Varieties of Justification in Machine Learning

David Corfield

September 4, 2008

Abstract

Forms of justification for inductive machine learning techniques are discussed and classified into four types. This is done with a view to introduce some of these techniques and their justificatory guarantees to the attention of philosophers, and to initiate a discussion as to whether they must be treated separately or rather can be viewed consistently from within a single framework.

1 Introduction

It is common for philosophers once they have adopted a favourite mode of justifying inductive practices to hold it as being uniquely correct. Practitioners of machine learning, on the other hand, tend to be open to a range of such justifications, considering the breadth of this range to be a sign of the robustness of their inductive method. Taken together they present philosophers with an excellent opportunity to explore whether one position can make best sense of them all, or whether a pragmatic pluralism is called for.

In this paper I would like to present a way to classify such justifications into four varieties. Two of these four, ESTIMATE and BOUND, may be described as dealing with 'absolute' performance, in the sense that they indicate how accurately a learning device will perform in the future. Since we know that without making presuppositions about the world we can say nothing about the absolute future performance of an inductive device, what we must do then is embody some presuppositions in the inductive bias of the device, which we then employ to justify its output. By contrast, the other two forms of justification, RELATIVE and CONSISTENT, deal with the relative performance of a device compared, in the former case, to a given set of other devices, and in the latter, to other ways the same device might function.

Here then are the four forms I shall consider:

(ESTIMATE) Under the assumption that the world provides you with an independent and identically distributed sample from an unknown distribution, you can give an unbiased estimate of the generalisation error of the trained algorithm.

(BOUND) Under the assumption that the world provides you with an independent and identically distributed sample from an unknown distribution, you can give probabilistic bounds for the generalisation error of the trained algorithm in terms of its performance on a training set.

(RELATIVE) With no assumptions about the data generating process, you can bound online inaccuracy in terms of the error rate of the best of a specified set of competitors chosen with hindsight.

(CONSISTENT) The method is the only consistent scheme of inference in relation to your state of knowledge. If you behave otherwise, and you are acting consistently, it must be that your beliefs have been changed by extraneous factors.

I shall discuss the use of these criteria largely in relation to one of the simplest inductive tasks, that of binary classification. Assume that we are looking to find a classifier on a space X, a function which when presented with an element of X outputs either $+1$ or -1. For example, X could be the set of all 30×30 pixellated grayscale images. We also assume that there is a fixed unknown distribution over $X \times \{+1, -1\}$. This might represent, for example, a population of images of handwritten 3s and 5s. Then an algorithm is given a sample from this population, called the training data. By means of this, it will yield a function to classify unseen data, perhaps through the intermediary of a probability function, so $f(x) = +1$ if $p(x) \geq 0.5, f(x) = -1$ otherwise. We can then test this function by seeing how accurately it classifies data from another sample, the test data.

In this paper I shall use the opportunity of discussing justication criteria to highlight some notable discoveries made in machine learning, in particular I shall pay attention to so-called 'kernel' methods, which have come to dominate a large part of the field over the past 15 years. These methods involve the selection of a classifier from a very rich collection of functions, requiring that a means to *regularise* the choice be found to prevent overfitting.

2 ESTIMATE

ESTIMATE and BOUND are most often appealed to, especially by non-Bayesians, although it is possible to give them a Bayesian gloss. We can quickly deal with this first type of justification, the main example of which is the widely used cross-validation (CV). For example, in LOO (leave one out)-CV, we take each data point of the training sample in turn, train our algorithm on the remainder of the points, and then compute the prediction of our trained algorithm on the missing point. We can then compare this prediction with the true label. The error rate for these predictions is an almost unbiased estimator of the generalisation error, that is, the misclassification rate of the algorithm trained on all the data relative to the unknown distribution generating the data, a result due to Luntz and Brailovsky (See Vapnik 1998).

Cross-validation can be, and commonly is, used to select algorithm parameters. It appears to bear characteristics of frequentist ways of operating, but it is also used by Bayesians as a short-cut when the calculations involved in marginalising over parameter values is intractable or too costly. I am unaware of any analysis having been done, but it it is not hard to imagine that certain conditions of ignorance would imply that parameter values for which the cross validation score is highest correspond to the maximum of the posterior distribution on these parameters. A Bayesian might also think to use this technique to check on the plausibility of her model. Often, for reasons of tractability, she must use a model which does not fully capture her prior beliefs. Cross-validation

can then provide assurance that she has not strayed too far from these beliefs.

3 BOUND

A key notion in the field of machine learning is that if the function space from which the classifier is selected is very rich, then we risk overfitting. Imagine a collection of points in a plane, some coloured red, some blue. If I am allowed total freedom, of course I can draw a curve, however jagged, which separates the data perfectly, so that all red points fall on one side of the line and all blue points fall on the other. But I shall not expect this curve to perform well at separating the unseen test data. In other words, I cannot expect the true error rate to be close to the training error rate, zero in this case. What the BOUND form of justification seeks to achieve is a probablistic guarantee of the difference between the training error and the true error rates.

We shall need then to say something about the 'capacity' of the set of functions to make discriminations. One way goes via the Vapnik-Chervonenkis dimension, or VC-dimension. The VC-dimension of a set of classifying hypotheses is the largest natural number, n, such that there is a set of n points, for which however they are labelled $+$ or $-$, there is a hypothesis which agrees with this labelling. The set of points is said to be 'shattered' by the hypothesis class. For example, the set of half-planes has VC-dimension three because any set of three non-collinear points is shattered by these classifiers, while any set of four points is not shatterable by the same class. Arranged in a square, labelling diagonally opposite points the same prevents separation. In general, the set of hyperplanes in a d dimensional space has VC-dimension $d+1$. On the other hand, the class of convex polygons has infinite VC-dimension, since for any number of points on the circumference of a circle, however they are labelled, it is possible to draw a polygon to separate them.

With this dimension in hand we can now derive what are called PAC results, where PAC stands for probably approximately correct. For a class of classifiers or hypotheses H, then for any h in H, with probability at least $(1-\delta)$,

$$error_{true}(h) < error_{train}(h) + \frac{VC(H)(ln\frac{2m}{VC(H)}+1) + ln\frac{4}{\delta}}{m}.$$

This result rests on the assumption that the training data is an independent and identically distributed sample from the target distribution. Of course, we may wonder how reasonable this assumption is. If I have trained a digit classifier on digits written by people in one time and place, I should expect it to do less well elsewhere and at other times. An obvious reason might be that 7s are written with a bar across them in France but not in the UK.

We should also note that this result has the flavour of a confidence interval, with its less than straightforward interpretation. Just because we have a training error of, say, 2%, and the second and third terms of the right hand side amount to, say, 3%, for $\delta = 1\%$, does not mean we should say we're 99% certain that the true error is less than 5%. It is quite possible that the gap between training and true errors might be correlated with the training error, so that only when the training error is away from zero is the error gap small. On the other hand, one might argue that without evidence to the contrary it is reasonable to assume no correlation.

So-called 'empirical risk minimization' has us merely choose h from a given H with the smallest training error. A Bayesian might again point to some ignorance conditions (see Minka 2001) which make this a reasonable thing to do. But we can also use the PAC theorem in a more interesting way to select the hypothesis class H itself, by employing 'structural risk minimization'. Imagine we have a chain of hypotheses classes, H_i, of increasing VC-dimension. Then we can find the class and the classifier within that class which minimises the right hand side of the bound.

$$Min_{\{h,H_i:h\in H_i\}}(error_{train}(h) + \frac{VC(H)(ln\frac{2m}{VC(H)}+1) + ln\frac{4}{\delta}}{m}).$$

In practice, the PAC bound is usually rather loose, and we may have reason to doubt that a minimum will correspond to the best classifier.

Structural Risk Minimization is an instance of a general scheme where an algorithm minimises an *objective function* made up of an inaccuracy term and a regularization term to penalise complicated hypotheses:

$$\text{Loss + Penalty = Badness of fit + Complexity of classifier class}$$

4 Kernels

Now, one of the biggest events to happen in machine learning over the past decade and a half has been the introduction of 'kernel methods'. Such are the advantages of working with linear algebra that we don't mind mapping our data into a potentially very high, or even infinite, dimensional space, so long as we can cast our algorithm there in linear algebra terms. We may choose to do this even if our data domain is a vector space already.

We do this by mapping our space X into what is called a reproducing kernel Hilbert space, where the size of the inner product in the latter will reflect how 'similar' are the points in X. This is done by means of a kernel. A kernel on a set X is a function $K : X \times X \to \mathbb{R}$, which is symmetric and positive definite, in the sense that for any $N \geq 1$ and any $x_1, ..., x_N \in X$, the matrix $K_{ij} = K(x_i, x_j)$ is positive definite, i.e., $\sum_{i,j} c_i c_j K_{ij} \geq 0$ for all $c_1, ..., c_N \in \mathbb{R}$. (Complex-valued kernels are possible.)

Another way of looking at this situation is to reformulate it as a mapping $\phi : X \to H$, where H is a reproducing kernel Hilbert space, a function space in which pointwise evaluation is a continuous linear functional. The 'kernel' and 'feature map to Hilbert space' stories relate to each other as follows: $K(x, .) = \phi(x)$. The reproducing aspect of H means that $\langle f(.), K(x, .)\rangle = f(x)$, and this is continuous. So we have $K(x, y) = \langle \phi(x), \phi(y)\rangle$.

H is the span of the set of functions $\{K(x, .)|x \in X\}$, and, under certain conditions, when we find the $f \in H$ for which a functional is minimised, it takes the form

$$f(x) = \sum_{i=1}^{m} c_i K(x_i, x).$$

The x_i are the *support vectors*. Ideally the number of these would be low, so that the majority of the information in the training set labels is encoded efficiently.

An unlabelled data point is now classified by a kind of weighted sum of votes from the support vectors, those very close to the point, according to the kernel, having the largest say. The classification boundary in X is curved, while its image in H is a hyperplane. Many linear algebra techniques in H just involve the inner product, allowing the performance of a form of nonlinear algebra back in X.

Consider binary classification again. The support vector machine approach to this task looks to find the hyperplane in H which best separates the images of data points $\phi(x_i)$, so that those with the same label (y_i) fall in the same half space. In view of the PAC result mentioned in the previous section, it may appear strange to be mapping to a higher-dimensional space, where linear classifiers will have high or infinite VC-dimension. One solution is to control for overfitting by finding such a hyperplane that classifies the training data accurately with the largest perpendicular distance, or margin, to the nearest of them (the support vectors). The support vector machine (SVM) does precisely this, and as such it is known as a maximum-margin classifier. Now, the VC-dimension of classifiers whose margin is greater than a given value is finite, although data dependent. However, a PAC result is still possible.

Kernels, which characterise the similarity of the input set, are key to the SVM approach. But how do they get chosen? A common choice is the Gaussian radial-basis function (RBF) kernel:

$$K(x,y) = (4\pi t)^{-n/2} exp\left(-\frac{\|x-y\|^2}{4t}\right),$$

even though this can lead to an odd notion of similarity. Consider again handwritten digits, pixellated 30×30 so that their grayscale values form a vector in $\mathbb{R}^{900}$. The image of a '1' shifted a couple of pixels to the right will appear very different in this representation. The classifying algorithm is not fed any information to let it know that it ought to classify the shifted image as the same digit. However, if it is trained on sufficient data this may well not matter since there will be a training example 'close enough' to allow for correct classification.

This does seem wasteful, though, of useful background knowledge. Indeed, you can see that if you operate on the training images with the same permutation of the 900 pixels, it makes no difference to the SVM, even though it makes an easy task much more difficult for a human. One way of exploiting this background knowledge is to train a support vector machine with the given data, find the support vectors, then use translations of them as new 'virtual' data. This does have the effect of producing a more accurate classifier, but is seen by some as *ad hoc*. Ideally, the expected invariances would be encapsulated in the kernel.

A second very import motivation for the use of kernels is that they allow us to work with non-vectorial data, for instance strings, trees or graphs. Much information gathered today is encoded in these forms. For example, biochemists have produced vast amounts of data concerning protein sequences and the functions they perform. In the case of protein sequences of a given length (AGTTGACAG, GCTTGACTC,...) we might consider a kernel which counts as similar strings containing long identical substrings. The use of kernel methods in bioinformatics is a burgeoning field of application (see Schölkopf *et al.*, 2004).

5 Bayesian Machine Learning

Now where are the Bayesians in all this? Rather than merely classifying an object, surely it would be better if we were to give an assessment of the probability of each label. Well, attempts have been made to give SVMs a probabilistic interpretation, but the consensus is that these are contrived. More evidence of this incompatibility comes from the fact that we can also view trained SVMs as successfully compressing the training data, since the support vectors largely carry the information of the training sample, and the number of these vectors may be considerably lower than the sample size. But then theoretical work shows that "sparseness and the ability to estimate conditional probabilities seem to be incompatible." (Bartlett, Jordan, McAuliffe 2006). If you don't wish to bear the cost of computing the full probability distribution, you must find a way to ignore some of the training data. However, there are ways of treading this balance which are more acceptable to the Bayesian.

Some have questioned the need to compute the full probability distribution. One argument relies on a translation between probabilities and energy levels. In statistical mechanics the Gibbs distribution is used to assign a probability distribution to a collection of states and their energy levels. $P(x) = \frac{1}{Z}e^{-\beta E(x)}$, where $Z = \frac{\int d^d x \mu(x) e^{-\beta E(x)}}{\int d^d x \mu(x)}$, and $\beta = 1/T$, the inverse of the temperature. So we can pass in both directions between relative probabilities and normalisable energies of states. In the machine learning setting, the states correspond to the parameter values of the algorithm, for example, the weights of a neural network. Several ideas from statistical mechanics have found uses in machine learning. However, Yann LeCun et al. (2006) argue that in light of this translation we can see that thinking in terms of energies is more flexible than confining ourselves to probabilities. For one thing, it is possible to use unnormalised energies, which may make calculations tractable. Moreover, if you are not going to treat the whole space of states, but rather just concentrate on the region of states of least energy, that is, those closely competing with the optimal setting at a given stage, then there is no need to bear the full computational burden of performing calculations over the full space of states. Furthermore, some unnormalisable energy functions are very useful.

What of the Bayesian use of kernels? Introduced a little after SVMs hit the scene, Bayesians found a similar resource in Gaussian processes, inviting the charge that they had copied the SVM use of kernels. The Bayesian account of this discovery runs differently, however, as they point to an independent source of inspiration. It seems that people working on neural networks had found that it is very difficult to train a many layered network. Typically one hidden layer was doing most of the work. It was observed that in the limit of making this single layer large (with certain conditions on the connection strengths), you end up with a Gaussian Process (Rasmussen and Williams, 2006).

A Gaussian Process on X is a set of real-valued random variables on X, all finite subsets of which have (consistent) Gaussian distributions:

$$\mathbf{f} = (f_1, ..., f_n)^T \sim N(\mu, \Sigma),$$

where Σ is the covariance matrix. It can also be thought of as a distribution over functions on X, favouring those with certain qualities, e.g., smoothness. $f(x) \sim GP(m(x), k(x, x'))$. The Bayesian may be said to be regularizing by her

choice of prior.

In the case of classification, one is looking to find the posterior distribution of the value of functions on X evaluated at a test point, after conditioning on the training sample.

$$p(f_*|\mathbf{x}, \mathbf{y}, x_*) = \int p(f_*|\mathbf{x}, \mathbf{y}, f)p(f|\mathbf{x}, \mathbf{y})df.$$

This value is then scaled to give a probability. Unfortunately, whereas in the case of regression the relevant integral is Gaussian, this is no longer the case for classification. However, one may find a Gaussian process which best fits the maximum a posteriori solution, by using the Laplace approximation. A more adequate Bayesian approximation, one defined not in terms of the mode of the posterior but rather the mean, is to find a Gaussian process which matches the first two moments of the posterior distribution. This process is known as the Expectation Propagation (EP) algorithm.

Now, we can find a bayesian form of ESTIMATE which can be used for Gaussian process classification, known as the PAC-Bayesian theorem. We begin with a prior distribution over parameter values, $P(w)$. After training on a sample of size m, we arrive at a posterior distribution $Q(w)$. Now for any choice of P and Q, and for any δ, then with probability at least $(1-\delta)$ over the choice of $S \sim D^m$,

$$KL_B(l(Q,S)||l(Q,D)) \leq \frac{KL(Q||P)+log\frac{m+1}{\delta}}{m},$$

where $KL(.||.)$ is the Kullback-Leibler divergence and $l(Q,S)$ is the expected misclassification rate on the sample relative to distribution Q, while $l(Q,D)$ is the expected misclassification rate on the true distribution.

We can interpret this result by imagining that we are dealing with a collection of experts labelled by the w. We begin with P a probability distribution over the experts. After seeing the training data, we choose Q, a new distribution over experts. We choose our Q so that on average our experts have classified the sample well. Then we have a probabilistic bound for the divergence of their average true error rates from what they achieved on the training sample. The bound includes the term $KL(Q||P)$ which penalizes you for how picky you have been in your choice of experts. This will be large if you opted for a very narrow range of experts. If on the other hand, you could keep reasonably close to the original distribution P while remaining reasonably accurate on the training sample, then the bound will be tighter.

Now, one might object to this result that it is rather odd to have a Bayesian result rely on what looks like a frequentist interpretation of probability theory. Here, we imagine a long sequence of attempts to calculate the difference between test and training error, we would find that in $x\%$ of samples the gap between expected sample error and expected true error is less than $y\%$. But again, it seems likely that some conditions of ignorance would allow us to say that in any single instance of this training one's degree of belief in a error gap less than $y\%$, should be at least $x\%$. This ignorance would relate to the possible correlation between errors terms.

Most applications of ESTIMATE bounds to practical situations are very loose, indeed they are often vacuous, only telling us with high certainty that the generalisation error will be less than some figure over, say, 30%, sometimes even

100%. However, Seeger using Gaussian Process Classification on the MNIST handwritten digit set, and training on a sample of 5000 digits, arrived at the following result:

empirical error 1.87%
generalization error 1.95%
upper bound 7.6%
$\delta = 1\%$.

This is a startling result in view of the much larger amount of data that would have been necessary to achieve a similar result using the PAC-theorem, more data than it is possible to handle with present day machines.

6 RELATIVE

Here the idea is that we guarantee the performance of our inductive procedure by bounding the gap between its performance and that of the best of a set of competitors chosen with hindsight. Imagine we are fed data one point at a time. On the basis of information we have seen, we must estimate a certain quantity relating to the next unlabelled data point, such as the probability of its label. We are given a penalty according to our inaccuracy. If we adopt certain strategies, we can show that, whatever the data, our average penalty per data point will only exceed that of the best of a set of competitors, as selected after seeing all the data at once, by at most a specified amount. This amount tends to zero as the sample size increases. In many cases, Bayesian strategies fall into this category. (See Grunwald 2007.)

An example using Gaussian processes, which holds for regression as well as classification, is as follows. The penalty for a probabilistic estimate is the negative log likelihood of the correct value. So, for example, if we think a certain label is unlikely and we are wrong, we receive a large penalty. Now, we can compare the penalty for Bayesian online decision making with the penalty for a classifier chosen from a competitor set, after the input sequence has been given:

$$-logP_{bs}(y_{\leq n}|x_{\leq n}) \leq -logP(y_{\leq n}|f, x_{\leq n}) + \frac{1}{2}||f||_K^2 + \frac{1}{2}log|I + cK|$$

where P_{bs} is the Bayesian Gaussian Process prediction method, and f is any function from the Hilbert space associated with the kernel K. c is a constant which depends on K.

In other words, for any function, f, from the Gaussian process, our Bayesian total penalty is bounded by the sum of the penalty earned by that function, a term relating to the prior of the Gaussian process at f, and a constant term which depends on the kernel. It can be shown that the radial-basis function kernel has good properties in this last regard. Only the first term on the right hand side depends on n, so for given f, the average penalty per data point of the Bayesian strategy will exceed that of f by at most a quantity which tends to 0 as n grows large.

7 CONSISTENT

While the Bayesian may be pleased to find support for her methods from RELATIVE, their justification type of first resort is CONSISTENT. Here the advocate of a piece of inductive reasoning makes the claim that in view of what they know and learn, it is the only consistent way to reason. Bayesians commonly appeal to this type of justification through a variety of representation theorems which assume that beliefs can be represented as probability distributions. For example, de Finetti's result for exchangeable sequences, ones for which permuted sequences are seen as equiprobable, tells us that in the case of a binary sequence our beliefs can be represented as a distribution over p in $[0, 1]$, which is updated by conditioning on the numbers of 0s and 1s seen. If you do not act in this way, it must be that you did not wholly hold with exchangeability. Perhaps you had reserved some of your belief for the possibility of sequences where the generating mechanism changes the chance of a 0 appearing along the sequence. After a long series of 1s followed by a long series of 0s, this belief may start to dominate.

A more general, but less known, result of this form runs as follows (see Snoussi and Mohammad-Djafari 2002): my task is to find the best estimator taking values in a class of probability distributions over a space Z, based on a sample from Z. Best is meant here in the sense of a least 'distance' from the true distribution to my estimate, the distance often being the Kullback-Leibler divergence. Now, there's no reason I shouldn't be able to describe my estimator before I see any data, in other words my state of belief should tell me how I'll act whatever data I meet with. So let τ be my estimator, taking any finite set of data to a probability distribution. Then define the generalisation error for this estimator as follows:

$$E(\tau) = \int_p P(p) \int_{\mathbf{z}} P(\mathbf{z}|p) D(p, \tau(\mathbf{z})),$$

where $D(p, .)$ is the cost of misspecifying the true distribution p. As you can see, this quantity is what you expect your loss to be, based on your belief about the true distribution p. What we seek is an estimator which minimises this expected error. Snoussi and Mohammead-Djafari explain (p. 312) how, in the case where D is the Kullback-Leibler divergence, the best estimator is the function which sends $\mathbf{z}$ to $\int pP(p|\mathbf{z})$, the mean of the posterior distribution. This shows the coherence of the Bayesian approach. The idea then is, if you do not behave in a Bayesian way here, either not all relevant beliefs were encoded in your prior, or you received extraneous information, or you are behaving inconsistently with respect to the cost of misspecification.

More broadly yet, very many machine learning algorithms can be conceived as forms of maximum entropy reasoning under constraints. Some so-called 'objective' Bayesians consider this way of construing their reasoning as the best justification of their practice. (See, for example, Williamson 2007.) As mentioned earlier, however, frequently the Bayesian is not able properly to reflect her beliefs, due perhaps to issues of tractability or when no kernel can be devised which respects background knowledge. Techniques associated with other forms of justification, such as cross validation, may then be required for pragmatic reasons.

8 Conclusion

I have given outlines of four kinds of justification offered by machine learning theorists for their inductive algorithms. Many practitioners appear content to adopt a pluralistic attitude. But I believe it would be very worthwhile to probe their mutual coherence. I have given some indications that, if anyone can unify them within a single framework, it is the Bayesian camp. The ESTIMATE and BOUND type results admittedly have a classical flavour to them, but it would appear quite possible to interpret them in a Bayesian light, with a reasonable understanding of an agent's state of ignorance.

Some philosophers may see simple binary classification tasks as lacking the richness necessary to be able illuminate the most important aspects of inductive practice as takes place in a mature natural science. Be that as it may, I believe that there is sufficient complexity to warrant the devoting of philosophical attention to the wide range of types of justification used in machine learning today. Some interest has been shown in the VC-dimension and its use in PAC results, notably by Gilbert Harman, but for the most part the field remains largely unexplored. In particular, there is important work to be done in further understanding the relationship between minimum description length and Bayesian approaches, and in examining the idea that many machine learning techniques are forms of maximum entropy reasoning.

9 Bibliography

P. Bartlett, M. Jordan, J. McAuliffe, Comment on 'Support vector machines with applications', Statistical Science, 21, 341-346, 2006.
P. Grunwald, The Minimum Description Length Principle, MIT Press 2007.
Y. LeCun et al., 'A Tutorial on Energy-Based Learning',
http://yann.lecun.com/exdb/publis/index.html#lecun-06, 2006.
T. Minka, Empirical Risk Minimization is an Incomplete Inductive Principle, http://research.microsoft.com/ minka/papers/minka-erm.pdf, 2001.
Carl Rasmussen and Chris Williams, Gaussian Processes for Machine Learning, MIT Press, 2006.
Bernhard Schölkopf, Koji Tsuda and Jean-Philippe Vert (eds.), Kernel Methods in Computational Biology, MIT Press 2004.
H. Snoussi and A. Mohammad-Djafari 'Information geometry and Prior Selection.', in Bayesian Inference and Maximum Entropy Methods. MaxEnt Workshops, Aug. 2002, pp 307–327.
V. Vapnik, Statistical Learning Theory, Wiley 1998.
J. Williamson, 'Philosophies of probability: objective Bayesianism and its challenges'. In Irvine, A., editor, Handbook of the philosophy of mathematics. Elsevier, Amsterdam. Handbook of the Philosophy of Science volume 4, 2007.