE transactions on pattern analysis and machine intelligence, 2012, 35(8): 1915-1929.

[24] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[J]. Advances in neural information processing systems, 2012, 25: 1097-1105.

[25] Yang X, Qian X, Mei T. Learning salient visual word for scalable mobile image retrieval[J]. Pattern Recognition, 2015, 48(10): 3093-3101.

[26] Othman E, Bazi Y, Alajlan N, et al. Using convolutional features and a sparse autoencoder for land-use scene classification[J]. International Journal of Remote Sensing, 2016, 37(10): 2149-2167.

[27] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.

[28] Ravuri S, Stolcke A. Recurrent neural network and LSTM models for lexical utterance classification[C]//Sixteenth Annual Conference of the International Speech Communication Association. 2015.

[29] McAuley J, Pandey R, Leskovec J. Inferring networks of substitutable and complementary products[C]//Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining. 2015: 785-794.

[30] Haumahulu J P, Permana S D H, Yaddarabullah Y. Fake news classification for Indonesian news using Extreme Gradient Boosting (XGBoost)[C]//IOP Conference Series: Materials Science and Engineering. IOP Publishing, 2021, 1098(5): 052081.

[31] Pang B, Lee L. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales[J]. arXiv preprint cs/0506075, 2005.

[32] Nitze I, Schultethes U, Asche H. Comparison of machine learning algorithms random forest, artificial neural network and support vector machine to maximum likelihood for supervised crop type classification[J]. Proc. of the 4th GEOBIA, 2012, 35.
[33] Memon N, Patel S B, Patel D P. Comparative analysis of artificial neural network and XGBoost algorithm for PolSAR image classification[C]//International Conference on Pattern Recognition and Machine Intelligence. Springer, Cham, 2019: 452-460.

[34] Segovia F, Górriz J M, Ramírez J, et al. Classification of functional brain images using a GMM-based multi-variate approach[J]. Neuroscience Letters, 2010, 474(1): 58-62.

[35] Liu Q, Zhou F, Hang R, et al. Bidirectional-convolutional LSTM based spectral-spatial feature learning for hyperspectral image classification[J]. Remote Sensing, 2017, 9(12): 1330.

[36] Jiang X, Wang Y, Liu W, et al. Capsnet, cnn, fcn: Comparative performance evaluation for image classification[J]. International Journal of Machine Learning and Computing, 2019, 9(6): 840-848.