

Machine learning

Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task.^{[1][2]} Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop a conventional algorithm for effectively performing the task.

Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a field of study within machine learning, and focuses on exploratory data analysis through unsupervised learning.^{[3][4]} In its application across business problems, machine learning is also referred to as predictive analytics.

Contents

Overview

- Machine learning tasks

History and relationships to other fields

- Relation to data mining
- Relation to optimization
- Relation to statistics

Theory

Approaches

- Types of learning algorithms
 - Supervised learning
 - Unsupervised learning
 - Reinforcement learning
 - Self learning
 - Feature learning
 - Sparse dictionary learning
 - Anomaly detection
 - Association rules

Models

- Artificial neural networks
- Decision trees
- Support vector machines
- Regression analysis
- Bayesian networks
- Genetic algorithms

Training models

- Federated learning

Applications

Limitations

Bias

Model assessments

Ethics

Software

Free and open-source software

Proprietary software with free and open-source editions

Proprietary software

Journals

Conferences

See also

References

Further reading

External links

Overview

The name *machine learning* was coined in 1959 by Arthur Samuel.^[5] Tom M. Mitchell provided a widely quoted, more formal definition of the algorithms studied in the machine learning field: "A computer program is said to learn from experience *E* with respect to some class of tasks *T* and performance measure *P* if its performance at tasks in *T*, as measured by *P*, improves with experience *E*."^[6] This definition of the tasks in which machine learning is concerned offers a fundamentally operational definition rather than defining the field in cognitive terms. This follows Alan Turing's proposal in his paper "Computing Machinery and Intelligence", in which the question "Can machines think?" is replaced with the question "Can machines do what we (as thinking entities) can do?".^[7] In Turing's proposal the various characteristics that could be possessed by a *thinking machine* and the various implications in constructing one are exposed.

Machine learning tasks

Machine learning tasks are classified into several broad categories. In supervised learning, the algorithm builds a mathematical model from a set of data that contains both the inputs and the desired outputs. For example, if the task were determining whether an image contained a certain object, the training data for a supervised learning algorithm would include images with and without that object (the input), and each image would have a label (the output) designating whether it contained the object. In special cases, the input may be only partially available, or restricted to special feedback. Semi-supervised learning algorithms develop mathematical models from incomplete training data, where a portion of the sample input doesn't have labels.

Classification algorithms and regression algorithms are types of supervised learning. Classification algorithms are used when the outputs are restricted to a limited set of values. For a classification algorithm that filters emails, the input would be an incoming email, and the output would be the name of the folder in which to file the email. For an algorithm that identifies spam emails, the output would be the

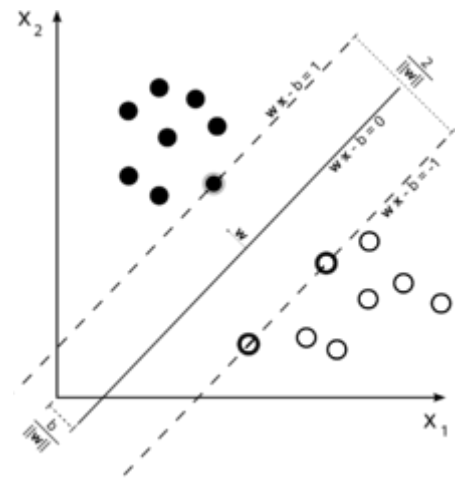
prediction of either "spam" or "not spam", represented by the Boolean values true and false. Regression algorithms are named for their continuous outputs, meaning they may have any value within a range. Examples of a continuous value are the temperature, length, or price of an object.

In unsupervised learning, the algorithm builds a mathematical model from a set of data that contains only inputs and no desired output labels. Unsupervised learning algorithms are used to find structure in the data, like grouping or clustering of data points. Unsupervised learning can discover patterns in the data, and can group the inputs into categories, as in feature learning. Dimensionality reduction is the process of reducing the number of "features", or inputs, in a set of data.

Active learning algorithms access the desired outputs (training labels) for a limited set of inputs based on a budget and optimize the choice of inputs for which it will acquire training labels. When used interactively, these can be presented to a human user for labeling. Reinforcement learning algorithms are given feedback in the form of positive or negative reinforcement in a dynamic environment and are used in autonomous vehicles or in learning to play a game against a human opponent.^{[2]:3} Other specialized algorithms in machine learning include topic modeling, where the computer program is given a set of natural language documents and finds other documents that cover similar topics. Machine learning algorithms can be used to find the unobservable probability density function in density estimation problems. Meta learning algorithms learn their own inductive bias based on previous experience. In developmental robotics, robot learning algorithms generate their own sequences of learning experiences, also known as a curriculum, to cumulatively acquire new skills through self-guided exploration and social interaction with humans. These robots use guidance mechanisms such as active learning, maturation, motor synergies, and imitation.

History and relationships to other fields

Arthur Samuel, an American pioneer in the field of computer gaming and artificial intelligence, coined the term "Machine Learning" in 1959 while at IBM.^[8] A representative book of the machine learning research during the 1960s was the Nilsson's book on Learning Machines, dealing mostly with machine learning for pattern classification.^[9] The interest of machine learning related to pattern recognition continued during the 1970s, as described in the book of Duda and Hart in 1973.^[10] In 1981 a report was given on using teaching strategies so that a neural network learns to recognize 40 characters (26 letters, 10 digits, and 4 special symbols) from a computer terminal.^[11] As a scientific endeavor, machine learning grew out of the quest for artificial intelligence. Already in the early days of AI as an academic discipline, some researchers were interested in having machines learn from data. They attempted to approach the problem with various symbolic methods, as well as what were then termed "neural networks"; these were mostly perceptrons and other models that were later found to be reinventions of the generalized linear models of statistics.^[12] Probabilistic reasoning was also employed, especially in automated medical diagnosis.^{[13]:488}



A support vector machine is a supervised learning model that divides the data into regions separated by a linear boundary. Here, the linear boundary divides the black circles from the white.

However, an increasing emphasis on the logical, knowledge-based approach caused a rift between AI and machine learning. Probabilistic systems were plagued by theoretical and practical problems of data acquisition and representation.^{[13]:488} By 1980, expert systems had come to dominate AI, and statistics was out of favor.^[14] Work on symbolic/knowledge-based learning did continue within AI, leading to inductive logic programming, but the more statistical line of research was now outside the field of AI proper, in pattern recognition and information retrieval.^{[13]:708–710; 755} Neural networks research had been abandoned by AI and computer science around the same time. This line, too, was continued outside the AI/CS field, as "connectionism", by researchers from other disciplines including Hopfield, Rumelhart and Hinton. Their main success came in the mid-1980s with the reinvention of backpropagation.^{[13]:25}

Machine learning, reorganized as a separate field, started to flourish in the 1990s. The field changed its goal from achieving artificial intelligence to tackling solvable problems of a practical nature. It shifted focus away from the symbolic approaches it had inherited from AI, and toward methods and models borrowed from statistics and probability theory.^[14] It also benefited from the increasing availability of digitized information, and the ability to distribute it via the Internet.

Relation to data mining

Machine learning and data mining often employ the same methods and overlap significantly, but while machine learning focuses on prediction, based on *known* properties learned from the training data, data mining focuses on the discovery of (previously) *unknown* properties in the data (this is the analysis step of knowledge discovery in databases). Data mining uses many machine learning methods, but with different goals; on the other hand, machine learning also employs data mining methods as "unsupervised learning" or as a preprocessing step to improve learner accuracy. Much of the confusion between these two research communities (which do often have separate conferences and separate journals, ECML PKDD being a major exception) comes from the basic assumptions they work with: in machine learning, performance is usually evaluated with respect to the ability to *reproduce known* knowledge, while in knowledge discovery and data mining (KDD) the key task is the discovery of previously *unknown* knowledge. Evaluated with respect to known knowledge, an uninformed (unsupervised) method will easily be outperformed by other supervised methods, while in a typical KDD task, supervised methods cannot be used due to the unavailability of training data.

Relation to optimization

Machine learning also has intimate ties to optimization: many learning problems are formulated as minimization of some loss function on a training set of examples. Loss functions express the discrepancy between the predictions of the model being trained and the actual problem instances (for example, in classification, one wants to assign a label to instances, and models are trained to correctly predict the pre-assigned labels of a set of examples). The difference between the two fields arises from the goal of generalization: while optimization algorithms can minimize the loss on a training set, machine learning is concerned with minimizing the loss on unseen samples.^[15]

Relation to statistics

Machine learning and statistics are closely related fields in terms of methods, but distinct in their principal goal: statistics draws population inferences from a sample, while machine learning finds generalizable predictive patterns.^[16] According to Michael I. Jordan, the ideas of machine learning, from

methodological principles to theoretical tools, have had a long pre-history in statistics.^[17] He also suggested the term data science as a placeholder to call the overall field.^[17]

Leo Breiman distinguished two statistical modeling paradigms: data model and algorithmic model,^[18] wherein "algorithmic model" means more or less the machine learning algorithms like Random forest.

Some statisticians have adopted methods from machine learning, leading to a combined field that they call *statistical learning*.^[19]

Theory

A core objective of a learner is to generalize from its experience.^{[2][20]} Generalization in this context is the ability of a learning machine to perform accurately on new, unseen examples/tasks after having experienced a learning data set. The training examples come from some generally unknown probability distribution (considered representative of the space of occurrences) and the learner has to build a general model about this space that enables it to produce sufficiently accurate predictions in new cases.

The computational analysis of machine learning algorithms and their performance is a branch of theoretical computer science known as computational learning theory. Because training sets are finite and the future is uncertain, learning theory usually does not yield guarantees of the performance of algorithms. Instead, probabilistic bounds on the performance are quite common. The bias-variance decomposition is one way to quantify generalization error.

For the best performance in the context of generalization, the complexity of the hypothesis should match the complexity of the function underlying the data. If the hypothesis is less complex than the function, then the model has under fitted the data. If the complexity of the model is increased in response, then the training error decreases. But if the hypothesis is too complex, then the model is subject to overfitting and generalization will be poorer.^[21]

In addition to performance bounds, learning theorists study the time complexity and feasibility of learning. In computational learning theory, a computation is considered feasible if it can be done in polynomial time. There are two kinds of time complexity results. Positive results show that a certain class of functions can be learned in polynomial time. Negative results show that certain classes cannot be learned in polynomial time.

Approaches

Types of learning algorithms

The types of machine learning algorithms differ in their approach, the type of data they input and output, and the type of task or problem that they are intended to solve.

Supervised learning

Supervised learning algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs.^[22] The data is known as training data, and consists of a set of training examples. Each training example has one or more inputs and the desired output, also known as a supervisory signal. In the mathematical model, each training example is represented by an array or vector, sometimes called

a feature vector, and the training data is represented by a matrix. Through iterative optimization of an objective function, supervised learning algorithms learn a function that can be used to predict the output associated with new inputs.^[23] An optimal function will allow the algorithm to correctly determine the output for inputs that were not a part of the training data. An algorithm that improves the accuracy of its outputs or predictions over time is said to have learned to perform that task.^[6]

Supervised learning algorithms include classification and regression.^[24] Classification algorithms are used when the outputs are restricted to a limited set of values, and regression algorithms are used when the outputs may have any numerical value within a range. Similarity learning is an area of supervised machine learning closely related to regression and classification, but the goal is to learn from examples using a similarity function that measures how similar or related two objects are. It has applications in ranking, recommendation systems, visual identity tracking, face verification, and speaker verification.

In the case of semi-supervised learning algorithms, some of the training examples are missing training labels, but they can nevertheless be used to improve the quality of a model. In weakly supervised learning, the training labels are noisy, limited, or imprecise; however, these labels are often cheaper to obtain, resulting in larger effective training sets.^[25]

Unsupervised learning

Unsupervised learning algorithms take a set of data that contains only inputs, and find structure in the data, like grouping or clustering of data points. The algorithms, therefore, learn from test data that has not been labeled, classified or categorized. Instead of responding to feedback, unsupervised learning algorithms identify commonalities in the data and react based on the presence or absence of such commonalities in each new piece of data. A central application of unsupervised learning is in the field of density estimation in statistics,^[26] though unsupervised learning encompasses other domains involving summarizing and explaining data features.

Cluster analysis is the assignment of a set of observations into subsets (called *clusters*) so that observations within the same cluster are similar according to one or more predesignated criteria, while observations drawn from different clusters are dissimilar. Different clustering techniques make different assumptions on the structure of the data, often defined by some *similarity metric* and evaluated, for example, by *internal compactness*, or the similarity between members of the same cluster, and *separation*, the difference between clusters. Other methods are based on *estimated density* and *graph connectivity*.

Semi-supervised Learning

Semi-supervised learning falls between unsupervised learning (without any labeled training data) and supervised learning (with completely labeled training data). Many machine-learning researchers have found that unlabeled data, when used in conjunction with a small amount of labeled data, can produce a considerable improvement in learning accuracy.

Reinforcement learning

Reinforcement learning is an area of machine learning concerned with how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward. Due to its generality, the field is studied in many other disciplines, such as game theory, control theory, operations research, information theory, simulation-based optimization, multi-agent systems, swarm intelligence, statistics

and genetic algorithms. In machine learning, the environment is typically represented as a Markov Decision Process (MDP). Many reinforcement learning algorithms use dynamic programming techniques.^[27] Reinforcement learning algorithms do not assume knowledge of an exact mathematical model of the MDP, and are used when exact models are infeasible. Reinforcement learning algorithms are used in autonomous vehicles or in learning to play a game against a human opponent.

Self learning

Self-learning as machine learning paradigm was introduced in 1982 along with a neural network capable of self-learning named Crossbar Adaptive Array (CAA).^[28] It is a learning with no external rewards and no external teacher advices. The CAA self-learning algorithm computes, in a crossbar fashion, both decisions about actions and emotions (feelings) about consequence situations. The system is driven by the interaction between cognition and emotion.^[29] The self-learning algorithm updates a memory matrix $W = ||w(a,s)||$ such that in each iteration executes the following machine learning routine:

```
In situation s perform action a;  
Receive consequence situation s';  
Compute emotion of being in consequence situation v(s');  
Update crossbar memory  $w'(a,s) = w(a,s) + v(s')$ .
```

It is a system with only one input, situation s, and only one output, action (or behavior) a. There is neither a separate reinforcement input nor an advice input from the environment. The backpropagated value (secondary reinforcement) is the emotion toward the consequence situation. The CAA exists in two environments, one is behavioral environment where it behaves, and the other is genetic environment, wherefrom it initially and only once receives initial emotions about situations to be encountered in the behavioral environment. After receiving the genome (species) vector from the genetic environment, the CAA learns a goal seeking behavior, in an environment that contains both desirable and undesirable situations.^[30]

Feature learning

Several learning algorithms aim at discovering better representations of the inputs provided during training.^[31] Classic examples include principal components analysis and cluster analysis. Feature learning algorithms, also called representation learning algorithms, often attempt to preserve the information in their input but also transform it in a way that makes it useful, often as a pre-processing step before performing classification or predictions. This technique allows reconstruction of the inputs coming from the unknown data-generating distribution, while not being necessarily faithful to configurations that are implausible under that distribution. This replaces manual feature engineering, and allows a machine to both learn the features and use them to perform a specific task.

Feature learning can be either supervised or unsupervised. In supervised feature learning, features are learned using labeled input data. Examples include artificial neural networks, multilayer perceptrons, and supervised dictionary learning. In unsupervised feature learning, features are learned with unlabeled input data. Examples include dictionary learning, independent component analysis, autoencoders, matrix factorization^[32] and various forms of clustering.^{[33][34][35]}

Manifold learning algorithms attempt to do so under the constraint that the learned representation is low-dimensional. Sparse coding algorithms attempt to do so under the constraint that the learned representation is sparse, meaning that the mathematical model has many zeros. Multilinear subspace learning algorithms aim to learn low-dimensional representations directly from tensor representations for

multidimensional data, without reshaping them into higher-dimensional vectors.^[36] Deep learning algorithms discover multiple levels of representation, or a hierarchy of features, with higher-level, more abstract features defined in terms of (or generating) lower-level features. It has been argued that an intelligent machine is one that learns a representation that disentangles the underlying factors of variation that explain the observed data.^[37]

Feature learning is motivated by the fact that machine learning tasks such as classification often require input that is mathematically and computationally convenient to process. However, real-world data such as images, video, and sensory data has not yielded to attempts to algorithmically define specific features. An alternative is to discover such features or representations through examination, without relying on explicit algorithms.

Sparse dictionary learning

Sparse dictionary learning is a feature learning method where a training example is represented as a linear combination of basis functions, and is assumed to be a sparse matrix. The method is strongly NP-hard and difficult to solve approximately.^[38] A popular heuristic method for sparse dictionary learning is the K-SVD algorithm. Sparse dictionary learning has been applied in several contexts. In classification, the problem is to determine the class to which a previously unseen training example belongs. For a dictionary where each class has already been built, a new training example is associated with the class that is best sparsely represented by the corresponding dictionary. Sparse dictionary learning has also been applied in image de-noising. The key idea is that a clean image patch can be sparsely represented by an image dictionary, but the noise cannot.^[39]

Anomaly detection

In data mining, anomaly detection, also known as outlier detection, is the identification of rare items, events or observations which raise suspicions by differing significantly from the majority of the data.^[40] Typically, the anomalous items represent an issue such as bank fraud, a structural defect, medical problems or errors in a text. Anomalies are referred to as outliers, novelties, noise, deviations and exceptions.^[41]

In particular, in the context of abuse and network intrusion detection, the interesting objects are often not rare objects, but unexpected bursts in activity. This pattern does not adhere to the common statistical definition of an outlier as a rare object, and many outlier detection methods (in particular, unsupervised algorithms) will fail on such data, unless it has been aggregated appropriately. Instead, a cluster analysis algorithm may be able to detect the micro-clusters formed by these patterns.^[42]

Three broad categories of anomaly detection techniques exist.^[43] Unsupervised anomaly detection techniques detect anomalies in an unlabeled test data set under the assumption that the majority of the instances in the data set are normal, by looking for instances that seem to fit least to the remainder of the data set. Supervised anomaly detection techniques require a data set that has been labeled as "normal" and "abnormal" and involves training a classifier (the key difference to many other statistical classification problems is the inherently unbalanced nature of outlier detection). Semi-supervised anomaly detection techniques construct a model representing normal behavior from a given normal training data set and then test the likelihood of a test instance to be generated by the model.

Association rules

Association rule learning is a rule-based machine learning method for discovering relationships between variables in large databases. It is intended to identify strong rules discovered in databases using some measure of "interestingness".^[44]

Rule-based machine learning is a general term for any machine learning method that identifies, learns, or evolves "rules" to store, manipulate or apply knowledge. The defining characteristic of a rule-based machine learning algorithm is the identification and utilization of a set of relational rules that collectively represent the knowledge captured by the system. This is in contrast to other machine learning algorithms that commonly identify a singular model that can be universally applied to any instance in order to make a prediction.^[45] Rule-based machine learning approaches include learning classifier systems, association rule learning, and artificial immune systems.

Based on the concept of strong rules, Rakesh Agrawal, Tomasz Imieliński and Arun Swami introduced association rules for discovering regularities between products in large-scale transaction data recorded by point-of-sale (POS) systems in supermarkets.^[46] For example, the rule **{onions, potatoes} \Rightarrow {burger}** found in the sales data of a supermarket would indicate that if a customer buys onions and potatoes together, they are likely to also buy hamburger meat. Such information can be used as the basis for decisions about marketing activities such as promotional pricing or product placements. In addition to market basket analysis, association rules are employed today in application areas including Web usage mining, intrusion detection, continuous production, and bioinformatics. In contrast with sequence mining, association rule learning typically does not consider the order of items either within a transaction or across transactions.

Learning classifier systems (LCS) are a family of rule-based machine learning algorithms that combine a discovery component, typically a genetic algorithm, with a learning component, performing either supervised learning, reinforcement learning, or unsupervised learning. They seek to identify a set of context-dependent rules that collectively store and apply knowledge in a piecewise manner in order to make predictions.^[47]

Inductive logic programming (ILP) is an approach to rule-learning using logic programming as a uniform representation for input examples, background knowledge, and hypotheses. Given an encoding of the known background knowledge and a set of examples represented as a logical database of facts, an ILP system will derive a hypothesized logic program that entails all positive and no negative examples. Inductive programming is a related field that considers any kind of programming languages for representing hypotheses (and not only logic programming), such as functional programs.

Inductive logic programming is particularly useful in bioinformatics and natural language processing. Gordon Plotkin and Ehud Shapiro laid the initial theoretical foundation for inductive machine learning in a logical setting.^{[48][49][50]} Shapiro built their first implementation (Model Inference System) in 1981: a Prolog program that inductively inferred logic programs from positive and negative examples.^[51] The term *inductive* here refers to philosophical induction, suggesting a theory to explain observed facts, rather than mathematical induction, proving a property for all members of a well-ordered set.

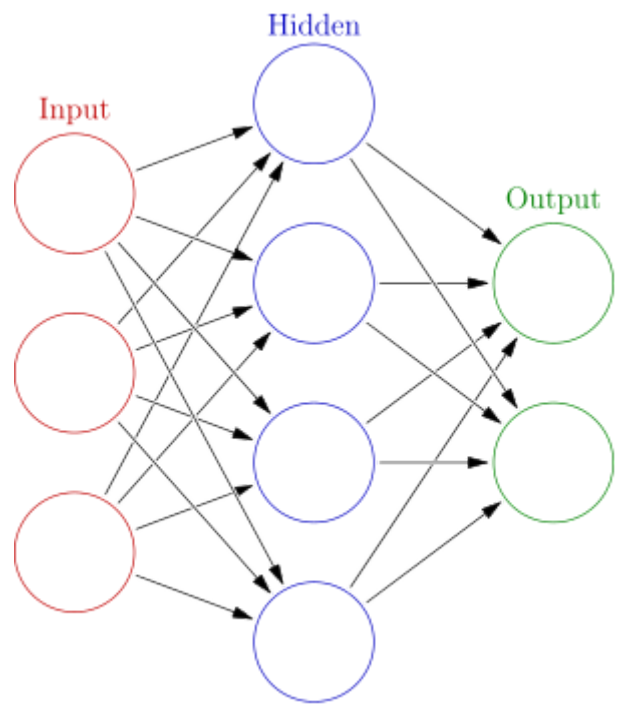
Models

Performing machine learning involves creating a model, which is trained on some training data and then can process additional data to make predictions. Various types of models have been used and researched for machine learning systems.

Artificial neural networks

Artificial neural networks (ANNs), or connectionist systems, are computing systems vaguely inspired by the biological neural networks that constitute animal brains. Such systems "learn" to perform tasks by considering examples, generally without being programmed with any task-specific rules.

An ANN is a model based on a collection of connected units or nodes called "artificial neurons", which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit information, a "signal", from one artificial neuron to another. An artificial neuron that receives a signal can process it and then signal additional artificial neurons connected to it. In common ANN implementations, the signal at a connection between artificial neurons is a real number, and the output of each artificial neuron is computed by some non-linear function of the sum of its inputs. The connections between artificial neurons are called "edges". Artificial neurons and edges typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Artificial neurons may have a threshold such that the signal is only sent if the aggregate signal crosses that threshold. Typically, artificial neurons are aggregated into layers. Different layers may perform different kinds of transformations on their inputs. Signals travel from the first layer (the input layer) to the last layer (the output layer), possibly after traversing the layers multiple times.



An artificial neural network is an interconnected group of nodes, akin to the vast network of neurons in a brain. Here, each circular node represents an artificial neuron and an arrow represents a connection from the output of one artificial neuron to the input of another.

The original goal of the ANN approach was to solve problems in the same way that a human brain would. However, over time, attention moved to performing specific tasks, leading to deviations from biology. Artificial neural networks have been used on a variety of tasks, including computer vision, speech recognition, machine translation, social network filtering, playing board and video games and medical diagnosis.

Deep learning consists of multiple hidden layers in an artificial neural network. This approach tries to model the way the human brain processes light and sound into vision and hearing. Some successful applications of deep learning are computer vision and speech recognition.^[52]

Decision trees

Decision tree learning uses a decision tree as a predictive model to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). It is one of the predictive modeling approaches used in statistics, data mining and machine learning. Tree models where the target variable can take a discrete set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to

those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. In data mining, a decision tree describes data, but the resulting classification tree can be an input for decision making.

Support vector machines

Support vector machines (SVMs), also known as support vector networks, are a set of related supervised learning methods used for classification and regression. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that predicts whether a new example falls into one category or the other.^[53] An SVM training algorithm is a non-probabilistic, binary, linear classifier, although methods such as Platt scaling exist to use SVM in a probabilistic classification setting. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

Regression analysis

Regression analysis encompasses a large variety of statistical methods to estimate the relationship between input variables and their associated features. Its most common form is linear regression, where a single line is drawn to best fit the given data according to a mathematical criterion such as ordinary least squares. The latter is oftentimes extended by regularization (mathematics) methods to mitigate overfitting and high bias, as can be seen in ridge regression. When dealing with non-linear problems, go-to models include polynomial regression (e.g. used for trendline fitting in Microsoft Excel ^[54]), Logistic regression (often used in statistical classification) or even kernel regression, which introduces non-linearity by taking advantage of the kernel trick to implicitly map input variables to higher dimensional space.

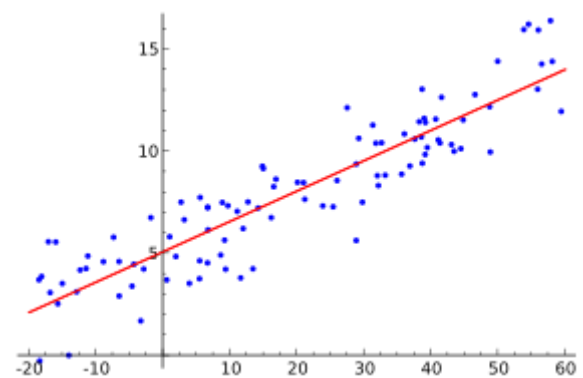
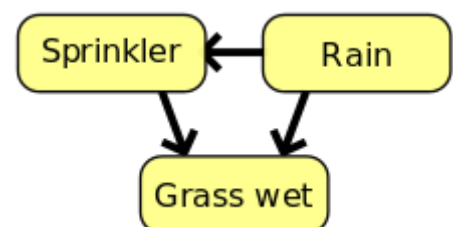


Illustration of linear regression on a data set.

Bayesian networks

A Bayesian network, belief network or directed acyclic graphical model is a probabilistic graphical model that represents a set of random variables and their conditional independence with a directed acyclic graph (DAG). For example, a Bayesian network could represent the probabilistic relationships between diseases and symptoms. Given symptoms, the network can be used to compute the probabilities of the presence of various diseases. Efficient algorithms exist that perform inference and learning. Bayesian networks that model sequences of variables, like speech signals or protein sequences, are called dynamic Bayesian networks. Generalizations of Bayesian networks that can represent and solve decision problems under uncertainty are called influence diagrams.



A simple Bayesian network. Rain influences whether the sprinkler is activated, and both rain and the sprinkler influence whether the grass is wet.

Genetic algorithms

A genetic algorithm (GA) is a search algorithm and heuristic technique that mimics the process of natural selection, using methods such as mutation and crossover to generate new genotypes in the hope of finding good solutions to a given problem. In machine learning, genetic algorithms were used in the 1980s and 1990s.^{[55][56]} Conversely, machine learning techniques have been used to improve the performance of genetic and evolutionary algorithms.^[57]

Training models

Usually, machine learning models require a lot of data in order for them to perform well. Usually, when training a machine learning model, one needs to collect a large, representative sample of data from a training set. Data from the training set can be as varied as a corpus of text, a collection of images, and data collected from individual users of a service. Overfitting is something to watch out for when training a machine learning model.

Federated learning

Federated learning is a new approach to training machine learning models that decentralizes the training process, allowing for users' privacy to be maintained by not needing to send their data to a centralized server. This also increases efficiency by decentralizing the training process to many devices. For example, Gboard uses federated machine learning to train search query prediction models on users' mobile phones without having to send individual searches back to Google.^[58]

Applications

There are many applications for machine learning, including:

- Agriculture
- Anatomy
- Adaptive websites
- Affective computing
- Banking
- Bioinformatics
- Brain–machine interfaces
- Cheminformatics
- Citizen science
- Computer networks
- Computer vision
- Credit-card fraud detection
- Data quality
- DNA sequence classification
- Economics
- Financial market analysis ^[59]
- General game playing
- Handwriting recognition
- Information retrieval
- Insurance
- Internet fraud detection
- Linguistics
- Machine learning control
- Machine perception
- Machine translation
- Marketing
- Medical diagnosis
- Natural language processing
- Natural language understanding
- Online advertising
- Optimization
- Recommender systems
- Robot locomotion
- Search engines
- Sentiment analysis
- Sequence mining
- Software engineering
- Speech recognition
- Structural health monitoring
- Syntactic pattern recognition

- Telecommunication
- Theorem proving
- Time series forecasting
- User behavior analytics

In 2006, the media-services provider Netflix held the first "Netflix Prize" competition to find a program to better predict user preferences and improve the accuracy on its existing Cinematch movie recommendation algorithm by at least 10%. A joint team made up of researchers from AT&T Labs-Research in collaboration with the teams Big Chaos and Pragmatic Theory built an ensemble model to win the Grand Prize in 2009 for \$1 million.^[60] Shortly after the prize was awarded, Netflix realized that viewers' ratings were not the best indicators of their viewing patterns ("everything is a recommendation") and they changed their recommendation engine accordingly.^[61] In 2010 The Wall Street Journal wrote about the firm Rebellion Research and their use of machine learning to predict the financial crisis.^[62] In 2012, co-founder of Sun Microsystems, Vinod Khosla, predicted that 80% of medical doctors' jobs would be lost in the next two decades to automated machine learning medical diagnostic software.^[63] In 2014, it was reported that a machine learning algorithm had been applied in the field of art history to study fine art paintings, and that it may have revealed previously unrecognized influences among artists.^[64] In 2019 Springer Nature published the first research book created using machine learning.^[65]

Limitations

Although machine learning has been transformative in some fields, machine-learning programs often fail to deliver expected results.^{[66][67][68]} Reasons for this are numerous: lack of (suitable) data, lack of access to the data, data bias, privacy problems, badly chosen tasks and algorithms, wrong tools and people, lack of resources, and evaluation problems.^[69]

In 2018, a self-driving car from Uber failed to detect a pedestrian, who was killed after a collision.^[70] Attempts to use machine learning in healthcare with the IBM Watson system failed to deliver even after years of time and billions of investment.^{[71][72]}

Bias

Machine learning approaches in particular can suffer from different data biases. A machine learning system trained on current customers only may not be able to predict the needs of new customer groups that are not represented in the training data. When trained on man-made data, machine learning is likely to pick up the same constitutional and unconscious biases already present in society.^[73] Language models learned from data have been shown to contain human-like biases.^{[74][75]} Machine learning systems used for criminal risk assessment have been found to be biased against black people.^{[76][77]} In 2015, Google photos would often tag black people as gorillas,^[78] and in 2018 this still was not well resolved, but Google reportedly was still using the workaround to remove all gorillas from the training data, and thus was not able to recognize real gorillas at all.^[79] Similar issues with recognizing non-white people have been found in many other systems.^[80] In 2016, Microsoft tested a chatbot that learned from Twitter, and it quickly picked up racist and sexist language.^[81] Because of such challenges, the effective use of machine learning may take longer to be adopted in other domains.^[82] Concern for fairness in machine learning, that is, reducing bias in machine learning and propelling its use for human good is increasingly expressed by artificial intelligence scientists, including Fei-Fei Li, who reminds engineers that "There's nothing artificial about AI...It's inspired by people, it's created by people, and—most importantly—it impacts people. It is a powerful tool we are only just beginning to understand, and that is a profound responsibility."^[83]

Model assessments

Classification machine learning models can be validated by accuracy estimation techniques like the Holdout method, which splits the data in a training and test set (conventionally 2/3 training set and 1/3 test set designation) and evaluates the performance of the training model on the test set. In comparison, the K-fold-cross-validation method randomly partitions the data into K subsets and then K experiments are performed each respectively considering 1 subset for evaluation and the remaining K-1 subsets for training the model. In addition to the holdout and cross-validation methods, bootstrap, which samples n instances with replacement from the dataset, can be used to assess model accuracy.^[84]

In addition to overall accuracy, investigators frequently report sensitivity and specificity meaning True Positive Rate (TPR) and True Negative Rate (TNR) respectively. Similarly, investigators sometimes report the False Positive Rate (FPR) as well as the False Negative Rate (FNR). However, these rates are ratios that fail to reveal their numerators and denominators. The Total Operating Characteristic (TOC) is an effective method to express a model's diagnostic ability. TOC shows the numerators and denominators of the previously mentioned rates, thus TOC provides more information than the commonly used Receiver Operating Characteristic (ROC) and ROC's associated Area Under the Curve (AUC).^[85]

Ethics

Machine learning poses a host of ethical questions. Systems which are trained on datasets collected with biases may exhibit these biases upon use (algorithmic bias), thus digitizing cultural prejudices.^[86] For example, using job hiring data from a firm with racist hiring policies may lead to a machine learning system duplicating the bias by scoring job applicants against similarity to previous successful applicants.^{[87][88]} Responsible collection of data and documentation of algorithmic rules used by a system thus is a critical part of machine learning.

Because human languages contain biases, machines trained on language corpora will necessarily also learn these biases.^{[89][90]}

Other forms of ethical challenges, not related to personal biases, are more seen in health care. There are concerns among health care professionals that these systems might not be designed in the public's interest but as income-generating machines. This is especially true in the United States where there is a perpetual ethical dilemma of improving health care, but also increase profits. For example, the algorithms could be designed to provide patients with unnecessary tests or medication in which the algorithm's proprietary owners hold stakes in. There is huge potential for machine learning in health care to provide professionals a great tool to diagnose, medicate, and even plan recovery paths for patients, but this will not happen until the personal biases mentioned previously, and these "greed" biases are addressed.^[91]

Software

Software suites containing a variety of machine learning algorithms include the following:

Free and open-source software

- CNTK
- Deeplearning4j
- ELKI
- Keras
- Caffe
- ML.NET

- [Mahout](#)
- [Mallet](#)
- [mlpack](#)
- [MXNet](#)
- [Neural Lab](#)
- [GNU Octave](#)
- [OpenNN](#)
- [Orange](#)
- [scikit-learn](#)
- [Shogun](#)
- [Spark MLlib](#)
- [Apache SystemML](#)
- [TensorFlow](#)
- [ROOT](#) (TMVA with ROOT)
- [Torch / PyTorch](#)
- [Weka / MOA](#)
- [Yooreeka](#)
- [R](#)

Proprietary software with free and open-source editions

- [KNIME](#)
- [RapidMiner](#)

Proprietary software

- [Amazon Machine Learning](#)
- [Angoss KnowledgeSTUDIO](#)
- [Azure Machine Learning](#)
- [Ayasdi](#)
- [IBM Data Science Experience](#)
- [Google Prediction API](#)
- [IBM SPSS Modeler](#)
- [KXEN Modeler](#)
- [LIONsolver](#)
- [Mathematica](#)
- [MATLAB](#)
- [Microsoft Azure](#)
- [Neural Designer](#)
- [NeuroSolutions](#)
- [Oracle Data Mining](#)
- [Oracle AI Platform Cloud Service](#)
- [RCASE](#)
- [SAS Enterprise Miner](#)
- [SequenceL](#)
- [Splunk](#)
- [STATISTICA](#) Data Miner

Journals

- [Journal of Machine Learning Research](#)
- [Machine Learning](#)
- [Nature Machine Intelligence](#)
- [Neural Computation](#)

Conferences

- [Conference on Neural Information Processing Systems](#)
- [International Conference on Machine Learning](#)

See also

- [Automated machine learning](#)
- [Big data](#)
- [Explanation-based learning](#)
- [Important publications in machine learning](#)

- List of datasets for machine learning research
- Predictive analytics
- Quantum machine learning
- Machine-learning applications in bioinformatics
- Seq2seq
- Fairness (machine learning)

References

1. The definition "without being explicitly programmed" is often attributed to Arthur Samuel, who coined the term "machine learning" in 1959, but the phrase is not found verbatim in this publication, and may be a paraphrase that appeared later. Confer "Paraphrasing Arthur Samuel (1959), the question is: How can computers learn to solve problems without being explicitly programmed?" in Koza, John R.; Bennett, Forrest H.; Andre, David; Keane, Martin A. (1996). *Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming*. Artificial Intelligence in Design '96. Springer, Dordrecht. pp. 151–170. doi:10.1007/978-94-009-0279-4_9 (https://doi.org/10.1007%2F978-94-009-0279-4_9).
2. Bishop, C. M. (2006), *Pattern Recognition and Machine Learning*, Springer, ISBN 978-0-387-31073-2
3. Machine learning and pattern recognition "can be viewed as two facets of the same field." [2]:vii
4. Friedman, Jerome H. (1998). "Data Mining and Statistics: What's the connection?". *Computing Science and Statistics*. **29** (1): 3–9.
5. Samuel, Arthur (1959). "Some Studies in Machine Learning Using the Game of Checkers". *IBM Journal of Research and Development*. **3** (3): 210–229. CiteSeerX 10.1.1.368.2254 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.368.2254>). doi:10.1147/rd.33.0210 (<https://doi.org/10.1147%2Frd.33.0210>).
6. Mitchell, T. (1997). *Machine Learning*. McGraw Hill. p. 2. ISBN 978-0-07-042807-2.
7. Harnad, Stevan (2008), "The Annotation Game: On Turing (1950) on Computing, Machinery, and Intelligence" (<http://eprints.ecs.soton.ac.uk/12954/>), in Epstein, Robert; Peters, Grace (eds.), *The Turing Test Sourcebook: Philosophical and Methodological Issues in the Quest for the Thinking Computer*, Kluwer, pp. 23–66, ISBN 9781402067082
8. R. Kohavi and F. Provost, "Glossary of terms," *Machine Learning*, vol. 30, no. 2–3, pp. 271–274, 1998.
9. Nilsson N. *Learning Machines*, McGraw Hill, 1965.
10. Duda, R., Hart P. *Pattern Recognition and Scene Analysis*, Wiley Interscience, 1973
11. S. Bozinovski "Teaching space: A representation concept for adaptive pattern classification" COINS Technical Report No. 81-28, Computer and Information Science Department, University of Massachusetts at Amherst, MA, 1981. <https://web.cs.umass.edu/publication/docs/1981/UM-CS-1981-028.pdf>
12. Sarle, Warren (1994). "Neural Networks and statistical models". CiteSeerX 10.1.1.27.699 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.27.699>).
13. Russell, Stuart; Norvig, Peter (2003) [1995]. *Artificial Intelligence: A Modern Approach* (2nd ed.). Prentice Hall. ISBN 978-0137903955.
14. Langley, Pat (2011). "The changing science of machine learning". *Machine Learning*. **82** (3): 275–279. doi:10.1007/s10994-011-5242-y (<https://doi.org/10.1007%2Fs10994-011-5242-y>).

15. Le Roux, Nicolas; Bengio, Yoshua; Fitzgibbon, Andrew (2012). "Improving+First+and+Second-Order+Methods+by+Modeling+Uncertainty "Improving First and Second-Order Methods by Modeling Uncertainty" (<https://books.google.com/?id=JPQx7s2L1A8C&pg=PA403&dq=>). In Sra, Suvrit; Nowozin, Sebastian; Wright, Stephen J. (eds.). *Optimization for Machine Learning*. MIT Press. p. 404. ISBN 9780262016469.
16. Bzdok, Danilo; Altman, Naomi; Krzywinski, Martin (2018). "Statistics versus Machine Learning" (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6082636>). *Nature Methods*. 15 (4): 233–234. doi:10.1038/nmeth.4642 (<https://doi.org/10.1038%2Fnmeth.4642>). PMC 6082636 (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6082636>). PMID 30100822 (<https://pubmed.ncbi.nlm.nih.gov/30100822>).
17. Michael I. Jordan (2014-09-10). "statistics and machine learning" (https://www.reddit.com/r/MachineLearning/comments/2fxi6v/ama_michael_i_jordan/ckelmtt?context=3). reddit. Retrieved 2014-10-01.
18. Cornell University Library. "Breiman: Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author)" (http://projecteuclid.org/download/pdf_1/euclid.ss/1009213726). Retrieved 8 August 2015.
19. Gareth James; Daniela Witten; Trevor Hastie; Robert Tibshirani (2013). *An Introduction to Statistical Learning* (<http://www-bcf.usc.edu/~gareth/ISL/>). Springer. p. vii.
20. Mohri, Mehryar; Rostamizadeh, Afshin; Talwalkar, Ameet (2012). *Foundations of Machine Learning*. USA, Massachusetts: MIT Press. ISBN 9780262018258.
21. Alpaydin, Ethem (2010). *Introduction to Machine Learning* (https://archive.org/details/introductiontoma00alpa_0). London: The MIT Press. ISBN 978-0-262-01243-0. Retrieved 4 February 2017.
22. Russell, Stuart J.; Norvig, Peter (2010). *Artificial Intelligence: A Modern Approach* (Third ed.). Prentice Hall. ISBN 9780136042594.
23. Mohri, Mehryar; Rostamizadeh, Afshin; Talwalkar, Ameet (2012). *Foundations of Machine Learning*. The MIT Press. ISBN 9780262018258.
24. Alpaydin, Ethem (2010). *Introduction to Machine Learning* (<https://books.google.com/books?id=7f5bBAAAQBAJ&printsec=frontcover#v=onepage&q=classification&f=false>). MIT Press. p. 9. ISBN 978-0-262-01243-0.
25. Alex Ratner; Stephen Bach; Paroma Varma; Chris. "Weak Supervision: The New Programming Paradigm for Machine Learning" (https://hazyresearch.github.io/snorkel/blog/ws_blog_post.html). *hazyresearch.github.io*. referencing work by many other members of Hazy Research. Retrieved 2019-06-06.
26. Jordan, Michael I.; Bishop, Christopher M. (2004). "Neural Networks". In Allen B. Tucker (ed.). *Computer Science Handbook, Second Edition (Section VII: Intelligent Systems)*. Boca Raton, Florida: Chapman & Hall/CRC Press LLC. ISBN 978-1-58488-360-9.
27. van Otterlo, M.; Wiering, M. (2012). *Reinforcement learning and markov decision processes. Reinforcement Learning. Adaptation, Learning, and Optimization*. 12. pp. 3–42. doi:10.1007/978-3-642-27645-3_1 (https://doi.org/10.1007%2F978-3-642-27645-3_1). ISBN 978-3-642-27644-6.
28. Bozinovski, S. (1982). "A self-learning system using secondary reinforcement" . In Trappl, Robert (ed.). *Cybernetics and Systems Research: Proceedings of the Sixth European Meeting on Cybernetics and Systems Research*. North Holland. pp. 397–402. ISBN 978-0-444-86488-8.
29. Bozinovski, Stevo (2014) "Modeling mechanisms of cognition-emotion interaction in artificial neural networks, since 1981." *Procedia Computer Science* p. 255-263
30. Bozinovski, S. (2001) "Self-learning agents: A connectionist theory of emotion based on crossbar value judgment." *Cybernetics and Systems* 32(6) 637-667.

31. Y. Bengio; A. Courville; P. Vincent (2013). "Representation Learning: A Review and New Perspectives". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **35** (8): 1798–1828. [arXiv:1206.5538](https://arxiv.org/abs/1206.5538) (<https://arxiv.org/abs/1206.5538>). doi:10.1109/tpami.2013.50 (<https://doi.org/10.1109%2Ftpami.2013.50>). PMID 23787338 (<https://pubmed.ncbi.nlm.nih.gov/23787338>).
32. Nathan Srebro; Jason D. M. Rennie; Tommi S. Jaakkola (2004). *Maximum-Margin Matrix Factorization*. *NIPS*.
33. Coates, Adam; Lee, Honglak; Ng, Andrew Y. (2011). *An analysis of single-layer networks in unsupervised feature learning* (http://machinelearning.wustl.edu/mlpapers/paper_files/AISTA-TS2011_CoatesNL11.pdf) (PDF). Int'l Conf. on AI and Statistics (AISTATS).
34. Csurka, Gabriella; Dance, Christopher C.; Fan, Lixin; Willamowski, Jutta; Bray, Cédric (2004). *Visual categorization with bags of keypoints* (<https://www.cs.cmu.edu/~efros/course/s/LBMV07/Papers/csurka-eccv-04.pdf>) (PDF). ECCV Workshop on Statistical Learning in Computer Vision.
35. Daniel Jurafsky; James H. Martin (2009). *Speech and Language Processing*. Pearson Education International. pp. 145–146.
36. Lu, Haiping; Plataniotis, K.N.; Venetsanopoulos, A.N. (2011). "A Survey of Multilinear Subspace Learning for Tensor Data" (http://www.dsp.utoronto.ca/~haiping/Publication/SurveyMSL_PR2011.pdf) (PDF). *Pattern Recognition*. **44** (7): 1540–1551. doi:10.1016/j.patcog.2011.01.004 (<https://doi.org/10.1016%2Fj.patcog.2011.01.004>).
37. Yoshua Bengio (2009). *Learning Deep Architectures for AI* (<https://books.google.com/books?id=cq5ewg7FniMC&pg=PA3>). Now Publishers Inc. pp. 1–3. ISBN 978-1-60198-294-0.
38. Tillmann, A. M. (2015). "On the Computational Intractability of Exact and Approximate Dictionary Learning". *IEEE Signal Processing Letters*. **22** (1): 45–49. [arXiv:1405.6664](https://arxiv.org/abs/1405.6664) (<https://arxiv.org/abs/1405.6664>). Bibcode:2015ISPL...22...45T (<https://ui.adsabs.harvard.edu/abs/2015ISPL...22...45T>). doi:10.1109/LSP.2014.2345761 (<https://doi.org/10.1109%2FLSP.2014.2345761>).
39. Aharon, M, M Elad, and A Bruckstein. 2006. "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation (<http://sites.fas.harvard.edu/~cs278/papers/ksvd.pdf>)." *Signal Processing, IEEE Transactions on* 54 (11): 4311–4322
40. Zimek, Arthur; Schubert, Erich (2017), "Outlier Detection", *Encyclopedia of Database Systems*, Springer New York, pp. 1–5, doi:10.1007/978-1-4899-7993-3_80719-1 (https://doi.org/10.1007%2F978-1-4899-7993-3_80719-1), ISBN 9781489979933
41. Hodge, V. J.; Austin, J. (2004). "A Survey of Outlier Detection Methodologies" (<http://eprints.whiterose.ac.uk/767/1/hodgevj4.pdf>) (PDF). *Artificial Intelligence Review*. **22** (2): 85–126. CiteSeerX 10.1.1.318.4023 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.318.4023>). doi:10.1007/s10462-004-4304-y (<https://doi.org/10.1007%2Fs10462-004-4304-y>).
42. Dokas, Paul; Ertöz, Levent; Kumar, Vipin; Lazarevic, Aleksandar; Srivastava, Jaideep; Tan, Pang-Ning (2002). "Data mining for network intrusion detection" (<http://www.csee.umbc.edu/~kolari1/Mining/ngdm/dokas.pdf>) (PDF). *Proceedings NSF Workshop on Next Generation Data Mining*.
43. Chandola, V.; Banerjee, A.; Kumar, V. (2009). "Anomaly detection: A survey". *ACM Computing Surveys*. **41** (3): 1–58. doi:10.1145/1541880.1541882 (<https://doi.org/10.1145%2F1541880.1541882>).
44. Piatetsky-Shapiro, Gregory (1991), *Discovery, analysis, and presentation of strong rules*, in Piatetsky-Shapiro, Gregory; and Frawley, William J.; eds., *Knowledge Discovery in Databases*, AAAI/MIT Press, Cambridge, MA.

45. Bassel, George W.; Glaab, Enrico; Marquez, Julietta; Holdsworth, Michael J.; Bacardit, Jaume (2011-09-01). "Functional Network Construction in Arabidopsis Using Rule-Based Machine Learning on Large-Scale Data Sets" (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3203449>). *The Plant Cell*. **23** (9): 3101–3116. doi:10.1105/tpc.111.088153 (<https://doi.org/10.1105%2Ftpc.111.088153>). ISSN 1532-298X (<https://www.worldcat.org/issn/1532-298X>). PMC 3203449 (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3203449>). PMID 21896882 (<https://pubmed.ncbi.nlm.nih.gov/21896882>).
46. Agrawal, R.; Imieliński, T.; Swami, A. (1993). "Mining association rules between sets of items in large databases". *Proceedings of the 1993 ACM SIGMOD international conference on Management of data - SIGMOD '93*. p. 207. CiteSeerX 10.1.1.40.6984 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.40.6984>). doi:10.1145/170035.170072 (<https://doi.org/10.1145%2F170035.170072>). ISBN 978-0897915922.
47. Urbanowicz, Ryan J.; Moore, Jason H. (2009-09-22). "Learning Classifier Systems: A Complete Introduction, Review, and Roadmap". *Journal of Artificial Evolution and Applications*. **2009**: 1–25. doi:10.1155/2009/736398 (<https://doi.org/10.1155%2F2009%2F736398>). ISSN 1687-6229 (<https://www.worldcat.org/issn/1687-6229>).
48. Plotkin G.D. Automatic Methods of Inductive Inference (<https://www.era.lib.ed.ac.uk/bitstream/handle/1842/6656/Plotkin1972.pdf;sequence=1>), PhD thesis, University of Edinburgh, 1970.
49. Shapiro, Ehud Y. Inductive inference of theories from facts (<http://ftp.cs.yale.edu/publication/s/techreports/tr192.pdf>), Research Report 192, Yale University, Department of Computer Science, 1981. Reprinted in J.-L. Lassez, G. Plotkin (Eds.), *Computational Logic*, The MIT Press, Cambridge, MA, 1991, pp. 199–254.
50. Shapiro, Ehud Y. (1983). *Algorithmic program debugging*. Cambridge, Mass: MIT Press. ISBN 0-262-19218-7
51. Shapiro, Ehud Y. "The model inference system (<http://dl.acm.org/citation.cfm?id=1623364>)."
Proceedings of the 7th international joint conference on Artificial intelligence-Volume 2. Morgan Kaufmann Publishers Inc., 1981.
52. Honglak Lee, Roger Grosse, Rajesh Ranganath, Andrew Y. Ng. "Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations (<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.149.802&rep=rep1&type=pdf>)" Proceedings of the 26th Annual International Conference on Machine Learning, 2009.
53. Cortes, Corinna; Vapnik, Vladimir N. (1995). "Support-vector networks". *Machine Learning*. **20** (3): 273–297. doi:10.1007/BF00994018 (<https://doi.org/10.1007%2FBF00994018>).
54. Stevenson, Christopher. "Tutorial: Polynomial Regression in Excel" (<https://facultystaff.richmond.edu/~cstevens/301/Excel4.html>). *facultystaff.richmond.edu*. Retrieved 22 January 2017.
55. Goldberg, David E.; Holland, John H. (1988). "Genetic algorithms and machine learning" (https://deepblue.lib.umich.edu/bitstream/2027.42/46947/1/10994_2005_Article_422926.pdf) (PDF). *Machine Learning*. **3** (2): 95–99. doi:10.1007/bf00113892 (<https://doi.org/10.1007%2Fbf00113892>).
56. Michie, D.; Spiegelhalter, D. J.; Taylor, C. C. (1994). "Machine Learning, Neural and Statistical Classification". *Ellis Horwood Series in Artificial Intelligence*. Bibcode:1994mlns.book.....M (<https://ui.adsabs.harvard.edu/abs/1994mlns.book.....M>).
57. Zhang, Jun; Zhan, Zhi-hui; Lin, Ying; Chen, Ni; Gong, Yue-jiao; Zhong, Jing-hui; Chung, Henry S.H.; Li, Yun; Shi, Yu-hui (2011). "Evolutionary Computation Meets Machine Learning: A Survey". *Computational Intelligence Magazine*. **6** (4): 68–75. doi:10.1109/mci.2011.942584 (<https://doi.org/10.1109%2Fmci.2011.942584>).
58. "Federated Learning: Collaborative Machine Learning without Centralized Training Data" (<http://ai.googleblog.com/2017/04/federated-learning-collaborative.html>). *Google AI Blog*. Retrieved 2019-06-08.

59. Machine learning is included in the CFA Curriculum (discussion is top down); see: Kathleen DeRose and Christophe Le Lanno (2020). "Machine Learning" (<https://www.cfainstitute.org/-/media/documents/study-session/2020-l2-ss3.ashx>).
60. "BelKor Home Page" (<https://web.archive.org/web/20151110062742/http://www2.research.att.com/~volinsky/netflix/>) research.att.com
61. "The Netflix Tech Blog: Netflix Recommendations: Beyond the 5 stars (Part 1)" (<http://techblog.netflix.com/2012/04/netflix-recommendations-beyond-5-stars.html>). 2012-04-06. Retrieved 8 August 2015.
62. Scott Patterson (13 July 2010). "Letting the Machines Decide" (<https://www.wsj.com/articles/SB10001424052748703834604575365310813948080>). *The Wall Street Journal*. Retrieved 24 June 2018.
63. Vinod Khosla (January 10, 2012). "Do We Need Doctors or Algorithms?" (<https://techcrunch.com/2012/01/10/doctors-or-algorithms/>). Tech Crunch.
64. When A Machine Learning Algorithm Studied Fine Art Paintings, It Saw Things Art Historians Had Never Noticed (<https://medium.com/the-physics-arxiv-blog/when-a-machine-learning-algorithm-studied-fine-art-paintings-it-saw-things-art-historians-had-never-b8e4e7bf7d3e>), *The Physics at ArXiv blog*
65. Vincent, James (2019-04-10). "The first AI-generated textbook shows what robot writers are actually good at" (<https://www.theverge.com/2019/4/10/18304558/ai-writing-academic-research-book-springer-nature-artificial-intelligence>). *The Verge*. Retrieved 2019-05-05.
66. "Why Machine Learning Models Often Fail to Learn: QuickTake Q&A" (<https://web.archive.org/web/20170320225010/https://www.bloomberg.com/news/articles/2016-11-10/why-machine-learning-models-often-fail-to-learn-quicktake-q-a>). *Bloomberg.com*. 2016-11-10. Retrieved 2017-04-10.
67. "The First Wave of Corporate AI Is Doomed to Fail" (<https://hbr.org/2017/04/the-first-wave-of-corporate-ai-is-doomed-to-fail>). *Harvard Business Review*. 2017-04-18. Retrieved 2018-08-20.
68. "Why the A.I. euphoria is doomed to fail" (<https://venturebeat.com/2016/09/17/why-the-a-i-euphoria-is-doomed-to-fail/>). *VentureBeat*. 2016-09-18. Retrieved 2018-08-20.
69. "9 Reasons why your machine learning project will fail" (<https://www.kdnuggets.com/2018/07/why-machine-learning-project-fail.html>). *www.kdnuggets.com*. Retrieved 2018-08-20.
70. "Why Uber's self-driving car killed a pedestrian" (<https://www.economist.com/the-economist-explains/2018/05/29/why-ubers-self-driving-car-killed-a-pedestrian>). *The Economist*. Retrieved 2018-08-20.
71. "IBM's Watson recommended 'unsafe and incorrect' cancer treatments - STAT" (<https://www.statnews.com/2018/07/25/ibm-watson-recommended-unsafe-incorrect-treatments/>). *STAT*. 2018-07-25. Retrieved 2018-08-21.
72. Hernandez, Daniela; Greenwald, Ted (2018-08-11). "IBM Has a Watson Dilemma" (<https://www.wsj.com/articles/ibm-bet-billions-that-watson-could-improve-cancer-treatment-it-hasnt-worked-1533961147>). *Wall Street Journal*. ISSN 0099-9660 (<https://www.worldcat.org/issn/0099-9660>). Retrieved 2018-08-21.
73. Garcia, Megan (2016). "Racist in the Machine". *World Policy Journal*. **33** (4): 111–117. doi:10.1215/07402775-3813015 (<https://doi.org/10.1215/07402775-3813015>). ISSN 0740-2775 (<https://www.worldcat.org/issn/0740-2775>).
74. Caliskan, Aylin; Bryson, Joanna J.; Narayanan, Arvind (2017-04-14). "Semantics derived automatically from language corpora contain human-like biases". *Science*. **356** (6334): 183–186. arXiv:1608.07187 (<https://arxiv.org/abs/1608.07187>). Bibcode:2017Sci...356..183C (<https://ui.adsabs.harvard.edu/abs/2017Sci...356..183C>). doi:10.1126/science.aal4230 (<https://doi.org/10.1126/science.aal4230>). ISSN 0036-8075 (<https://www.worldcat.org/issn/0036-8075>). PMID 28408601 (<https://pubmed.ncbi.nlm.nih.gov/28408601>).

75. Wang, Xinan; Dasgupta, Sanjoy (2016), Lee, D. D.; Sugiyama, M.; Luxburg, U. V.; Guyon, I. (eds.), "An algorithm for L1 nearest neighbor search via monotonic embedding" (<http://papers.nips.cc/paper/6227-an-algorithm-for-l1-nearest-neighbor-search-via-monotonic-embedding.pdf>) (PDF), *Advances in Neural Information Processing Systems 29*, Curran Associates, Inc., pp. 983–991, retrieved 2018-08-20
76. Julia Angwin; Jeff Larson; Lauren Kirchner; Surya Mattu (2016-05-23). "Machine Bias" (<http://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing>). *ProPublica*. Retrieved 2018-08-20.
77. "Opinion | When an Algorithm Helps Send You to Prison" (<https://www.nytimes.com/2017/10/26/opinion/algorithm-compas-sentencing-bias.html>). *New York Times*. Retrieved 2018-08-20.
78. "Google apologises for racist blunder" (<https://www.bbc.co.uk/news/technology-33347866>). *BBC News*. 2015-07-01. Retrieved 2018-08-20.
79. "Google 'fixed' its racist algorithm by removing gorillas from its image-labeling tech" (<https://www.theverge.com/2018/1/12/16882408/google-racist-gorillas-photo-recognition-algorithm-ai>). *The Verge*. Retrieved 2018-08-20.
80. "Opinion | Artificial Intelligence's White Guy Problem" (<https://www.nytimes.com/2016/06/26/opinion/sunday/artificial-intelligences-white-guy-problem.html>). *New York Times*. Retrieved 2018-08-20.
81. Metz, Rachel. "Why Microsoft's teen chatbot, Tay, said lots of awful things online" (<https://www.technologyreview.com/s/601111/why-microsoft-accidentally-unleashed-a-neo-nazi-sexbot/>). *MIT Technology Review*. Retrieved 2018-08-20.
82. Simonite, Tom. "Microsoft says its racist chatbot illustrates how AI isn't adaptable enough to help most businesses" (<https://www.technologyreview.com/s/603944/microsoft-ai-isnt-yet-adaptable-enough-to-help-businesses/>). *MIT Technology Review*. Retrieved 2018-08-20.
83. Hempel, Jessi (2018-11-13). "Fei-Fei Li's Quest to Make Machines Better for Humanity" (<https://www.wired.com/story/fei-fei-li-artificial-intelligence-humanity/>). *Wired*. ISSN 1059-1028 (<https://www.worldcat.org/issn/1059-1028>). Retrieved 2019-02-17.
84. Kohavi, Ron (1995). "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection" (<http://web.cs.iastate.edu/~jtian/cs573/Papers/Kohavi-IJCAI-95.pdf>) (PDF). *International Joint Conference on Artificial Intelligence*.
85. Pontius, Robert Gilmore; Si, Kangping (2014). "The total operating characteristic to measure diagnostic ability for multiple thresholds". *International Journal of Geographical Information Science*. **28** (3): 570–583. doi:10.1080/13658816.2013.862623 (<https://doi.org/10.1080/13658816.2013.862623>).
86. Bostrom, Nick (2011). "The Ethics of Artificial Intelligence" (<http://www.nickbostrom.com/ethics/artificial-intelligence.pdf>) (PDF). Retrieved 11 April 2016.
87. Edionwe, Tolulope. "The fight against racist algorithms" (<https://theoutline.com/post/1571/the-fight-against-racist-algorithms>). *The Outline*. Retrieved 17 November 2017.
88. Jeffries, Adrienne. "Machine learning is racist because the internet is racist" (<https://theoutline.com/post/1439/machine-learning-is-racist-because-the-internet-is-racist>). *The Outline*. Retrieved 17 November 2017.
89. Prates, Marcelo O. R. (11 Mar 2019). "Assessing Gender Bias in Machine Translation -- A Case Study with Google Translate". arXiv:1809.02208 (<https://arxiv.org/abs/1809.02208>) [cs.CY (<https://arxiv.org/archive/cs.CY>)].
90. Narayanan, Arvind (August 24, 2016). "Language necessarily contains human biases, and so will machines trained on language corpora" (<https://freedom-to-tinker.com/2016/08/24/language-necessarily-contains-human-biases-and-so-will-machines-trained-on-language-corpora/>). *Freedom to Tinker*.

91. Char, D. S.; Shah, N. H.; Magnus, D. (2018). "Implementing Machine Learning in Health Care—Addressing Ethical Challenges" (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5962261>). *New England Journal of Medicine*. **378** (11): 981–983. doi:10.1056/nejmp1714229 (<https://doi.org/10.1056%2Fnejmp1714229>). PMC 5962261 (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5962261>). PMID 29539284 (<https://pubmed.ncbi.nlm.nih.gov/29539284>).

Further reading

- Nils J. Nilsson, *Introduction to Machine Learning* (<https://ai.stanford.edu/people/nilsson/mlbook.html>).
- Trevor Hastie, Robert Tibshirani and Jerome H. Friedman (2001). *The Elements of Statistical Learning* (<https://web.stanford.edu/~hastie/ElemStatLearn/>), Springer. ISBN 0-387-95284-5.
- Pedro Domingos (September 2015), *The Master Algorithm*, Basic Books, ISBN 978-0-465-06570-7
- Ian H. Witten and Eibe Frank (2011). *Data Mining: Practical machine learning tools and techniques* Morgan Kaufmann, 664pp., ISBN 978-0-12-374856-0.
- Ethem Alpaydin (2004). *Introduction to Machine Learning*, MIT Press, ISBN 978-0-262-01243-0.
- David J. C. MacKay. *Information Theory, Inference, and Learning Algorithms* (<http://www.inference.phy.cam.ac.uk/mackay/itila/book.html>) Cambridge: Cambridge University Press, 2003. ISBN 0-521-64298-1
- Richard O. Duda, Peter E. Hart, David G. Stork (2001) *Pattern classification* (2nd edition), Wiley, New York, ISBN 0-471-05669-3.
- Christopher Bishop (1995). *Neural Networks for Pattern Recognition*, Oxford University Press. ISBN 0-19-853864-2.
- Stuart Russell & Peter Norvig, (2009). *Artificial Intelligence – A Modern Approach* (<http://aima.cs.berkeley.edu/>). Pearson, ISBN 9789332543515.
- Ray Solomonoff, *An Inductive Inference Machine*, IRE Convention Record, Section on Information Theory, Part 2, pp., 56–62, 1957.
- Ray Solomonoff, *An Inductive Inference Machine* (<http://world.std.com/~rjs/indinf56.pdf>) A privately circulated report from the 1956 Dartmouth Summer Research Conference on AI.

External links

- International Machine Learning Society (<https://web.archive.org/web/20171230081341/http://machinelearning.org:80/>)
- [mloss](https://mloss.org/) (<https://mloss.org/>) is an academic database of open-source machine learning software.
- Machine Learning Crash Course (<https://developers.google.com/machine-learning/crash-course/>) by Google. This is a free course on machine learning through the use of TensorFlow.

Retrieved from "https://en.wikipedia.org/w/index.php?title=Machine_learning&oldid=936209083"

This page was last edited on 17 January 2020, at 10:48 (UTC).

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.