Noname manuscript No.
(will be inserted by the editor)

The Big Data razor

Ezequiel LÓPEZ-RUBIO

the date of receipt and acceptance should be inserted later

Abstract Classic conceptions of model simplicity for machine learning are mainly based on the analysis of the structure of the model. Bayesian, Frequentist, information theoretic and expressive power concepts are the best known of them, which are reviewed in this work, along with their underlying assumptions and weaknesses. These approaches were developed before the advent of the Big Data deluge, which has overturned the importance of structural simplicity. The computational simplicity concept is presented, and it is argued that it is more encompassing and closer to actual machine learning practices than the classic ones. In order to process the huge datasets which are commonplace nowadays, the computational complexity of the learning algorithm is the decisive factor to assess the viability of a machine learning strategy, while the classic accounts of simplicity play a surrogate role. Some of the desirable features of computational simplicity derive from its reliance on the learning system concept, which integrates key aspects of machine learning that are ignored by the classic concepts. Moreover, computational simplicity is directly associated with energy efficiency. In particular, the question of whether the maximum possibly achievable predictive accuracy should be attained, no matter the economic cost of the associated energy consumption pattern, is considered.

Keywords model simplicity · machine learning · Bayesianism · information theory · energy efficiency

This is a preprint of the paper published in the European Journal for Philosophy of Science. The final published version is at https://dx.doi.org/10.1007/s13194-020-00288-8

Ezequiel LÓPEZ-RUBIO
Departamento de Lenguajes y Ciencias de la Computación, Universidad de Málaga (UMA), Bulevar Louis Pasteur 35, 29071 Málaga, Spain
Departamento de Lógica, Historia y Filosofía de la Ciencia, Universidad Nacional de Educación a Distancia (UNED), Paseo de Senda del Rey 7, 28040 Madrid, Spain
email: ezeqlr@lcc.uma.es
Telephone: +34 95 213 71 55
Fax: +34 95 213 13 97

1 Introduction

Automated model selection is one of the most relevant features of machine learning. It gives the scientist a powerful tool to quantitatively assess the merits of several possible models in the light of their predictive performance when fitted to large volumes of data. In the current Big Data age, many scientific and engineering endeavors involve the execution of machine learning software to obtain a fitted model, with little or no human intervention. This calls for an analysis of the criteria which are employed in such software to choose one model over another. Simplicity is often employed to justify these selections, but the concept of simplicity has different meanings depending on the school of thought that a machine learning practitioner adheres to. Here we aim to explain such differences and their associated underlying assumptions about the goals of model selection. Furthermore, we describe and discuss the computational concept of simplicity, which is frequently applied in practice but often neglected in the literature. Finally, we claim that the computational concept of simplicity is more appropriate to understand current practices in machine learning than the classic ones. In this work, we focus on model selection for machine learning, and not for general scientific inference.

In what follows we will use the following terminology. A model is a mathematical structure that aims to fit some experimental data. A model contains zero, one or more adjustable (also called learnable) parameters, which are real numbers that must be determined. Therefore, a model with one or more parameters has infinite possible realizations[1], which we call instantiations. A machine learning algorithm takes a model and a set of training data as inputs and returns a set of values for the adjustable parameters of the model, i.e. an instantiation of the model. We also say that the algorithm fits the model to the data so that a fitted (instantiated) model is obtained.

We start by presenting four classic concepts of simplicity which have been applied to machine learning models (Section 2). Then the relations among machine learning models, the learnable parameters that they contain, and the learning algorithms which are used to adjust the parameters are discussed, along with the concept of a learning system, which contains them (Section 3). After that, the proposed concept of computational simplicity is detailed, and its epistemic advantages with respect to the classic ones (Section 4). The relations among the classic and computational approaches of simplicity are studied in Section 5. Next, some non epistemic justifications of the proposed computational simplicity concept are outlined (Section 6). Finally, Section 7 concludes this work by highlighting the relevance of computational simplicity to state of the art machine learning.

2 Four brands of simplicity for machine learning models

Ockham's razor is invoked by machine learning theorists to prefer simpler models:

> Entities should not be multiplied beyond necessity.

[1] An uncountable infinity, since the parameters are real numbers. In the particular case that all adjustable parameters are constrained to be integers, the number of possible realizations is a countable infinity.

It must be understood as a search strategy to extract good models from data, and not as a statement that Nature must actually be simple (de Rooij and Grünwald, 2011, p. 893). It has been argued that, even if it does not lead to the true model, it finds models that yield reliable predictions (de Rooij and Grünwald, 2011, p. 894).

As an example, let us consider a typical application of machine learning, namely the modeling of customers of an e-commerce site. Given the complexity of human behavior, it is almost impossible to build machine learning models that capture all the details of this problem. Therefore, machine learning practitioners assume that the true model is not attainable. Nevertheless, under the Ockam's razor principle, it is possible that a simple model can be obtained by learning from customer activity data collected by the web site software. Such a model might be good enough to produce accurate predictions of customer actions like buying a certain product or revisiting the site.

For instance, it could be found in the collected data that 90% of the customers that have already purchased items in the site for a total amount of more than 2,000$ actually revisit the site. On the basis of such an observation, a simple decision tree model might be learned by a machine learning algorithm that contains a rule which states that if a customer has purchased more than 2,000$ on the site, then she is predicted to revisit. Here the amount 2,000$ is an adjustable parameter which is learned from the data.

However, there are many possible ways to define simplicity in the machine learning context (Domingos, 1999, p. 409; Kelly, 2007, p. 270). Next, we review four of them.

2.1 The Bayesian concept

The standard view in the Bayesian school understands probabilities as degrees of belief in the truth of a statement (Sober, 2015, p. 64; Forster, 2001, p. 88). Bayesian model selection chooses the model which has the highest probability given the data (Wasserman, 2000, p. 93). In order to compare two models, their Bayes factor can be computed, which is the evidence of one of the models versus the other (Wasserman, 2000, p. 98). This calculation requires the assumption of prior probability distributions for the models and their parameters.

The Bayesian Information Criterion (BIC) is the criterion of choice according to Bayesianism (Sober, 2015, p. 135). It is an approximation to the log Bayes factor (Wasserman, 2000, p. 100) that has the advantage that no prior information must be supplied since only the maximum likelihood instantiation of each model is necessary (Claeskens and Hjort, 2008, p. 81). The BIC is an unbiased estimator of the probability of observing the data given the model (Sober, 2015, p. 139). After that, the model associated with the highest posterior probability of observing the data is chosen (Claeskens and Hjort, 2008, p. 79). The maximization of this posterior probability automatically leads to a balance between the goodness of fit of the model to the data and the complexity of the model, which is known as the Bayesian Ockam's razor (Huang and Beck, 2018, pp. 712-713; Murphy, 2012, p. 157). The Bayesian framework favors models with fewer adjustable parameters because they concentrate the prior probability mass in a smaller range of options (Sober, 2015, p. 125). In other words, simpler models have a smaller number of

possible instantiations, which means that more prior probability mass is allocated to each instantiation (Henderson et al, 2010, pp. 186-187).

It is sometimes interpreted that the Bayes factors or the BIC select the model which is believed to be true with the highest probability (Grünwald, 2007, p. 540). However, this interpretation fails when a model is compared to a restriction of it (a submodel). From the axioms of probability, it follows that the submodel cannot have a higher probability than the model, but Bayesian model selection sometimes chooses the submodel (Forster, 2001, p. 95).

Bayesian model selection might be employed to predict whether a customer will revisit our example e-commerce site, given her past activity on the site. Typically the competing models have several learnable parameters that are adjusted from the data collected by the web site software about many customers over time. Given the history of a customer, each model will output the probability that the customer revisits the site. Then the acquired data about the customers who actually revisited the site could be used, so that the BIC applies the Bayesian Ockam's razor to choose the model which attains the best balance between the model accuracy, i.e. how the predicted revisit probabilities match the actual revisit data, and the model complexity, i.e. the amount of parameters to be adjusted from the customer activity data. This selection is based on the maximization of the posterior probability of observing the revisit data.

2.2 The Frequentist concept

The Frequentist approach holds that probability only has a meaning when it refers to a repeatable experiment (de Rooij and Grünwald, 2011, p. 895). Therefore, the Bayesian evaluation of a model, based on the probability of the model given the observed data, does not make sense under this interpretation. Frequentism conducts model assessment without considering subjectively assigned prior probabilities, which enhances the perceived objectivity of its conclusions, as compared to Bayesianism (Dawid, 2017, pp. 378-381). The AIC is the best known Frequentist criterion (Sober, 2015, pp. 128-135). The AIC does not assign probabilities to models since it is based on an unbiased estimation of the predictive accuracy of each model.

The AIC favors models with fewer adjustable parameters because it is based on an unbiased estimate of the predictive accuracy of a model which contains the number of adjustable parameters with a negative sign. This means that the lower the number of adjustable parameters, the higher (better) the AIC. This mathematical fact reflects the experience of scientists when they try to employ a model which fits the training data very well but performs poorly on new test data (Sober, 2015, pp. 130-131). In machine learning terms, it is said that a model with too many adjustable parameters overfits the data.

While both AIC and BIC favor models with fewer adjustable parameters, they diverge in their interpretation of Ockam's razor. The main difference between Frequentist and Bayesian model selection is that the Bayesian approach intends to maximize the probability of the model given the data so that it assigns probabilities to models. In the Bayesian context, more adjustable parameters mean lower model probability, and this is provided as the Bayesian foundation of Ockam's

razor (Sober, 2015, p. 141). The Frequentist approach refrains from model probability assignments, which means that models with more adjustable parameters are penalized on the grounds that they are estimated to have lower predictive accuracy, irrespective of their probability. Therefore, the Frequentist justification of Ockam's razor refers to the estimated accuracy of the models. As mentioned in (Sober, 2015, p. 149), model parsimony is used as a surrogate for accuracy. Following our previous account of Bayesian model selection for e-commerce customer revisit prediction, the AIC differs from the BIC in that the AIC chooses the model that is estimated to have the highest predictive accuracy for future customer revisits, irrespective of any model probabilities.

Accuracy is what really matters, and this explains why the AIC and the BIC are less used than direct estimations of predictive accuracy obtained by cross validation. The AIC is a parsimony based, indirect estimator of predictive accuracy (Bandyopadhyay and Forster, 2011, p. 3). The BIC does not even aim to maximize predictive accuracy, but it evaluates the evidence for a model given the observed data (Wasserman, 2000, pp. 99-100). In contrast to these criteria, cross validation actually measures the predictive accuracy over validation data sets. The predictive accuracy measured by cross validation over validation data sets often turns out to be a better estimator of the performance of a model on a test set than BIC or AIC (Hastie et al, 2009, p. 254). Moreover, cross validation can be used for non probabilistic models (Murphy, 2012, p. 370). All of this assumes that predictive accuracy is the primary goal. Therefore, machine learning practitioners mostly adhere to an instrumentalist conception of science (Sober, 2015, pp. 143-144). Parsimony is seen as a secondary goal so that if the predictive accuracy is similar among several candidate models, the candidate with the smallest number of adjustable parameters is chosen (James et al, 2014, p. 214).

2.3 The information theoretic concept

Information coding is at the root of the Minimum Description Length (MDL) concept of simplicity, see (Montañez, 2017, pp. 73-75; Domingos, 1999, pp. 412-413). Under the MDL framework, probability distributions are equivalent to codes, so that the truth or the belief in the truth of a model is not relevant (de Rooij and Grünwald, 2011, p. 895). MDL aims to minimize the sum of the number of bits that are required to represent the data plus those required to represent the model (Domingos, 1999, p. 412; Pothos and Wolff, 2006, p. 213; Grünwald, 2007, p. 132). Hence MDL can be decomposed into a model fit term plus a model simplicity term (Montañez, 2017, p. 73). The MDL principle is based on the observation that any regularities on the data enable efficient compression of such data. Models with fewer adjustable parameters require fewer bits to be coded. This means that more parsimonious models are preferred, provided that the models adequately capture the underlying patterns in the data so that the data are efficiently compressed.

Both MDL and BIC have been proved to behave suboptimally when the true model does not belong to the set of models to choose from (Grünwald, 2007, p. 530), a situation that occurs very frequently in practice (Grünwald and Langford, 2007, p. 139). The main drawback of MDL is that, while models that are good at prediction implicitly compress the training data by identifying data patterns (Grünwald, 2007, p. 595), there are models which compress the training data ad-

equately but do not perform well at predicting new test data. In other words, implicit training data compression is necessary, but not sufficient, to attain good predictive performance. For the customer revisit example, this is a significant impediment to employ MDL since the true model does not belong to the set of models under comparison. This means that MDL could choose a model which summarizes the observed customer behavior data very efficiently, but predicts future revisits poorly.

2.4 The expressive power concept

Perhaps the most typical task in machine learning is classification. It consists of predicting a class label given an input vector which lies in some input space comprising several features. Classification problems greatly vary in their difficulty. The most difficult problems are those where nearby vectors in the input space have different class labels. This intricacy of the distribution of the labels must be matched by the classification models which aim to solve the problem. That is, a classification model must have enough expressive power to represent the intricate boundaries among the classes in the input space. This observation has led to some measures of model simplicity which are based on the richness of the kind of class boundaries that they can represent.

The best known of such measures is the Vapnik-Chervonenkis (VC) dimension. The VC dimension of a classification model is defined as the largest number of input vectors which can be shattered by the model, where the set of points is shattered if the model can learn a boundary which perfectly separates them no matter how the class labels are assigned to the points (Hastie et al, 2009, p. 238). This way, a researcher can evaluate the relative simplicity of two classification models by comparing their VC dimensions. Models with excessive VC dimension should be avoided because they might overfit (Vapnik, 2000, pp. 297-298), i.e. they could focus on irrelevant features of the input dataset.

A key shortcoming of the VC dimension is that it does not assume any form of the distribution of the input vectors and their labels. This implies that it does not consider any kind of regularity in such distribution, while real problems do exhibit strong regularities. For example, e-commerce customer activity contains significant patterns, such as daily and seasonal activity trends, which are ignored by the VC dimension theory. Therefore the estimation by the VC dimension of the capability of a model to solve a classification problem is often too conservative, which greatly limits its applicability (Bishop, 2006, pp. 344-345; Shalev-Shwartz and Ben-David, 2014, p. 116).

Another quantitative measure of the flexibility of a classification model to learn an input distribution is Rademacher complexity. It measures to which extent, in average terms, a model can fit the random noise that may corrupt a given input distribution (Mohri et al, 2014, pp. 34-35). A higher Rademacher complexity indicates that the model is more rich and flexible so that it can accommodate more intricate input distributions. It is defined with respect to a certain input distribution, so it takes into account the regularities of the input, unlike the VC dimension. However, Rademacher complexity is known to be hard to compute (Oneto et al, 2018, p. 4660). In particular, it is more difficult to bound or estimate than the VC

dimension, and in some cases it is NP-hard, i.e. it requires an exponential number of calculations (Mohri et al, 2014, pp. 33, 38).

3 Models, learnable parameters and learning algorithms

In this section, I will argue that in machine learning the complexity of models is inextricably linked to the complexity of the algorithms that learn them. Software is made of algorithms plus data structures (Wirth, 1985):

> Programs, after all, are concrete formulations of abstract algorithms based on particular representations and structures of data.

In prediction software, the data structures are the model. In this work I refer to the computational complexity of a specific algorithm, also called algorithm efficiency (Levitin, 2011, p. 42), and not to the (more abstract and elusive) computational complexity of a problem, also called intrinsic complexity of a problem (Moore and Mertens, 2011, p. 23). Computational efficiency is acknowledged as a criterion to justify the suitability of software (Kelly, 2007, pp. 271, 273). The mere fact that the problem of learning a certain model can be solved, which is all that the simplicity concepts considered in Section 2 care about in terms of computation, says very little about the potential of the model to be applied in practice. There are many possible algorithms to learn a model, and for each learning algorithm, there is an infinity of possible programs that implement it, each with its own computational complexity. Therefore, in machine learning, the sole specification of a learnable model is not enough to assess simplicity (Kelly, 2007, p. 273). Model evaluation, model selection, and algorithm selection are seen as different aspects of the same task (Raschka, 2018, p. 1). The number of parameters of the model and the speed with which those parameters can be learned are seen as two equally important factors (Lin and Tegmark, 2016, p. 2). The object whose simplicity should be assessed is the overall *learning system*, which is composed (at least) of:

1. The data acquisition process.
2. The hardware where the computations are performed.
3. The learnable model.
4. The learning algorithm (software) which is executed on the hardware, with the acquired data as input, and outputs the learned instantiation of the model.
5. The prediction procedure (software), executed on the hardware, which accepts the learned instantiation of the model and a query as inputs, and outputs a prediction.

Hardware is also important to assess the simplicity of a learning system, because models which are unfeasible to learn on a certain kind of hardware, become easy to use on other kinds of hardware. The differences in the computational complexity among implementations of the same algorithm depending on hardware considerations can be enormous (Parashar et al, 2019, p. 305). This is what happens with deep learning neural networks, which are too complex to be learned on standard hardware, but affordable on specific graphics hardware. However, in this work, I will focus on the importance of learning algorithms to evaluate simplicity.

Let us consider an example of a learning system, namely a recommender system for our example e-commerce site. The data acquisition consists of recording

successive purchase transactions by the same customer so that a register of the purchase history of many customers is collected. This data acquisition process requires dedicated software, databases, and computers. Some additional hardware and software are necessary to run the machine learning methods themselves. Next, a data scientist chooses a learnable model that might be adequate to predict, given the past behavior of a customer, which items she would be interested to buy in the future. After that, the learnable model is trained on the collected purchase histories by a suitable learning algorithm. Finally, the trained model is employed to execute a prediction procedure to show a visiting customer some items which she might like to purchase.

Inspired on the Inference to the Best Explanation (IBE) principle (Cabrera, 2017, p. 1248), we can define the Refinement to the Best Prediction (RBP) scheme:

1. D is a set of data.
2. Software S_1 predicts D sufficiently well.
3. No competing software $S_2, ..., S_N$ predicts D better than S_1.
4. One is justified in using S_1 to predict D.

Please note that truth does not have an explicit role here. It is not a logical inference process, but a software refinement one. Maybe the model which exists in S_1 is closest to the underlying real process which D comes from, but this is not a goal. In other words, machine learning allows that the models inside the S_i are black boxes with little or no resemblance to the physical process which generated the data D. Once the data are generated in step 1, the underlying physical process is irrelevant to the subsequent steps of the RBP scheme. In this context, the computational efficiency of the prediction software becomes a key factor to choose a learning system over another.

Learnable models contain a certain number of learnable parameters whose values must be estimated from the data. The first three classic approaches considered in Section 2 (Bayesian, frequentist and information-theoretic) evaluate the simplicity of the learned model with all their adjusted learnable parameters with respect to a space of possible models, but they completely ignore the computational procedure by which such learnable parameters have been determined. Furthermore, these three classic approaches are designed to measure the complexity of what are known as parametric models, as opposed to nonparametric models. On one hand, parametric models are based on the assumption that the observed data can be explained and summarized by a probability distribution of a specific mathematical form which is characterized by a relatively small number of learnable parameters. On the other hand, nonparametric models do not make assumptions about the mathematical form of the probability distribution which underlies the observed data nor they try to summarize the available dataset. This means that nonparametric models grow with the size of the dataset, while they have very few learnable parameters, often just one learnable parameter. The neglection of nonparametric models causes the failure of classic simplicity concepts to account for nonparametric model selection (Rochefort-Maranda, 2016, p. 270; de Rooij and Grünwald, 2011, p. 892). Moreover, these three classic conceptions of simplicity are useless to explain how scientists choose among a range of parametric and nonparametric models. That is, parametric and nonparametric models are incommensurable under these classic simplicity models. The situation only gets worse for machine learning models which do not define any probability distribution. Notably enough,

this class of models includes many of the most celebrated machine learning models[2]. In these cases, the AIC and the BIC cannot even be defined.

The fourth kind of classic simplicity measures, namely the expressive power ones (Subsection 2.4) also has important shortcomings. Like the previous ones, they do not consider the computational procedure that is followed to learn a model. The VC dimension completely ignores the strong regularities which often exist in real datasets D, which leads to inaccurate assessments of the capability of the models to learn such real datasets. Rademacher complexity is hard to compute, which hampers its practical applicability. Furthermore, both VC dimension and Rademacher complexity are defined for supervised learning problems where the desired output is provided by a supervisor, so that the simplicity of models for many fundamental tasks in machine learning such as unsupervised clustering, dimensionality reduction, density estimation, and reinforcement learning cannot be analyzed with them. In contrast to this, computational efficiency can be employed to assess the simplicity of all kinds of machine learning models.

4 Computational simplicity

As seen in Section 3, classic conceptions of simplicity are incomplete because they do not address the computational burden of learning the adjustable parameters of a model. Here we propose an alternative concept of simplicity which works for parametric, nonparametric, non probabilistic, unsupervised and reinforcement learning models. It is founded on a new version of Ockham's razor, that we may call the Big Data razor:

Definition 1 *Big Data razor.* Computations should not be multiplied beyond necessity.

Computational simplicity has already been recognized as one of several alternative concepts of simplicity (Rochefort-Maranda, 2016, pp. 271-272). It has also been pointed out that, while favoring simplicity does not necessarily lead to the true model, it speeds up the search process (Kelly, 2011, p. 1000), which highlights the importance of the computational load associated with model selection. Here we aim to argue that computational simplicity is a critical criterion for current machine learning.

Computational limitations have always played a role in machine learning. The novelty is that before Big Data, the scalability of algorithms as the number of samples grows was not so important because most datasets were small. This implies that the relative importance of computational complexity was moderate, as compared to classic measures of model complexity. Nowadays sample sizes N of 10^9 to 10^{11} are common (Cahsai et al, 2017, p. 1419, 1426; Hestness et al, 2017, pp. 5-10), and they are becoming even more frequent. For example, more than 10^9 items are shipped by the same e-commerce site per year (Carman, 2018). This means that machine learning models whose computational requirements scale badly to

[2] The class of non probabilistic machine learning models includes: Support Vector Machines, k-means, decision trees, random forests, artificial neural networks (including deep learning neural networks) and many others. Most of them can be adapted to output probabilities, but they are usually employed without such adaptation, i.e. the model selection is carried out without any reference to probabilities.

those sample sizes are simply discarded, no matter how simple their structures may be, because only computationally cheaper models can harness the incoming data deluge.

An example of the Big Data razor is the success of two deep learning neural networks to detect objects in images: FasterRCNN (Ren et al, 2017) is more accurate than Yolo (Redmon and Farhadi, 2018), but Yolo is faster (Dhiraj and Jain, 2019, p. 118). FasterRCNN follows a two step approach (Ren et al, 2017, pp. 1138-1139). First, it generates a relatively large number of region proposals, i.e. rectangles (also called bounding boxes) which might enclose an object. After that, it analyzes the region proposals to estimate how likely they are to actually contain an object. In contrast to this, Yolo performs the object detection in a single step, since it directly generates regions which are very likely to enclose an object. Furthermore, Yolo comes in several versions that accept images of different resolutions. The higher the resolution, the more accurate the object detection, but the larger the network and its associated computational load. That is, Yolo versions are in a relation of computational load versus accuracy tradeoff (Redmon and Farhadi, 2018, p. 4). Due to its two step architecture, FasterRCNN is slower than Yolo, but FasterRCNN is more accurate because its analysis of the input image is more detailed. Depending on the time requirements of the application, one of them is chosen: FasterRCNN or some of the Yolo versions. Even for the same network, some implementations are more or less accurate depending on the object detection accuracy that you wish to attain (Kang et al, 2018; Ma et al, 2018). In general terms, it can be said that the more accuracy, the more calculations are required (Huang et al, 2017, p. 7315). A linear relationship has been experimentally found between the accuracy and the number of images that can be processed per unit time (Canziani et al, 2016, p. 6). A similar tradeoff is found in classification. The more calculations, the more accuracy which can be attained (Kpotufe and Verma, 2017, pp. 1, 16; Jose et al, 2013, pp. 1, 8).

The above examples illustrate the fact that in machine learning often predictive accuracy and computational simplicity are competing goals which stand in a relation of tradeoff, while classic concepts of simplicity (Bayesian, Frequentist, information theoretic and expressive power) are not considered.

There is a practical reason for this quest for computational simplicity. Computation has an economic cost in terms of hardware, software, and energy. In our example recommender system for an e-commerce site, there is a need for machine learning algorithms that can be executed with a small computational effort, so that cheaper web servers are required and less electrical power is employed to supply them. All of this reduces the monetary cost of running the web site, i.e. the profits are increased. Therefore less computation means the cheaper application of machine learning to an ever growing range of tasks (Agrawal et al, 2018, ch. 3). In turn, this economic rationale directs researchers and practitioners towards computationally simple models. This justification of simplicity departs from the epistemic justifications of the classic concepts of simplicity. These classic justifications are problematic and suggest that parsimony is a surrogate goal (Sober, 2015, p. 149).

Since predictive accuracy comes at a computational (and hence economic) cost, for the Big Data era the models which are obtained by a learning algorithm with a computational complexity which is higher than linear are not practical (Hong et al, 2019, p. 1; Burkov, 2019, ch. 8). It is not only that the space of possible models is

infinite (Korb, 2004, p. 437), which was already known before the advent of Big Data. It is also that many of the possible models cannot be learned (adjusted) within practical computer resource limits.

The old (Bayesian, statistical) balance was between overfitting and underfitting, i.e. between the simplicity of the model and its predictive accuracy. While the old balance is still valid for parametric models, there is a new balance between the predictive accuracy of the learned model and the computational complexity of the associated learning algorithm, measured in terms of computation time and memory usage (Jiang et al, 2019, p. 201). The new balance is not restricted to parametric models since it applies to nonparametric and non probabilistic models too. The computational complexity of the test phase, i.e. when the learned model is employed to generate predictions, is also very important. This is because the test phase might be more computationally demanding than the training phase if the learned model is to be maintained for some time to yield many predictions. In other words, the relative importance of the computational complexity of the training and test phases depends on the expendability of the learned model.

The computational simplicity concept, considered in the context of the learning system depicted in Section 3, can help to explain why Big Data calls for models whose learning algorithms have linear complexity (or lower). To a first approximation, it can be assumed that the energy (and monetary cost) of acquiring N samples of data is directly proportional to N, i.e. it is linear with N. If the learning algorithm has linear complexity, this implies that the ratio between the energy required to learn the model and the energy required to acquire the data is a constant independent of N. However, if the learning algorithm has a complexity which is higher than linear, then the ratio tends to infinity as N grows. Hence the learning phase absorbs an unsustainable fraction of the energy devoted to run the entire learning system, as the number of available data samples N increases. For our example recommender system, this means that the energy employed to predict the future behavior of the customers grows much larger than the energy devoted to processing their actual purchase orders. This situation is characteristic of the Big Data era since the rate of growth of the amount of computation required to learn state of the art machine learning models has dramatically accelerated since 2012 (Amodei et al, 2019).

In the light of the above considerations, it can be inferred that a triple balance must be attained among three variables: the cost of acquiring the data, the cost of learning the model, and the cost of using the model to make accurate predictions. The computational simplicity concept can provide a unified framework to understand the two last variables, while the first one (the data acquisition cost) depends on the scientific or technological field that the models are applied to.

Parsimony in terms of the number of adjustable parameters is not a goal in itself, although it affects the memory usage and indirectly the computation time. In other words, each adjustable parameter occupies some memory space, so models with more parameters have more memory usage. Besides, each adjustable parameter requires some computational effort to learn it, so adding more parameters to a given model usually results in higher computation time. Truth is not a goal either since it is assumed that none of the competing black box models reflects the actual structure of the problem. As mentioned before, the exact behavior of the customers of an e-commerce site is too complex to be captured by a machine learning model. There was a probabilistic turn (Sober, 2015, p. 152) in the 20th

century which meant that we did not assume anymore that nature is simple or that it is probably simple. The Big Data turn of the 21st century means that machine learning practitioners assume that models whose computational complexity is small enough to manage large and ever increasing volumes of data have the best chances to generate good predictions. Computationally simple algorithms which performed poorly with small amounts of data yield excellent results when supplied with large datasets, and it is believed that some problems can be essentially solved as soon as enough data is provided (Pereira et al, 2009, p. 9; Sun et al, 2017, p. 8). It is acknowledged that these simple models are not true. In other words, after the Big Data turn model selection is not driven by purely theoretical motivations, because the cost of running the overall learning system has become a fundamental factor to choose one model over another. It must be noted that before the advent of massive data processing by machine learning methods, the computational cost of adjusting the parameters of a model was not a pressing concern for machine learning practitioners. Nowadays, a standard strategy to cope with a given dataset is to try to learn a set of computationally cheap models, and then see which one yields the best predictive accuracy. The cheaper the models, the more tries that can be attempted for a fixed computational budget.

In this new context, the key questions are:

1. *How accurate can we get within our current computational limits?* Prediction is seen as a process of progressive refinement, which advances at the pace of the improvements in the computational resources. The discovery of the true model is not a concern because predictive accuracy is all that matters for most application fields of machine learning.
2. *Is there a limit to the accuracy increase of these simple models as the number of data samples N grows?* As seen above, the true model is not searched for, while it is not clear how close we can get to the truth by approximate models. This question can not be answered by theoretical reasoning. It can only be ascertained by experimentation on ever growing datasets provided by Big Data techniques. This strategy is driven by the observation that larger datasets lead to better results, although it is also observed that the marginal performance decreases as the size of the dataset increases (Sun et al, 2017, p. 850). There is a subjective perception that for a given model, there is a maximum possible accuracy which can not be surpassed no matter how big the training set is (Fernández-Delgado et al, 2014, pp. 3134-3135; Hestness et al, 2017, p. 11). For the e-commerce site example, this means that customer behavior is not fully predictable by any particular model, even if an unlimited amount of data is available. This leaves the question of whether radically new models could outperform the current best models to generate accurate predictions for a particular problem (Huang et al, 2017, p. 7315).

We may call this the *computational turn*. The state of affairs in machine learning is that researchers no longer aim to obtain the true model, but an approximate one which can be learned with a small computational load, while providing high accuracy.

The main reason behind this computational turn is the recent increase by several orders of magnitude of the number of available data samples N, which has dramatically emphasized the relevance of computational simplicity with respect to classic measures of model simplicity. Often machine learning algorithms whose

computational complexity is higher than linear cannot be applied to very large datasets (Witten et al, 2017, p. 507), which did not happen before the Big Data deluge.

Let us consider the example of comparing a machine learning model *Quad* that is associated to a quadratic complexity learning algorithm[3] with a model *Lin* that is associated to a linear complexity learning algorithm[4]. The average execution times of both algorithms can be written as follows:

$$T_{Quad} = N^2 K_{Quad} F_{Quad} \tag{1}$$

$$T_{Lin} = N K_{Lin} F_{Lin} \tag{2}$$

where N is the number of training samples; F_{Quad} and F_{Lin} are the numbers of free parameters of *Quad* and *Lin*, respectively; and K_{Quad} and K_{Lin} are constants that depend on the models, the software implementations of the training algorithms, and the hardware where the algorithms are executed on. It must be pointed out that for parametric models, F_{Quad} and F_{Lin} must not depend on N, i.e. the size of the model must not depend on the number of available training samples (Russell and Norvig, 2016, p. 737). Moreover, K_{Quad} and K_{Lin} must not depend on N either since we focus on a particular setup of software and hardware.

Now let us define the execution time ratio between *Quad* and *Lin*:

$$R = \frac{T_{Quad}}{T_{Lin}} = \frac{N^2 K_{Quad} F_{Quad}}{N K_{Lin} F_{Lin}} = \frac{N K_{Quad} F_{Quad}}{K_{Lin} F_{Lin}} \tag{3}$$

which is the number of times that *Lin* is faster than *Quad*.

If the dataset has a moderate size N, then it makes sense to compare *Quad* and *Lin* by means of classic model simplicity measures, which focus on the numbers of free parameters F_{Quad} and F_{Lin}. For example, it could be the case that $F_{Quad} < F_{Lin}$ so that *Quad* is judged to be simpler than *Lin*, provided that the execution time ratio R is not too large. In other words, the classic simplicity measures and the execution time are complementary simplicity measures which can be collectively assessed by the machine learning practitioner. However, the advent of Big Data means that N grows by several orders of magnitude. Before Big Data, N was in the hundreds or thousands of samples, while nowadays N can be in the millions or billions. This implies that the execution time ratio R also grows by several orders of magnitude, as seen in equation (3). As N grows to infinity, the execution time ratio R also tends to infinity. Here is where the Big Data razor shaves off the *Quad* model since it does not matter how simple *Quad* might be in terms of the classic simplicity measures, because the difference in the execution times as measured by R is just too large. The situation becomes even more dramatic for learning algorithms whose execution time is proportional to N^3, i.e. cubic

[3] This means that the execution time of the learning algorithm is proportional to N^2. For example, kernelized Support Vector Machines (Bishop, 2006, p. 349), and decision tree induction by the C4.5 algorithm with numeric attributes (Witten et al, 2017, pp. 219-220, 508).

[4] This means that the execution time of the learning algorithm is proportional to N. For example, Naive Bayes classifiers (Bishop, 2006, p. 380), and logistic regression (Hastie et al, 2009, p. 120).

complexity[5]. That is, the relevance of model simplicity criteria that do not consider the execution time decays as N grows in the Big Data era.

5 The interplay among the classic and computational notions of simplicity

Here the relations among the classic and computational accounts of simplicity are explored. The classic ones have not been abandoned in current machine learning practices, although they play a secondary role:

Claim In the Big Data era, the classic notions of simplicity are surrogate goals of computational simplicity.

That is, the classic notions which deal with the structural simplicity of the models are considered by machine learning practitioners because they are indirect indicators of the computational load required to train the model, and not by their intrinsic value.

Next, the above claim is justified. The classic notions outlined in Section 2 are mainly devoted to assessing the simplicity of the machine learning models. As seen in Section 1, a machine learning model is a mathematical structure with one or more learnable parameters. The computational complexity of a learning algorithm is related to the structural complexity of the model that the algorithm is applied to. For example, we may take equations (1) and (2), and put the dependence with respect to the number of training samples N into a function $\mathcal{F}(N)$, so that the average execution time of a generic learning algorithm reads as follows:

$$T_{Generic} = \mathcal{F}(N) K_{Generic} F_{Generic} \tag{4}$$

where $F_{Generic}$ is the number of free parameters of the model, and $K_{Generic}$ is a constant that depends on the model, the software implementation of the training algorithm, and the hardware where the algorithm is executed on. Equation (4) means that the average execution time is directly proportional to the number of free parameters, which is associated with the structural complexity of the model. Now, depending on the number of free parameters $F_{Generic}$ of the chosen model, the average execution time $T_{Generic}$ varies. Consequently, structural complexity is often correlated with computational complexity, so the former is an indirect indicator of the latter.

The preference for simple models of classic notions can be interpreted as a search strategy in the space of the possible models (de Rooij and Grünwald, 2011, p. 893). However, a model must be instantiated in order to yield predictions, which are the ultimate goal of machine learning activity. Therefore, it is mandatory to learn the parameters of the model in order to instantiate it, prior to the extraction of predictions and their evaluation to measure predictive performance. This means that the relevant search space for machine learning is the space of possible model instantiations, which comprises all instantiations of all the considered models. In order to carry out a search in such space, classic notions of model complexity can play an auxiliary role. Classic notions can direct the search to models whose

[5] This means that the execution time of the learning algorithm is proportional to N^3. For example, kernel ridge regression (Witten et al, 2017, p. 508).

structure is smaller. But the key factor is the computational load of instantiating (training) the models, as seen in Section 4. This is managed by the computational notion of complexity that speeds up the process of instantiation of the models by choosing learning algorithms with a small computational complexity. In other words, computational simplicity governs the overall search process, while classic notions can help in the prioritization of some search directions over others in the space of all model instantiations.

This framework where classic and computational notions of simplicity work together is associated with scenarios where there are many possible models and many learning algorithms to choose from. The same algorithm can be employed to train several different models. This is the case of the Expectation Maximization (EM), which can be employed to train probabilistic mixtures of Gaussian (Bishop, 2006, p. 435) or Bernoulli distributions (Bishop, 2006, p. 444), or Bayesian linear regression models (Bishop, 2006, p. 448). Conversely, a model can be trained by several different algorithms. For example, Bayesian networks can be learned by a wide range of algorithms (Acid et al, 2004, p. 219).

In order to avoid overfitting in current applications of machine learning to Big Data, computational techniques such as cross validation, bootstrap, regularization and early stopping are commonly employed (Hastie et al, 2009, p. 241, 253, 398; Russell and Norvig, 2016, p. 708, 713; Witten et al, 2017, p. 162, 169, 393, 419, 431). This tendency is due to their lack of assumptions about the datasets, which widens their applicability.

6 Non epistemic justifications of computational simplicity

In this section, we investigate how the previously explained computational simplicity concept (Section 4) relates to the energy consumption and economic cost of the application of machine learning to real problems.

Some operations within the model learning stage of a learning system (Section 3) spend a disproportionately high amount of energy. This is the case of the optimization of the neural architecture for deep learning artificial neural networks since each step in this optimization often requires training of a full neural network. Sometimes the performance increment is small, so it must be evaluated whether it is worth the huge amounts of extra computation and the associated energy cost. The manager of our example e-commerce site may discover that it is not profitable to employ the most accurate customer behavior prediction model if the monetary value of the additional customer purchases is smaller than the extra cost of running the prediction software. Energy efficiency is acknowledged as a design criterion for the proposal of new deep neural network architectures, where a balance must be attained between energy consumption and predictive accuracy (Yang et al, 2017, pp. 1916, 1920; Li et al, 2016, p. 477), since the energy cost of deep learning is notoriously high (Ganguly et al, 2019, p. 335). This is directly connected to the computational simplicity criterion since energy consumption is to be measured in terms of the number of computational operations required to complete a learning task (Strubell et al, 2019). An optimization with at least two independent goals is established, where one of the goals is the minimization of the economic cost and the other goal is the maximization of the predictive accuracy.

In the particular case of deep learning neural networks, substantial energy savings can be attained by using the fine tuning technique. First, a neural network is trained with a standard set of training samples. This is a computationally heavy task since the entire network must be trained from scratch. Then the neural network is tuned to accomplish a specific task, which involves retraining a small portion of the network. This way, most of the initial training effort is reused, thereby drastically reducing the overall energy expenditure. This technique departs from classic machine learning approaches, which usually involved retraining the models for each new application. In the case of deep learning, the computational requirements are so huge that the classic naive retraining approach is simply not feasible for many organizations because they can not afford the costs. That is, typically our example e-commerce site software is based on some readily available pre-trained neural network, which is then tuned to learn the behavior of the customers of this specific web site so that the computational effort and the associated energy budget devoted to learning the predictive model are kept as low as possible.

There are reasons for the past neglection of computational simplicity and other energy consumption criteria. Machine learning is seen as a rapidly expanding branch of science with great potential to enable the automation of many tasks in the near future. Therefore, advanced machine learning systems are not subject to the criticisms that older technologies may have, because automation is recognized as a possible way to save human time and effort. While it is true that many tasks might be more efficiently executed by learning machines, optimal efficiency can only be attained by constraining the energy consumption of these machines. Otherwise, it would still be more energy efficient to employ humans for the task than replacing them with machines. In other words, machine learning systems should not be granted an unlimited amount of computational resources. In order to make informed economic decisions about this matter, the benefits and costs of different strategies to implement machine learning methods must be elicited.

7 Conclusion

Classic concepts of simplicity aim to capture the parsimony of the machine learning model. The number of adjustable parameters is the most straightforward way to measure model complexity, and it is considered both by the Bayesian and Frequentist approaches. These two approaches differ mainly in their interpretation of the concept of probability. Bayesian model selection intends to choose the model that is estimated to have the highest probability of having generated the data, with the help of suitable choices for the prior probabilities, while the Frequentist field does not assign probabilities to models and focuses on estimating the predictive accuracy of the models. Minimum Description Length measures model complexity as the number of bits that are required to encode the model, under a suitable coding system. On their part, expressive power approaches measure the expressive power of a model, i.e. its ability to fit the intrinsic structure of the problem at hand. As seen, the four classic approaches to simplicity considered here only analyze the structure of the model, so that they ignore other aspects that are essential to employ machine learning models in practice. These additional aspects are covered by the learning system concept, which integrates all the relevant conceptual and physical structures required to successfully apply machine learning methods.

The computational load necessary to train and test a model is a reliable, end to end measure of the effort which is devoted to accomplishing a machine learning task. Moreover, it is more encompassing than any of the classic concepts, since computational simplicity can directly be employed to compare parametric, non parametric, non probabilistic, unsupervised and reinforcement machine learning models.

Current practices in machine learning suggest that computational simplicity is what practitioners really seek to minimize since the simplicity of the structure of the model or its expressivity are not by themselves direct indicators of the real effort required to learn and apply the model. Therefore, structural simplicity can be regarded as a surrogate goal of computational simplicity. Computational resources are always limited, and this implies that a tradeoff must be found between computational effort and predictive accuracy. This tradeoff can be seen as an economic decision since computational load translates directly into energy and hardware costs. Choices about computational effort are associated with various energy consumption patterns. In most cases, there is a technological limit for the predictive accuracy that machine learning can attain at the current state of the art so that the tradeoff between effort and accuracy is bounded within the computational resource limits and the predictive accuracy limits. In other words, in many cases, it is not optimal to raise the energy consumption levels without limits until the maximum technologically possible predictive accuracy is achieved. Therefore, purely theoretical criteria about the structure of the models play a less important role for model selection in the Big Data era, since the size of the datasets has grown by several orders of magnitude. These aspects of machine learning activity are no longer overlooked by scientists and practitioners.

References

Acid S, de Campos LM, Fernández-Luna JM, Rodríguez S, Rodríguez JM, Salcedo JL (2004) A comparison of learning algorithms for Bayesian networks: a case study based on data from an emergency medical service. Artificial Intelligence in Medicine 30(3):215 – 232

Agrawal A, Gans J, Goldfarb A (2018) Prediction Machines: The Simple Economics of Artificial Intelligence. Harvard Business Review Press

Amodei D, Hernandez D, Sastry G, Clark J, Brockman G, Sutskever I (2019) AI and compute. https://openai.com/blog/ai-and-compute/

Bandyopadhyay PS, Forster MR (2011) Philosophy of Statistics: An Introduction, North Holland, pp 1–52

Bishop CM (2006) Pattern Recognition and Machine Learning. Springer

Burkov A (2019) The Hundred-Page Machine Learning Book. Andriy Burkov

Cabrera F (2017) Can there be a Bayesian explanationism? On the prospects of a productive partnership. Synthese 194(4):1245–1272

Cahsai A, Ntarmos N, Anagnostopoulos C, Triantafillou P (2017) Scaling k-nearest neighbours queries (the right way). In: 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), pp 1419–1430

Canziani A, Paszke A, Culurciello E (2016) An analysis of deep neural network models for practical applications. CoRR abs/1605.07678, URL `http://arxiv.org/abs/1605.07678`

Carman A (2018) Amazon shipped over 5 billion items worldwide through Prime in 2017. https://www.theverge.com/2018/1/2/16841786/amazon-prime-2017-users-ship-five-billion

Claeskens G, Hjort NL (2008) Model Selection and Model Averaging. Cambridge University Press, Cambridge, UK

Dawid R (2017) Bayesian perspectives on the discovery of the Higgs particle. Synthese 194(2):377–394

Dhiraj, Jain DK (2019) An evaluation of deep learning based object detection strategies for threat object detection in baggage security imagery. Pattern Recognition Letters 120:112 – 119

Domingos P (1999) The role of Occam's razor in knowledge discovery. Data Mining and Knowledge Discovery 3(4):409–425

Fernández-Delgado M, Cernadas E, Barro S, Amorim D (2014) Do we need hundreds of classifiers to solve real world classification problems? Journal of Machine Learning Research 15:3133–3181

Forster MR (2001) The new science of simplicity, Cambridge University Press, pp 83–119

Ganguly A, Muralidhar R, Singh V (2019) Towards energy efficient non-von Neumann architectures for deep learning. In: 20th International Symposium on Quality Electronic Design (ISQED), pp 335–342

Grünwald P, Langford J (2007) Suboptimal behavior of Bayes and MDL in classification under misspecification. Machine Learning 66(2):119–149

Grünwald PD (2007) The Minimum Description Length Principle. The MIT Press

Hastie T, Tibshirani R, Friedman J (2009) The Elements of Statistical Learning, 2nd edn. Springer

Henderson L, Goodman N, Tenenbaum J, Woodward J (2010) The structure and dynamics of scientific theories: A hierarchical Bayesian perspective. Philosophy of Science 77(2):172–200

Hestness J, Narang S, Ardalani N, Diamos GF, et al (2017) Deep learning scaling is predictable, empirically. CoRR abs/1712.00409, URL `http://arxiv.org/abs/1712.00409`, `1712.00409`

Hong J, Wang Z, Niu W (2019) A simple approximation algorithm for the diameter of a set of points in an Euclidean plane. PLOS ONE 14(2):1–13

Huang J, Rathod V, Sun C, Zhu M, Korattikara A, Fathi A, Fischer I, Wojna Z, Song Y, Guadarrama S, Murphy K (2017) Speed/accuracy trade-offs for modern convolutional object detectors. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 3296–3297

Huang Y, Beck JL (2018) Full Gibbs sampling procedure for Bayesian system identification incorporating sparse Bayesian learning with automatic relevance determination. Computer-Aided Civil and Infrastructure Engineering 33(9):712–730

James G, Witten D, Hastie T, Tibshirani R (2014) An Introduction to Statistical Learning with Applications in R. Springer

Jiang L, Zhang L, Li C, Wu J (2019) A correlation-based feature weighting filter for Naive Bayes. IEEE Transactions on Knowledge and Data Engineering 31(2):201–213

Jose C, Goyal P, Aggrwal P, Varma M (2013) Local deep kernel learning for efficient non-linear SVM prediction. In: Dasgupta S, McAllester D (eds) Proceedings of the 30th International Conference on Machine Learning, PMLR, Atlanta,

Georgia, USA, Proceedings of Machine Learning Research, vol 28, pp 486–494
Kang D, Kang D, Kang J, Yoo S, Ha S (2018) Joint optimization of speed, accuracy, and energy for embedded image recognition systems. In: Proceedings of the 2018 Design, Automation and Test in Europe Conference and Exhibition, DATE 2018, vol 2018-January, pp 715–720
Kelly KT (2007) Ockham's razor, empirical complexity, and truth-finding efficiency. Theoretical Computer Science 383(2):270 – 289
Kelly KT (2011) Simplicity, truth and probability, North Holland, pp 983–1026
Korb KB (2004) Introduction: Machine learning as philosophy of science. Minds and Machines 14(4):433–440
Kpotufe S, Verma N (2017) Time-accuracy tradeoffs in kernel prediction: Controlling prediction quality. Journal of Machine Learning Research 18(44):1–29
Levitin A (2011) The Design and Analysis of Algorithms, 3rd edn. Pearson Education
Li D, Chen X, Becchi M, Zong Z (2016) Evaluating the energy efficiency of deep convolutional neural networks on CPUs and GPUs. In: 2016 IEEE International Conferences on Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom) (BDCloud-SocialCom-SustainCom), pp 477–484
Lin HW, Tegmark M (2016) Why does deep and cheap learning work so well? https://arxiv.org/abs/1608.08225
Ma J, Chen L, Gao Z (2018) Hardware implementation and optimization of tiny-YOLO network. Communications in Computer and Information Science 815:224–234
Mohri M, Rostamizadeh A, Talwalkar A (2014) Foundations of Machine Learning. The MIT Press, Cambridge, Massachusetts
Montañez GD (2017) Why machine learning works. URL `https://www.cs.cmu.edu/~gmontane/montanez_dissertation.pdf`
Moore C, Mertens S (2011) The Nature of Computation. Oxford University Press
Murphy KP (2012) Machine Learning: A Probabilistic Perspective. The MIT Press
Oneto L, Navarin N, Donini M, Ridella S, Sperduti A, Aiolli F, Anguita D (2018) Learning with kernels: A local Rademacher complexity-based analysis with application to graph kernels. IEEE Transactions on Neural Networks and Learning Systems 29(10):4660–4671
Parashar A, Raina P, Shao YS, Chen Y, Ying VA, Mukkara A, Venkatesan R, Khailany B, Keckler SW, Emer J (2019) Timeloop: A systematic approach to DNN accelerator evaluation. In: 2019 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), pp 304–315
Pereira F, Norvig P, Halevy A (2009) The unreasonable effectiveness of data. IEEE Intelligent Systems 24(2):8–12
Pothos EM, Wolff JG (2006) The simplicity and power model for inductive inference. Artificial Intelligence Review 26(3):211–225
Raschka S (2018) Model evaluation, model selection, and algorithm selection in machine learning. CoRR abs/1811.12808, URL `http://arxiv.org/abs/1811.12808`
Redmon J, Farhadi A (2018) YOLOv3: An incremental improvement. CoRR abs/1804.02767, URL `http://arxiv.org/abs/1804.02767`
Ren S, He K, Girshick R, Sun J (2017) Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis

and Machine Intelligence 39(6):1137–1149
Rochefort-Maranda G (2016) Simplicity and model selection. European Journal for Philosophy of Science (6):261–279
de Rooij S, Grünwald PD (2011) Luckiness and regret in Minimum Description Length inference, North Holland, pp 865–900
Russell SJ, Norvig P (2016) Artificial Intelligence: A Modern Approach. Pearson Education Limited, Harlow, Essex, England
Shalev-Shwartz S, Ben-David S (2014) Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press, Cambridge, UK
Sober E (2015) Ockham's razor: a user manual. Cambridge University Press
Strubell E, Ganesh A, McCallum A (2019) Energy and policy considerations for deep learning in NLP. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics
Sun C, Shrivastava A, Singh S, Gupta A (2017) Revisiting unreasonable effectiveness of data in deep learning era. In: The IEEE International Conference on Computer Vision (ICCV), pp 843–852
Vapnik VN (2000) The Nature of Statistical Learning Theory. Springer, New York
Wasserman L (2000) Bayesian model selection and model averaging. Journal of Mathematical Psychology 44(1):92 – 107
Wirth N (1985) Algorithms + Data Structures = Programs, 2nd edn. Prentice-Hall
Witten IH, Frank E, Hall MA, Pal CJ (2017) Data Mining: Practical Machine Learning Tools and Techniques, 4th edn. Morgan Kaufmann, Cambridge, MA
Yang T, Chen Y, Emer J, Sze V (2017) A method to estimate the energy consumption of deep neural networks. In: 2017 51st Asilomar Conference on Signals, Systems, and Computers, pp 1916–1920