American Journal of Epidemiology

Vol. 188, No. 12
DOI: 10.1093/aje/kwz189
Advance Access publication: October 21, 2019

Practice of Epidemiology

What is Machine Learning? A Primer for the Epidemiologist

Qifang Bi, Katherine E. Goodman, Joshua Kaminsky, and Justin Lessler*

* Correspondence to Dr. Justin Lessler, Department of Epidemiology, Johns Hopkins Bloomberg School of Public Health, 615 N. Wolfe Street, Room E6545, Baltimore, MD 21231 (e-mail: justin@jhu.edu).

Initially submitted November 16, 2018; accepted for publication August 14, 2019.

Machine learning is a branch of computer science that has the potential to transform epidemiologic sciences. Amid a growing focus on "Big Data," it offers epidemiologists new tools to tackle problems for which classical methods are not well-suited. In order to critically evaluate the value of integrating machine learning algorithms and existing methods, however, it is essential to address language and technical barriers between the two fields that can make it difficult for epidemiologists to read and assess machine learning studies. Here, we provide an overview of the concepts and terminology used in machine learning literature, which encompasses a diverse set of tools with goals ranging from prediction to classification to clustering. We provide a brief introduction to 5 common machine learning algorithms and 4 ensemble-based approaches. We then summarize epidemiologic applications of machine learning techniques in the published literature. We recommend approaches to incorporate machine learning in epidemiologic research and discuss opportunities and challenges for integrating machine learning and existing epidemiologic research methods.

Big Data; ensemble models; machine learning

Abbreviations: ANN, artificial neural networks; BMA, Bayesian model averaging; BMI, body mass index; CART, classification and regression trees; SVM, support vector machine.

Machine learning is a branch of computer science that broadly aims to enable computers to "learn" without being directly programmed (1). It has origins in the artificial intelligence movement of the 1950s and emphasizes practical objectives and applications, particularly prediction and optimization. Computers "learn" in machine learning by improving their performance at tasks through "experience" (2, p. xv). In practice, "experience" usually means fitting to data; hence, there is not a clear boundary between machine learning and statistical approaches. Indeed, whether a given methodology is considered "machine learning" or "statistical" often reflects its history as much as genuine differences, and many algorithms (e.g., least absolute shrinkage and selection operator (LASSO), stepwise regression) may or may not be considered machine learning depending on who you ask. Still, despite methodological similarities, machine learning is philosophically and practically distinguishable. At the liberty of (considerable) oversimplification, machine learning generally emphasizes predictive accuracy over hypothesis-driven inference, usually focusing on large, high-dimensional (i.e., having many covariates) data sets (3, 4). Regardless of the precise distinction between approaches, in practice, machine learning offers epidemiologists important tools. In particular, a growing focus on "Big Data" emphasizes problems and data sets for which machine learning algorithms excel while more commonly used statistical approaches struggle.

This primer provides a basic introduction to machine learning with the aim of providing readers a foundation for critically reading studies based on these methods and a jumping-off point for those interested in using machine learning techniques in epidemiologic research. The "Concepts and Terminology" section of this paper presents concepts and terminology used in the machine learning literature. The "Machine Learning Algorithms" section provides a brief introduction to 5 common machine learning algorithms: artificial neural networks, decision trees, support vector machines, naive Bayes, and *k*-means clustering. These are important and commonly used algorithms that epidemiologists are likely to encounter in practice, but they are by no means comprehensive of this large and highly diverse field. The following two sections, "Ensemble Methods" and "Epidemiologic Applications," extend this examination to ensemble-based approaches and epidemiologic applications in the published literature. "Brief

Recommendations" provides some recommendations for incorporating machine learning into epidemiologic practice, and the last section discusses opportunities and challenges.

CONCEPTS AND TERMINOLOGY

For epidemiologists seeking to integrate machine learning techniques into their research, language and technical barriers between the two fields can make reading source materials and studies challenging. Some machine learning concepts lack statistical or epidemiologic parallels, and machine learning terminology often differs even where the underlying concepts are the same. Here we briefly review basic machine learning principles and provide a glossary of machine learning terms and their statistical/epidemiologic equivalents (Table 1).

Supervised, unsupervised, and semisupervised learning

Machine learning is broadly classifiable by whether the computer's learning (i.e., model-fitting) is "supervised" or "unsupervised." *Supervised learning* is akin to the type of model-fitting that is standard in epidemiologic practice: The value of the outcome (i.e., the dependent variable), often called its "label" in machine learning, is known for each observation. Data with specified outcome values are called "labeled data." Common supervised learning techniques include standard epidemiologic approaches such as linear and logistic regression, as well as many of the most popular machine learning algorithms (e.g., decision trees, support vector machines).

In *unsupervised learning*, the algorithm attempts to identify natural relationships and groupings within the data without reference to any outcome or the "right answer" (5, p. 517). Unsupervised learning approaches share similarities in goals and structure with statistical approaches that attempt to identify unspecified subgroups with similar characteristics (e.g., "latent" variables or classes) (6). Clustering algorithms, which group observations on the basis of similar data characteristics (e.g., both oranges and beach balls are round), are common unsupervised learning implementations. Examples may include *k*-means clustering and expectation-maximization clustering using Gaussian mixture models (7, 8).

Semisupervised learning fits models to both labeled and unlabeled data. Labeling data (outcomes) is often time-consuming and expensive, particularly for large data sets. Semisupervised learning supplements limited labeled data with an abundance of unlabeled data with the goal of improving model performance (studies show that unlabeled data can help build a better classifier, but appropriate model selection is critical) (9). For example, in a study of Web page classification, Nigam et al. (10) fit a naive Bayes classifier to labeled data and then used the same classifier to probabilistically label unlabeled observations (i.e., fill in missing outcome data). They then retrained a new classifier on the resulting, fully labeled data set, thereby achieving a 30% increase in Web page classification accuracy on data outside of the training set. Semisupervised learning can bear some similarity to statistical approaches for missing data and censoring (e.g., multiple imputation), but as an approach that focuses on imputing missing outcomes rather than missing covariates.

Classification versus regression algorithms

Within the domain of supervised learning, machine learning algorithms can be further divided into classification or regression applications, depending upon the nature of the response variable. In general, in the machine learning literature, *classification* refers to prediction of categorical outcomes, while *regression* refers to prediction of continuous outcomes. We use this terminology throughout this primer and are explicit when referring to specific regression algorithms (e.g., logistic regression). Many machine learning algorithms that were developed to perform classification have been adapted to also address regression problems, and vice versa.

Generative versus discriminative algorithms

Machine learning algorithms, both supervised and unsupervised, can be discriminative or generative (11, 12). *Discriminative algorithms* directly model the conditional probability of an outcome, $\Pr(y|x)$ (the probability of y given x), in a set of observed data—for example, the probability that a subject has type 2 diabetes mellitus given a certain body mass index (BMI; weight (kg)/height $(m)^2$). Most statistical approaches familiar to epidemiologists (e.g., linear and logistic regression) are discriminative, as are most of the algorithms discussed in this primer.

In contrast, while *generative algorithms* can also compute the conditional probability of an outcome, this computation occurs indirectly. Generative algorithms first model the joint probability distribution, $\Pr(x, y)$ (the probabilities associated with all possible combinations of x and y), or, continuing our example, a probabilistic model that accounts for all observed combinations of BMIs and diabetes outcomes (Table 2). This joint probability distribution can be transformed into a conditional probability distribution in order to classify data, as $\Pr(y|x) = \Pr(x, y)/\Pr(x)$. Because the joint probability distribution models the underlying data-generating process, generative models can also be used, as their name suggests, for directly generating new simulated data points reflecting the distribution of the covariates and outcome in the modeled population (11). However, because they model the full joint distribution of outcomes and covariates, generative models are generally more complex and require more assumptions to fit than discriminative algorithms (12, 13). Examples of generative algorithms include naive Bayes and hidden Markov models (11).

Reinforcement learning

In reinforcement learning, systems learn to excel at a task over time through trial and error (14). Reinforcement learning techniques take an iterative approach to learning by obtaining positive or negative feedback based on performance of a given task on some data (whether prediction, classification, or another action) and then self-adapting and attempting the task again on new data (though old data may be reencountered) (15). Depending on how it is implemented, this approach can be akin to supervised learning, or it may represent a semisupervised approach (as in generative adversarial neural networks (16)). Reinforcement learning algorithms often optimize the use of early, "exploratory" versions of a model—that is, task attempts—that perform poorly to gain information to perform better on future attempts, and then

Table 1. Glossary of Machine Learning and Epidemiology Terminology

Machine Learning Term(s)	Epidemiology Term(s)	Definition and Notes	Example
Attribute, feature, predictor, or field	Independent variable	Machine learning uses various terms to reference what epidemiologists would consider an "independent variable," including *attribute*, *feature*, *predictor*, and *field*.	In a data set with 4 independent variables (BMI[a], age, race, and SES) and a dependent variable (diabetes mellitus), BMI, age, race, and SES are attributes.
Domain	Range of possible variable values	The domain is the set of possible values of an attribute. It can be continuous or categorical/binary.	If race is recorded in a data set as "1 = Caucasian, 2 = African-American, and 3 = other," its domain is categorical and includes only the 3 referenced categories.
Input and output	Independent (exposure) and dependent (outcome) variables	In machine learning, "input" refers to all of the predictors or independent variables that enter the model, and "output" generally refers to the predicted value (whether a number, classification, etc.) of the dependent variable or outcome.	BMI, age, race, and SES are model input. In a binary classification algorithm, the model output is a prediction of whether a subject does ($D = 1$) or does not ($D = 0$) have diabetes.
Classifier, estimator	Model	"Classifiers" or "estimators" are used generally in the machine learning literature to refer to algorithms that perform a prediction or classification of interest. Their less common, though more technical, usage specifically refers to fully parameterized models that are used to predict or classify.	A decision tree is one type of machine learning classifier (general usage). The more specific usage of this term would refer only to a parameterized decision tree that has been fit in a data set (e.g., that predicts diabetes outcomes from BMI, age, sex, and SES).
Learner	Model-fitting algorithm	A learner inputs a training set and outputs a classifier. Usually, but not always, *learner* refers to the fitting algorithm, while *classifier* refers to the fitted model.	In decision tree learning, the classification and regression trees (CART) algorithm, developed by Breiman et al. (27) in 1984, is one of multiple available learners for developing a decision tree classifier.
Dimensionality	No. of covariates	No. of independent variables under consideration in a model.	A data set with 4 independent variables (BMI, age, race, and SES) and a dependent variable (diabetes) has 4 dimensions.
Label	Value of dependent variables, outcomes	A variable's label is its value for each observation (e.g., 0 or 1). Although labels can technically describe any variable, common shorthand is that "labeled data" refers to data in which the dependent variable assumes a value for all observations.	In a data set for which an investigator has collected information on diabetes status (outcome) for all subjects, this is "labeled" data. The label for diabetes is 0 or 1. Partially labeled data would have diabetes status missing for some subjects.
Imbalanced data	Data set in which some cases or risk categories occur much less frequently than the others	In imbalanced machine learning data sets, the outcome or another risk category of interest occurs much less frequently, either because of the intrinsic nature of the problem (e.g., a rare disease in a database of medical records) or because of the sampling strategy (e.g., prevalence of cases in the study population is much lower than that in the target/source population). Heavily imbalanced data may pose challenges in some classification algorithms and require tuning parameters in order to correct for or otherwise address this imbalance. One method for addressing imbalanced data sets is to "balance" them artificially, either by oversampling instances of the minority class or undersampling instances of the majority class.	Assume a hypothetical data set of pediatric, normal-weight patients in which the prevalence of diabetes is 2%. This data set is imbalanced because the outcome is very rare, which can lead to poor sensitivity of classification algorithms without parameter tuning or other corrective methods. This imbalance is due to the intrinsic nature of the population we are evaluating (i.e., healthy children) and not due to the sampling strategy or other bias.
Loss function	Error measure	In machine learning, a loss function is generally considered a penalty for misclassification when assessing a model's predictive performance.	A simple loss function may be the absolute value of (predicted value minus true value). If a model predicts that a subject has diabetes ($D = 1$) and the subject does not ($D = 0$), the value of the loss function for this prediction is "1."

Abbreviations: BMI, body mass index; SES, socioeconomic status.
[a] Weight (kg)/height $(m)^2$.

Table 2. Matrix of Joint Probabilities for Body Mass Index[a] (x) and Diabetes Mellitus (y) in a Data Set With 4 Dichotomized Observations: (0, 1), (0, 1), (0, 1), and (0, 0)

Diabetes Status	BMI Status	
	Overweight BMI = 1	Overweight BMI = 0
$D = 1$	0/4	1/4
$D = 0$	2/4	1/4

Abbreviation: BMI, body mass index.
[a] Weight (kg)/height (m)^2.

become less labile as the model "learns" more (15). Medical and epidemiologic applications of reinforcement learning have included modeling the effect of sequential clinical treatment decisions on disease progression (17) (e.g., optimizing first- and second-line therapy decisions for schizophrenia management (18)) and personalized, adaptive medication dosing strategies. For example, Nemati et al. (19) used reinforcement learning with artificial neural networks in a cohort of intensive-care-unit patients to develop individualized heparin dosing strategies that evolve as a patient's clinical phenotype changes, in order to maximize the amount of time that blood drug levels remain within the therapeutic window.

MACHINE LEARNING ALGORITHMS

In this section, we introduce 5 common machine learning algorithms: artificial neural networks, decision trees, support vector machines, naive Bayes, and k-means clustering. For each, we include a brief description, summarize strengths and limitations, and highlight implementations available on common statistical computing platforms. This section is intended to provide a high-level introduction to these algorithms, and we refer interested readers to the cited references for further information.

Artificial neural networks

Artificial neural networks (ANNs) are inspired by the signaling behavior of neurons in biological neural networks. ANNs, which consist of a population of neurons interconnected through complex signaling pathways, use this structure to analyze complex interactions between a group of measurable covariates in order to predict an outcome. ANNs possess layers of "neurons" connected by "axons" (20) (Figure 1A). These layers are grouped into 1) an input layer, 2) one or more middle "hidden" layers, and 3) an output layer. The neurons in the input and output layers correspond to the independent and dependent variables, respectively. Neurons in adjacent layers communicate with each other through activation functions, which convert the weighted sum of a neuron's inputs into an output (Figure 1B). Depending on the type of activation function, the output can be dichotomous ("1" when the weighted sum exceeds a given threshold and "0" otherwise) or continuous. The weighted sum of a neuron's inputs is somewhat analogous to coefficients in linear or logistic regression.

Figure 1 illustrates a simple neural network with a single hidden layer and a feed-forward structure (i.e., signals progress

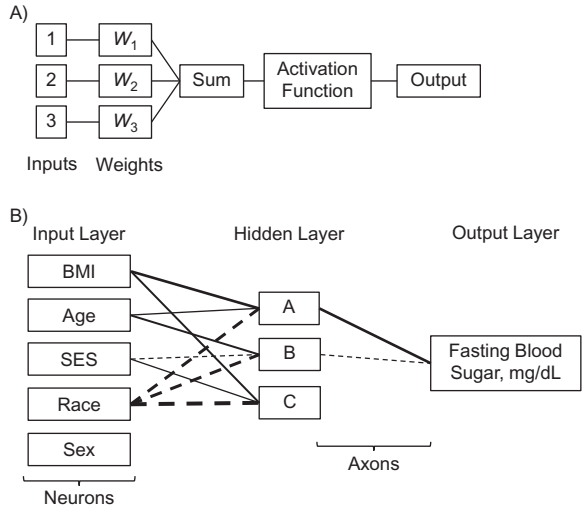

Figure 1. A single artificial neuron, also called a perceptron (panel A), and a feed-forward neural network (a collection of multiple neurons organized in layers) that examines the hypothetical relationship between clinical and demographic predictors and a numerical outcome, fasting blood sugar level (panel B). Line (axon) thickness reflects input weight, and line type indicates direction of effect (solid = excitatory or positive; dashed = inhibitory or negative). Lack of a line (e.g., connecting "sex" to neuron C) indicates no input. Connections between input and output layers are exclusively mediated through the hidden layer (more complex artificial neural networks can have multiple hidden layers). At hidden layer neuron A, we observe that both body mass index (BMI; weight (kg)/height (m)^2) and age exert positive inputs, and they demonstrate interactive effects with each other and race (the latter's input is negative, as indicated by the dashed line). The weighted sum of these inputs results in activation of neuron A and positive output. In contrast, neuron B converts inputs from age, socioeconomic status (SES), and race into negative output (inversely correlated with fasting blood sugar), while neuron C's inputs fail to surpass the activation function threshold; that is, there is no effect on the outcome mediated through neuron C.

unidirectionally from input to output layers). For supervised learning applications, once the numbers of layers and neurons are selected, the connection weights of the ANN are fit on a training set of labeled data through a reinforcement learning approach. Initial connection weights are generally selected randomly, and network output is compared with the correct output (class labels) using a loss function, which is based on the difference between the predicted and true values of the outcome. The goal is to reduce the loss function to zero—that is, to make the ANN's predicted output match truth as closely as possible, albeit while also protecting against overfitting. In response, 1) resulting error values are distributed backwards through the network, from output to input, in order to assign an error value contribution to each hidden and input layer neuron (called "back-propagation"; for additional technical information on this process, see, for example, Rumelhart et al. (21)), and 2) connection weights are updated in order to minimize the loss function ("weight adjustment"). This 2-fold optimization process repeats for a number of "epochs" or iterations until the network meets a prespecified stopping rule or error rate threshold (22, 23).

Strengths and limitations. Strengths of ANNs include their ability to accommodate variable interactions and nonlinear associations without user specification (22). The primary

limitation of ANNs is that, although it is arguably not completely a "black box" (23, p. 1112), the underlying model nevertheless remains largely opaque. Effects are mediated exclusively through hidden layer(s), making interpreting relationships between input and output layers challenging, especially for "deep" ANNs, which include multiple hidden layers. This lack of transparency complicates commonsense or etiological interpretation of individual variable effects and connection weights, although there are continuing efforts to enhance ANN interpretability (20, 24, 25). ANN training parameters can also be complex, and setting and tuning these parameters generally necessitates technical expertise. Moreover, complex ANNs, including deep networks, can require large data sets (potentially in the tens or hundreds of thousands, although there is no hard-and-fast rule) in order to achieve optimal model performance, which may be prohibitive for some epidemiologic applications (26).

Sample statistical packages and modules. Available software includes neuralnet, nnet, deepnet, and TensorFlow in R (R Foundation for Statistical Computing, Vienna, Austria); Enterprise Miner Neural Network and AutoNeural in SAS (SAS Institute, Inc., Cary, North Carolina); and sklearn and TensorFlow in Python (Python Software Foundation, Wilmington, Delaware).

Decision trees

Decision trees (i.e., classification and regression trees (CART)) create a series of decision rules based on continuous and/or categorical input variables to predict an outcome (5, 27). Classification trees predict categorical outcomes, and regression trees predict continuous outcomes. CART analysis has been popularized as an umbrella term for any decision tree learning method (27). However, "CART" is also a common implementation algorithm in the epidemiologic and medical literature, although a number of other decision tree algorithms have also been developed (e.g., ID3, CHAID) (28–30).

Figure 2 presents a hypothetical classification tree for a binary outcome, diabetes. To derive a decision tree, the algorithm applies a splitting rule on successively smaller partitions of data, with each partition being a node on the tree. The partition consisting of all data is the root node; in Figure 2 this node is split on the basis of BMI. Splits are selected to minimize some measure of node impurity (i.e., diversity of classes) or heterogeneity (i.e., variance) in each resulting partition (the "daughter nodes") (5, 27). The splitting process repeats on each branch of the tree until additional splits yield no further reductions in node impurity, or some other stopping criterion is reached (e.g., a specified minimum number of observations in terminal nodes or the value at which error is minimized in cross-validation (31)). In many algorithms, this splitting is often followed by a "pruning" step in which partitions are remerged (i.e., some bottom nodes are removed, making the final tree smaller) based on some criterion designed to increase generalizability (32).

Strengths and limitations. Decision trees are generally easy to understand—its having been said that "[o]n interpretability, trees rate an A+" (4, p. 206)—making their output ideal for a range of target audiences. They are also flexible to nonlinear

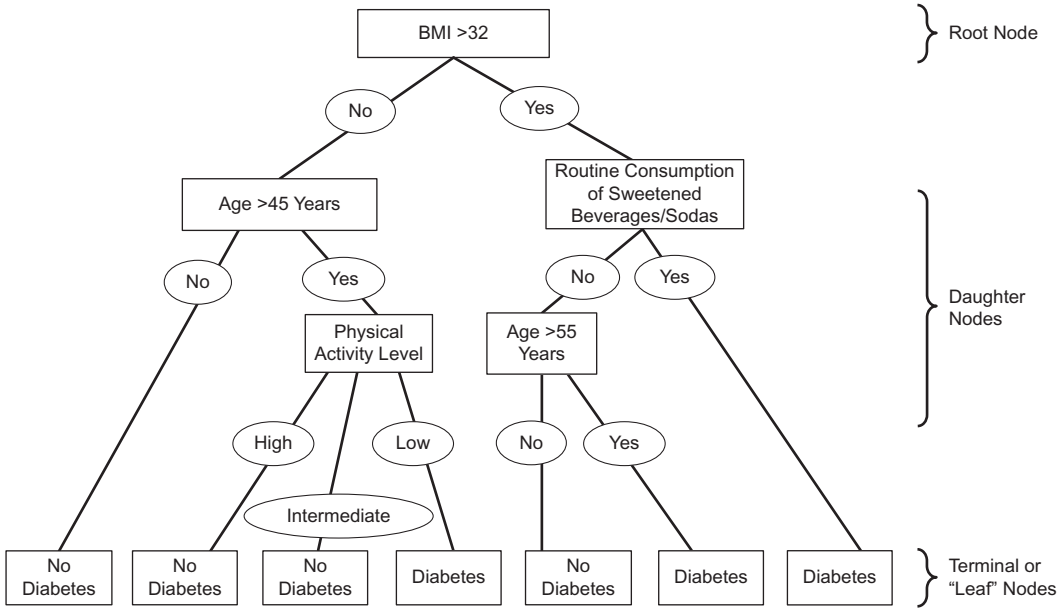

Figure 2. A hypothetical classification decision tree for predicting a binary outcome, type 2 diabetes mellitus. Body mass index (BMI; weight (kg)/height (m)^2) occupies the root node (the most discriminatory variable in the data set); age, consumption of sweetened beverages, and physical activity occupy daughter nodes; and predicted diabetes status (yes/no) is reflected in the terminal or "leaf" nodes. Terminal node predictions proceed on the basis of simple majority rule (e.g., if 60% of patients in a terminal node are diabetes-positive, the entire terminal node will be classified as "Diabetes"). The cutpoints for the continuous variables, BMI and age, are algorithm-derived. The presence of age at different cutpoints in 2 different daughter nodes reflects likely interaction effects: The relationship between age and diabetes differs in patients with BMI ≤32 compared with patients with BMI >32 who do not routinely consume sweetened beverages.

covariate effects and can incorporate higher-order interactions between covariates (27, 33). Trees may lose information by dichotomizing or categorizing variables where associations are continuous, and they can be unstable to even small data changes. Because most decision tree algorithms are "greedy" (splitting decisions are locally optimized at nodes), through a domino effect, dramatically different trees can result if even a single higher-level node shifts to a different variable (34). Hence, decision trees can be highly sensitive to small perturbations in data. Perhaps most fundamentally, decision trees are prone to overfitting, and their ultimate utility depends heavily on appropriately implemented pruning and/or stopping criteria. Ensemble-based decision trees (e.g., random forests) can address some of these concerns (see "Ensemble Methods" section), but they do not produce a single, easily interpretable tree.

Sample statistical packages and modules. Available software includes rpart, caret, ctree, and randomForest (ensemble decision trees) in R; CART (failure-time data only), CHAID, and CHAIDFOREST (ensemble decision trees) in Stata (StataCorp LLC, College Station, Texas); Enterprise Miner Decision Tree in SAS; and sklearn in Python.

Support vector machines

Support vector machines (SVMs) are a set of supervised learning methods used for classification and regression problems (35, 36). SVMs construct an optimal boundary, called a hyperplane, that best separates observations of different classes. In 1 dimension, this boundary is a point; in 2 dimensions, a line; and in 3, a plane (Figure 3). However, many observations often need to be transformed before they can be separated by a hyperplane. SVMs address this problem by applying a data transformation called a "kernel function" to the data (3). Kernel functions project the data into a higher-dimensional space where the input variables are separable (Figure 3). The optimal kernel function is usually chosen from a set of commonly used kernel functions selected through cross-validation. Popular kernel functions include polynomial kernel, gaussian kernel, and sigmoid kernel. Following kernel function transformation, the best hyperplane maximizes the separation between the different classes (i.e., the margin, defined as the distance from the hyperplane to the closest data point), while tolerating a specified level of misclassification. SVMs are traditionally used for binary classification, but multiple pairwise comparison can be applied for multiclass classification (36). Extensions to SVM techniques have also been developed that can be used to predict continuous outcomes (called support vector regression) (37).

In Figure 3, persons with and without diabetes cannot be separated by a line in the 2-dimensional space based upon the predictors, age and BMI (Figure 3A). However, when we project the data into a 3-dimensional space by applying a kernel given by $\varphi((\text{age}, \text{BMI}) = (\text{age}, \text{BMI}, (\text{BMI} - a) \times (\text{age} - b))$, where a and b are fixed parameters estimated from the data, the data are now separable in the 3-dimensional space by a plane (Figure 3B).

Strengths and limitations. SVMs generally demonstrate low misclassification error and scale well to high-dimensional data (38). SVMs have reasonable interpretability, especially when a kernel function is not used. Where a kernel function is necessary, however, selecting the optimal kernel function typically requires experimenting with a set of standard functions. This approach can be time-consuming and does not guarantee that the set of standard kernel functions that were evaluated included the optimal function, and in some cases hand-crafted kernel functions are used instead.

Sample statistical packages and modules. Available software includes e1071, kernlab, and caret in R; svmachines (39) in Stata; PROC SVM in SAS; and sklearn in Python.

Naive Bayes algorithms

A *naive Bayes* algorithm is a simple probabilistic classification algorithm based upon Bayes' theorem that makes the "naive" assumption of independence between predictive variables (40). Naive Bayes calculates the probability associated with each possible class conditional on a set of covariates—that is, the product of the prior probability and the likelihood function. The classifier then selects the class with the highest probability as the "correct" class (Figure 4). The prior probability

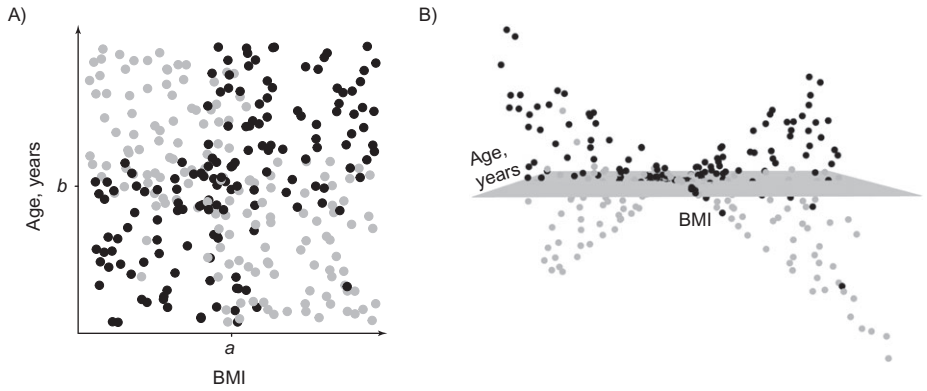

Figure 3. An illustration of data transformation with a support vector machine for predicting diabetes status. A) Hypothetical age and body mass index (BMI; weight (kg)/height (m)^2) distribution of diabetic (black dots) and nondiabetic (gray dots) patients in 2-dimensional space. *a* and *b* are fixed parameters estimated from the data (see text). B) After transformation, these dots/patients who are not linearly separable in 2-dimensional space become linearly separable in 3-dimensional space. A hyperplane in 3-dimensional space is shown as a surface.

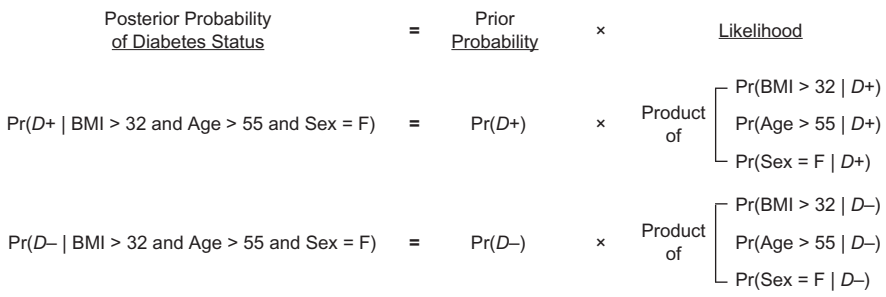

Figure 4. A hypothetical naive Bayes algorithm for predicting a binary outcome, type 2 diabetes mellitus, in the subpopulation whose body mass index (BMI; weight (kg)/height (m)^2) is over 32, whose age is over 55 years, and who are female. The prior probability of the class (e.g., diabetes status) and a product of the likelihood functions, one for each patient characteristic, determine the class assignment. If the posterior probability of being diabetic ($D+$) in this population, Pr($D+$|BMI > 32, age > 55, female), is larger than the posterior probability of not being diabetic ($D-$) in this population, Pr($D-$|BMI > 32, age > 55, female), then this population would be classified as having diabetes. The prior probability of being diabetic, Pr($D+$), approximates the overall diabetes prevalence. Because of the independence assumption, the likelihood of observing people with this set of attributes—BMI > 32, age > 55 years, and female sex—among the persons with diabetes (i.e., Pr(BMI > 32, age > 55, female|$D+$)) can be approximated by the product of the likelihood of observing each attribute among persons with diabetes (i.e., Pr(BMI > 32|$D+$) × Pr(age > 55|$D+$) × Pr(female|$D+$)). For example, Pr(BMI > 32|$D+$) represents, among persons with diabetes, the likelihood of observing people with BMI >32.

typically reflects one's belief about the outcome, either based on the study itself or from other published literature. The independence assumption in naive Bayes greatly simplifies the calculation by decomposing the likelihood function into a product of likelihood functions, one for each covariate. Though adaptations of naive Bayes for regression exist (41), the algorithm is most commonly used for classification.

Continuing our diabetes example, a naive Bayes classifier would calculate the likelihood of each observation (e.g., BMI > 32, age > 55 years, and female sex) among people who are and are not diabetic (Figure 4). Assuming equal prior probability for diabetes, an individual would be assigned to the class (i.e., diabetic vs. not diabetic) that had the highest likelihood of independently producing each observation.

Strengths and limitations. The simplicity of the naive Bayes approach contributes to the popularity of these algorithms. It has been shown to perform relatively well in the presence of noise, missing data, and irrelevant features (42). Because of the independence assumption, naive Bayes requires estimation of fewer parameters, and thus a smaller training set, than more complex algorithms (43, 44).

Arguably the most important limitation of naive Bayes is that its independence assumption is often violated in the real world. In addition, the most probable class may weigh heavily on the chosen prior. Thus, proper adjustment for underlying class frequencies is necessary when prior probability in the training set is not representative of the general population. In addition, when data are correlated, naive Bayes gives more influence to the likelihood function of highly correlated features and may bias the prediction (43). These limitations will not affect classification performance, however, so long as the ordering of the biased probabilities is the same as that of the correct ones. Naive Bayes probability outputs nevertheless should *never* be interpreted as actual probabilities of class membership.

Sample statistical packages and modules. Available software includes e1071, klaR, bnlearn, H2O, and naivebayes in R; multinomial mixture models in StataStan (45); PROC HPBNET in SAS; and sklearn in Python.

K-means clustering

K-means clustering is one of the simplest unsupervised learning algorithms (46). It partitions observations into a prespecified number of distinct clusters (k), such that within-cluster variation (e.g., squared Euclidean distance) is as small as possible (47). *K*-means clustering first randomly selects k centroids, with each centroid defining 1 cluster (i.e., each observation is assigned to its closest centroid). Following k selection, the algorithm iteratively alternates between 2 steps until classification remains unchanged: 1) assign each observation to its nearest centroid, typically defined by squared Euclidean distance, and 2) move the location of the centroid to the mean of all data points assigned to that centroid's cluster (Figure 5). There are a variety of methods for selecting k. Often investigators prespecify k based on background knowledge or visual examination of the data; however, likelihood and error-based approaches to selecting k have been developed (48).

Strengths and limitations. *K*-means clustering is simple, easy to interpret, and computationally efficient. However, one important limitation is that the number of clusters needs to be prespecified. A slight difference in k can produce very different results, and methods for estimating k (49) do not necessarily agree with each other (50). In addition, when the distance between observations and cluster centroids is calculated with Euclidean metrics, the algorithm assumes that clusters have the same within-cluster variance. If some clusters are much larger than others, *k*-means can produce nonintuitive results (50) (Figure 6).

Sample statistical packages and modules. Available software includes ClusterR, fpc, akmeans, and kmeans in base R; cluster kmeans in Stata; FASTCLUS and HPCLUS in SAS; and sklearn in Python.

ENSEMBLE METHODS

Ensemble methods utilize information from multiple models to improve predictive performance in comparison with a single model. The idea is that even though any individual model

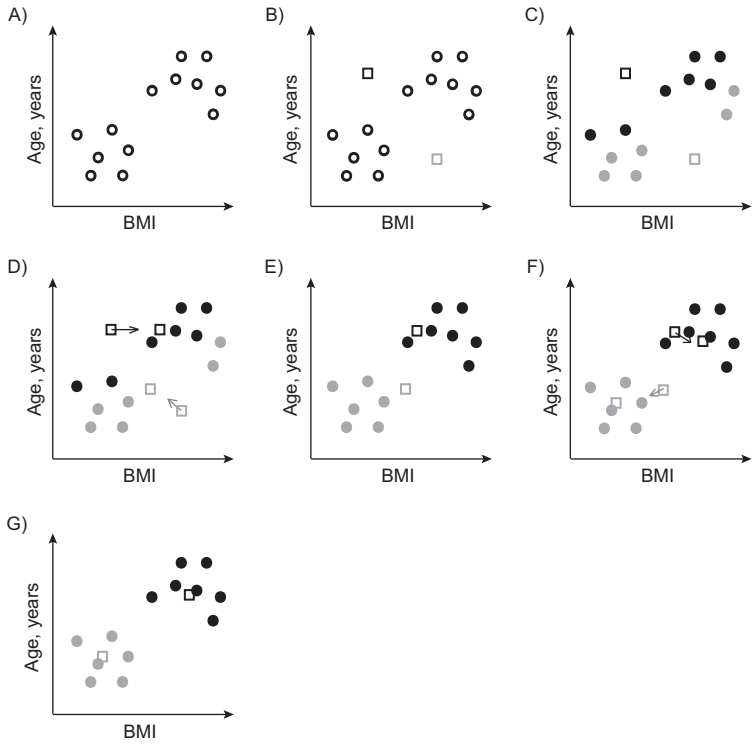

Figure 5. A hypothetical *k*-means algorithm for dichotomizing ($k = 2$) patients on the basis of their age and body mass index (BMI; weight (kg)/height $(m)^2$). Each unclassified observation (hollow dots) is assigned to a diabetes classification (solid dots), with black and gray representing the predicted diabetes classifications (black, diabetic; gray, nondiabetic) at each step. Squares are centroids, with a single centroid per cluster. The steps of a *k*-means algorithm classifying hypothetical data are: A) obtain unclassified data; B) randomly select $k = 2$ centroids; C) assign each observation to its nearest centroid and predict its diabetes status (black dots are closer to the black square and gray dots are closer to the gray square); C) move the black centroid to the mean of all black dots, and similarly for the gray dots, as represented by centroid arrows; D) reclassify observations to the nearest, updated centroid; E) repeat step C; F) repeat step D; and G) perform final classifications, assuming that clusters have stabilized.

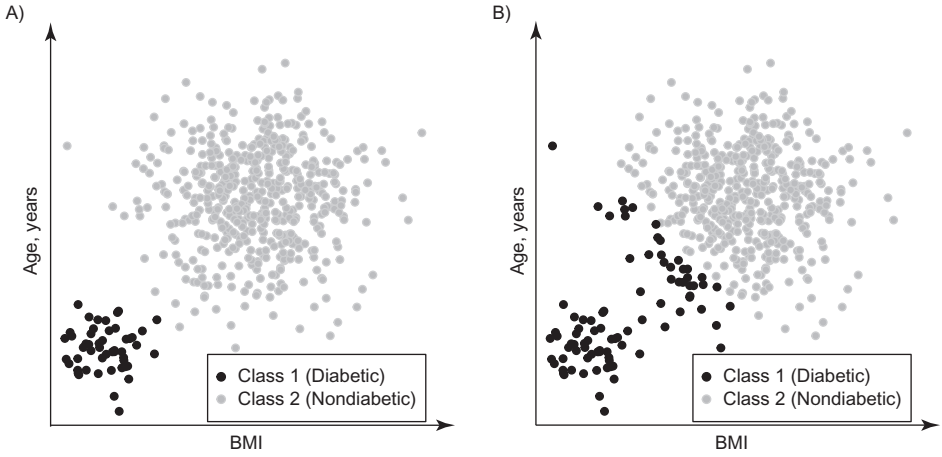

Figure 6. One limitation of the *k*-means algorithm, as illustrated with simulated data on age and body mass index (BMI; weight (kg)/height $(m)^2$). When one cluster (upper right) is much larger than the other (lower left), *k*-means can produce counterintuitive classifications (A). The more intuitive classification is shown in the right-hand panel (B).

within an ensemble is not adequate to capture the characteristics of the entire phenomenon, so long as the models perform better than they would with random class assignment, once combined they can borrow strength from each other and achieve high predictive accuracy. Broadly, ensemble methods improve performance by creating a population of models through either 1) training the same underlying algorithm to different versions of a data set (e.g., bagging and boosting) or 2) training qualitatively different models on the same data set (e.g., Bayesian model averaging, Super Learner) (see Web Figure 1, available at https://academic.oup.com/aje) and then combining results across these models on the basis of a defined algorithm. While the primary objective of bagging and boosting is to minimize overfitting, multiple algorithm ensembles capitalize on different models' strengths and avoid the need for model preselection. These alternative ensemble approaches are often used in combination, either as part of the same algorithm or through nested approaches.

Bagging

Bagging (or bootstrap aggregating) fits the same underlying algorithm to each bootstrapped copy of the original training data and then creates a final prediction based on outputs from the resulting, parameterized, models (51). The final prediction for a quantitative outcome is obtained by averaging the predictions. For a qualitative outcome, the final prediction either takes the majority vote among the classifiers or averages probabilities across the number of bootstrap fits. Bagging reduces model variance significantly without affecting bias (52, 53).

Feature bagging attempts to further reduce overfitting. It trains models on random subsets of variables/features instead of all variables in an attempt to reduce correlation between models in an ensemble. When applied to tree-based methods, the resulting models are called *random forests*, which force each split to consider a random subset of predictors (54), giving other weak predictors a greater chance to be selected as split candidates. Otherwise, when there is a strong predictor for the outcome, many trees would choose to first split on that predictor, creating highly correlated predictions regardless of the variables chosen at the subsequent splits.

Aside from *k*-fold cross-validation, one way to estimate prediction errors specifically for random forests is to compute *out-of-bag error* (55). Out-of-bag error is the mean prediction error for each observation, using only the models that did not include the observation in their bootstrapped samples. *Variable importance rankings* summarize the relative importance of each predictor across all fitted trees. These rankings reflect the importance of a variable for predicting outcomes by averaging the impurity decrease for all nodes where the variable is used across all trees in the forest (51). Impurity decrease measures changes in the accuracy of a tree and can be described by, for example, Gini impurity (a measure of the probability of mistaken categorization within a node) for bagging classification trees or the residual sum of squares for bagging regression trees. "Important" variables change the accuracy of the trees the most. Importance rankings can be used to assess the relative impact of individual predictors, as well as the interaction between predictors, in predicting the outcome (56, 57).

Available software includes caret, randomForest, and adabag in R; CHAIDFOREST for random forests in Stata; SAS (58); and sklearn.ensemble in Python.

Boosting

Like bagging, *boosting* also trains models on subsets of data, but it does it in a sequential fashion and improves the classifiers by analyzing prediction errors (59, 60). AdaBoost is a well-known boosting method that sets weights to both observations and classifiers (61, 62). Observations are given weights, initially equal, that increase if incorrectly classified by the last iteration of the classifier; hence, subsequent iterations will prioritize correctly classifying these observations. The final output classifier is a weighted average from the classifier built in each iteration, with higher weight given to classifiers with higher predictive accuracy (i.e., lower error rates on training data). *Gradient boosting* is a generalization of AdaBoost that uses gradient descent to optimize any differentiable loss function (i.e., a measure of classifier performance other than simple classification error) (63, 64).

Available software includes gbm, adabag, fastAdaboost, xgboost, ada, and caret in R; Stata (65); SAS (58); and sklearn.ensemble in Python.

Bayesian model averaging

Bayesian model averaging (BMA) estimates the posterior distribution of a predicted value (or the parameters defining a parametric relationship) by calculating the weighted average of model-specific estimates, where the weights are driven by how much the data support each competing model (66). BMA has been applied to many statistical models, including linear regression, generalized linear models, and Cox proportional hazards models, and it provides better predictive ability than using any single model (66). Its variants, such as *Bayesian model combination*, have emerged to further tackle the issue of overfitting, as BMA has a tendency to place too much weight on the most probable model (67). Bayesian model combination creates a set of ensembles, each representing a combination of individual models, and weights the ensemble-specific estimate of the effect size (as opposed to estimates based on the most probable model in BMA) by the probability that the ensemble is correct given the data (67, 68).

Available software includes BMS/BAS/BMA in R; SAS (69); and pyBMA in Python.

Super Learner

Super Learner is a prediction algorithm that uses cross-validation to determine the optimal weighted combination of predictions from a group of candidate learners (70–72). Building on the "stacked generalization" approach proposed by Wolpert (72), this approach allows the use of machine learning algorithms (e.g., random forests) in addition to standard parametric algorithms (e.g., logistic regression). *K*-fold cross-validation (Web Figure 1) is used to assign weights to each of a user-defined pool of component algorithms based on out-of-training set performance, and then the component models are fit to the entire data set. Model outputs are based on the predictions of

these candidate models weighted by the cross-validation–derived weights. It has been applied to predict the drug susceptibility of human immunodeficiency virus as a function of its mutations (71), and it has been used as part of procedures to estimate causal effects (see "Epidemiologic Applications" section).

Available software includes SuperLearner in R and scikit-learn in Python.

EPIDEMIOLOGIC APPLICATIONS

In this section, we give an overview of the way in which machine learning algorithms have been used in various applications related to epidemiologic practice. While this is not a comprehensive review and we do not intend to discuss every limitation and nuance of these approaches, we hope to direct readers to areas of active research in the literature.

Causal inference

Relative to classical statistical or epidemiologic approaches, machine learning algorithms have historically placed less emphasis on causal inference. Indeed, machine learning has been described as a "black box" method because it is difficult to draw etiological inferences from the output of some algorithms (e.g., ANNs). However, machine learning techniques can still be an important component of approaches to estimating causal effects in observational studies, with sometimes superior performance for reducing bias and controlling for confounding (73).

Propensity score weighting is a common approach for estimating causal effects in observational studies (74). Propensity scores have traditionally been estimated with logistic regression, but this approach requires assumptions that, if unmet, may render biased effect estimates despite propensity score conditioning. Machine learning algorithms often deal implicitly with interactions and nonlinearities, whereas such high-order terms must be explicitly specified (and are commonly ignored) in logistic regression. Machine learning algorithms also perform well in estimating propensity scores in the presence of high-dimensional data and can reduce underlying model misspecification (75). Although these machine learning benefits may exist at the expense of easy interpretability, these concerns are not pertinent to propensity score estimation, as the interpretability of propensity scores is not relevant to their performance. Multiple studies have empirically demonstrated bias reductions where propensity scores are generated with machine learning methods, particularly ensemble-based approaches (75–78). Under certain conditions, however, bias may persist or be exacerbated by machine learning methods (79–81). Studies in which researchers calculated propensity scores with machine learning approaches have included those assessing the effects of early sexual initiation on young adult health (82), vaccination on birth outcomes (83), and combination antibiotic treatment on Gram-negative bacteremia (84).

Likewise, machine learning algorithms can be used as a component of any causal inference framework where an estimate of the likelihood (or distribution) of an outcome is an important component of the inferential process but need not be directly interpretable. For example, the Super Learner algorithm has been used as a component of targeted maximum likelihood estimation and marginal structural model approaches (85). Examples include investigations of the relationship between alcohol outlet density and alcohol consumption patterns (86) and the relationship between childhood adversity and mental disorders by race/ethnicity (87).

Machine learning methods have been used more directly to attempt to understand heterogeneity in treatment effects across subpopulations. For example, Athey et al. (88) have developed an approach to building "casual trees" that create decision trees where groupings are based on treatment effect and provide principled estimates of treatment effects within these strata using an approach they call "honest estimation." This approach has been extended to apply the random forest algorithm to these trees, creating so-called "casual forests" that can be used to estimate treatment effects in persons with particular covariate profiles (89).

Another application of machine learning more directly related to problems of causal inference is causal structure learning, which has grown as a distinct branch of machine learning. Causal structure learning encompasses a group of exploratory techniques that identify an optimal directed acyclic graph consistent with conditional independence relationships in the data and provided background knowledge. Approaches to causal structure learning include Bayesian network approaches (see Scutari and Denis (90) for an overview) and linear, nongaussian, acyclic models (LiNGAMs) (91–93). The former have been applied to derive causal influences in cellular signaling pathways (94) and to infer causal associations between gene expression and disease (95), while the latter have been used to estimate causal directionality between sleep disorders and depression (96) and to explore causality between television viewing habits and weight change (97).

Diagnostics, prognostic predictive models, and other clinical decision support tools

Disease diagnosis and prognosis are perhaps the oldest clinical utilizations of machine learning techniques (98) and remain common applications in the epidemiologic literature. Machine learning is particularly well-suited to certain diagnostic questions (e.g., those that involve imaging and/or high-dimensional data), and it can enhance prognostic models and clinical decision support tools through, for example, automation and ease of use.

Diagnostics that involve imaging, where each pixel can be conceptualized as a feature, and other high-dimensional data are problems well suited for machine learning approaches. SVMs have been utilized extensively in oncology for diagnosis and disease staging from radiological and tissue data (99–107). They have also been utilized for tumor typing from tissue microarray gene expression data, which, because of their high dimensionality, can be problematic for traditional statistical models (108–111). Outside of oncology, SVMs have shown promise for neuroimaging diagnostics, including for dementia (112) and autism spectrum disorder (113–115).

Machine learning techniques are also well-suited to prognostic models and other clinical decision support tools where accurate diagnosis (i.e., low classification error) is the primary objective, or where automation is desired. For example, Palaniappan and Awang (116) employed a combination of methods (ANNs, naive Bayes, and decision trees) in order to develop an automated, Web-based prediction tool, the Intelligent Heart Disease Prediction System. Incorporation into hospital and emergency room

operations research is also common. ANNs have been used in emergency room populations, for example, to predict death among sepsis patients (117) and prolonged hospital stays among the elderly (118), and random forests have been used to build electronic triage models for risk-stratifying patients (119, 120). Because many machine learning methods can accommodate complex variable interactions without a priori specification, they may also uncover previously unknown prognostic subgroups (121). For example, Brims et al.'s (121) application of CART algorithms to a data set of malignant pleural mesothelioma cases revealed 4 distinct prognostic groups based upon clinical characteristics.

Conversely, machine learning methodologies can also be helpful where manual use, rather than automation, is contemplated. In particular, decision trees are popular clinical prediction tools for both diagnosis and prognosis due to their simplicity and interpretability. Because their output uses branching logic rather than calculations, decision trees are generally user-friendly for clinicians to apply at the bedside (e.g., to predict the likelihood that an infection is drug-resistant while awaiting microbiological confirmation (122, 123)).

Genome-wide association studies

Genome-wide association studies seek to identify genetic variants that influence disease risk. Genome data sets generally contain large numbers of genes and single nucleotide polymorphisms of interest but, because of sequencing costs and other practical constraints, are of limited sample size. These high-dimensional data are the types of data on which machine learning algorithms perform well. Hence, ensemble machine learning approaches such as random forests are commonly used. Random forests can rank the most important single nucleotide polymorphisms for a disease outcome. For example, they have been used to predict drug response in epilepsy patients based on clinical and genetic information (124); to identify genetic variants associated with Parkinson disease and other neurological disorders (125); and to "data-mine" high-density genetic data to predict Alzheimer disease risk (126).

Other, nonensemble algorithms are also popular in genome-wide association studies. Researchers have applied SVMs with Bayesian model averaging to genome-wide data to predict late-onset Alzheimer disease (127) and *k*-nearest neighbors (a relatively simple unsupervised classification algorithm (128)) to predict the heritable genetic susceptibility of common cancers (129). Microbiome studies, which also involve high-dimensional (albeit bacterial) genetic data, have likewise utilized machine learning to identify disease risk factors among microbiota/microbiome signatures (130). Moreover, because interactions do not require a priori specification under many machine learning algorithms, machine learning approaches are well-suited to identification of complex gene-gene (131) and gene-environment interactions that may modulate disease risk (e.g., use of ANNs to explore interactions between nutrient intake and metabolic pathway polymorphisms in breast cancer susceptibility (132)).

Geospatial applications

Machine learning can help to predict and map disease occurrence and health indicators in areas where data are limited. Its ability to efficiently process high-dimensional data sets from heterogeneous contexts and multiple geographic scales makes it particularly suitable for this task. A major focus is the development of the WorldPop Project (www.worldpop.org), which is an open-source archive of demographic parameters on fine spatial scales (133). It uses random forests to map the global population distribution on a per-pixel scale by combining remote sensing data (e.g., satellite) across multiple geospatial scales (133). Beyond WorldPop's use of random forests, another type of ensemble machine learning algorithm, boosted regression trees, has also been widely used to map environmental suitability for disease transmission, including dengue (134), leishmaniasis (135), Ebola (136), Crimean-Congo hemorrhagic fever (137), and Zika virus (138, 139). In general, investigators in these studies 1) chose a set of known or proposed environmental and socioeconomic covariates, 2) incorporated global assessments regarding whether the disease(s) of interest was circulating in the country or region, and 3) with these data, built boosted regression tree models. The resulting models were used to predict infection probabilities on a pixel-by-pixel scale.

Text mining

Electronic health records provide an unprecedented amount of clinical information for research, but utilization of these data sources effectively in studies or for surveillance is generally cost-prohibitive without some form of automated data extraction. Machine learning offers automated tools for extracting unstructured information from textual clinical documents. For example, i2b2 (Informatics for Integrating Biology and the Bedside) Challenges address a range of projects aiming to develop and evaluate information extraction methods for clinical text (140). The 2009 Medication Challenge focused on providing a schema with which to extract information including medications, dosages, modes (routes) of administration, frequencies, durations, and reasons for administration from discharge summaries (141).

Other applications include deidentifying personal health information, research subject recruitment, coding, and surveillance. Machine learning has been used to remove personal health information from clinical records, such that deidentified records may be made public for research purposes without obtaining individual informed consent (142). Studies have also used textual data and machine learning algorithms to identify patients who may qualify for and benefit from participation in clinical studies (143). Furthermore, text mining can improve the efficiency of systematic reviews by facilitating the identification, rapid categorization, and summarization of relevant literature (144). Finally, natural language processing of clinical documents can supplement manual surveillance and has been used to identify a range of reportable postoperative complications (145).

In addition to clinical settings, text mining algorithms have been incorporated into automated infectious disease surveillance systems that acquire, classify, and process Web-accessible data. These algorithms can improve detection of early outbreaks and complement traditional surveillance efforts performed by government and international organizations. For example, HealthMap graphically displays areas where diseases are circulating by combining search query data, social media data, validated official

reports, and expert-curated accounts (e.g., ProMED-mail) (146, 147). Similarly, the BioCaster system tracks infectious disease outbreaks on Google maps (Google, Inc., Mountain View, California) on the basis of residual sum-of-squares feeds (148).

Prediction and forecasting of infectious disease

Machine learning methods have been incorporated into prediction and forecasting models for infectious disease. For example, SVMs have been used to predict whether dengue incidence exceeded a chosen threshold using Google search terms (149). Researchers have also used SVMs to predict levels of influenza-like illness from Twitter data (Twitter, Inc., San Francisco, California) 1–2 weeks before official reports (150). In addition, infectious disease forecasters have adopted ensemble-based methods traditionally used for meteorological and oceanographic predictions. For example, climate forecasting from multimodel ensembles has been adapted to produce early malaria warning systems (151). Moreover, ensemble-based forecasting methods based on sequential data assimilation approaches are increasingly common infectious disease forecasting tools, because of their ability to correct for various sources of uncertainty in mathematical simulations as compared with traditional linear time-series models such as negative binomial models and autoregressive integrated moving average (ARIMA) models. One type of sequential ensemble filtering, the ensemble adjustment Kalman filter, has been used to forecast seasonal outbreaks of influenza (152), to reconstruct the transmission network of the 2014–2015 Ebola epidemic in Sierra Leone (153), and to retrospectively "forecast" cases of West Nile virus (154) and respiratory syncytial virus (155).

BRIEF RECOMMENDATIONS

In this primer, we have discussed several important algorithms, but this is only the tip of the iceberg. We refer readers to the machine learning textbooks referenced herein for a more comprehensive review (2, 5, 31). The choice of an algorithm is highly tied to the research goals associated with its use, and there is no single recommendation for all projects. However, epidemiologists interested in adopting machine learning methodologies will often be most interested in accurate prediction in the context of a large number of covariates. In these cases, we encourage them to start with ensemble-based boosting or bagging approaches. Through refitting the same underlying model to different versions of a data set, these ensembles are less susceptible to overfitting and less sensitive to tuning parameters. They are also easy to implement with many commonly available tools and packages, with random forests analysis being a popular choice. The Super Learner approach, which fits many different models to a data set, is also attractive since it allows simultaneous consideration of multiple algorithms and automates many of the best practices for fitting and validating machine learning models. However, as with traditional epidemiologic or statistical approaches, a rigorous approach to assessing performance and appropriate matching of a model to its use are more important than the specific algorithm used.

Despite the benefits of boosting and bagging, as a general rule, these ensemble approaches add another stage to modeling, making their results harder to interpret. Investigators should carefully consider their primary objective: Is it predictive accuracy or interpretability? Where interpretability is important, as in many clinical applications, researchers might consider single, more easily understandable algorithms such as decision trees. However, many machine learning algorithms, particularly nonensemble approaches, are prone to overfitting.

Measures of fit alone (e.g., R^2) should be interpreted with caution, as they can be effectively meaningless for some machine learning applications (156). Without a likelihood function, techniques such as Akaike Information Criterion evaluation are not available metrics for assessing the generalizability of machine learning models; hence, cross-validation (whether *k*-fold, leave-one-out, or another approach) is a critical tool for evaluating model performance. These methods must be used appropriately, however, or they can fool the researcher. The testing and validation plan should be specified a priori and must be applied to the full algorithm: For example, if there is a data-based variable selection step, it should be executed in each data partition used in cross-validation, not in the full data set prior to the cross-validation. It is important that researchers understand clearly that these cross-validation approaches give expected out-of-data-set performance given the algorithm used, not an assessment of the particular fitted model, and that they recognize that the quality of these measures depends on the representativeness of the population and the correlation between observations in the training and testing sets (i.e., if there is high correlation, cross-validated performance will be deceptively high).

OPPORTUNITIES AND CHALLENGES

The field of machine learning is rapidly developing and can make any technical review seem obsolete within months. Growing interest in the field from the general public, as reflected in extensive coverage of self-driving cars and AlphaGo (Alphabet, Inc., Mountain View, California) in the mainstream media, is accompanied by efforts from the machine learning community to make advanced machine learning technologies more accessible. Educational companies such as Udacity (Udacity, Inc., Mountain View, California) and Coursera (Coursera, Inc., Mountain View, California) have partnered with companies like Google and academic institutions to create online and freely available courses on machine learning and deep learning.

In addition to the growing educational resources, large technological companies, including Google, IBM (International Business Machines Corporation, Armonk, New York), and Amazon Web Services (Amazon Web Services, Inc., Seattle, Washington), are heavily investing in open-source machine learning that uses data-flow graphs to build models (e.g., TensorFlow (Google, Inc.) (157)). The use of data-flow graphs in TensorFlow enables developers and data scientists to focus on the high-level overall logic of the algorithms rather than the technical coding details, which greatly increases the reproducibility and optimizability of the models. Models built with TensorFlow can be integrated into mobile devices, making on-device/bedside diagnosis practical when combined with mobile sensors. The ability of TensorFlow to build and run models on the cloud also dramatically increases processing power and storage ability, which is particularly helpful for analyzing large data sets with complex

algorithms. These machine learning developments continue to ease the entry barriers for epidemiologists interested in using advanced machine learning technologies, and they have the potential to transform epidemiologic research.

Yet, there continue to be challenges that impede greater integration of machine learning into epidemiologic research. Classically trained epidemiologists often lack the skills to take full advantage of machine learning technologies, partly because of the continued popularity of closed-source programming languages (e.g., SAS, Stata) in epidemiology. In addition, despite the promise of "Big Data," logistical roadblocks to sharing de-identified patient data and amassing large health-care data sets can make it challenging for epidemiologists to leverage these opportunities, particularly compared with the private sector. Even when data are available, epidemiologists should be mindful of the class-imbalance issue (see Table 1) often inherent in health-care and surveillance data, which can pose challenges for many standard algorithms (158). Most importantly, a general lack of working knowledge on machine learning algorithms, despite their substantial methodological overlap with statistical methods, reduces the practical uptake of these techniques in the epidemiologic literature.

Ultimately, advanced machine learning algorithms offer epidemiologists new tools for tackling problems that classical methods are not well-suited for, but they by no means serve as a cure-all for poor study design or poor data quality. Further eroding the cultural and language barriers between machine learning and epidemiology serves as an essential first step toward understanding the value of, and achieving greater integration with, machine learning and existing epidemiologic research methods.

ACKNOWLEDGMENTS

Author affiliations: Department of Epidemiology, Bloomberg School of Public Health, Johns Hopkins University, Baltimore, Maryland (Qifang Bi, Katherine E. Goodman, Joshua Kaminsky, Justin Lessler).

Q.B. and K.E.G. contributed equally to this paper.

We thank Dr. Maria Glymour for her very helpful suggestions.

Conflict of interest: none declared.

REFERENCES

1. Samuel AL. Some studies in machine learning using the game of checkers. *IBM J Res Dev*. 1959;3(3):210–229.
2. Mitchell TM. *Machine Learning*. 1st ed. New York, NY: McGraw-Hill Education; 1997.
3. Rasmussen CE, Williams CKI. *Gaussian Processes for Machine Learning*. 1st ed. Cambridge, MA: MIT Press; 2006.
4. Breiman L. Statistical modeling: the two cultures (with comments and a rejoinder by the author). *Stat Sci*. 2001;16(3):199–231.
5. Duda RO, Hart PE, Stork DG. *Pattern Classification*. 2nd ed. Hoboken, NJ: John Wiley & Sons, Inc.; 2012:517.
6. Bartholomew DJ, Knott M, Moustaki I. *Latent Variable Models and Factor Analysis: A Unified Approach*. 3rd ed. Hoboken, NJ: John Wiley & Sons, Inc.; 2011.
7. Hennig C, Meila M, Murtagh F, et al. *Handbook of Cluster Analysis*. 1st ed. Boca Raton, FL: CRC Press; 2015:34.
8. Bishop CM. *Pattern Recognition and Machine Learning*. 1st ed. New York, NY: Springer Publishing Compnay; 2006:424.
9. Zhu X, Goldberg AB. *Introduction to Semi-Supervised Learning*. 1st ed. San Rafael, CA: Morgan & Claypool Publishers; 2009:11.
10. Nigam K, McCallum AK, Thrun S, et al. Text classification from labeled and unlabeled documents using EM. *Mach Learn*. 2000;39(2):103–134.
11. Ng AY, Jordan MI. On discriminative vs. generative classifiers: a comparison of logistic regression and naive Bayes. In: Dietterich TG, Becker S, Ghahramani Z, eds. *Advances in Neural Information Processing Systems 14*. Cambridge, MA: MIT Press; 2002:841–848.
12. Vapnik VN. *Statistical Learning Theory*. 1st ed. Hoboken, NJ: Wiley-Interscience; 1998:12–21.
13. Pernkopf F, Bilmes J. Discriminative versus generative parameter and structure learning of Bayesian network classifiers. In: *Proceedings of the 22nd International Conference on Machine Learning—ICML '05*. New York, NY: Association for Computing Machinery; 2005:657–664. https://dl.acm.org/citation.cfm?id=1102434. Accessed July 4, 2019.
14. *Reinforcement Learning: An Introduction*—Richard S. Sutton and Andrew G. Bartow [book review]. *IEEE Trans Neural Netw*. 1998;9(5):1054.
15. RS Sutton, AG Barto. *Reinforcement Learning: An Introduction*. 2nd ed. Cambridge, MA: MIT Press; 2018.
16. Ganguly K. *Learning Generative Adversarial Networks*. 1st ed. Birmingham, United Kingdom: Packt Publishing; 2017.
17. Asoh H, Shiro M, Akaho S, et al. An application of inverse reinforcement learning to medical records of diabetes treatment. Presented at the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD), Prague, Czech Republic, September 23–27, 2013. https://pdfs.semanticscholar.org/5f44/548a3cd1932fc5035236b9be9018df5103c5.pdf. Accessed July 3, 2019.
18. Shortreed SM, Laber E, Lizotte DJ, et al. Informing sequential clinical decision-making through reinforcement learning: an empirical study. *Mach Learn*. 2011;84(1-2):109–136.
19. Nemati S, Ghassemi MM, Clifford GD. Optimal medication dosing from suboptimal clinical examples: a deep reinforcement learning approach. *Conf Proc IEEE Eng Med Biol Soc*. 2016;2016:2978–2981.
20. Olden JD, Jackson DA. Illuminating the "black box": a randomization approach for understanding variable contributions in artificial neural networks. *Ecol Modell*. 2002;154(1-2):135–150.
21. Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. *Nature*. 1986;323(6088):533–536.
22. McCulloch WS, Pitts W. A logical calculus of the ideas immanent in nervous activity. *Bull Math Biol*. 1990;52(1-2):99–115. [Reprinted from *Bull Math Biophys*. 1943;5:115–133].
23. Duh MS, Walker AM, Ayanian JZ. Epidemiologic interpretation of artificial neural networks. *Am J Epidemiol*. 1998;147(12):1112–1122.

24. Papadokonstantakis S, Lygeros A, Jacobsson SP. Comparison of recent methods for inference of variable influence in neural networks. *Neural Netw*. 2006;19(4):500–513.
25. Beck MW. Variable importance in neural networks. In: *R-Bloggers*. https://www.r-bloggers.com/variable-importance-in-neural-networks/. Published August 12, 2013. Accessed June 19, 2018.
26. Hershey S, Chaudhuri S, Ellis DPW, et al. CNN architectures for large-scale audio classification. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2017)*. Piscataway, NJ: Institute of Electrical and Electronics Engineers; 2017:131–135. https://ai.google/research/pubs/pub45611. Accessed July 15, 2019.
27. Breiman L, Friedman J, Stone CJ, et al. *Classification and Regression Trees*. 1st ed. London, United Kingdom: Chapman & Hall Ltd.; 1984.
28. Kass GV. An exploratory technique for investigating large quantities of categorical data. *Appl Stat*. 1980;29(2):119–127.
29. Biggs D, De Ville B, Suen E. A method of choosing multiway partitions for classification and decision trees. *J Appl Stat*. 1991;18(1):49–62.
30. Quinlan JR. Induction of decision trees. *Mach Learn*. 1986;1(1):81–106.
31. James G, Witten D, Hastie T, et al. *An Introduction to Statistical Learning: With Applications in R*. New York, NY: Springer Publishing Company; 2013.
32. Almuallim H. An efficient algorithm for optimal pruning of decision trees. *Artif Intell*. 1996;83(2):347–362.
33. Boulesteix AL, Janitza S, Hapfelmeier A, et al. Letter to the editor: on the term "interaction" and related phrases in the literature on random forests [letter]. *Brief Bioinform*. 2015;16(2):338–345.
34. Aluja-Banet T, Nafria E. Stability and scalability in decision trees. *Comput Stat*. 2003;18(3):505–520.
35. Boser BE, Guyon IM, Vapnik VN. A training algorithm for optimal margin classifiers. In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. New York, NY: Association for Computing Machinery; 1992:144–152. http://www.svms.org/training/BOGV92.pdf. Accessed July 3, 2019.
36. Noble WS. What is a support vector machine? *Nat Biotechnol*. 2006;24(12):1565–1567.
37. Smola AJ, Schölkopf B. A tutorial on support vector regression. *Stat Comput*. 2004;14(3):199–222.
38. Belousov AI, Verzakov SA, von Frese J. A flexible classification approach with optimal generalisation performance: support vector machines. *Chemometr Intell Lab Syst*. 2002;64(1):15–25.
39. Guenther N, Schonlau M. Support vector machines. *Stata J*. 2016;16(4):917–937.
40. Lewis DD. Naive (Bayes) at forty: the independence assumption in information retrieval. In: Nédellec C, Rouveirol C, eds. *Machine Learning: ECML-98*. Berlin, Germany: Springer Berlin-Heidelberg; 1998:4–15.
41. Frank E, Trigg L, Holmes G, et al. Technical note: naive Bayes for regression. *Mach Learn*. 2000;41(1):5–25.
42. Rish I. An empirical study of the naive Bayes classifier. In: Hoos HH, Stützle T, eds. *Proceedings of the IJCAI-01 Workshop on Empirical Methods in Artificial Intelligence*. Stanford, CA: International Joint Conferences on Artificial Intelligence Organization; 2001:41–46. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.330.2788&rep=rep1&type=pdf. Accessed July 3, 2019.
43. Russek E, Kronmal RA, Fisher LD. The effect of assuming independence in applying Bayes' theorem to risk estimation and classification in diagnosis. *Comput Biomed Res*. 1983;16(6):537–552.
44. Hand DJ. Statistical methods in diagnosis. *Stat Methods Med Res*. 1992;1(1):49–67.
45. Stan Development Team. Naive Bayes classification and clustering. In: *Stan User's Guide. Version 2.19*. https://mc-stan.org/docs/2_19/stan-users-guide/naive-bayes-classification-and-clustering.html. Accessed July 20, 2019.
46. Jain AK. Data clustering: 50 years beyond K-means. *Pattern Recognit Lett*. 2010;31(8):651–666.
47. Lloyd S. Least squares quantization in PCM. *IEEE Trans Inf Theory*. 1982;28(2):129–137.
48. Pham DT, Dimov SS, Nguyen CD. Selection of K in K-means clustering. *Proc Inst Mech Eng Pt C J Mechan Eng Sci*. 2005;219(1):103–119.
49. Tibshirani R, Walther G, Hastie T. Estimating the number of clusters in a data set via the gap statistic. *J R Stat Soc Series B Stat Methodol*. 2001;63(2):411–423.
50. Raykov YP, Boukouvalas A, Baig F, et al. What to do when K-means clustering fails: a simple yet principled alternative algorithm. *PLoS One*. 2016;11(9):e0162259.
51. Breiman L. Bagging predictors. *Mach Learn*. 1996;24(2):123–140.
52. Schapire RE, Freund Y, Bartlett P, et al. Boosting the margin: a new explanation for the effectiveness of voting methods. *Ann Stat*. 1998;26(5):1651–1686.
53. Breiman L. *Bias, Variance, and Arcing Classifiers*. (Technical report 460). Berkeley, CA: Department of Statistics, University of California, Berkeley; 1996. http://www.stat.berkeley.edu/~breiman/arcall96.pdf. Accessed July 3, 2019.
54. Breiman L. Random forests. *Mach Learn*. 2001;45(1):5–32.
55. Breiman L. *Manual on Setting Up, Using, and Understanding Random Forests V3.1*. Berkeley, CA: Department of Statistics, University of California, Berkeley; 2002. https://www.stat.berkeley.edu/~breiman/Using_random_forests_V3.1.pdf. Accessed July 3, 2019.
56. Friedman J, Hastie T, Tibshirani R. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. New York, NY: Springer Publishing Company; 2009.
57. van der Laan MJ. Statistical inference for variable importance. *Int J Biostat*. 2006;2(1):1557–4679.
58. Maldonado M, Dean J, Czika W, et al. Leveraging ensemble models in SAS® Enterprise MinerTM. (Paper SAS133-2014). In: *Proceedings of the SAS® Global Forum 2014 Conference*. Cary, NC: SAS Institute, Inc.; 2014. https://support.sas.com/resources/papers/proceedings14/SAS133-2014.pdf. Accessed July 3, 2019.
59. Schapire RE. The strength of weak learnability. *Mach Learn*. 1990;5(2):197–227.
60. Freund Y. Boosting a weak learning algorithm by majority. *Inf Comput*. 1995;121(2):256–285.
61. Schapire RE. A brief introduction to boosting. In: *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*. Stanford, CA: International Joint Conferences on Artificial Intelligence Organization; 1999:1401–1406. http://rob.schapire.net/papers/Schapire99c.pdf. Accessed July 3, 2019.
62. Freund Y, Schapire RE. A desicion-theoretic generalization of on-line learning and an application to boosting. *J Comput Syst Sci*. 1997;55(1):119–139.
63. Friedman JH. Greedy function approximation: a gradient boosting machine. *Ann Stat*. 2001;29(5):1189–1232.
64. Friedman JH. Stochastic gradient boosting. *Comput Stat Data Anal*. 2002;38(4):367–378.

65. Schonlau M. Boosted regression (boosting): an introductory tutorial and a Stata plugin. *Stata J*. 2005;5(3):330–354.
66. Hoeting JA, Madigan D, Raftery AE, et al. Bayesian model averaging: a tutorial. *Stat Sci*. 1999;14(4):382–417.
67. Domingos P. Bayesian averaging of classifiers and the overfitting problem. In: *Proceedings of the 17th International Conference on Machine Learning*. Burlington, MA: Morgan Kaufman Publishers; 2000:223–230. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.34.9147. Accessed August 27, 2017.
68. Monteith K, Carroll JL, Seppi K, et al. Turning Bayesian model averaging into Bayesian model combination. In: *The 2011 International Joint Conference on Neural Networks (IJCNN 2011—San Jose)*. Piscataway, NJ: Institute of Electrical and Electronics Engineers; 2011:2657–2663. http://axon.cs.byu.edu/papers/Kristine.ijcnn2011.pdf. Accessed July 3, 2019.
69. Whitney M, Ngo L. Bayesian model averaging using SAS® software. (Paper 203-29). In: *Proceedings of the Twenty-Ninth Annual SAS® Users Group International Conference*. Cary, NC: SAS Institute, Inc.; 2004:203–229. http://www2.sas.com/proceedings/sugi29/203-29.pdf. Accessed July 3, 2019.
70. van der Laan MJ, Polley EC, Hubbard AE. Super Learner. *Stat Appl Genet Mol Biol*. 2007;6:Article 25.
71. Sinisi SE, Polley EC, Petersen ML, et al. Super learning: an application to the prediction of HIV-1 drug resistance. *Stat Appl Genet Mol Biol*. 2007;6:Article 7.
72. Wolpert DH. Stacked generalization. *Neural Netw*. 1992;5(2):241–259.
73. Hernán MA, Robins JM. Using big data to emulate a target trial when a randomized trial is not available. *Am J Epidemiol*. 2016;183(8):758–764.
74. Rosenbaum PR, Rubin DB. Reducing bias in observational studies using subclassification on the propensity score. *J Am Stat Assoc*. 1984;79(387):516–524.
75. Westreich D, Lessler J, Funk MJ. Propensity score estimation: neural networks, support vector machines, decision trees (CART), and meta-classifiers as alternatives to logistic regression. *J Clin Epidemiol*. 2010;63(8):826–833.
76. Lee BK, Lessler J, Stuart EA. Improving propensity score weighting using machine learning. *Stat Med*. 2010;29(3):337–346.
77. Pirracchio R, Petersen ML, van der Laan M. Improving propensity score estimators' robustness to model misspecification using Super Learner. *Am J Epidemiol*. 2015;181(2):108–119.
78. Watkins S, Jonsson-Funk M, Brookhart MA, et al. An empirical comparison of tree-based methods for propensity score estimation. *Health Serv Res*. 2013;48(5):1798–1817.
79. Schnitzer ME, Lok JJ, Gruber S. Variable selection for confounder control, flexible modeling and collaborative targeted minimum loss-based estimation in causal inference. *Int J Biostat*. 2016;12(1):97–115.
80. Moodie EEM, Stephens DA. Treatment prediction, balance, and propensity score adjustment. *Epidemiology*. 2017;28(5):e51–e53.
81. Bahamyirou A, Blais L, Forget A, et al. Understanding and diagnosing the potential for bias when using machine learning methods with doubly robust causal estimators. *Stat Methods Med Res*. 2019;28(6):1637–1650.
82. Kugler KC, Vasilenko SA, Butera NM, et al. Long-term consequences of early sexual initiation on young adult health: a causal inference approach. *J Early Adolesc*. 2017;37(5):662–676.
83. Oppermann M, Fritzsche J, Weber-Schoendorfer C, et al. A(H1N1)v2009: a controlled observational prospective cohort study on vaccine safety in pregnancy. *Vaccine*. 2012;30(30):4445–4452.
84. Tamma PD, Turnbull AE, Harris AD, et al. Less is more: combination antibiotic therapy for the treatment of gram-negative bacteremia in pediatric patients. *JAMA Pediatr*. 2013;167(10):903–910.
85. Schuler MS, Rose S. Targeted maximum likelihood estimation for causal inference in observational studies. *Am J Epidemiol*. 2017;185(1):65–73.
86. Ahern J, Balzer L, Galea S. The roles of outlet density and norms in alcohol use disorder. *Drug Alcohol Depend*. 2015;151:144–150.
87. Ahern J, Karasek D, Luedtke AR, et al. Racial/ethnic differences in the role of childhood adversities for mental disorders among a nationally representative sample of adolescents. *Epidemiology*. 2016;27(5):697–704.
88. Athey S, Imbens G. Recursive partitioning for heterogeneous causal effects. *Proc Natl Acad Sci U S A*. 2016;113(27):7353–7360.
89. Wager S, Athey S. Estimation and inference of heterogeneous treatment effects using random forests. *J Am Stat Assoc*. 2018;113(523):1228–1242.
90. Scutari M, Denis JB. *Bayesian Networks: With Examples in R*. 1st ed. London, United Kingdom: Chapman & Hall Ltd.; 2014.
91. Shimizu S, Hoyer PO, Hyvärinen A, et al. A linear non-gaussian acyclic model for causal discovery. *J Mach Learn Res*. 2006;7:2003–2030.
92. Hoyer PO, Shimizu S, Kerminen AJ, et al. Estimation of causal effects using linear non-Gaussian causal models with hidden variables. *Int J Approx Reason*. 2008;49(2):362–378.
93. Shimizu S. LiNGAM: non-Gaussian methods for estimating causal structures. *Behaviormetrika*. 2014;41(1):65–98.
94. Sachs K, Perez O, Pe'er D, et al. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*. 2005;308(5721):523–529.
95. Schadt EE, Lamb J, Yang X, et al. An integrative genomics approach to infer causal associations between gene expression and disease. *Nat Genet*. 2005;37(7):710–717.
96. Rosenström T, Jokela M, Puttonen S, et al. Pairwise measures of causal direction in the epidemiology of sleep problems and depression. *PLoS One*. 2012;7(11):e50841.
97. Helajärvi H, Rosenström T, Pahkala K, et al. Exploring causality between TV viewing and weight change in young and middle-aged adults. The Cardiovascular Risk in Young Finns Study. *PLoS One*. 2014;9(7):e101860.
98. Warner HR, Toronto AF, Veasey LG, et al. A mathematical approach to medical diagnosis. Application to congenital heart disease. *JAMA*. 1961;177:177–183.
99. Blumenthal DT, Artzi M, Liberman G, et al. Classification of high-grade glioma into tumor and nontumor components using support vector machine. *AJNR Am J Neuroradiol*. 2017;38(5):908–914.
100. Artzi M, Liberman G, Nadav G, et al. Differentiation between treatment-related changes and progressive disease in patients with high grade brain tumors using support vector machine classification based on DCE MRI. *J Neurooncol*. 2016;127(3):515–524.
101. Zarinabad N, Abernethy LJ, Avula S, et al. Application of pattern recognition techniques for classification of pediatric brain tumors by in vivo 3T ^{1}H-MR spectroscopy—a multi-center study. *Magn Reson Med*. 2018;79(4):2359–2366.

102. Chang Y, Paul AK, Kim N, et al. Computer-aided diagnosis for classifying benign versus malignant thyroid nodules based on ultrasound images: a comparison with radiologist-based assessments. *Med Phys*. 2016;43(1):554–567.
103. El-Naqa I, Yang Y, Wernick MN, et al. A support vector machine approach for detection of microcalcifications. *IEEE Trans Med Imaging*. 2002;21(12):1552–1563.
104. Polat K, Güneş S. Breast cancer diagnosis using least square support vector machine. *Digit Signal Process*. 2007;17(4): 694–701.
105. Akay MF. Support vector machines combined with feature selection for breast cancer diagnosis. *Expert Syst Appl*. 2009; 36(2):3240–3247.
106. Wang ZL, Zhou ZG, Chen Y, et al. Support vector machines model of computed tomography for assessing lymph node metastasis in esophageal cancer with neoadjuvant chemotherapy. *J Comput Assist Tomogr*. 2017;41(3): 455–460.
107. Zhang XP, Wang ZL, Tang L, et al. Support vector machine model for diagnosis of lymph node metastasis in gastric cancer with multidetector computed tomography: a preliminary study. *BMC Cancer*. 2011;11:Article 10.
108. Brown MP, Grundy WN, Lin D, et al. Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc Natl Acad Sci U S A*. 2000;97(1): 262–267.
109. Furey TS, Cristianini N, Duffy N, et al. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*. 2000; 16(10):906–914.
110. Golub TR, Slonim DK, Tamayo P, et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*. 1999;286(5439): 531–537.
111. Pomeroy SL, Tamayo P, Gaasenbeek M, et al. Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*. 2002;415(6870):436–442.
112. Orrù G, Pettersson-Yeo W, Marquand AF, et al. Using Support Vector Machine to identify imaging biomarkers of neurological and psychiatric disease: a critical review. *Neurosci Biobehav Rev*. 2012;36(4):1140–1152.
113. Costafreda SG, Chu C, Ashburner J, et al. Prognostic and diagnostic potential of the structural neuroanatomy of depression. *PLoS One*. 2009;4(7):e6353.
114. Costafreda SG, Khanna A, Mourao-Miranda J, et al. Neural correlates of sad faces predict clinical remission to cognitive behavioural therapy in depression. *Neuroreport*. 2009;20(7): 637–641.
115. Gong Q, Wu Q, Scarpazza C, et al. Prognostic prediction of therapeutic response in depression using high-field MR imaging. *Neuroimage*. 2011;55(4):1497–1503.
116. Palaniappan S, Awang R. Intelligent heart disease prediction system using data mining techniques. *IJCSNS Int J Comput Sci Netw Secur*. 2008;8(8):343–350.
117. Jaimes F, Farbiarz J, Alvarez D, et al. Comparison between logistic regression and neural networks to predict death in patients with suspected sepsis in the emergency room. *Crit Care*. 2005;9(2):R150–R156.
118. Launay CP, Rivière H, Kabeshova A, et al. Predicting prolonged length of hospital stay in older emergency department users: use of a novel analysis method, the artificial neural network. *Eur J Intern Med*. 2015;26(7):478–482.
119. Demšar J, Zupan B, Aoki N, et al. Feature mining and predictive model construction from severe trauma patient's data. *Int J Med Inform*. 2001;63(1-2):41–50.
120. Levin S, Toerper M, Hamrock E, et al. Machine-learning-based electronic triage more accurately differentiates patients with respect to clinical outcomes compared with the emergency severity index. *Ann Emerg Med*. 2018;71(5): 565–574.e2.
121. Brims FJH, Meniawy TM, Duffus I, et al. A novel clinical prediction model for prognosis in malignant pleural mesothelioma using decision tree analysis. *J Thorac Oncol*. 2016;11(4):573–582.
122. Goodman KE, Lessler J, Cosgrove SE, et al. A clinical decision tree to predict whether a bacteremic patient is infected with an extended-spectrum β-lactamase-producing organism. *Clin Infect Dis*. 2016;63(7):896–903.
123. Dias CC, Pereira Rodrigues P, Fernandes S, et al. The risk of disabling, surgery and reoperation in Crohn's disease—a decision tree-based approach to prognosis. *PLoS One*. 2017; 12(2):e0172165.
124. Silva-Alves MS, Secolin R, Carvalho BS, et al. A prediction algorithm for drug response in patients with mesial temporal lobe epilepsy based on clinical and genetic information. *PLoS One*. 2017;12(1):e0169214.
125. Nguyen TT, Huang J, Wu Q, et al. Genome-wide association data classification and SNPs selection using two-stage quality-based random forests. *BMC Genomics*. 2015;16(suppl 2):Article S5.
126. Briones N, Dinu V. Data mining of high density genomic variant data for prediction of Alzheimer's disease risk. *BMC Med Genet*. 2012;13:Article 7.
127. Wei W, Visweswaran S, Cooper GF. The application of naive Bayes model averaging to predict Alzheimer's disease from genome-wide data. *J Am Med Inform Assoc*. 2011;18(4): 370–375.
128. Altman NS. An introduction to kernel and nearest-neighbor nonparametric regression. *Am Stat*. 1992;46(3):175–185.
129. Kim BJ, Kim SH. Prediction of inherited genomic susceptibility to 20 common cancer types by a supervised machine-learning method. *Proc Natl Acad Sci U S A*. 2018; 115(6):1322–1327.
130. Montassier E, Al-Ghalith GA, Ward T, et al. Pretreatment gut microbiome predicts chemotherapy-related bloodstream infection. *Genome Med*. 2016;8:Article 49.
131. Upstill-Goddard R, Eccles D, Fliege J, et al. Machine learning approaches for the discovery of gene-gene interactions in disease data. *Brief Bioinform*. 2013;14(2):251–260.
132. Naushad SM, Janaki Ramaiah M, Pavithrakumari M, et al. Artificial neural network-based exploration of gene-nutrient interactions in folate and xenobiotic metabolic pathways that modulate susceptibility to breast cancer. *Gene*. 2016;580(2): 159–168.
133. Stevens FR, Gaughan AE, Linard C, et al. Disaggregating census data for population mapping using random forests with remotely-sensed and ancillary data. *PLoS One*. 2015; 10(2):e0107042.
134. Bhatt S, Gething PW, Brady OJ, et al. The global distribution and burden of dengue. *Nature*. 2013;496(7446):504–507.
135. Pigott DM, Bhatt S, Golding N, et al. Global distribution maps of the leishmaniases. *ELife*. 2014;3:e02851.
136. Pigott DM, Golding N, Mylne A, et al. Mapping the zoonotic niche of Ebola virus disease in Africa. *ELife*. 2014;3:e04395.
137. Messina JP, Pigott DM, Golding N, et al. The global distribution of Crimean-Congo hemorrhagic fever. *Trans R Soc Trop Med Hyg*. 2015;109(8):503–513.
138. Messina JP, Kraemer MU, Brady OJ, et al. Mapping global environmental suitability for Zika virus. *ELife*. 2016;5: pii:15272.

139. Perkins TA, Siraj AS, Ruktanonchai CW, et al. Model-based projections of Zika virus infections in childbearing women in the Americas. *Nat Microbiol*. 2016;1(9):16126.
140. i2b2 tranSMART Foundation. i2b2: Informatics for Integrating Biology & the Bedside. Overview. https://www.i2b2.org/about/index.html. Accessed May 20, 2018.
141. Uzuner O, Solti I, Cadag E. Extracting medication information from clinical text. *J Am Med Inform Assoc*. 2010;17(5):514–518.
142. Meystre SM, Friedlin FJ, South BR, et al. Automatic de-identification of textual documents in the electronic health record: a review of recent research. *BMC Med Res Methodol*. 2010;10:Article 70.
143. Pakhomov S, Weston SA, Jacobsen SJ, et al. Electronic medical records for clinical research: application to the identification of heart failure. *Am J Manag Care*. 2007;13(6):281–288.
144. Thomas J, McNaught J, Ananiadou S. Applications of text mining within systematic reviews. *Res Synth Methods*. 2011;2(1):1–14.
145. Murff HJ, FitzHenry F, Matheny ME, et al. Automated identification of postoperative complications within an electronic medical record using natural language processing. *JAMA*. 2011;306(8):848–855.
146. Brownstein JS, Freifeld CC, Reis BY, et al. Surveillance sans frontières: Internet-based emerging infectious disease intelligence and the HealthMap project. *PLoS Med*. 2008;5(7):e151.
147. Freifeld CC, Mandl KD, Reis BY, et al. HealthMap: global infectious disease monitoring through automated classification and visualization of Internet media reports. *J Am Med Inform Assoc*. 2008;15(2):150–157.
148. Collier N, Doan S, Kawazoe A, et al. BioCaster: detecting public health rumors with a Web-based text mining system. *Bioinformatics*. 2008;24(24):2940–2941.
149. Althouse BM, Ng YY, Cummings DA. Prediction of dengue incidence using search query surveillance. *PLoS Negl Trop Dis*. 2011;5(8):e1258.
150. Signorini A, Segre AM, Polgreen PM. The use of Twitter to track levels of disease activity and public concern in the US during the influenza A H1N1 pandemic. *PLoS One*. 2011;6(5):e19467.
151. Thomson MC, Doblas-Reyes FJ, Mason SJ, et al. Malaria early warnings based on seasonal climate forecasts from multi-model ensembles. *Nature*. 2006;439(7076):576–579.
152. Shaman J, Karspeck A. Forecasting seasonal outbreaks of influenza. *Proc Natl Acad Sci U S A*. 2012;109(50):20425–20430.
153. Yang W, Zhang W, Kargbo D, et al. Transmission network of the 2014–2015 Ebola epidemic in Sierra Leone. *J R Soc Interface*. 2015;12(112):20150536.
154. DeFelice NB, Little E, Campbell SR, et al. Ensemble forecast of human West Nile virus cases and mosquito infection rates. *Nat Commun*. 2017;8:14592.
155. Reis J, Shaman J. Retrospective parameter estimation and forecast of respiratory syncytial virus in the United States. *PLoS Comput Biol*. 2016;12(10):e1005133.
156. Mountford MD. *Principles and Procedures of Statistics with Special Reference to the Biological Sciences* by R. G. D. Steel, J. H. Torrie [book review]. *Biometrics*. 1962;18(1):127.
157. Abadi M, Agarwal A, Barham P, et al. TensorFlow: large-scale machine learning on heterogeneous distributed systems. *arXiv*. 2015. (doi: arXiv:1603.04467). Accessed September 10, 2019.
158. Khalilia M, Chakraborty S, Popescu M. Predicting disease risks from highly imbalanced data using random forest. *BMC Med Inform Decis Mak*. 2011;11:Article 51.
159. Jain AK, Mao J, Mohiuddin KM. Artificial neural networks: a tutorial. *Computer*. 1996;29(3):31–44.
160. Olden JD, Lawler JJ, Poff NL. Machine learning methods without tears: a primer for ecologists. *Q Rev Biol*. 2008;83(2):171–193.
161. Therneau TM, Atkinson EJ. *An Introduction to Recursive Partitioning Using the RPART Routines*. Rochester, MN: Division of Biomedical Statistics and Informatics, Mayo Clinic; 2018. https://cran.r-project.org/web/packages/rpart/vignettes/longintro.pdf. Accessed July 3, 2019.
162. Polley EC, van der Laan MJ. *Super Learner in Prediction*. (U.C. Berkeley Division of Biostatistics Working Paper Series, working paper 266). Berkeley, CA: Division of Biostatistics, University of California, Berkeley; 2010. http://biostats.bepress.com/ucbbiostat/paper266/. Accessed May 20, 2018.
163. Opitz D, Maclin R. Popular ensemble methods: an empirical study. *J Artif Intell Res*. 1999;11:169–198.
164. Markham K. In-depth introduction to machine learning in 15 hours of expert videos. In: *R-Bloggers*. https://www.r-bloggers.com/in-depth-introduction-to-machine-learning-in-15-hours-of-expert-videos/. Published September 23, 2014. Accessed July 3, 2019.

(Appendix follows)

APPENDIX

Further Reading: Machine Learning Resources for Epidemiologists

Many machine learning articles and textbooks are written for an audience with a computer science background, and as a consequence, the language and terminology can be unfamiliar to epidemiologists. In order to help interested readers further explore these topics, we have selected a sample of relatively easily accessible articles that introduce the algorithms and ensemble models reviewed in this primer in greater detail:

- Artificial neural networks: Jain et al. (159); Olden et al. (160)
- Decision trees: Atkinson and Therneau (161); Olden et al. (160)
- Support vector machines: Noble (36)
- Naive Bayes: Lewis (40)
- *K*-means clustering: Jain (46)
- Bayesian model averaging: Hoeting et al. (66)
- Super Learner: Polley and van der Laan (162)
- Boosting and bagging: Opitz and Maclin (163)

In addition, *An Introduction to Statistical Learning* by James et al. (31) provides an accessible overview of popular machine learning algorithms and discusses them in parallel with traditional statistical approaches. A supplemental 15-hour online tutorial by Markham (164) discusses much of the same material in further detail and offers an alternative learning format. It is available at https://www.r-bloggers.com/in-depth-introduction-to-machine-learning-in-15-hours-of-expert-videos/. Both resources are open-access.