Notes on decision theory for machine-learning algorithms

Luca <pgl@portamana.org>

6 November 2021; updated 6 November 2021

1 Inferential vs decision algorithms

Machine-learning algorithms can be insightfully interpreted as inferential algorithms[1], which in turn can be seen as exchangeable probability models[2]. By 'inference' we mean the assessment of a probability distribution for some quantity or scenario of interest, *without commitment* to any specific value of such quantity.

The output of a trained machine-learning algorithm, however, is often a specific value taken at face value; that is, treated as "the truth"[3]. From this point of view the algorithm is making a *decision*: choosing a specific value to be used in the problem at hand. Moreover, once the algorithm has been trained it is often used for future decisions without any further training; that is, its internal parameters (the weights of a neural net, for example) remain fixed at some specific value. This also represents a choice made on our part.

When we not only assess the probabilities for the values of a quantity, but also choose a specific value or other course of action based on such assessment, we enter the domain of decision theory[4]. We shall consider this theory as normative and use its principles to approach the problem of training and choosing the internal parameters of a machine-learning algorithm that performs regression or classification, that is, that outputs a specific value y (continuous or categorical) given an input x (which can be real, categorical, or belong to some general manifold). A brief summary of decision theory is given in the next section.

[1] Tishby et al. 1989; Levin et al. 1990; MacKay 1992a,b,c,d; 2005 esp. Part V; Neal 1996.
[2] Bernardo & Smith 2000 ch. 4. [3] cf. MacKay 1992b § 3. [4] Savage 1972; Raiffa & Schlaifer 2000; Berger 1985; Bernardo & Smith 2000 ch. 2; Pratt et al. 1996; Jaynes 2003 chs 13–14; for a charming introduction see Raiffa 1970.

2 A simplified overview of decision theory

to be written

3 Training, parameter choice, and algorithm choice as a combined decision problem

From a decision-theoretic perspective our problem is as follows.

3.1 Decisions

We must choose one among several machine-learning algorithms, and for each of them, one among several internal parameter values. Note that this nested choice can actually be combined into a single choice. Label the candidates algorithms with 1, 2, etc., and denote their parameter spaces by Θ_1, Θ_2, etc.. The choice of algorithm and internal parameter can then be seen as the choice of a parameter value θ in the union space $\Theta_1 \cup \Theta_2 \cup \cdots$, denoted Θ. A machine-learning algorithm with internal parameter θ typically outputs a value that is a function $t(x \mid \theta)$ of the input x and of the (fixed) parameter θ. From our general point of view it is implicitly understood that "t" can actually have different functional forms depending on whether $\theta \in \Theta_1$ or $\theta \in \Theta_2$ and so on (for example, t can be a composition of nonlinear functions of x if θ belongs to the space of weights of a neural net, or it can be a linear function of x if θ belongs to the space of coefficients of a linear-regression algorithm).

For the moment we work with this general point of view, and later explore how the separation into two different kinds of choices is made. Our possible choices or decisions therefore consist of the possible values $\theta \in \Theta$.

3.2 Scenarios

Besides the space of choices we must specify the space of possible scenarios, of which only one will turn out to be true. Our scenarios consist of all possible sequences of pairs $\big((x_1, y_1), (x_2, y_2), \dots\big)$ that our algorithm will encounter in its lifetime: the x_n will be the known inputs fed to the algorithm, and the y_n the unknown values that the algorithm will try to predict. Let us assume that this sequence is finite, although very large. Denote $\bar{x} \coloneqq (x_1, x_2, \dots)$, analogously for $\bar{y}$, and $\bar{z} \coloneqq (\bar{x}, \bar{y}) \coloneqq$

$\big((x_1, y_1), (x_2, y_2), \dots\big)$. If the quantity x takes values in the manifold X and y in Y, our possible scenarios live in the space $\prod_n (X \times Y)$.

3.3 Utilities

For each combination of decision (algorithm & internal parameter) θ and scenario (future data) $(\bar{x}, \bar{y})$ we must now specify the utility $U(\theta \mid \bar{z})$.

We make the realistic assumptions that this utility is the sum of utilities $u[\theta \mid (x_n, y_n)]$ from each single application n of the algorithm to the sequence of data, and that such individual utilities have identical functional forms:

$$U(\theta \mid \bar{z}) = \sum_n u(\theta \mid x_n, y_n)\,. \tag{1}$$

This assumption simplifies the calculations to follow. A more general approach, where the utilities change with time, is also possible and may be realistic in particular situation.

The functional form of the single-instance utility $u(\theta \mid x_n, y_n)$ depends on the specific problem – in fact it is no less problem-specific than the choice of machine-learning algorithm – so we do not make any assumptions about it for the moment.

3.4 Probabilities for the scenarios

Lastly we need to assess the distribution of probability $\mathrm{p}(\bar{z})\,\mathrm{d}\bar{z}$ over the possible scenarios $\bar{z}$. This probability distribution is conditional on some hypotheses or background knowledge H, and on a sequence of known inputs (ξ_ν) and corresponding *known* outputs (υ_ν). Analogously to the scenarios we denote $\overline{\xi} \coloneqq (\xi_1, \xi_2, \dots)$, analogously for $\overline{\upsilon}$, and $\overline{\zeta} \coloneqq (\overline{\xi}, \overline{\upsilon}) \coloneqq \big((\xi_1, \upsilon_1), (\xi_2, \upsilon_2), \dots\big)$. So our probability distribution is

$$\mathrm{p}(\bar{z} \mid \overline{\zeta}, H)\,\mathrm{d}\bar{z} \equiv \mathrm{p}(\bar{x}, \bar{y} \mid \overline{\zeta}, H)\,\mathrm{d}\bar{x}\,\mathrm{d}\bar{y}\,. \tag{2}$$

We will shortly further specify its form.

3.5 Expected utilities

According to decision theory every action θ has an associated expected utility

$$\begin{aligned} E(\theta \mid \bar{z}, H) &\coloneqq \iint U(\theta \mid \bar{x}, \bar{y})\, \mathrm{p}(\bar{x}, \bar{y} \mid \bar{\zeta}, H)\, \mathrm{d}\bar{x}\, \mathrm{d}\bar{y} \\ &= \iint \sum_n u(\theta \mid x_n, y_n)\, \mathrm{p}(\bar{x}, \bar{y} \mid \bar{\zeta}, H)\, \mathrm{d}\bar{x}\, \mathrm{d}\bar{y} \end{aligned} \tag{3}$$

and we must choose the algorithm and internal parameters θ^* given by

$$\theta^* \coloneqq \arg\sup_\theta \iint \sum_n u(\theta \mid x_n, y_n)\, \mathrm{p}(\bar{x}, \bar{y} \mid \bar{\zeta}, H)\, \mathrm{d}\bar{x}\, \mathrm{d}\bar{y}\,. \tag{4}$$

Bibliography

("de X" is listed under D, "van X" under V, and so on, regardless of national conventions.)

Berger, J. O. (1985): *Statistical Decision Theory and Bayesian Analysis*, 2nd ed. (Springer, New York). DOI:10.1007/978-1-4757-4286-2. First publ. 1980.

Bernardo, J.-M., Smith, A. F. (2000): *Bayesian Theory*, repr. (Wiley, New York). DOI:10.1002/9780470316870, https://archive.org/details/bayesiantheory0000bern. First publ. 1994.

Jaynes, E. T. (2003): *Probability Theory: The Logic of Science*. (Cambridge University Press, Cambridge). DOI:10.1017/CBO9780511790423. Ed. by G. Larry Bretthorst. First publ. 1994. https://archive.org/details/XQUHIUXHIQUHIQXUIHX2, http://www-biba.inrialpes.fr/Jaynes/prob.html.

Levin, E., Tishby, N., Solla, S. A. (1990): *A statistical approach to learning and generalization in layered neural networks*. Proc. IEEE **78**[10], 1568–1574. DOI:10.1109/5.58339.

MacKay, D. J. C. (1992a): *Bayesian interpolation*. Neural Comput. **4**[3], 415–447. http://www.inference.phy.cam.ac.uk/mackay/PhD.html, DOI:10.1162/neco.1992.4.3.415.

— (1992b): *A practical Bayesian framework for backpropagation networks*. Neural Comput. **4**[3], 448–472. http://www.inference.phy.cam.ac.uk/mackay/PhD.html, DOI:10.1162/neco.1992.4.3.448.

— (1992c): *Information-based objective functions for active data selection*. Neural Comput. **4**[4], 590–604. http://www.inference.phy.cam.ac.uk/mackay/PhD.html.

— (1992d): *The evidence framework applied to classification networks*. Neural Comput. **4**[5], 720–736. http://www.inference.phy.cam.ac.uk/mackay/PhD.html, DOI:10.1162/neco.1992.4.5.720.

— (2005): *Information Theory, Inference, and Learning Algorithms*, Version 7.2 (4th pr.) (Cambridge University Press, Cambridge). http://www.inference.phy.cam.ac.uk/mackay/itila. First publ. 1995.

Neal, R. M. (1996): *Bayesian Learning for Neural Networks*. (Springer, New York). DOI:10.1007/978-1-4612-0745-0, https://www.cs.toronto.edu/~radford/bnn.book.html.

Pratt, J. W., Raiffa, H., Schlaifer, R. (1996): *Introduction to Statistical Decision Theory*, 2nd pr. (MIT Press, Cambridge, USA). First publ. 1995.

Raiffa, H. (1970): *Decision Analysis: Introductory Lectures on Choices under Uncertainty*, 2nd pr. (Addison-Wesley, Reading, USA). First publ. 1968.

Raiffa, H., Schlaifer, R. (2000): *Applied Statistical Decision Theory*, repr. (Wiley, New York). First publ. 1961.

Savage, L. J. (1972): *The Foundations of Statistics*, 2nd rev. and enl. ed. (Dover, New York). First publ. 1954.

Tishby, N., Levin, E., Solla, S. A. (1989): *Consistent inference of probabilities in layered networks: predictions and generalizations*. Int. Joint Conf. Neural Networks **1989**, II-403–II-409. DOI:10.1109/IJCNN.1989.118274.