Chapter 1
General Elements of Genomic Selection and Statistical Learning

1.1 Data as a Powerful Weapon

Thanks to advances in digital technologies like electronic devices and networks, it is possible to automatize and digitalize many jobs, processes, and services, which are generating huge quantities of data. These "big data" are transmitted, collected, aggregated, and analyzed to deliver deep insights into processes and human behavior. For this reason, data are called the new oil, since "data are to this century what oil was for the last century"—that is, a driver for change, growth, and success. While statistical and machine learning algorithms extract information from raw data, information can be used to create knowledge, knowledge leads to understanding, and understanding leads to wisdom (Sejnowski 2018). We have the tools and expertise to collect data from diverse sources and in any format, which is the cornerstone of a modern data strategy that can unleash the power of artificial intelligence. Every single day we are creating around 2.5 quintillion bytes of data (McKinsey Global Institute 2016). This means that almost 90% of the data in the world has been generated over the last 2 years. This unprecedented capacity to generate data has increased connectivity and global data flows through numerous sources like tweets, YouTube, blogs, sensors, internet, Google, emails, pictures, etc. For example, Google processes more than 40,000 searches every second (and 3.5 billion searches per day), 456,000 tweets are sent, and 4,146,600 YouTube videos are watched per minute, and every minute, 154,200 Skype calls are made, 156 million emails are sent, 16 million text messages are written, etc. In other words, the amount of data is becoming bigger and bigger (big data) day by day in terms of volume, velocity, variety, veracity, and "value."

The nature of international trade is being radically transformed by global data flows, which are creating new opportunities for businesses to participate in the global economy. The following are some ways that these data flows are transforming international trade: (a) businesses can use the internet (i.e., digital platforms) to export goods; (b) services can be purchased and consumed online; (c) data collection

O. A. Montesinos López et al., *Multivariate Statistical Machine Learning Methods for Genomic Prediction*, https://doi.org/10.1007/978-3-030-89010-0_1

and analysis are allowing new services (often also provided online) to add value to exported goods; and (d) global data flows underpin global value chains, creating new opportunities for participation.

According to estimates, by 2020, 15–20% of the gross domestic product (GDP) of countries all over the world will be based on data flows. Companies that adopt big data practices are expected to increase their productivity by 5–10% compared to companies that do not, and big data practices could add 1.9% to Europe's GDP between 2014 and 2020. According to McKinsey Global Institute (2016) estimates, big data could generate an additional $3 trillion in value every year in some industries. Of this, $1.3 trillion would benefit the United States. Although these benefits do not directly affect the GDP or people's personal income, they indirectly help to improve the quality of life.

But once we have collected a large amount of data, the next question is: how to make sense of it? A lot of businesses and people are collecting data, but few are extracting useful knowledge from them. As mentioned above, nowadays there have been significant advances for measuring and collecting data; however, obtaining knowledge from (making sense of) these collected data is still very hard and challenging, since there are few people in the market with the expertise and training needed to extract knowledge from huge amounts of data. For this reason, to make sense of data, new disciplines were recently created, such as data science (the commercial name of statistics), business intelligence and analytics, that use a combination of statistics, computing, machine learning, and business, among other domains, to analyze and discover useful patterns to improve the decision-making process. In general, these tools help to (1) rationalize decisions and the decision-making process, (2) minimize errors and deal with a range of scenarios, (3) get a better understanding of the situation at hand (due to the availability of historical data), (4) assess alternative solutions (based on key performance indicators), and (5) map them against the best possible outcome (benchmarking), which helps make decision-making more agile. For this reason, data analysis has been called "the sexiest job of the twenty-first century." For example, big data jobs in the United States are estimated to be 500,000. But the McKinsey Global Institute (2016) estimates that there is still a shortage of between 140,000 and 190,000 workers with a background in statistics, computer engineering, and other applied fields, and that 1.5 million managers are needed to evaluate and make decisions on big data. This means that 50–60% of the required staff was lacking in the year 2018 in the United States alone (Dean 2018).

Some areas and cases where statistical and machine learning techniques have been successfully applied to create knowledge and make sense of data are given next. For example, in astronomy these techniques have been used for the classification of exoplanets using thousands of images taken of these celestial bodies. Banks use them to decide if a client will be granted credit or not, using as predictors many socioeconomic variables they ask of clients. Banks also use them to detect credit card fraud. On the internet, they are used to classify emails as spam or ham (not spam) based on previous emails and the text they contain. In genomic selection, they are used to predict grain yield (or another trait) of non-phenotyped plants using

information, including thousands of markers and environmental data. In Google, they are used for recommending books, movies, products, etc., using previous data on the characteristics (age, gender, location, etc.) of the people who have used these services. They are also used in self-driving cars, that is, cars capable of sensing their environment and moving with little or no human input, based on thousands of images and information from sensors that perceive their surroundings. These examples give more evidence that the appropriate use of data is a powerful weapon for getting knowledge of the target population.

However, as is true of any new technology, this data-related technology can be used against society. One example is the Facebook–Cambridge Analytica data scandal that was a major political scandal in early 2018 when it was revealed that Cambridge Analytica had harvested personal data from the Facebook profiles of millions of people without their consent and used them for political purposes. This scandal was described as a watershed moment in the public understanding of personal data and precipitated a massive fall in Facebook's stock price and calls for tighter regulation of tech companies' use of data. For these reasons, some experts believe that governments have a responsibility to create and enforce rules on data privacy, since data are a powerful weapon, and weapons should be controlled, and because privacy is a fundamental human right. All these are very important to avoid the weaponization of data against people and society.

To take advantage of the massive data collected in Genomic Selection (GS) and many other domains, it is really important to train people in statistical machine learning methods and related areas to perform precise prediction, extract useful knowledge, and find hidden data patterns. This means that experts in statistical machine learning methods should be able to identify the statistical or machine learning method that is most relevant to a given problem, since there is no universal method that works well for all data sets, cleans the original data, implements these methods in statistical machine learning software, and interprets the output of statistical machine learning methods correctly to translate the big data collected into insights and operational value quickly and accurately.

1.2 Genomic Selection

Plant breeding is a key scientific area for increasing the food production required to feed the people of our planet. The key step in plant breeding is selection, and conventional breeding is based on phenotypic selection. Breeders choose good offspring using their experience and the observed phenotypes of crops, so as to achieve genetic improvement of target traits (Wang et al. 2018). Thanks to this area (and related areas of science), the genetic gain nowadays has reached a near-linear increase of 1% in grain yield yearly (Oury et al. 2012; Fischer et al. 2014). However, a linear increase of at least 2% is needed to cope with the 2% yearly increase in the world population, which relies heavily on wheat products as a source of food (FAO 2011). For this reason, genomic selection (GS) is now being implemented in many

plant breeding programs around the world. GS consists of genotyping (markers) and phenotyping individuals in the reference (training) population and, with the help of statistical machine learning models, predicting the phenotypes or breeding values of the candidates for selection in the testing (evaluation) population that were only genotyped. GS is revolutionizing plant breeding because it is not limited to traits determined by a few major genes and allows using a statistical machine learning model to establish the associations between markers and phenotypes and also to make predictions of non-phenotyped individuals that help make a more comprehensive and reliable selection of candidate individuals. In this way, it is essential for accelerating genetic progress in crop breeding (Montesinos-López et al. 2019).

1.2.1 Concepts of Genomic Selection

The development of different molecular marker systems that started in the 1980s drastically increased the total number of polymorphic markers available to breeders and molecular biologists in general. The single nucleotide polymorphism (SNP) that has been intensively used in QTL discovery is perhaps the most popular high-throughput genotyping system (Crossa et al. 2017). Initially, by applying marker-assisted selection (MAS), molecular markers were integrated with traditional phenotypic selection. In the context of simple traits, MAS consists of selecting individuals with QTL-associated markers with major effects; markers not significantly associated with a trait are not used (Crossa et al. 2017). However, after many attempts to improve complex quantitative traits by using QTL-associated markers, there is not enough evidence that this method really can be helpful in practical breeding programs due to the difficulty of finding the same QTL across multiple environments (due to QTL $\times$ environment interaction) or in different genetic backgrounds (Bernardo 2016). Due to this difficulty of the MAS approach, in the early 2000s, an approach called association mapping appeared with the purpose of overcoming the insufficient power of linkage analysis, thus facilitating the detection of marker–trait associations in non-biparental populations and fine-mapping chromosome segments with high recombination rates (Crossa et al. 2017). However, even the fine-mapping approach was unable to increase the power to detect rare variants that may be associated with economically important traits.

For this reason, Meuwissen et al. (2001) proposed the GS methodology (that was initially used in animal science), which is different from association mapping and QTL analysis, since GS simultaneously uses all the molecular markers available in a training data set for building a prediction model; then, with the output of the trained model, predictions are performed for new candidate individuals not included in the training data set, but only if genotypic information is available for those candidate individuals. This means that the goal of GS is to predict breeding and/or genetic values. Because GS is implemented in a two-stage process, to successfully implement it, the data must be divided into a training (TRN) and a testing (TST) set, as can be observed in Fig. 1.1. The training set is used in the first stage, while the testing set

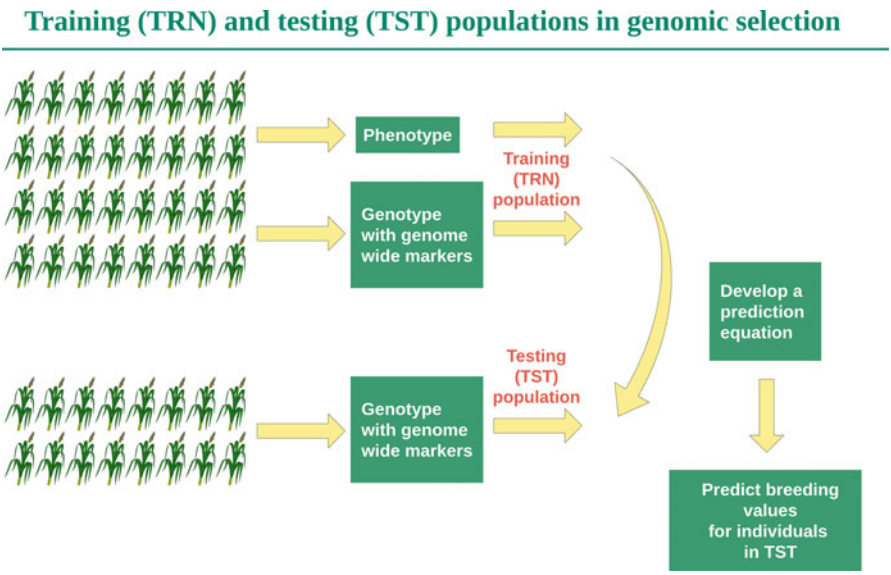

Fig. 1.1 Schematic representation of TRN and TST sets required for implementing GS (Crossa et al. 2017)

is used in the second stage. The main characteristics of the training set are (a) it combines molecular (independent variables) and phenotypic (dependent variables) data and (b) it contains enough observations (lines) and predictors (molecular data) to be able to train a statistical machine learning model with high generalized power (able to predict data not used in the training process) to predict new lines. The main characteristic of the testing set is that it only contains genotypic data (markers) for a sample of observations (lines) and the goal is to predict the phenotypic or breeding values of lines that have been genotyped but not phenotyped.

The two basic populations in a GS program are shown in Fig. 1.1: the training (TRN) data whose genotype and phenotype are known and the testing (TST) data whose phenotypic values are to be predicted using their genotypic information. GS substitutes phenotyping for a few selection cycles. Some advantages of GS over traditional (phenotypic) selection are that it: (a) reduces costs, in part by saving the resources required for extensive phenotyping, (b) saves time needed for variety development by reducing the cycle length, (c) has the ability to substantially increase the selection intensity, thus providing scenarios for capturing greater gain per unit time, (d) makes it possible to select traits that are very difficult to measure, and (e) can improve the accuracy of the selection process. Of course, successful implementation of GS depends strongly on the quality of the training and testing sets.

GS has great potential for quickly improving complex traits with low heritability, as well as significantly reducing the cost of line and hybrid development. Certainly, GS could also be employed for simple traits with higher heritability than complex traits, and high genomic prediction (GP) accuracy is expected. Application of GS in

plant breeding could be limited by some of the following factors: (i) genotyping cost, (ii) unclear guidelines about where in the breeding program GS could be efficiently applied (Crossa et al. 2017), (iii) insufficient number of lines or animals in the reference (training) population, (iv) insufficient number of SNPs in the panels, and (v) the reference population contains very heterogeneous individuals (plants or animals).

1.2.2 Why Is Statistical Machine Learning a Key Element of Genomic Selection?

GS is challenging and very interesting because it aims to improve crop productivity to satisfy humankind's need for food. Addressing the current challenges to increase crop productivity by improving the genetic makeup of plants and avoiding plant diseases is not new, but it is of paramount importance today to be able to increase crop productivity around the world without the need to increase the arable land. Statistical machine learning methods can help improve GS methodology, since they are able to make computers learn patterns that could be used for analysis, interpretation, prediction, and decision-making. These methods learn the relationships between the predictors and the target output using statistical and mathematical models that are implemented using computational tools to be able to predict (or explain) one or more dependent variables based on one or more independent variables in an efficient manner. However, to do this successfully, many real-world problems are only approximated using the statistical machine learning tools, by evaluating probabilistic distributions, and the decisions made using these models are supported by indicators like confidence intervals. However, the creation of models using probability distributions and indicators for evaluating prediction (or association) performance is a field of statistical machine learning, which is a branch of artificial intelligence, understanding as statistical machine learning the application of statistical methods to identify patterns in data using computers, but giving computers the ability to learn without being explicitly programmed (Samuel 1959). However, artificial intelligence is the field of science that creates machines or devices that can mimic intelligent behaviors.

As mentioned above, statistical machine learning allows learning the relationship between two types of information that are assumed to be related. Then one part of the information (input or independent variables) can be used to predict the information lacking (output or dependent variables) in the other using the learned relationship. The information we want to predict is defined as the response variable (y), while the information we use as input are the predictor variables ($\boldsymbol{X}$). Thanks to the continuous reduction in the cost of genotyping, GS nowadays is implemented in many crops around the world, which has caused the accumulation of large amounts of biological data that can be used for prediction of non-phenotyped plants and animals. However, GS implementation is still challenging, since the quality of the data (phenotypic and

genotypic) needs to be improved. Many times the genotypic information available is not enough to make high-quality predictions of the target trait, since the information available has a lot of noise. Also, since there is no universal best prediction model that can be used under all circumstances, a good understanding of statistical machine learning models is required to increase the efficiency of the selection process of the best candidate individuals with GS early in time. This is very important because one of the key components of genomic selection is the use of statistical machine learning models for the prediction of non-phenotyped individuals. For this reason, statistical machine learning tools have the power to help increase the potential of GS if more powerful statistical machine learning methods are developed, if the existing methods can deal with larger data sets, and if these methods can be automatized to perform the prediction process with only a limited knowledge of the subject.

For these reasons, statistical machine learning tools promise considerable benefits for GS and agriculture through their contribution to productivity growth and gain in the genetic makeup of plants and animals without the need to increase the arable land. At the same time, with the help of statistical machine learning tools, GS is deeply impacting the traditional way of selecting candidate individuals in plant and animal breeding. Since GS can reduce by at least half the time needed to select candidate individuals, it has been implemented in many crops around the globe and is radically changing the traditional way of developing new varieties and animals all over the world.

Although GS is not the dominant paradigm for developing new plants and animals, it has the potential to transform the way they are developed due to the following facts: (a) the massive amounts of data being generated in plant breeding programs are now available to train the statistical machine learning methods, (b) new technologies such as sensors, satellite technology, and robotics allow scientists to generate not only genomic data but also phenotypic data that can capture a lot of environmental and phenotypic information that can be used in the modeling process to increase the performance of statistical machine learning methods, (c) increased computational power now allows complex statistical machine learning models with larger data sets to be implemented in less time, and (d) there is now greater availability of user-friendly statistical machine learning software for implementing a great variety of statistical machine learning models.

However, there are still limitations for the successful implementation of GS with the help of statistical machine learning methods because much human effort is required to collect a good training data set for supervised learning. Although nowadays it is possible to measure a lot of independent variables (markers, environmental variables) due to the fact that the training set should be measured in real-world experiments conducted in different environments and locations, this is expensive and subject to nature's random variability. This means that GS data are hampered by issues such as multicollinearity among markers (adjacent markers are highly correlated) and by a problem that consists of having a small number of observations and a large number of independent variables (commonly known as "large p small n"), which poses a statistical challenge. For this reason, obtaining data sets that are large and comprehensive enough to be used for training—for example,

creating or obtaining sufficient plant trial data to predict yield, plant height, grain quality, and presence or absence of disease outcomes more accurately—is also often challenging.

Another challenge is that of building statistical machine learning techniques that are able to generalize the unseen data, since statistical machine learning methods continue to have difficulty carrying their experiences from one set of circumstances to another. This is known as transfer learning, and it focuses on storing knowledge gained when training a particular machine learning algorithm and then using this stored knowledge for solving another related problem. In other words, transfer learning is still very challenging and occurs when a statistical machine learning model is trained to accomplish a certain task and then quickly apply that learning exercise to a different activity.

Another disadvantage is that even though today there are many software programs for implementing statistical machine learning tools for GS, the computational resources required for learning from moderate to large data sets are very expensive and most of the time it is not possible to implement them in commonly used computers, since servers with many cores and considerable computational resources are required. However, the rapid increase in computational power will change this situation in the coming years.

1.3 Modeling Basics

1.3.1 What Is a Statistical Machine Learning Model?

A model is a simplified description, using mathematical tools, of the processes we think that give rise to the observations in a set of data. A model is deterministic if it explains (completely) the dependent variables based on the independent ones. In many real-world scenarios, this is not possible. Instead, statistical (or stochastic) models try to approximate exact solutions by evaluating probabilistic distributions. For this reason, a statistical model is expressed by an equation composed of a *systematic* (deterministic) and a *random part* (Stroup 2012) as given in the next equation:

$$y_i = f(\boldsymbol{x_i}) + \epsilon_i, \text{for } i = 1, 2, \ldots, n, \tag{1.1}$$

where y_i represents the response variable in individual i and $f(\boldsymbol{x}_i)$ is the systematic part of the model because it is *determined* by the explanatory variables (predictors). For these reasons, the systematic part of the statistical learning model is also called the deterministic part of the model, which gives rise to an unknown mathematical function (f) of $\boldsymbol{x}_i = x_{i1}, \ldots, x_{ip}$ not subject to random variability. ϵ_i is the ith random element (error term) which is independent of $\boldsymbol{x_i}$ and has mean zero. The ϵ_i term tells us that observations are assumed to vary at random about their mean, and it also defines the uniqueness of each individual. In theory (at least in some philosophical

domains), if we know the mechanism that gives rise to the uniqueness of each individual, we can write a completely deterministic model. However, this is rarely possible because we use probability distributions to characterize the observations measured in the individuals. Most of the time, the error term (ϵ_i) is assumed to follow a normal distribution with mean zero and variance σ^2 (Stroup 2012).

As given in Eq. (1.1), the f function that gives rise to the systematic part of a statistical learning model is not restricted to a unique input variable, but can be a function of many, or even thousands, of input variables. In general, the set of approaches for estimating f is called statistical learning (James et al. 2013). Also, the functions that f can take are very broad due to the huge variety of phenomena we want to predict and due to the fact that there is no universally superior f that can be used for all processes. For this reason, to be able to perform good predictions out of sample data, many times we need to fit many models and then choose the one most likely to succeed with the help of cross-validation techniques. However, due to the fact that models are only a simplified picture of the true complex process that gives rise to the data at hand, many times it is very hard to find a good candidate model. For this reason, statistical machine learning provides a catalog of different models and algorithms from which we try to find the one that best fits our data, since there is no universally best model and because there is evidence that a set of assumptions that works well in one domain may work poorly in another—this is called the *no free lunch theorem* by Wolpert (1996). All these are in agreement with the famous aphorism, "all models are wrong, but some are useful," attributed to the British statistician George Box (October 18, 1919–March 28, 2013) who first mentioned this aphorism in his paper "Science and Statistics" published in the *Journal of the American Statistical Association* (Box 1976). As a result of the *no free lunch theorem*, we need to evaluate many models, algorithms, and sets of hyperparameters to find the best model in terms of prediction performance, speed of implementation, and degree of complexity. This book is concerned precisely with the appropriate combination of data, models, and algorithms needed to reach the best possible prediction performance.

1.3.2 The Two Cultures of Model Building: Prediction Versus Inference

The term "two cultures" in statistical model building was coined by Breiman (2001) to explain the difference between the two goals for estimating f in Eq. (1.1): prediction and inference. These definitions are provided in order to clarify the distinct scientific goals that follow inference and empirical predictions, respectively. A clear understanding and distinction between these two approaches is essential for the progress of scientific knowledge. Inference and predictive modeling reflect the process of using data and statistical (or data mining) methods for inferring or predicting, respectively. The term modeling is intentionally chosen over model to

highlight the entire process involved, from goal definition, study design, and data collection to scientific use (Breiman 2001).

Prediction

The prediction approach can be defined as the process of applying a statistical machine learning model or algorithm to data for the purpose of predicting new or future observations. For example, in plant breeding a set of inputs (marker information) and the outcome Y (disease resistance: yes or no) are available for some individuals, but for others only marker information is available. In this case, marker information can be used as a predictor and the disease status should be used as the response variable. When scientists are interested in predicting new plants not used to train the model, they simply want an accurate model to predict the response using the predictors. However, when scientists are interested in understanding the relationship between each individual predictor (marker) and the response variable, what they really want is a model for inference. Another example is when forest scientists are interested in developing models to predict the number of fire hotspots from an accumulated fuel dryness index, by vegetation type and region. In this context, it is obvious that scientists are interested in future predictions to improve decision-making in forest fire management. Another example is when an agro-industrial engineer is interested in developing an automated system for classifying mango species based on hundreds of mango images taken with digital cameras, mobile phones, etc. Here again it is clear that the best approach to build this system should be based on prediction modeling since the objective is the prediction of new mango species, not any of those used for training the model.

Inference

Many areas of science are devoted mainly to testing causal theories. Under this framework, scientists are interested in testing the validity of a theoretical causal relationship between the causal variables (X; underlying factors) and the measured variable (Y) using statistical machine learning models and collected data to test the causal hypotheses. The type of statistical machine learning models used for testing causal hypotheses are usually association-based models applied to observational data (Shmueli 2012). For example, regression models are one type of association-based models used for testing causal hypotheses. This practice is justified by the theory itself, which assumes the causality. In this context, the role of the theory is very strong and the reliance on data and statistical modeling is strictly through the lens of the theoretical model. The theory–data relationship varies in different fields. While the social sciences are very theory-heavy, in areas such as bioinformatics and natural language processing, the emphasis on a causal theory is much weaker. Hence, given this reality, Shmueli (2012) defined *explaining* as causal explanation and *explanatory modeling* as the use of statistical models for testing causal explanations.

Next, we provide some great examples used for testing causal hypotheses: for example, between 1911 and 1912, Austrian physicist Victor Hess made a series of ten balloon ascents to study why metal plates tend to charge spontaneously. At that

time, it was assumed that the cause was the presence, in small quantities, of radioactive materials in rocks. If this is the case, as one moves away from the ground, the tendency of metal plates to be charged should decrease. Hess brought with him three electroscopes, which are instruments composed basically of two metal plates enclosed in a glass sphere. When charging the plates, they separated one from the other. Hess observed that from a certain height, the three electroscopes tended to be charged to a greater extent. On August 7, 1912, together with a flight commander and a meteorologist, he made a 6-hour flight in which he ascended to more than 5000 m in height (Schuster 2014). Hess published his results in 1913, where he presented his conclusion that the cause of the charge of the electroscopes was radiation of cosmic origin that penetrates the atmosphere from above. The discovery of this cosmic radiation, for which Hess received the Nobel Prize in 1936, opened a new window for the study of the universe (Schuster 2014). It was in 1925 when American physicist Robert Andrew Millikan introduced the term "cosmic rays" to describe this radiation, and what it was made of was still unknown. In this example, it is clear that the goal of the analysis was association.

No one suspected that tobacco was a cause of lung tumors until the final decade of the nineteenth century. In 1898, Hermann Rottmann (a medical student) in Würzburg proposed that tobacco dust—not smoke—might be causing the elevated incidence of lung tumors among German tobacco workers. This was a mistake corrected by Adler (1912) who proposed that smoking might be to blame for the growing incidence of pulmonary tumors. Lung cancer was still a very rare disease; so rare, in fact, that medical professors, when confronted with a case, sometimes told their students they might never see another. However, in the 1920s, surgeons were already faced with a greater incidence of lung cancer, and they began to get confused about its possible causes. In general, smoking was blamed, along with asphalt dust from recently paved roads, industrial air pollution, and the latent effects of poisonous gas exposure during World War I or the global influenza pandemic of 1918–1919. These and many other theories were presented as possible explanations for the increase in lung cancer, until evidence from multiple research sources made it clear that tobacco was the main culprit (Proctor 2012). Here again it is clear that the goal of the analysis should be related to inference.

1.3.3 Types of Statistical Machine Learning Models and Model Effects

1.3.3.1 Types of Statistical Machine Learning Models

Statistical machine learning models are most commonly classified as parametric models, semiparametric models, and nonparametric models. Next, we define each type of statistical machine learning models and provide examples that help to understand each one.

Parametric Model It is a type of statistical machine learning model in which all the predictors take predetermined forms with the response. Linear models (e.g., multiple regression: $y = \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \epsilon$), generalized linear models [Poisson regression: $E(y|x) = \exp(\beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3)$], and nonlinear models (nonlinear regression: $y = \beta_1 x_1 + \beta_2 x_2 + \beta_3 e^{\beta_4 x_3} + \epsilon$) are examples of parametric statistical machine learning models because we know the function that describes the relationship between the response and the explanatory variables. These models are very easy to interpret but very inflexible.

Nonparametric Model It is a type of statistical machine learning model in which none of the predictors take predetermined forms with the response but are constructed according to information derived from data. Two common statistical machine learning models are kernel regression and smoothing spline. Kernel regression estimates the conditional expectation of y at a given value x using a weighted filter on the data ($y = m(x) + \epsilon$, with $\widehat{m}(x_0) = \frac{\sum_{i=1}^{n} K\left(\frac{x_i - x_0}{h}\right) y_i}{\sum_{i=1}^{n} K\left(\frac{x_i - x_0}{h}\right)}$), where h is the bandwidth (this estimator of $m(x)$ is called the *Nadaraya–Watson (NW) kernel estimator*) and K is a kernel function. While smoothing splines minimize the sum of squared residuals plus a term which penalizes the roughness of the fit [$y = \beta_0 + \beta_1 x + \beta_2 x^2 + 3x^3 + \sum_{j=1}^{J} \beta_{1j} \left(x - \theta_j\right)_+^3$, where $(x - \theta_j)_+ = x - \theta_j$, $x > \theta_j$ and 0 otherwise], this model in brackets is a spline of degree 3 which is represented as a power series. These models are very difficult to interpret but are very flexible. Nonparametric statistical machine learning models differ from parametric models in that the shape of the functional relationships between the response (dependent) and the explanatory (independent) variables are not predetermined but can be adjusted to capture unusual or unexpected features of the data. Nonparametric statistical machine learning models can reduce modeling bias (the difference between estimated and true values) by imposing no specific model structure other than certain smoothness assumptions, and therefore they are particularly useful when we have little information or we want to be flexible about the underlying statistical machine learning model. In general, nonparametric statistical machine learning models are very flexible and are better at fitting the data than parametric statistical machine learning models. However, these models require larger samples than parametric statistical machine learning models because the data must supply the model structure as well as the model estimates.

Semiparametric Model It is a statistical machine learning model in which *part* of the predictors do not take predetermined forms while the other part takes known forms with the response. Some examples are (a) $y = \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + m(x) + \epsilon$ and (b) $y = \exp(\beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3) + m(x) + \epsilon$. This means that semiparametric models are a mixture of parametric and nonparametric models.

When the relationship between the response and explanatory variables is known, parametric statistical machine learning models should be used. If the relationship is

unknown and nonlinear, nonparametric statistical machine learning models should be used. When we know the relationship between the response and part of the explanatory variables, but do not know the relationship between the response and the other part of the explanatory variables, we should use semiparametric statistical machine learning models. Any application area that uses statistical machine learning analysis could potentially benefit from semi/nonparametric regression.

1.3.3.2 Model Effects

Many statistical machine learning models are expressed as models that incorporate *fixed effects*, which are parameters associated with an entire population or with certain levels of experimental factors of interest. Other models are expressed as *random effects*, where individual experimental units are drawn at random from a population, while a model with *fixed effects* and *random effects* is called a *mixed-effects* model (Pinheiro and Bates 2000).

According to Milliken and Johnson (2009), a factor is a *random effect* if its levels consist of a random sample of levels from a population of possible levels, while a factor is a *fixed effect* if its levels are selected by a nonrandom process or if its levels consist of the entire population of possible levels.

Mixed-effects models, also called multilevel models in the social science community (education, psychology, etc.), are an extension of regression models that allow for the incorporation of random effects; they are better suited to describe relationships between a response variable and some covariates in data that are grouped according to one or more classification factors. Examples of such grouped data include longitudinal data, repeated measures data, multilevel data, and block designs. One example of grouped data are animals that belong to the same herd; for example, assume we have 10 herds with 50 animals (observations) in each. By associating to observations (animals) sharing the same level of a classification factor (herd) a common random effect, mixed-effects models parsimoniously represent the covariance structure induced by the grouping of data (Pinheiro and Bates 2000). Most of the early work on mixed models was motivated by the animal science community driven by the need to incorporate heritabilities and genetic correlations in parsimonious fashion.

Next we provide an example to illustrate how to build these types of models. Assume that five environments were chosen at random from an agroecological area of Mexico. Then in each area, three replicates of a new variety (NV) of maize were tested to measure grain yield (GY) in tons per hectare. The data collected from this experiment are shown in Fig. 1.2.

Since the only factor that changes among the observations measured in this experiment is the environment, they are arranged in a one-way classification because they are classified according to a single characteristic: the environments in which the observations were made (Pinheiro and Bates 2000). The data structure is very simple since each row represents one observation for which the environment and GY were recorded, as can be seen in Table 1.1.

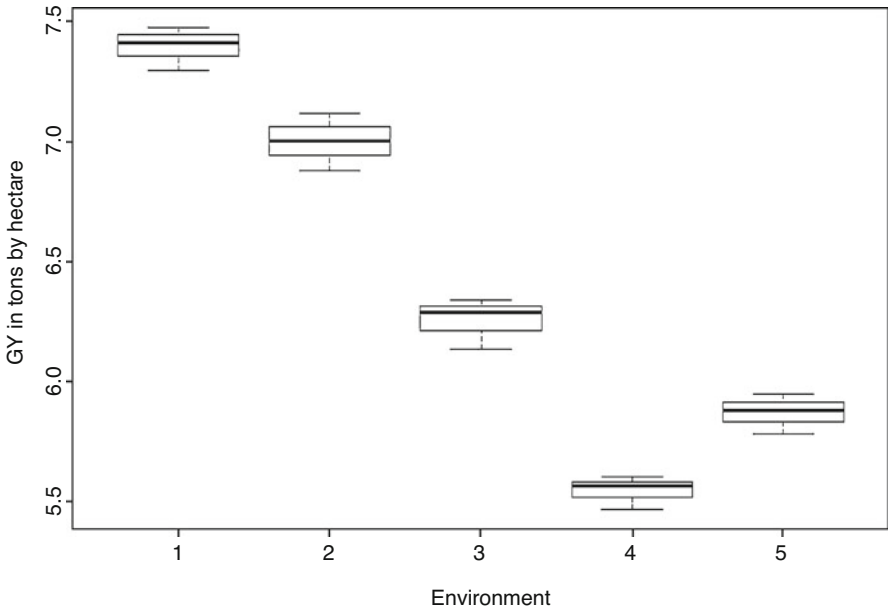

Fig. 1.2 Grain yield (GY) in tons per hectare by environment

Table 1.1 Grain yield (GY) measured in five environments (Env) with three repetitions (Rep) in each environment

Env	Rep	GY
1	1	7.476
1	2	7.298
1	3	7.414
2	1	7.117
2	2	6.878
2	3	7.004
3	1	6.136
3	2	6.340
3	3	6.288
4	1	5.600
4	2	5.564
4	3	5.466
5	1	5.780
5	2	5.948
5	3	5.881

The breeder who conducted this experiment was only interested in the average GY for a typical environment, that is, the expected GY, the variation in average GY among environments (between-environment variability), and the variation in the observed GY for a single environment (within-environment variability). Figure 1.2 shows that there is considerable variability in the mean GY for different

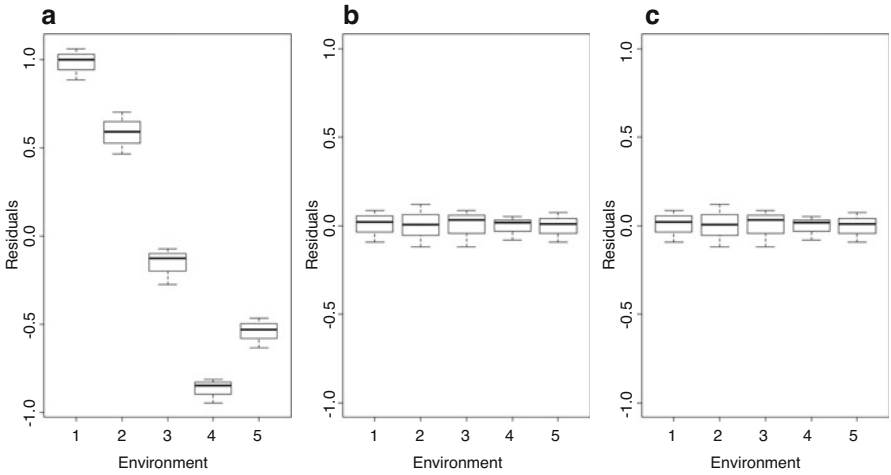

Fig. 1.3 Box plot of residuals by environments of the fit (**a**) of a single-mean model, (**b**) of a fixed-effects model with environment effect, and (**c**) of a mixed-effects model with environment effect

environments and that the within-environment variability is smaller than the between-environment variability.

Data from a one-way classification like the one given in Table 1.1 can be analyzed under both approaches with fixed effects or random effects. The decision on which approach to use depends basically on the goal of the study, since if the goal is to make inferences about the population for which these environments (levels) were drawn, then the random effect approach is the best option, but if the goal is to make inferences about the particular environments (levels) selected in this experiment, then the fixed-effects approach should be preferred.

Assuming a simple model that ignores the environment

$$GY_{ij} = \beta + e_{ij}, \qquad i = 1, .., 5, j = 1, 2, 3, \tag{1.2}$$

where GY_{ij} denotes the GY of environment i in replication j, and β is the mean GY across the population of environments sampled with $e_{ij} \sim N(0, \sigma^2)$. Using this model we estimate $\widehat{\beta} = 6.4127$ and the residual standard error is equal to $\widehat{\sigma} = 0.7197$. By fitting this model and observing the boxplot of the residuals (Fig. 1.3a) versus the environments, we can see that these residuals are very large and different for each environment, which can be attributed to the fact that this model ignores the environmental effect and was only fitted as a single-mean model, which implies that the environmental effects are included in the residuals. For this reason, we then incorporated the environmental effect in the model as a separate effect. This fixed-effects model is equal to a one-way classification model as

$$GY_{ij} = \beta_i + e_{ij}, \quad i = 1, .., 5, j = 1, 2, 3, \tag{1.3}$$

where β_i represents the fixed effect of environment i, while the other terms in the model are the same as those in Eq. (1.2). By fitting this model using least squares, we now have a beta coefficient for each environment equal to: $\widehat{\beta}_1 = 7.396$, $\widehat{\beta}_2 = 6.999$, $\widehat{\beta}_3 = 6.255$, $\widehat{\beta}_4 = 5.543$, $\widehat{\beta}_5 = 5.869$, while the residual standard error is equal to $\widehat{\sigma} = 0.095$. Figure 1.3b shows that the residuals are considerably lower and more centered around zero than in the first fitted model (single-mean model), which can be observed in the residual standard error that is 7.47 times smaller. Therefore, we have evidence that the model given in Eq. (1.3) successfully accounted for the environmental effects. Two drawbacks of the model with fixed effects given in Eq. (1.3) are that it is unable to provide an estimate of the between-environments variability and that the number of parameters in the model increases linearly with the number of environments. Fortunately, the random effects model circumvents these problems by treating the environmental effects as random variations around a population mean. Next we reparameterize model (1.3) as a random effects model. We write

$$GY_{ij} = \overline{\beta} + \left(\beta_i - \overline{\beta}\right) + e_{ij}, \tag{1.4}$$

where $\overline{\beta} = \sum_{i=1}^{5} \beta_i / 5$ represents the average grain yield for the environments in the experiment. The random effects model replaces $\overline{\beta}$ with the mean grain yield across the population of environments and replaces the deviations $\beta_i - \overline{\beta}$ with the random variables whose distribution is to be estimated. If $\beta_i - \overline{\beta}$ is not assumed random the model belongs to fixed effects. Therefore, the random effects version of the model given in Eq. (1.4) is equal to

$$GY_{ij} = \beta + b_i + e_{ij}, \tag{1.5}$$

where β is the mean grain yield across the population of environments, b_i is a random variable representing the deviation from the population mean of the grain yield for the ith environment, and e_{ij} is defined as before. To complete this statistical machine learning model, we must specify the distribution of the random variables b_i, with $i = 1, \ldots, 5$. It is common to assume that b_i is normally distributed with mean zero and variance between environments σ_b^2, that is, b_i is distributed $N\left(0, \sigma_b^2\right)$. It is also common to assume independence between the two random effects b_i and e_{ij}. Models with at least two sources of random variation are also called hierarchical models or multilevel models (Pinheiro and Bates 2000). The covariance between observations in the same environment is σ_b^2, which corresponds to a correlation of $\sigma_b^2 / \left(\sigma_b^2 + \sigma^2\right)$. The parameters of the mixed model given in Eq. (1.5) are β, σ_b^2, and σ^2, and irrespective of the number of environments in the experiment, the required number of parameters will always be three, although the random effects, b_i, behave like parameters. We will, however, require $\widehat{b}_i$ predictions of these random effects, given the observed data at hand. Note that when fitting this model (Eq. 1.5) for the environmental data, the parameter estimates were $\beta = 6.413$, $\sigma_b^2 = 0.594$, and $\widehat{\sigma} = 0.095$. It is evident that there was no improvement in terms of fitting since the plot of

residuals versus environment looks the same as the last fitted model (Fig. 1.3c) and the estimated residual standard error was the same as that under the fixed-effects model, which includes the effects of environment in the predictor, but as mentioned above, many times requires considerably fewer parameter estimates than a fixed-effects model.

1.4 Matrix Algebra Review

In this section, we provide the basic elements of linear algebra that are key to understanding the machinery behind the process of building statistical machine learning algorithms.

A ***matrix*** is a rectangular arrangement of numbers whose elements can be identified by the row and column in which they are located. For example, matrix $\boldsymbol{E}$, consisting of three rows and five columns, can be represented as follows:

$$\boldsymbol{E} = \begin{bmatrix} E_{11} & E_{12} & E_{13} & E_{14} & E_{15} \\ E_{21} & E_{22} & E_{23} & E_{24} & E_{25} \\ E_{31} & E_{32} & E_{33} & E_{34} & E_{35} \end{bmatrix}$$

For example, by replacing the matrix with numbers, we have

$$\boldsymbol{E} = \begin{bmatrix} 7 & 9 & 4 & 3 & 6 \\ 9 & 5 & 9 & 8 & 11 \\ 3 & 2 & 11 & 9 & 6 \end{bmatrix}$$

where the element E_{ij} is called the *ij*th element of the matrix; the first subscript refers to the row where the element is located and the second subscript refers to the column, for example, $E_{32} = 2$. The order of an array is the number of rows and columns. Therefore, a matrix with r rows and c columns has an order of $r \times c$. Matrix $\boldsymbol{E}$ has an order of 3×5 and is denoted as $\boldsymbol{E}_{3 \times 5}$.

In R, the way to establish an array is through the command *matrix(...)* with parameters of this function given by *matrix* (*data* = *NA, nrow* = *3, ncol* = *5, byrow* = *FALSE*) where *data* is the data for the matrix, *nrow* the number of rows, *ncol* the number of columns, and *byrow* is the way in which you will accommodate the data in the matrix by row or column. The data entered by default are FALSE, so they will fill the matrix by columns, while if you specified TRUE, they will fill the matrix by rows.

For example, to build matrix $\boldsymbol{E}$ in R, use the following R script:

$$\boldsymbol{E} = \begin{bmatrix} 7 & 9 & 4 & 3 & 6 \\ 9 & 5 & 9 & 8 & 11 \\ 3 & 2 & 11 & 9 & 6 \end{bmatrix}$$

```
E <- matrix(data= c(7,9,4,3,6,9,5,9,8,11,3,2,11,9,6), nrow = 3, ncol = 5, byrow = TRUE)
E
     [,1] [,2] [,3] [,4] [,5]
[1,]   7   9   4   3   6
[2,]   9   5   9   8  11
[3,]   3   2  11   9   6
```

To access all the values of a row, for example, the first row of matrix $\boldsymbol{E}$, you can use:

```
E[1,]
[1] 7 9 4 3 6
```

While, to access all the values of a column, for example, the first column of matrix $\boldsymbol{E}$, you can use:

```
E[,1]
[1] 7 9 3
```

To access a specific element, for example, row 3, column 2 of matrix $\boldsymbol{E}$, you can specify:

```
E[3,2]
[1] 2
```

A matrix consisting of a single row is called a vector. For example, a vector that has five elements can be represented as

$$\boldsymbol{e} = [8 \;\; 10 \;\; 5 \;\; 4 \;\; 7]$$

Here only a subscript is needed to specify the position of an element within the vector. Therefore, the i*th* element in vector $\boldsymbol{e}$ refers to the element in the ith column. For example, $e_3 = 5$.

To create a vector in R, we use the c (...) command that receives the data, separated by a comma. For example, the vector named $\boldsymbol{e}$ can be created using the following command:

```
e <- c(8,10,5,4,7)
e
[1] 8 10 5 4 7
```

To access a specific index, you specify the value between brackets, for example, index 3 of vector ***e***.

```
e[3]
```

```
[1] 5
```

Next we provide definitions and examples of some common types of matrices. We start with a ***square matrix***, which is a matrix with the same number of rows and columns. A matrix ***B*** of order 3×3 is shown below.

$$\boldsymbol{B} = \begin{bmatrix} 3 & 5 & 0 \\ 5 & 1 & 5 \\ -2 & -3 & 7 \end{bmatrix}$$

The *ij* elements in the square matrix where *i* equals *j* are called diagonal elements. The rest of the elements are known as elements that are outside the diagonal, so in this example, the elements of the diagonal of matrix ***B*** are 3, 1, and 7.

To create this type of matrix in R, simply use the *matrix* (...) command and specify the dimensions in the *ncol* and *nrow* parameters, as in the following command:

```
B <- matrix(data = c(3,5,-2,5,1,-3,0,5,7), nrow = 3, ncol = 3)
B
```

```
     [,1] [,2] [,3]
[1,]    3    5    0
[2,]    5    1    5
[3,]   -2   -3    7
```

Next we define a ***diagonal matrix*** as a square matrix that has zeros in all the elements that are outside the diagonal. For example, by extracting the diagonal elements of the above matrix (***B***), we can form the following diagonal matrix:

$$\boldsymbol{D} = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 7 \end{bmatrix}$$

Another special matrix is when the elements of the diagonal are 1; it is called an ***identity matrix***, and is usually denoted as $\boldsymbol{I}_r$, and *r* denotes the order of the matrix.

In R there is the command *diag (x = 1, ncol, nrow)* that works to create a diagonal matrix, but if you want to extract the diagonal of the ***B*** matrix, it can be extracted in the following way:

```
diag(B)
[1] 3 1 7
```

If we want to create an identity matrix, it is enough to specify the size of the matrix. For example, to create an identity matrix of order 5×5, the command to use would be the following:

```
I <- diag(5)
I
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    0    0    0    0
[2,]    0    1    0    0    0
[3,]    0    0    1    0    0
[4,]    0    0    0    1    0
[5,]    0    0    0    0    1
```

A square matrix with all the elements above the diagonal equal to 0 is known as a *lower triangular* matrix; when all the lower elements of the diagonal are 0, it is known as an *upper triangular* matrix. In the example given below, matrix F illustrates a lower triangular matrix and matrix G illustrates an upper triangular matrix.

$$F = \begin{bmatrix} 8 & 0 & 0 \\ 10 & 7 & 0 \\ 4 & 3 & 12 \end{bmatrix}; \qquad G = \begin{bmatrix} 8 & 10 & 5 \\ 0 & 6 & 10 \\ 0 & 0 & 11 \end{bmatrix}$$

The lower triangular matrix can be extracted using the following commands:

```
F <- matrix(data = c(8,10,4,0,7,3,0,0,12), nrow = 3, ncol = 3)
F=F[upper.tri(F)] <- 0
F
      [,1] [,2] [,3]
[1,]    8    0    0
[2,]   10    7    0
[3,]    4    3   12
```

The upper triangular matrix can be extracted using the following commands:

```
G <- matrix(data = c(8,10,5,0,6,10,0,0,11), nrow = 3, ncol = 3, ,byrow=T)
G[lower.tri(G)] <- 0
G
     [,1] [,2] [,3]
[1,]    8   10    5
[2,]    0    6   10
[3,]    0    0   11
```

Next we will illustrate some basic matrix operations. We start by illustrating the ***transpose*** of matrix $\boldsymbol{E}$, commonly indicated as $\boldsymbol{E}'$ or $\boldsymbol{E}^T$; in this type of matrix, the elements *ji* are the *ij* elements of the original matrix. That is, $E'_{ji} = E_{ij}$, where the columns of $\boldsymbol{E}'$ are the rows of $\boldsymbol{E}$, and the rows of $\boldsymbol{E}'$ are the columns of $\boldsymbol{E}$. Below we provide matrix $\boldsymbol{H}$ and its transpose $\boldsymbol{H}'$.

$$\boldsymbol{H} = \begin{bmatrix} 4 & 6 \\ 6 & 2 \\ 0 & -1 \end{bmatrix}; \qquad \boldsymbol{H}' = \begin{bmatrix} 4 & 6 & 0 \\ 6 & 2 & -1 \end{bmatrix}$$

To obtain the transpose of a matrix in $\boldsymbol{H}$, the command *t (...)* is used:

```
H <- matrix(data = c(4,6,0,6,2,-1), nrow = 3,   2)
H
     [,1] [,2]
[1,]    4    6
[2,]    6    2
[3,]    0   -1
```

```
t(H)
 [1,]   4   6   0
[2,]   6   2  -1
```

Two matrices can be added or subtracted only if they have the same number of rows and columns. To demonstrate the adding process, we use the following matrices:

$$\boldsymbol{J} = \begin{bmatrix} 15 & 15 \\ 25 & 35 \end{bmatrix}; \quad \boldsymbol{L} = \begin{bmatrix} 55 & 65 \\ 75 & 85 \end{bmatrix}$$

We form matrix $\boldsymbol{M}$ as the sum of matrices $\boldsymbol{J}$ and $\boldsymbol{L}$, so that $\boldsymbol{M}_{ij} = \boldsymbol{J}_{ij} + \boldsymbol{L}_{ij}$, and their sum is the following:

$$\boldsymbol{M} = \begin{bmatrix} 15+55 & 15+65 \\ 25+75 & 35+85 \end{bmatrix} = \begin{bmatrix} 70 & 80 \\ 100 & 120 \end{bmatrix}$$

Matrix $\boldsymbol{N}$ is reached by subtracting matrices $\boldsymbol{J}$ and $\boldsymbol{L}$, so $\boldsymbol{N}_{ij} = \boldsymbol{J}_{ij} - \boldsymbol{L}_{ij}$, and the subtraction is the following:

$$\boldsymbol{N} = \begin{bmatrix} 15-55 & 15-65 \\ 25-75 & 35-85 \end{bmatrix} = \begin{bmatrix} -40 & -50 \\ -50 & -50 \end{bmatrix}$$

To do the addition and subtraction of matrices in R, it is enough to use the addition or subtraction operator and fulfill the requirement that both matrices have the same dimensions. Next we reproduce the two previous addition and subtraction examples using the commands in R:

```
J <- matrix(data= c(15,15,25,35), ncol = 2, byrow = TRUE)
L <- matrix(data= c(55,65,75,85), ncol = 2, byrow = TRUE)

M<- J+L
M
     [,1] [,2]
[1,]   70   80
[2,]  100  120
```

While the subtraction of matrices is:

```
N<- J-L
N
      [,1]  [,2]
[1,]  -40  -50
[2,]  -50  -50
```

Two matrices can be multiplied only if the number of columns in the first matrix equals the number of rows in the second. The resulting matrix will be equal to the number of rows in the first matrix and the number of columns in the second. Since $\boldsymbol{O} = \boldsymbol{PQ}$, then

$$O = \sum_{j=1}^{m} \sum_{i=1}^{n} \sum_{k=1}^{z} P_{ik} Q_{kj},$$

where m is the number of columns in matrix $\boldsymbol{Q}$, n the number of rows in matrix $\boldsymbol{P}$, and z the number of rows in $\boldsymbol{Q}$ and number of columns in $\boldsymbol{P}$. To demonstrate the above, we have

$$\boldsymbol{P} = \begin{bmatrix} 6 & 8 & 10 \\ 5 & 6 & 8 \\ 4 & 6 & 9 \end{bmatrix}; \boldsymbol{Q} = \begin{bmatrix} 3 & 5 \\ 2 & 2 \\ 9 & 8 \end{bmatrix}$$

Then $\boldsymbol{S}$ is obtained as

$$\begin{aligned} S_{11} &= 6 \times 3 + 8 \times 2 + 10 \times 9 = 124 \\ S_{21} &= 5 \times 3 + 6 \times 2 + 8 \times 9 = 99 \\ S_{31} &= 4 \times 3 + 6 \times 2 + 9 \times 9 = 105 \\ S_{12} &= 6 \times 5 + 8 \times 2 + 10 \times 8 = 126 \\ S_{22} &= 5 \times 5 + 6 \times 2 + 8 \times 8 = 101 \\ S_{32} &= 4 \times 5 + 6 \times 2 + 9 \times 8 = 104 \end{aligned}$$

Therefore,

$$\boldsymbol{S} = \begin{bmatrix} 124 & 126 \\ 99 & 101 \\ 105 & 104 \end{bmatrix}$$

Note that $\boldsymbol{S}$ is of order 3×2, where 3 represents the number of rows in $\boldsymbol{P}$ and 2 equals the number of columns in $\boldsymbol{Q}$.

In order to multiply matrices in R, it is necessary to use the operator *%*%* between the two matrices, in addition to meeting the requirements mentioned above.

```
P <- matrix(data=c(6,5,4,8,6,6,10,8,9), ncol=3)
Q <- matrix(data=c(3,2,9,5,2,8), ncol=2)

S <- P%*%Q
S
     [,1]  [,2]
[1,]  124  126
[2,]   99  101
[3,]  105  104
```

The inverse of a matrix usually is denoted as $\boldsymbol{R}^{-1}$, and when it is multiplied by the original matrix, it results in an identity matrix, that is, $\boldsymbol{R}^{-1}\boldsymbol{R} = \boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix. Only square matrices are invertible.

If it is a diagonal matrix, its inverse can be calculated simply in the following way:

$$\boldsymbol{R} = \begin{bmatrix} 7 & 0 & 0 \\ 0 & 8 & 0 \\ 0 & 0 & 6 \end{bmatrix}$$

Then

$$\boldsymbol{R}^{-1} = \begin{bmatrix} \frac{1}{7} & 0 & 0 \\ 0 & \frac{1}{8} & 0 \\ 0 & 0 & \frac{1}{6} \end{bmatrix}$$

For a square matrix of 2×2, its inverse can be calculated by finding the determinant, which is the difference between the product of the two elements of the diagonals and the product of the two elements outside the diagonal ($R_{11}R_{22} - R_{12}R_{21}$). Then the position of the elements of the diagonals is reversed by multiplying the elements outside the diagonal by -1 and dividing all the elements by the determinant.

$$\boldsymbol{R} = \begin{bmatrix} 4 & 2 \\ 1 & 6 \end{bmatrix}$$

Then

$$\boldsymbol{R}^{-1} = \frac{1}{(4 \times 6) - (1 \times 2)} \begin{bmatrix} 6 & -2 \\ -1 & 4 \end{bmatrix} = \begin{bmatrix} 0.2727 & -0.0909 \\ -0.0455 & 0.1818 \end{bmatrix}$$

To obtain the inverse in $\boldsymbol{R}$, the solve(...) command is the function used to perform this process, as shown in the following example, where the inverse of the matrix $\boldsymbol{R}$ is obtained.

```
R <- matrix(data = c(4, 1, 2, 6), ncol = 2)
R
     [,1] [,2]
[1,]    4    2
[2,]    1    6
```

```
solve(R)
          [,1]      [,2]
[1,]  0.2727 -0.0909
[2,] -0.0455  0.1818
```

1.5 Statistical Data Types

1.5.1 Data Types

To use statistical learning methods correctly, it is very important to understand the classification of the types of data that exist. This is of paramount importance because data are the input to all statistical machine learning methods and because the data type *determines* the appropriate and valid analysis to be implemented; in addition, each statistical machine learning method is specific to a certain type of data. In general, data are most commonly classified as quantitative (numerical) or qualitative (categorical) (Fig. 1.4).

By quantitative (numerical) data, we understand that the result of the observation or the result of a measurement is a number. They are classified as

(a) Discrete. The variable can only have point values and no values in between, that is, the variable can only have a certain set of possible values and represent items that can be counted because they only have isolated numerical values. Examples: number of household members, number of surgical interventions, number of reported cases of a certain pathology, number of accidents per month, etc. Examples in the context of plant breeding are panicle number per plant, seed number per panicle, weed count per plot, number of infected spikelets per spike, etc. Also, discrete values are called as count responses and those models based on Poisson and negative binomial distribution are appropriate for this type of responses.

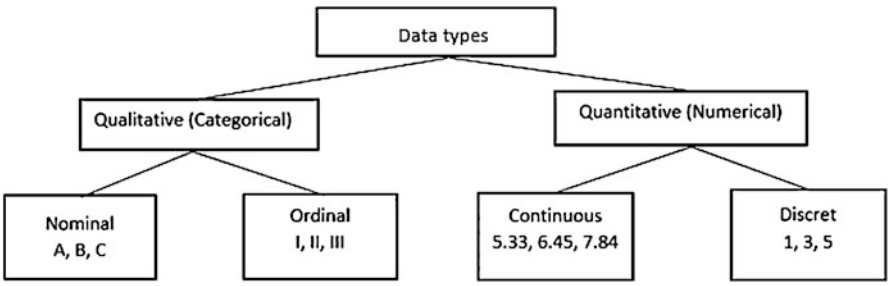

Fig. 1.4 Types of data

(b) Continuous. They are usually the result of a measurement that is expressed in particular units, and values are measured based on a zero point and are treated as real numbers. There are many types of mathematical operations that can be performed on this type of data. The measurements can theoretically have an infinite set of possible values within a range and they do not need transformation. In practice, the possible values of the variable are limited by the accuracy of the measurement method or by the recording mode. Examples: plant height, age, weight, grain yield, pH, blood cholesterol level, etc. The distinction between discrete and continuous data is important for deciding which statistical learning method to use for the analysis, since there are methods that assume that the data are continuous. Consider, for example, the age variable. Age is continuous, but if it is recorded in years, it turns out to be discrete. In studies with adults, in which the age ranges from 20 to 70 years, for example, there are no problems in treating age as continuous, since the number of possible values is large. But in the case of preschool children, if the age is recorded in years, it should be treated as discrete, while if it is recorded in months, it can be treated as continuous.

Similarly, the variable number of beats per minute is a discrete variable, but it is treated as continuous due to the large number of possible values. Numerical data (discrete or continuous) can be transformed into categorical and be treated as such. Although this is correct, it is not necessarily efficient because information is lost during the categorization process. It is always preferable to record the numerical value of the measurement, since this makes it possible to (a) analyze the variable as numerical because statistical analysis is simpler and more powerful and (b) form new categories using different criteria. Only in special cases it is preferable to record numerical data as categorical, for example, when the measurement is known to be imprecise (number of cigarettes per day, number of cups of coffee per week).

Categorical variables result from registering the presence of an attribute. The categories of a qualitative variable must be clearly defined during the design stage of the research and must be mutually exclusive and exhaustive. This means that each observation unit must be classified unambiguously in one, and only one, of the possible categories and that there is a category to classify each individual. In this sense, it is important to take into account all possibilities when constructing categorical variables, including a category such as “Do not know/No answer,” or “Not Registered,” or “Other,” which ensures that all the observed individuals will be classified based on the criteria that define the variable. Categorical data are also classified as (a) dichotomous, (b) nominal, and (c) ordinal.

(a) Two categories (dichotomous). The individual or observation unit can be assigned to only one of two categories. In general, it is about the presence or absence of the attribute and it is advantageous to assign code 0 to the absence and 1 to the presence.

(b) Examples: (1) resistance—no resistance, (2) disease—no disease, (3) tall—not tall, and (4) red color—no red color. It should be noted that examples 1 and 2 definitely cover all categories, while 3 and 4 are simplifications of more complex categories. In 3 and 4 it was necessary to establish a cutoff criterion to assemble a categorical variable from a numerical variable.
(c) More than two categories. When there are more than two categories, data can be nominal or ordinal. In nominal categories, there is no obvious order between the categories. These types of data values are distinct symbols, and these values serve as labels. The term "nominal" comes from the latin word for "name." Nominal attributes or labels have no relation to one another, nor is any order implied (Patterson and Gibson 2017). Some examples are religion: Catholicism, Islam, Judaism, etc.; race type: African, American, European, Asian, other; type of species; location; plant color; etc. In ordinal data, there is an obvious order between categories. Ordinal values have rank, giving us a notion of order but no concept of distance between the values. We can compare ordinal values with one another, but mathematical operations don't make sense in the context of these values (Patterson and Gibson 2017). Some examples are

1. Drought resistance: no resistance/low resistance/medium resistance/high resistance/total resistance
2. Disease severity: absent/mild/moderate/severe
3. Temperature of a process: hot/mild/cool
4. Social class: lower/middle/upper

Even when ordinal data can be coded as numbers as in the case of stages of drought resistance from 1 to 5 (1 = no resistance, 2 = low resistance, 3 = medium resistance, 4 = high resistance, 5 = total resistance), we cannot say that a plant in stage 4 has a drought resistance twice as strong as the resistance of a plant in stage 2, nor that the difference between stages 1 and 2 is the same as between stages 3 and 4. In contrast, when considering the age of a person, 40 years is twice 20 and a difference of 1 year is the same across the entire range of values. Therefore, we need to be aware that in ordinal data the difference between categories does not make sense.

Ordinal traits are very common in plant breeding programs for measuring disease incidence and severity and for sensory evaluation, such as the perceived quality of a product (e.g., taste, smell, color, decay) and plant development (e.g., developmental stages, maturity). These types of data are often partially subjective since the scale indicates only relative order and no absolute amounts; therefore, the intervals between successive categories may not be the same (Simko and Piepho 2011).

For this reason, we must be careful when dealing with qualitative variables, especially when they have been coded numerically, since they cannot be analyzed as numbers but must be analyzed as categories. It is incorrect to present, for example, the average stage of drought resistance in a group of plants.

In practice, scales are used to define degrees of a symptom or a disease, such as 1, 2, 3, 4, and 5. For this reason, it is important to operationally define this type of variables and study their reliability in order to ensure that two observers placed in front of the same plant will classify it in the same category.

Table 1.2 Examples of multivariate data

Units	Variables	Types of data
Plant	Several measurements of plant height on a single plant in time	All continuous
Animal	Measurement of three animal traits (average daily weight gain, muscularity, and calving)	Mixture of continuous and ordinal
Students	Grades in mathematics, physics, chemistry, biology	All continuous
People	Income, type of residence, gender, educational level, occupation	Mixture of nominal, ordinal, and continuous
Wheat plant	Measurement of four traits: grain yield, panicle number per plant, drought resistance, and type of wheat (common or durum)	Mixture of continuous, discrete, ordinal, and nominal
Country	Several measurements of school performance using the programme for international student assessment (PISA) test	Exam scores continuous in mathematics, reading and science

1.5.2 Multivariate Data Types

Practitioners and researchers in all applied disciplines often measure several variables in each observation, subject, unit, or experimental unit. That is, all variables are simultaneously measured in the same observation. Multivariate data are very common in all disciplines due to the need and facility for data collection in most fields. These variables can consist of only one type of data (for example, plant height measured using a continuous scale on each plant 12 times every 15 days) or a mixture of data types, for example, measuring, on each plant, four different traits: grain yield (on a continuous scale), disease resistance (on an ordinal scale), flower color (nominal scale), and days to flowering (discrete or count). Table 1.2 provides other examples of multivariate data measured using only one scale or a mixture of scales.

It is important to point out that here all measurements are done simultaneously in each observation. For this reason, they are classified as multivariate type of data and include data that will be used as dependent variables or independent variables in the process of training the statistical machine learning algorithms that will be studied here.

1.6 Types of Learning

The three most common ways of learning in statistical machine learning are (a) supervised learning, (b) unsupervised learning, and (c) semi-supervised learning. The three methods are explained below.

1.6.1 Definition and Examples of Supervised Learning

Supervised learning can be defined as the process of learning a function that maps an input to an output based on teaching the statistical machine learning method with input–output pairs. The training data consist of pairs of objects (usually vectors): one component of the pair is the input data (predictors = explanatory variable = input) and the other, the desired results (response variable = dependent variable = output). The output of the function can be a numerical value (as in regression problems) or a class label (as in multinomial regression). The goal of supervised learning is to learn a function that, given a sample of data and desired outputs, best approximates the relationship between input and output observable in the data. This function should be capable of predicting the value corresponding to any valid input object after having seen a series of examples of training data. Under optimal conditions, the algorithm correctly determines the class labels for unseen instances. This implies a learning algorithm that is able to generalize from the training data to unseen situations in a "reasonable" way.

Suppose you're teaching your child to distinguish between corn and tomato (Fig. 1.5). First you show him (her) a picture of an ear of corn and a picture of a tomato. In the learning process, your child must keep in mind that if the color is yellow and the shape is not round, then it is probably an ear of corn, but if the color is red and the shape is round, then it is probably a tomato. This is how your child learns. Then you can show a third picture and ask your child to classify the vegetable as either ear of corn or tomato. When you show the third picture, he (she) will very likely identify if the vegetable is ear of corn or tomato, due to the fact that we have already labeled the two pictures into categories, so your child knows what is an ear of

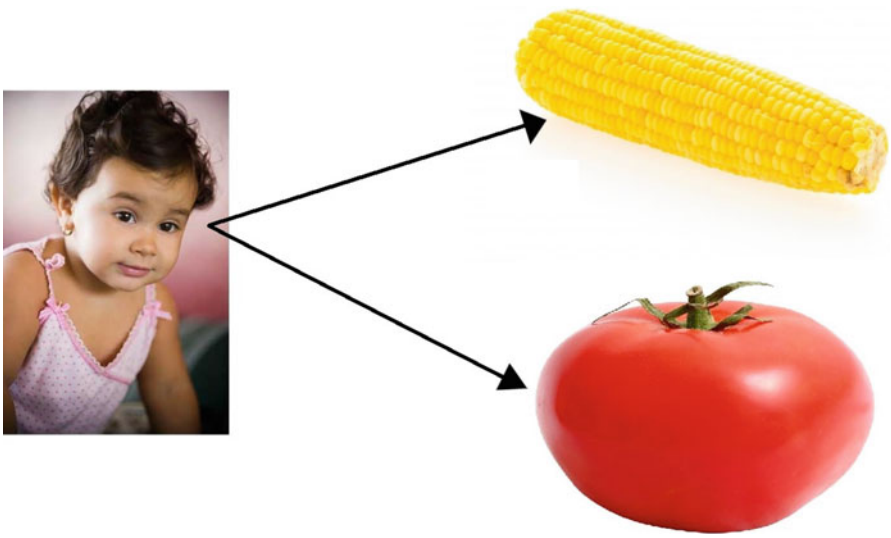

Fig. 1.5 Supervised learning process for teaching a child to distinguish tomato from corn

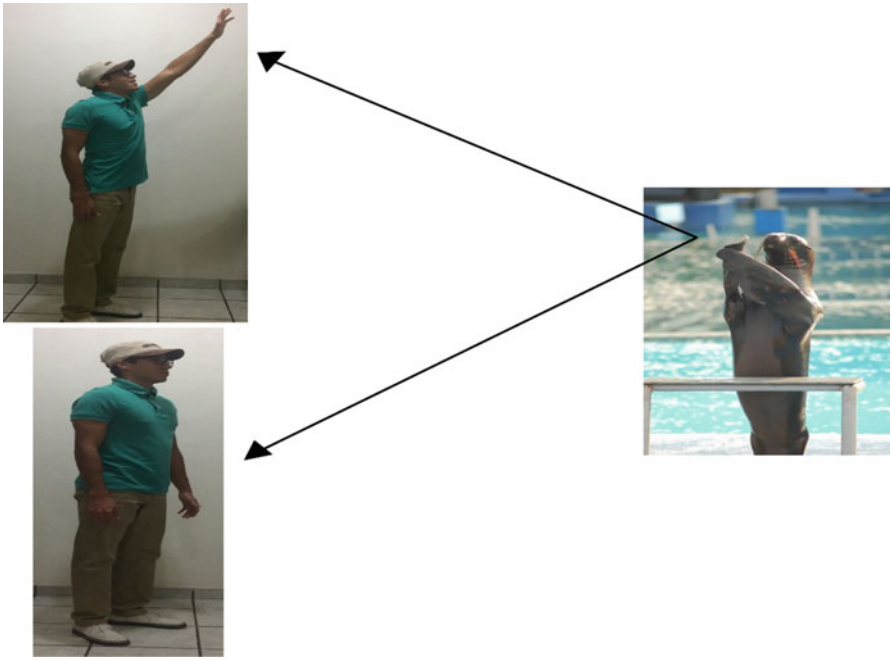

Fig. 1.6 Supervised learning process for teaching a seal to applaud

corn and what is a tomato. This example illustrates how supervised learning works using ground truth data that consist of having prior knowledge of what the output values of our samples should be.

To give another example, imagine that you're training a seal to applaud (Fig. 1.6). The goal is to make the seal applaud when you raise your right hand. The training process consists of presenting the seal with enough examples by raising your right hand and rewarding it with some great candy whenever it applauds when it sees your right hand is raised. In the same way, the seal may be "punished" if it applauds whenever your right hand is not raised, by doing something unpleasant for the seal but not harmful. Supervision involves stimulating the seal to respond to positive samples by rewarding it, and not to respond to negative samples by "punishing" it. Hopefully, the seal then obtains a built-in feeling (hypothesis) for applauding whenever you raise your hand right. The process is evaluated by presenting the seal with another person raising his/her right hand, someone who did not take part in the training process and who is unknown to the seal. However, based on its built-in feeling for what a person with his/her raised right hand looks like, the seal should be able to transfer this knowledge to the present person. It must then consider and decide whether or not it wants to signal the presence of a right hand raised by applauding.

Next we provide some real examples. In the first example, a scientist has thousands of molecules and information about which ones are drugs and he trains

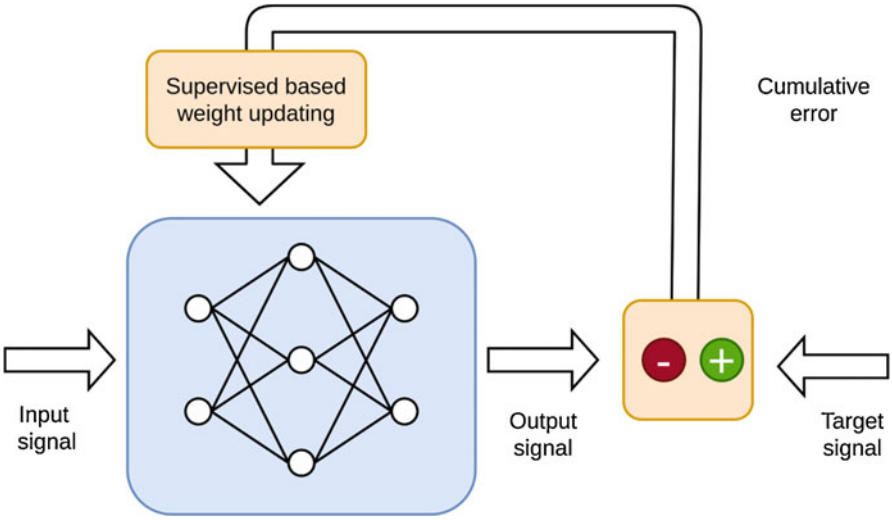

Fig. 1.7 Diagram illustrating a supervised learning process where there are inputs (X) and response variables (y) for each observation

a statistical machine learning model to determine whether a new molecule is also a drug. In the context of plant breeding, a scientist collects hundreds of markers (genetic information) and thousands of images of hundreds of plants. For this sample of plants, he also measures their phenotype (grain yield) because he wants to implement a statistical machine learning algorithm to estimate (predict) the grain yield of new plants not used in the training process. Another example is in environmental science, where scientists use historical data (as when it's sunny and the temperature is higher, or when it's cloudy and the humidity is higher, etc.) to train a statistical machine learning model to predict the weather for a given future time.

In a more mathematical way, under supervised learning, we usually have access to a set of p predictor (input) variables $X_1, X_2, \ldots, X_p$ measured in n observations, and a response variable (output) Y also measured in those same n observations (Fig. 1.7). The goal is to predict Y using a function of $X_1, X_2, \ldots, X_p$, that is, we use an algorithm to learn the mapping function from the input to the output $Y = f(X_1, X_2, \ldots, X_p)$, and we expect to estimate the mapping function so well that when we have new input data, we can predict the output variables for those data. The term supervised learning was coined because the learning process of any statistical machine learning method from the training dataset can be thought of as a teacher supervising the learning process. We know the correct outputs (response variables), the algorithm iteratively makes predictions on the training data and is corrected by the teacher. Learning stops when the statistical machine learning algorithm achieves an acceptable level of performance.

1.6.2 *Definitions and Examples of Unsupervised Learning*

Unsupervised learning is when you only have input (predictors = independent variables) data (X) and no previous knowledge of corresponding labeled outputs or response variables (Fig. 1.8). So its goal is to deduce the natural structure present within a set of data points. In other words, to extract the underlying structure or distribution in the data in order to learn more about the data, that is, the network uses training patterns to discover emerging collective properties and organizes the data into clusters. In unsupervised learning (unlike supervised learning), there is no correct answer (output = response variable = dependent variable) and there is no teacher. For this reason, we are not interested in prediction since we do not have an associated response variable Y. Statistical machine learning algorithms under unsupervised learning are left to their own devices to discover and present the interesting structure in the data. However, there is no way to determine if our work is correct since we don't know the right answer because the job was done without supervision. Unsupervised learning problems can be divided into clustering and association problems.

Clustering: A clustering problem is when you want to discover the inherent groupings in the data, such as grouping maize hybrids by their genetic architecture. Another example is grouping people according to their consumption behaviors. But in both cases we cannot check if the classifications are correct since we don't know the true grouping of each individual.

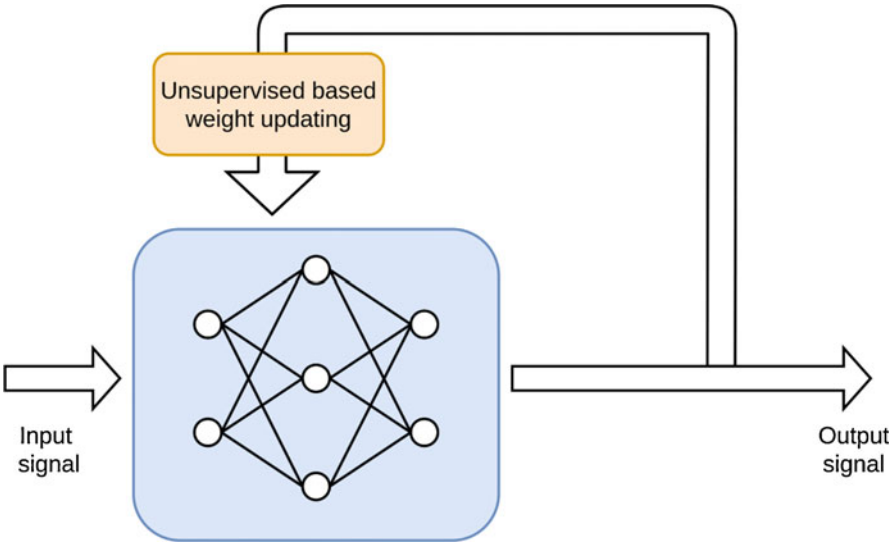

Fig. 1.8 Diagram illustrating an unsupervised learning process where there are only inputs (X), but no response variables (y) for each observation

Association: An association rule learning problem is when you want to discover rules that describe large portions of your data, such as people who buy X also tend to buy Y.

Some popular examples of unsupervised learning algorithms are

(a) Principal component analysis
(b) Multidimensional scaling for grouping
(c) A k-means algorithm for clustering problems
(d) An a priori algorithm for association rule learning problems

1.6.3 Definition and Examples of Semi-Supervised Learning

Semi-supervised learning problems are those that have a large amount of input data (X) available but only some of the data are labeled (Y). For this reason, these problems are positioned between supervised and unsupervised learning. A good example is plant species classification using thousands of images where only some of the images are labeled (e.g., species 1, species 2, species 3, etc.) and the majority are unlabeled. Another example is the classification of exoplanets (exoplanets are planets that are outside our solar system) also using thousands of photos where only a small fraction of the photos is labeled (four types of exoplanets). Many real-world problems in the context of statistical machine learning belong to this type of learning process. This is because it is more expensive and time-consuming to use labeled data than unlabeled data since many times this requires having access to domain experts, whereas it is cheap and easy to collect and store unlabeled data.

References

Adler I (1912) Primary malignant growths of the lungs and bronchi: a pathological and clinical study. Longmans, Green and Co, New York and London, p 325

Bernardo R (2016) Bandwagons I, too, have known. Theor Appl Genet. https://doi.org/10.1007/s00122-016-2772-5

Box GEP (1976) Science and statistics (PDF). J Am Stat Assoc 71:791–799. https://doi.org/10.1080/01621459.1976.10480949

Breiman L (2001) Statistical modeling: the two cultures. Stat Sci 16:199–215

Crossa J, Pérez-Rodríguez P, Cuevas J, Montesinos-López OA, Jarquín D, de Los Campos G, Burgueño J, González-Camacho JM, Pérez-Elizalde S, Beyene Y, Dreisigacker S, Singh R, Zhang X, Gowda M, Roorkiwal M, Rutkoski J, Varshney RK (2017) Genomic selection in plant breeding: methods, models, and perspectives. Trends Plant Sci 22(11):961–975

Dean J (2018) Big data, data mining, and machine learning. Value creation for business leaders and practitioners. John Wiley & Sons, Inc., Hoboken

FAO (2011) The state of the World's land and water resources for food and agriculture: managing Systems at Risk. Food and agriculture Organization of the United Nations. FAO, Rome

Fischer T, Byerlee D, Edmeades G (2014) Crop yields and global food security. ACIAR, Canberra

James G, Witten D, Hastie T, Tibshirani R (2013) An introduction to statistical learning: with applications in R. Springer, New York

McKinsey Global Institute (2016) The age of analytics: competing in a data-driven world. https://www.mckinsey.com/~/media/mckinsey/business%20functions/mckinsey%20analytics/our%20insights/the%20age%20of%20analytics%20competing%20in%20a%20data%20driven%20world/mgi-the-age-of-analytics-executive-summary.ashx

Meuwissen THE, Hayes BJ, Goddard ME (2001) Prediction of total genetic value using genome-wide dense marker maps. Genetics 157:1819–1829

Milliken GA, Johnson DE (2009) Analysis messy of data, volume 1 designed experiments. CRC Press Taylor & Francis Group, Boca Raton, London, New York

Montesinos-López A, Martín-Vallejo J, Crossa J, Gianola D, Hernández-Suárez CM, Montesinos-López OA, Juliana P, Singh R (2019) A benchmarking between deep learning, support vector machine and Bayesian threshold best linear unbiased prediction for predicting ordinal traits in plant breeding. G3 9(2):601–618

Oury F-X, Godin C, Mailliard A, Chassin A, Gardet O, Giraud A, Heumez E, Morlais J-Y, Rolland B, Rousset M, Trottet M, Charmet G (2012) A study of genetic progress due to selection reveals a negative effect of climate change on bread wheat yield in France. Eur J Agron 40:28–38

Patterson J, Gibson A (2017) Deep learning: a Practitioner's approach. O'Reilly Media, Beijing

Pinheiro JC, Bates DM (2000) Mixed-effects models in S and S-PLUS. Springer Verlag, New York

Proctor RN (2012) The history of the discovery of the cigarette-lung cancer link: evidentiary traditions, corporate denial, global toll. Tob Control 21(2):87–91

Samuel AL (1959) Some studies in machine learning using the game of checkers. IBM J Res Dev 3 (3):210–229

Schuster PM (2014) The scientific life of Victor Franz (Francis) Hess (June 24, 1883–December 17, 1964). Astropart Phys 53:33–49

Sejnowski TJ (2018) The deep learning revolution. The MIT Press, Cambridge, MA, London

Shmueli G (2012) To explain or to predict? Stat Sci 25(3):289–310. https://doi.org/10.1214/10-STS330

Simko I, Piepho H-P (2011) Combining phenotypic data from ordinal rating scales in multiple plant experiments. Trends Plant Sci 16:235–237

Stroup W (2012) Generalized linear mixed models: modern concepts, methods and applications. CRC Press, Boca Raton

Wang X, Xua Y, Hu Z, Hu C (2018) Genomic selection methods for crop improvement: current status and prospects. Crop J 6(4):330–340

Wolpert DH (1996) The lack of a priory distinction between learning algorithms. Neural Comput 8 (7):1341–1390