Hinxton, Cambridge CB10 1SA, UK.
e-mail: agf@sanger.ac.uk

1. Fire, A. *et al. Nature* **391,** 806–811 (1998).
2. Kamath, R. S. *et al. Nature* **421,** 231–237 (2003).
3. Kiger, A. *et al. J. Biol.* **2,** 27 (2003).
4. Paddison, P. J. *et al. Nature* **428,** 427–431 (2004).
5. Berns, K. *et al. Nature* **428,** 431–437 (2004).
6. Dykxhoorn, D. M., Novina, C. D. & Sharp, P. A. *Nature Rev. Mol. Cell Biol.* **4,** 457–467 (2003).
7. Hemann, M. T. *et al. Nature Genet.* **33,** 396–400 (2003).
8. Shoemaker, D. D., Lashkari, D. A., Morris, D., Mittmann, M. & Davis, R. W. *Nature Genet.* **14,** 450–456 (1996).
9. Jackson, A. L. *et al. Nature Biotechnol.* **21,** 635–637 (2003).
10. Chi, J. T. *et al. Proc. Natl Acad. Sci. USA* **100,** 6343–6346 (2003).

Learning theory

Past performance and future results

Carlo Tomasi

Learning from experience is hard, and predicting how well what we have learned will serve us in the future is even harder. The most useful lessons turn out to be those that are insensitive to small changes in our experience.

A hallmark of intelligent learning is that we can apply what we have learned to new situations. In the mathematical theory of learning, this ability is called generalization. On page 419 of this issue[1], Poggio *et al.* formulate an elegant condition for a learning system to generalize well.

As an illustration, consider practising how to hit a tennis ball. We see the trajectory of the incoming ball, and we react with complex motions of our bodies. Sometimes we hit the ball with the racket's sweet spot and send it where we want; sometimes we do less well. In the theory of supervised learning, an input–output pair exemplified by a trajectory and the corresponding reaction is called a training sample. A learning algorithm observes many training samples and computes a function that maps inputs to outputs. The learned function generalizes well if it does about as well on new inputs as on the old ones: if this is true, our performance during tennis practice is a reliable indication of how well we will play during the game.

Given an appropriate measure for the 'cost' of a poor hit, the algorithm could choose the least expensive function over the set of training samples, an approach to learning called empirical risk minimization. A classical result[2] in learning theory shows that the functions learned through empirical risk minimization generalize well only if the 'hypothesis space' from which they are chosen is simple enough. That there may be trouble in a poor choice of hypotheses is a familiar concept in most scientific disciplines. For instance, a high-degree polynomial fitted to a set of data points can swing wildly between them, and these swings decrease our confidence in the ability of the polynomial to make correct predictions about function values between available data points. For similar reasons, we have come to trust Kepler's simple description of the elliptical motion of heavenly bodies more than the elaborate system of deferents, epicycles and equants of Ptolemy's *Almagest*, no matter how well the latter fit the observations.

The classical definition of a 'simple enough' hypothesis space is brilliant but technically involved. For instance, the set of linear functions defined on the plane has a complexity (or Vapnik–Chervonenkis dimension[2]) of three because this is the greatest number of points that can be arranged on the plane so that suitable linear functions assume any desired combination of signs (positive or negative) when evaluated at the points. This definition is a mouthful already for this simple case. Although this approach has generated powerful learning algorithms[2], the complexity of hypothesis spaces for many realistic scenarios quickly becomes too hard to measure with this yardstick. In addition, not all learning problems can be formulated through empirical risk minimization, so classical results might not apply.

Poggio *et al.*[1] propose an elegant solution to these difficulties that builds on earlier intuitions[3–5] and shifts attention away from the hypothesis space. Instead, they require the learning algorithm to be stable if it is to produce functions that generalize well. In a nutshell, an algorithm is stable if the removal of any one training sample from any large set of samples results almost always in a small change in the learned function. *Post facto*, this makes intuitive sense: if removing one sample has little consequence (stability), then adding a new one should cause little surprise (generalization). For example, we expect that adding or removing an observation in Kepler's catalogue will usually not perturb his laws of planetary motion substantially.

FREDERIKE HEKURIG/GETTY IMAGES

In or out: success rests on learning algorithms that are stable against slight changes in input conditions[1].

The simplicity and generality of the stability criterion promises practical utility. For example, neuronal synapses in the brain may have to adapt (learn) with little or no memory of past training samples. In these cases, empirical risk minimization does not help, because computing the empirical risk requires access to all past inputs and outputs. In contrast, stability is a natural criterion to use in this context, because it implies predictable behaviour. In addition, stability could conceivably lead to a so-called online algorithm — that is, one that improves its output as new data become available.

Of course, stability is not the whole story, just as being able to predict our tennis performance does not mean that we will play well. If after practice we play as well as the best game contemplated in our hypothesis space, then our learning algorithm is said to be consistent. Poggio *et al.*[1] show that stability is equivalent to consistency for empirical risk minimization, whereas for other learning approaches stability only ensures good generalization. Even so, stability can become a practically important learning tool, as long as some key challenges are met. Specifically, Poggio *et al.*[1] define stability in asymptotic form, by requiring certain limits to vanish as the size of the training set becomes large. In addition, they require this to be the case for all possible probabilistic distributions of the training samples. True applicability to real situations will depend on how well these results can be rephrased for finite set sizes. In other words, can useful measures of stability and generalization be estimated from finite training samples? And is it feasible to develop statistical confidence tests for them? A new, exciting research direction has been opened. ■

Carlo Tomasi is in the Department of Computer Science, Duke University, Durham, North Carolina 27708, USA.
e-mail: tomasi@cs.duke.edu

1. Poggio, T., Rifkin, R., Mukherjee, S. & Niyogi, P. *Nature* **428,** 419–422 (2004).
2. Vapnik, V. N. *Statistical Learning Theory* (Wiley, New York, 1998).
3. Devroye, L. & Wagner, T. *IEEE Trans. Information Theory* **25,** 601–604 (1979).
4. Bousquet, O. & Elisseeff, A. J. *Machine Learning Res.* **2,** 499–526 (2002).
5. Kutin, S. & Niyogi, P. in *Proc. 18th Conf. Uncertainty in Artificial Intelligence, Edmonton, Canada,* 275–282 (Morgan Kaufmann, San Francisco, 2002).