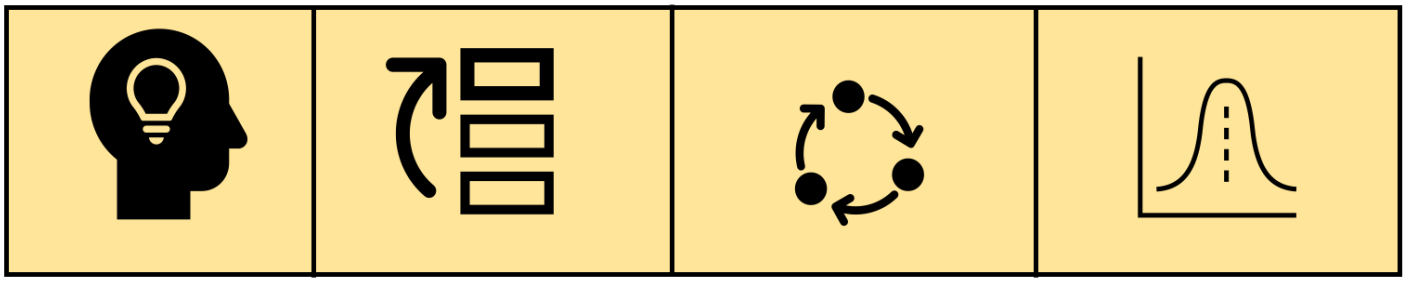
Machine Learning Algorithms: Advantages and Disadvantages

Trevor Andrus

Brigham Young University

Abstract

What are the advantages and disadvantages of machine learning algorithms in their applications?



Characteristics of Machine Learning | Classification Algorithms | Clustering Algorithms | Regression Algorithms

Keywords: Machine learning, Artificial Intelligence, Ensemble Learning, Supervised Learning, Unsupervised Learning, Reinforcement Learning, Classification, Regression, Clustering, Boosting, Stacking

Machine Learning Algorithms: Advantages and Disadvantages

In recent years, data analysis has increasingly become integral in many fields. Terms such as data science, machine learning, and artificial intelligence have become buzz words for innovation. Machine learning is a group of sophisticated algorithms that let computers improve results by self-regulating and optimizing (Yang, 2019). This process of self-optimization is often referred to as learning, and we therefore refer to machines with self-optimization as artificially intelligent.

The learning process is actually only a complex display of mathematics and statistics, but when these intelligent machines are properly implemented, researchers have found they can model probability distributions even more accurately than human experts (Kononenko, 2018). Here lies the power in artificial intelligence – designing computers that can optimize results independently minimizes human error (and effort). It allows for the processing of large quantities of data at super-human speeds. These intelligent algorithms, however, are subject to weaknesses.

Today, we answer the question "What are the advantages and disadvantages of machine learning algorithms in their applications?". We show that there is generally a consensus between experts on the strengths and weaknesses of algorithms – each individual algorithm must balance overfitting, computational efficiency, applicability, and interpretation.

As an introduction to the field of machine learning, we first introduce a general structure for the learning process and describe the three learning styles that classify algorithms. We then outline the three most common applications of machine learning: classification, clustering and regression. Within this context, we first look at the learning styles and structure employed by each algorithm, and then examples of methods with their associated strengths and weaknesses.

Machine Learning Structure

In order to understand the applications of machine learning, it is first necessary to understand the fundamentals of its structure. The machines being referred to in the term machine learning are often only algorithms. Paraphrasing the Oxford American Dictionary, algorithms are a process or set of rules to be followed in calculations or other problem-solving operations (Stevenson & Lindberg, 2010). Algorithms take an input and produce an output as a function of said input. To create artificially intelligent algorithms, these models must be put through a training process to develop self-optimization. This process looks different for each individual algorithm, but the concepts of training remain constant across all models.

Train - Test Split

In order to train, machine learning models split input data into two independent samples. In general, the larger portion of data is allocated to train the model (referred to as training data), and the remaining portion is left to test its efficacy (referred to as test data). The algorithms initially only utilize the training data and attempt to maximize their efficiency on these values. After the learning process, predictions can be made based on trained model parameters. Xu and Goodacre (2018), when trying to determine an optimal proportion for this split between testing and training data, found that there is no end-all-be-all answer to maximize model efficiency. However, Dobbin and Simon (2011) conducted a similar study and found that models with at least 2/3 of the data devoted to training performed the best. This training and testing process is the bread and butter of machine learning, and is often implemented in one of three ways: supervised, unsupervised, or reinforcement.

Learning Styles

In supervised learning models, data is labeled, meaning values are attached to tags that specify information about the data. In the training process, models assess data independent of these tags and then assess their accuracy retrospectively after predictions are made. After model accuracy is determined, models will adjust to align predictions more accurately with the previously provided labels. This method of learning is referred to as supervised because the model is given a task and supplied correct answers for assessment of accuracy. However, a literature survey from the computer science department at the University of Wisconsin Madison notes that labeled data is often difficult, expensive, or time consuming to obtain, as they require the efforts of experienced human annotators (Zhu, 2012). Sathya and Abraham (2013), in a study comparing supervised and unsupervised models, second this observation. Supervised learning methods are easy to train, but they are often hard to prepare.

In unsupervised learning methods, data comes unlabeled, with no tags or additional information attached. Both Kim and Lee (2020) in their application of unsupervised learning to physics problems, and Zhu (2012), agree that one of the greatest advantages to unsupervised learning is that data requires little manipulation. With this method, models are given raw data, and return inherent patterns within the data. During the training process, the model learns to identify and assign individual data points to specific groups, then applies the same groups to the test data (Zhu, 2012). Sathya and Abraham (2013) also note that supervised learning methods are effective in assessing real-time problems, but unsupervised learning methods often take longer to implement.

Reinforcement learning can be viewed as a synthesis of the two previously described methods. This learning style contains elements from both aforementioned styles. Like unsupervised learning, reinforcement learning algorithms do not require labeled data, and are good for exploring large amounts of raw data. However, much like supervised algorithms, this method also attempts to maximize efficiency through optimization of quality parameters. Models that follow a reinforcement learning approach start by identifying basic patterns within raw data, and then maximize their accuracy of identifying these patterns. However, none of these three learning methods is full proof.

Underfitting and Overfitting

Before moving on to specific applications of machine learning, it is necessary to mention perhaps the largest pitfall that most models encounter: overfitting and underfitting. As we previously mentioned, models learn by adapting to training data, and then use that adapted model on test data to perform analysis. Sheikh and Coolen (2019) performed a study identifying a specific algorithm's susceptibilities to overfitting. They defined overfitting as when model parameters seek to explain not only the patterns but also the 'noise' in training data. Overfitting results in the model learning the specific aspects of the training dataset, not just the inherent patterns within the population. In this case, the model has been conditioned to the intricacies of the training data. While this issue may seem minor, a model that is too accustomed to a specific data set loses applicability, and many are known to be biased or break down entirely due to overfitting (Sheikh & Coolen, 2019). Overfitting can be characterized by models performing well on training data, but poorly on test data.

Underfitting is the opposite of overfitting – the model produces poor results even on training data. The model cannot accurately account for patterns and variation, returning results that are essentially useless. Guyun and Yao (1999) comment on the need model designers have to constantly balance models between overfitting and underfitting. Here lies arguably the biggest struggle in training machine learning algorithms: to ensure the model learns patterns of the data but does not become too accustomed to the specific training data (Sheikh & Coolen, 2019).

In summary, machine learning is defined by creating self-updating algorithms. These algorithms learn through a training process, which can be conducted in one of three ways. Supervised learning, unsupervised learning, and reinforcement learning all have individual costs and benefits, but all three are subject to the problems of under and over fitting. Next, these three learning methods are mentioned to build a better understanding of the three applications of machine learning.

Classification

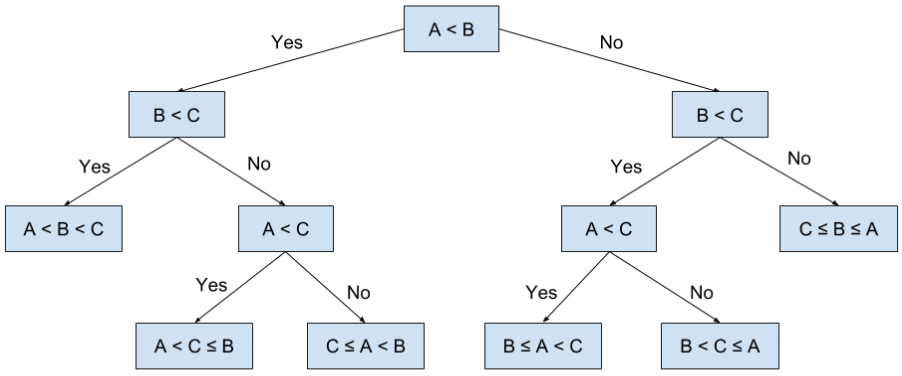
Classification is one of the most conceptually simple machine learning applications and can be viewed as a subcategory of either the supervised or reinforcement learning style (Sathya & Abraham, 2013). These algorithms are a type of predictive modeling that assigns class labels to input data. Classification algorithms first separate labels from input data before sorting them into categories. They then group together data with similar characteristics and assign labels to these groups. After initial assignments are made, the model compares assigned labels to previously provided labels to determine accuracy (Dobbin & Simon, 2011).

Supervised Method

A method that approaches classification from the supervised learning approach is a model called a decision tree. A decision tree is made up of many smaller parts called branches. In its simplest form, each branch can be viewed as a binary classifier (a method that sorts values into one of two categories). The model consecutively passes information from one branch to another, assessing different aspects of the data piece. Figure 1 shows a generic model for a binary tree. After multiple data points are passed into the tree, the tree assigns these data points into categories based on their characteristics.

Figure 1

Decision Tree Structure



Note. Binary trees are a type of classification algorithm that passes data from one branch to another, eventually designating a group to the data piece. Taken from "Classifying data with decision trees" by Oananiculaescu on Github, July 1, 2018 (retrieved from https://elf11.github.io/2018/07/01/python-decision-trees-acm.html)

Decision trees are fast at categorizing data and self-improving, but depending on the desired accuracy, they can also be prone to overfitting. Zeng et al. (2014) discuss the transparency in decision tree models because of their simplicity, but also note how this simplicity produces a lack of possibilities for adaptation. Research from Patel and Prajapati (2018) confirms this stipulation – decision trees are effective as a classifier, but their simplicity limits their application.

Reinforcement Method

A popular method that approaches classification from a reinforcement learning standpoint is a method known as a neural network. Neural networks (sometimes referred to as neural nets) are a complex combination of calculus and probabilistic statistics. Neural nets were initially designed to mimic human brain architecture in an attempt to create models that could solve tasks which more conventional algorithms failed (Schmidhuber, 2015). As such, the model contains artificial neurons to simulate biological neurons, and connections between the neurons to simulate brain synapses. Each neuron takes an input and conveys information to the next neuron based on that input. By increasing the number of layers (and therefore neurons) in a network, the complexity of the model increases exponentially (Schmidhuber, 2015). Following a reinforcement learning approach, a model can take raw data, categorize it, and then improve the accuracy of its predictions based on those categories. Neural nets have become increasingly popular recently with the production of computers with large computational capacity, but again are not immune to downfalls.

Experts agree that Neural networks are accurate, but considerably computationally expensive (Kim & Lee, 2020, Perrusquía & Yu, 2020). Additionally, because reinforcement

learning algorithms require so much data for training, they are more prone to overfitting. As the complexity of the model increases, the data required also increases. If repeated sampling from training data is used, models can become overfit (and therefore ineffective) easily (Schmidhuber, 2015, Sheikh & Coolen, 2019).

Clustering

Cluster analysis is a machine learning method based off of the unsupervised learning style. The idea of unsupervised learning is broad and loosely defined. Any model that can take in data and coerce it into groups based on the maximization of a proposed parameter without data labels can be considered an unsupervised learning algorithm (Kim & Lee, 2020). Cluster analysis algorithms do just that - take large amounts of raw untagged data, and group them into categories based on the data's characteristics. Unlike classification, there are no specific categories, and no correct answers for a clustering algorithm to use to assess its accuracy. While the use of untagged data alleviates some data wrangling in the preparation stages of the algorithm (Kim & Lee, 2020, Zhu, 2012), the lack of correct answers also creates additional work for the model designer during the learning process. If the returned clusters are not what the data scientist is looking for, he or she must manually adjust model parameters to adjust clustering.

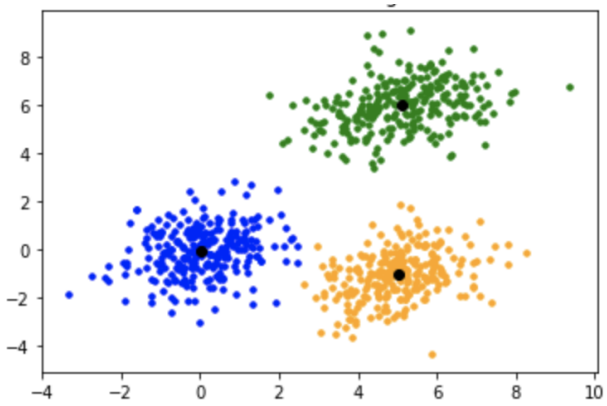
Unsupervised Method

One of the most common artificially intelligent clustering models is the K-means clustering algorithm. As an unsupervised method, this model takes in raw unlabeled data and splits it into a number of clusters specified by the model designer. The algorithm assigns each observation to the cluster with the closest mean. These cluster means are usually designated using the density of the points nearest to it. Referring to the scatterplot in figure 2, the graph

shows inherent patterns within the data when visualized. Because of the shape of the data, the algorithm will select values for clusters that align with that pattern. It will then likewise assign individual observations to these clusters. The K means clustering algorithm can be used only with numerical data and is often used in an exploratory sense to identify patterns within raw data.

Figure 2

Proper allocation of K means clusters



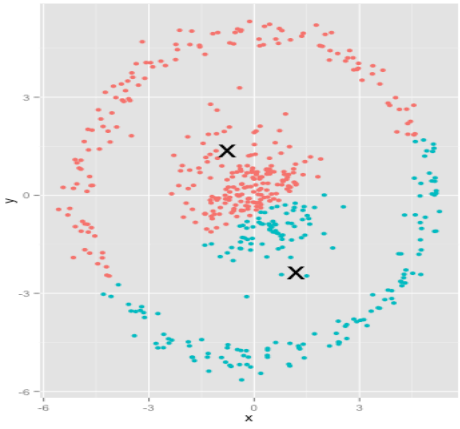
Note. K means clusters are most effective when underlying data distributions can be easily grouped into values with similar means. The figure shows an effective clustering method and proper data shape for K means analysis. Taken from "Ideal Clustering" by Savyalholsa on Geeks for Geeks, June 20, 2020 (retrieved from https://www.geeksforgeeks.org/ml-k-means-algorithm/)

In her study of clustering algorithms, Bhawana (2020) asserts that the simplicity of the K means method makes it computationally inexpensive. Problems of overfitting in clustering algorithms are also easily preventable (Sheikh & Coolen, 2019). Most often, overfitting only

occurs if the number of specified clusters is too large. For example, with a dataset of 350 observations, a K means clustering algorithm with 350 clusters would prove irrelevant because each cluster would represent a single data point. Because the number of clusters is chosen as an exogenous parameter by the model creator, this instance can be easily avoided. However, the K means algorithm is susceptible to underfitting in cases where the data has non-linear underlying relationships (Bhawana, 2020). As can be seen from figure 3, the K means algorithm assigned clusters to points of dense data, but these clusters do not fit the overall pattern of the data.

Figure 3

Improper allocation of K means clusters



Note. K means clusters are most effective when underlying data distributions can be easily grouped into values with similar means, when underlying patterns are non-linear, K means clustering is ineffective. Taken from "K-means clustering is not a free lunch" by David Robinson on Variance explained, January 16, 2015 (retrieved from http://varianceexplained.org/r/kmeans-free-lunch/)

Regression

Regression is arguably the most well-known application of machine learning. Ordinary least squares regression is taught in almost all introductory level statistics courses. Not many complex algorithms are required to perform regression on two variables, but multiple linear regression requires more advanced statistics and optimization models. Regression seeks to identify and quantify the strength of relationships between explanatory and response variables. In order to perform regression, data must include at least one causal variable, and one effect variable. Because data must come with tags, all forms of regression are considered either supervised or reinforcement learning algorithms (Sathya & Abraham, 2013). There are many types of regression, but the two most widely applied forms are linear and logistic regression.

Two Supervised Methods

Linear regression seeks to identify the strength of the linear relationship between two or more quantitative variables. This relationship is determined by deriving parameters to form a line of best fit. This line is created by finding the line which minimizes the squared distance between itself and all individual observations. Assuming variables already share a linear relationship, this line would be the line that most accurately represents the slope of the line formed by graphing the two variables. Once a model is properly fit on training data, predictions can be drawn for hypothetical data points along the same line of fit. This model (like most machine learning models) derives its greatest value in its prediction ability. Kumari and Yadav (2018) conducted a study of linear regression models and reported that the possible applications of linear regression cannot be understated.

Schneider et al. (2010) explored some of the downfalls of textbook linear regression, and how those downfalls may be present in medical applications. Schneider et al. (2010) and Kumari and Yadav (2018) agree in the advantages of the applications of linear regression, but Schneider et al. (2010) also raise the concerns of interpretation. They argue that the biggest downfalls of linear regression do not lie in computational efficiency or overfitting, but in human error in interpreting and applying the models.

Logistic regression is a special case of regression that is also a binary classifier. While linear regression uses a quantitative response variable, logistic regression uses a binary response variable. Because this response variable has only two possible outcomes, this model attempts to classify individual observations into those two categories. Unlike the previous classification algorithms mentioned, logistic regression creates a model based on the individual effects of each independent variable. These effects are independently assessed when fitting the model, and coefficients are often returned to represent the magnitude of each factor's effect.

Schneider et al. (2010) and Kumari and Yadav (2018) also comment on the efficacy of logistic regression, and the results are similar. The applications of regression are undisputed, but the interpretability of logistic regression may be even more abstract than that of linear regression.

Conclusion

Machine learning is a complex field of immense potential. Its possible applications exceed what we may now even imagine. While the world progresses exponentially forward in innovation, there is no doubt that machine learning and artificial intelligence will play an integral role in the process. As has been seen, machine learning is effective in its applications, but each application and algorithm has its individual downfalls. Within classification algorithms, methods

based on supervised learning techniques such as decisions trees are simple, but often lack applicability (Patel & Prajapati 2018, Zeng et al. 2014). Reinforcement classification algorithms improve on applicability but are often more computationally expensive and prone to overfitting (Kim & Lee, 2020, Perrusquía & Yu, 2020). Clustering methods such as K means algorithms are simple, computationally inexpensive, and resistant to overfitting, but require specific data (Bhawana 2020, Sheikh & Coolen, 2019). Finally, regression algorithms are hailed as being computationally inexpensive, and widely applicable, but often see faults in human interpretation (Kumari & Yadav 2018, Schneider et al. 2010).

Looking forward, an area of growing research is aiming to minimize the effects of over and under fitting by designing new learning methods. Hu et al. (2021) are currently conducting research on methods that may be more resistant to the previously discussed problems. In addition to supervised, unsupervised and reinforcement learning, they are developing a method called "Federation learning" which seeks to combine methods internationally and across machine learning classifications. Progress in this field of research will be pivotal in shaping the next generation of data analysis.

In summary, machine learning methods are created to fit specific needs, and these algorithms perform their assigned tasks efficiently within their respective fields. While downfalls such as overfitting, underfitting, computational efficiency, and interpretability may prove obstacles for these machines, the benefits of their proper application will no doubt outweigh their foreseen struggles.

References

Bhawana, K. (2020). A Comparative Study on K-Means Clustering and Agglomerative Hierarchical Clustering. *International Journal of Emerging Trends in Engineering Research*, *8*(5), 1600–1604. https://doi.org/10.30534/ijeter/2020/20852020

Classifying data with decision trees. (2018, July 1). [Graph]. Github. https://elf11.github.io/2018/07/01/python-decision-trees-acm.html

Dobbin, K. K., & Simon, R. M. (2011). Optimally splitting cases for training and testing high dimensional classifiers. *BMC Medical Genomics*, *4*(1), 1–10. https://doi.org/10.1186/1755-8794-4-31

Guyon, X., & Yao, J. F. (1999). On the Underfitting and Overfitting Sets of Models Chosen by Order Selection Criteria. *Journal of Multivariate Analysis*, *70*(2), 221–249. https://doi.org/10.1006/jmva.1999.1828

Hu, L., Yan, H., Li, L., Pan, Z., Liu, X., & Zhang, Z. (2021). MHAT: An efficient model-heterogenous aggregation training scheme for federated learning. *Information Sciences*, *560*, 493–503. https://doi.org/10.1016/j.ins.2021.01.046

Ideal Clustering. (2021, June 11). [Graph]. Geeks for Geeks. https://www.geeksforgeeks.org/ml-k-means-algorithm/

Kim, J., & Lee, C. (2020). Deep unsupervised learning of turbulence for inflow generation at various Reynolds numbers. *Journal of Computational Physics*, *406*, 109216. https://doi.org/10.1016/j.jcp.2019.109216

Kononenko, I. (2018). Early Machine Learning Research in Ljubljana. *An International Journal of Computing and Informatics*, *42*(1), 3–6. https://go-gale-com.erl.lib.byu.edu/ps/retrieve.do?tabID=T002&resultListType=RESULT_LIST&searchResultsType=SingleTab&hitCount=1&searchType=AdvancedSearchForm&currentPosition=1&docId=GALE%7CA538247466&docType=Report&sort=RELEVANCE&contentSegment=ZONE-MOD1&prodId=AONE&pageNum=1&contentSet=GALE%7CA538247466&searchId=R1&userGroupName=byuprovo&inPS=true

Kumari, K., & Yadav, S. (2018). Linear regression analysis study. *Journal of the Practice of Cardiovascular Sciences*, *4*(1), 33. https://doi.org/10.4103/jpcs.jpcs_8_18

Patel, H. H., & Prajapati, P. (2018). Study and Analysis of Decision Tree Based Classification Algorithms. *International Journal of Computer Sciences and Engineering*, *6*(10), 74–78. https://doi.org/10.26438/ijcse/v6i10.7478

Perrusquía, A., & Yu, W. (2020). Continuous-time reinforcement learning for robust control under worst-case uncertainty. *International Journal of Systems Science*, *52*(4), 770–784. https://doi.org/10.1080/00207721.2020.1839142

Robinson, D. (2015, January 16). *K means clustering is not a free lunch* [Graph]. Variance Explained. http://varianceexplained.org/r/kmeans-free-lunch/

Sathya, R., & Abraham, A. (2013). Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification. *International Journal of Advanced Research in Artificial Intelligence*, *2*(2), 1–10. https://doi.org/10.14569/ijarai.2013.020206

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, *61*, 85–117. https://doi.org/10.1016/j.neunet.2014.09.003

Schneider, A., Hommel, G., & Blettner, M. (2010). Linear Regression Analysis. *Deutsches Aerzteblatt Online*, 1–10. https://doi.org/10.3238/arztebl.2010.0776

Sheikh, M., & Coolen, A. C. C. (2019). Analysis of overfitting in the regularized Cox model. *Journal of Physics A: Mathematical and Theoretical*, *52*(38), 384002. https://doi.org/10.1088/1751-8121/ab375c

Stevenson, A., & Lindberg, C. A. (2010). *New Oxford American Dictionary 3rd Edition* (3rd ed.). Oxford University Press. https://www.oed.com

Xu, Y., & Goodacre, R. (2018). On Splitting Training and Validation Set: A Comparative Study of Cross-Validation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning. *Journal of Analysis and Testing*, *2*(3), 249–262. https://doi.org/10.1007/s41664-018-0068-2

Yang, X. (2019). *Introduction to Algorithms for Data Mining and Machine Learning* (1st ed.). Academic Press. https://www-sciencedirect-com.erl.lib.byu.edu/book/9780128172162/introduction-to-algorithms-for-data-mining-and-machine-learning

Zeng, X., Yuan, S., Li, Y., & Zou, Q. (2014). Decision Tree Classification Model for Popularity Forecast of Chinese Colleges. *Journal of Applied Mathematics*, *2014*, 1–7. https://doi.org/10.1155/2014/675806

Zhu, X. (2012, March 15). *Semi-Supervised Learning Literature Survey*. University of Wisconsin-Madison Department of Computer Sciences. https://minds.wisconsin.edu/handle/1793/60444