General conditions for predictivity in learning theory

Tomaso Poggio[1], **Ryan Rifkin**[1,4], **Sayan Mukherjee**[1,3] **& Partha Niyogi**[2]

[1] *Center for Biological and Computational Learning, McGovern Institute Computer Science Artificial Intelligence Laboratory, Brain Sciences Department, MIT, Cambridge, Massachusetts 02139, USA*

[2] *Departments of Computer Science and Statistics, University of Chicago, Chicago, Illinois 60637, USA*

[3] *Cancer Genomics Group, Center for Genome Research/Whitehead Institute, MIT, Cambridge, Massachusetts 02139, USA*

[4] *Honda Research Institute USA Inc., Boston, Massachusetts 02111, USA*

Developing theoretical foundations for learning is a key step towards understanding intelligence. 'Learning from examples' is a paradigm in which systems (natural or artificial) learn a functional relationship from a training set of examples. Within this paradigm, a learning algorithm is a map from the space of training sets to the hypothesis space of possible functional solutions. A central question for the theory is to determine conditions under which a learning algorithm will generalize from its finite training set to novel examples. A milestone in learning theory[1–5] was a characterization of conditions on the hypothesis space that ensure generalization for the natural class of empirical risk minimization (ERM) learning algorithms that are based on minimizing the error on the training set. Here we provide conditions for generalization in terms of a precise stability property of the learning process: when the training set is perturbed by deleting one example, the learned hypothesis does not change much. This stability property stipulates conditions on the learning map rather than on the hypothesis space, subsumes the classical theory for ERM algorithms, and is applicable to more general algorithms. The surprising connection between stability and predictivity has implications for the foundations of learning theory and for the design of novel algorithms, and provides insights into problems as diverse as language learning and inverse problems in physics and engineering.

One of the main impacts of learning theory is on engineering. Systems that learn from examples to perform a specific task have many applications[6]. For instance, a system may be needed to recognize whether an image contains a face or not. Such a system could be trained with positive and negative examples: images with and without faces. In this case, the input image is a point in a multidimensional space of variables such as pixel values; its associated output is a binary 'yes' or 'no' label.

In the auditory domain, one may consider a variety of problems. Consider speaker authentication. The input is an acoustic utterance, and the system has to determine whether it was produced by a particular target speaker or not. Training examples would then consist of a set of utterances each labelled according to whether or not they were produced by the target speaker. Similarly, in speech recognition, one wishes to learn a function that maps acoustic utterances to their underlying phonetic sequences. In learning the syntax of a language, one wishes to learn a function that maps sequences of words to their grammaticality values. These functions could be acquired from training data.

In another application in computational biology, algorithms have been developed that can produce a diagnosis of the type of cancer from a set of measurements of the expression level of many thousands of human genes in a biopsy of the tumour measured with a complementary DNA microarray containing probes for a number of genes. Again, the software learns the classification rule from a set of examples, that is, from examples of expression patterns in a number of patients with known diagnoses.

What we assume in the above examples is a machine that is trained, instead of programmed, to perform a task, given data of the form $S=(x_i,y_i)_{i=1}^n$. Training means synthesizing a function that best represents the relation between the inputs x_i and the corresponding outputs y_i.

The basic requirement for any learning algorithm is generalization: the performance on the training examples (empirical error) must be a good indicator of the performance on future examples (expected error), that is, the difference between the two must be 'small' (see Box 1 for definitions; see also Fig. 1).

Probably the most natural learning algorithm is ERM: the algorithm 'looks' at the training set S, and selects as the estimated function the one that minimizes the empirical error (training error) over the functions contained in a hypothesis space of candidate

Box 1

Formal definitions in supervised learning

Convergence in probability. A sequence of random variables $\{X_n\}$ converges in probability to a random variable X (for example, $\lim_{n\to\infty}|X_n-X|=0$ in probability) if and only if for every $\epsilon>0$, $\lim_{n\to\infty}\mathbb{P}(|X_n-X|>\epsilon)=0$.

Training data. The training data comprise input and output pairs. The input data X is assumed to be a compact domain in an euclidean space and the output data Y is assumed to be a closed subset of $\mathbb{R}^k$. There is an unknown probability distribution $\mu(x,y)$ on the product space $Z=X\times Y$. The training set S consists of n independent and identically drawn samples from the distribution on Z:

$$S=\{z_1=(x_1,y_1),\ldots,z_n=(x_n,y_n)\}$$

Learning algorithms. A learning algorithm takes as input a data set S and outputs a function f_S that represents the relation between the input x and output y. Formally the algorithm can be stated as a map $L:\cup_{n\geq 1}Z^n\to\mathcal{H}$ where $\mathcal{H}$, called the hypothesis space, is the space of functions the algorithm 'searches' to select f_S. We assume that the algorithm is symmetric, that is, f_S does not depend on the ordering of the samples in S. Most learning algorithms are either regression or classification algorithms depending on whether y is real-valued or binary valued.

Loss functions. We denote the price we pay with $V(f,z)$ when the prediction for a given x is $f(x)$ and the true value is y. We assume that $V(f,z)$ is always bounded. A classical example of a loss function is the square loss $V(f,z)=(f(x)-y)^2$.

Expected error. The expected error of a function f is defined as

$$I[f]=\int_z V(f,z)d\mu(z)$$

which is also the expected error of a new sample z drawn from the distribution. In the case of square loss:

$$I[f]=\int_{X,Y}(f(x)-y)^2 d\mu(x,y)$$

We would like to find functions for which $I[f]$ is small. However, we cannot compute $I[f]$ because we are not given the distribution μ.

Empirical error. The following quantity, called empirical error, can be computed given the training data S:

$$I_S[f]=\frac{1}{n}\sum_{i=1}^n V(f,z_i)$$

Generalization and consistency. An algorithm generalizes if the function f_S selected by it satisfies for all S ($|S|=n$) and uniformly for any probability distribution μ

$$\lim_{n\to\infty}|I[f_S]-I_S[f_S]|=0 \quad \text{in probability}$$

An algorithm is (universally) consistent if uniformly for any distribution μ and any $\epsilon>0$

$$\lim_{n\to\infty}\mathbb{P}\left\{I[f_S]>\inf_{f\in\mathcal{H}}I[f]+\varepsilon\right\}=0$$

functions. Classical learning theory was developed around the study of ERM. One of its main achievements is a complete characterization of the necessary and sufficient conditions for generalization of ERM and its consistency[1–5]. Consistency (see Box 2) requires that the expected error of the solution converges to the expected error of the most accurate function in the hypothesis class $\mathcal{H}$. For ERM, generalization is equivalent to consistency[3]. Generalization of ERM can be ensured by restricting sufficiently the hypothesis space $\mathcal{H}$ (ref. 7 provides a theory within a classical mathematical framework).

The basic intuition here is that if the class $\mathcal{H}$ is too large, in the sense of containing too many wild functions, it is impossible to construct any useful approximating function using empirical data. Without restrictions on $\mathcal{H}$ there are functions that minimize the empirical error by fitting the data S exactly (thus $I_S[f_S] = 0$) but are very far away from the 'true' underlying function and therefore have a large expected error (thus $I[f_S]$ is large).

Although the classical theory has achieved a complete characterization of ERM, there are many learning algorithms, some existing and some likely to be discovered, that are not ERM. Several of the most effective, state-of-the-art algorithms—from square-loss regularization[8] to support vector machines[2], from bagging[9] to boosting[10, 11], from k-nearest neighbour[12] to vicinal risk minimization[13]—are not, strictly, ERM, because ERM is defined as minimization of the empirical error (see Box 1) within a fixed—a priori—hypothesis space. Though specific theoretical results can be derived for some of these algorithms[7,14,15], general conditions, representing a broadly applicable approach for checking generalization properties of any learning algorithm, would be desirable. For the case of ERM algorithms, Vapnik and Červonenkis[2] asked: what property must the hypothesis space $\mathcal{H}$ have for good generalization of ERM? For the case of more general algorithms, we note that any learning algorithm is a map L from data sets to hypothesis functions. For a general theory, it therefore makes sense to ask: what property must the learning map L have for good generalization error? Ideally a general answer to the second question must include an answer to the first question as well.

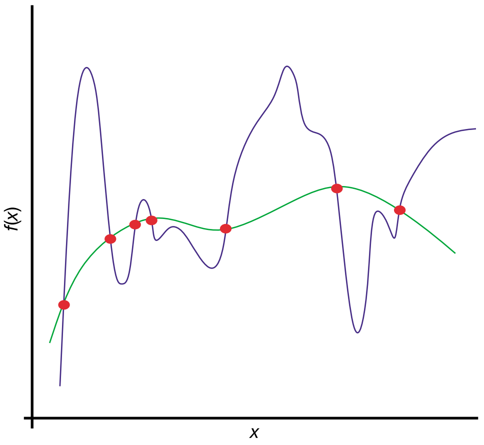

Figure 1 Example of an empirical minimizer with large expected error. In the case sketched here the data were generated from the 'true' green function. The blue function fits the data set and therefore has zero empirical error ($I_S[f_{\text{blue}}] = 0$). Yet it is clear that on future data, this function f_{blue} will perform poorly as it is far from the true function on most of the domain. Therefore $I[f_{\text{blue}}]$ is large. Generalization refers to whether the empirical performance on the training set ($I_S[f]$) will generalize to test performance on future examples ($I[f]$). If an algorithm is guaranteed to generalize, an absolute measure of its future predictivity can be determined from its empirical performance.

Recently, Bousquet and Elisseeff[15] proposed the notion of uniform stability to characterize the generalization properties of an algorithm (see Box 3). The idea—closely related to stability of well-posed problems[16]—is to look at the stability of the learning map L by considering how much the error at a point changes if the training set is perturbed. Uniform stability was too strong a stability requirement to subsume the classical theory (see Box 3), so the search remained open. Many different notions of algorithmic stability exist, going back at least to refs 14 and 17.

We first introduce the definition of cross-validation leave-one-out ($\mathrm{CV_{loo}}$) stability: the learning map L is distribution-independent, $\mathrm{CV_{loo}}$ stable if uniformly for all probability distributions μ $\lim_{n\to\infty} \sup_{i\in\{1,\ldots,n\}} |V(f_{S^i}, z_i) - V(f_S, z_i)| = 0$ in probability, where S^i denotes the training set S with the ith point removed.

$\mathrm{CV_{loo}}$ stability measures the difference in errors at a point z_i between a function obtained given the entire training set and one obtained given the same training set but with the point z_i left out (see Fig. 2). $\mathrm{CV_{loo}}$ stability is strictly weaker than uniform stability because the condition holds only on most (note the probabilistic quantification) training points and not for all possible points z (see also Box 3). For the supervised learning setting, the definition of $\mathrm{CV_{loo}}$ stability implements a specific and weak form of the general idea of stability of a well-posed problem: the function 'learned' from a training set should, with high probability, change little in its predictions for a small change in the training set, such as deletion of one of the examples.

The first crucial question is whether $\mathrm{CV_{loo}}$ stability is general enough to subsume the classical conditions of the consistency of ERM. The answer is surprising and positive[16]:

Theorem A. For 'good' loss functions the following statements are equivalent for ERM: (1) L is distribution-independent $\mathrm{CV_{loo}}$ stable; (2) ERM generalizes and is universally consistent; (3) $\mathcal{H}$ is uniform Glivenko–Cantelli (uGC; see Box 2).

Box 2

Classical results of empirical risk minimization algorithms

Empirical risk minimization (ERM) algorithms are defined as those satisfying:

$$I_S[f_S] = \min_{f\in\mathcal{H}} I_S[f]$$

The results described in this Letter for the special case of exact minimization are also valid in the general case of almost ERM, in which the minimum is not assumed to exist (see ref. 16), though the proofs are technically somewhat more complex.

The key theorem[1–5] of classical learning theory relates consistency of ERM to a constraint on the function classes $\mathcal{H}^2$.

Theorem. The following are equivalent for 'well-behaved' loss functions, such as the square-loss; (1) ERM generalizes and is consistent; (2) $\mathcal{H}$ is a uGC class.

A function class is a uGC class if universal, uniform convergence in probability holds

$$\lim_{n\to\infty} \sup_{\mu} \mathbb{P}\left(\sup_{f\in\mathcal{H}} \left| \frac{1}{n}\sum_{i=1}^{n} f(x_i) - \int_X f(x)\mathrm{d}\mu(x) \right| > \varepsilon \right) = 0.$$

The result extends to general loss functions if the functions induced by the composition of the loss V and hypothesis space $\mathcal{H}$ are uGC. For binary functions the uGC property reduces to the well known requirement of finite VC dimension.

Cucker and Smale[7] and Zhou[25] developed a complete and effective theory exploiting the property of compactness of $\mathcal{H}$, which is sufficient, but not necessary, for generalization and consistency of ERM, because compactness of $\mathcal{H}$ implies the uGC property of $\mathcal{H}$ but not vice versa.

We now ask whether CV$_{\text{loo}}$ stability is sufficient for generalization of any learning algorithm satisfying it. The answer to this question is negative (Theorem 11 in ref. 15 claims incorrectly that CV$_{\text{loo}}$ stability is sufficient for generalization, since there are counter-examples[16]).

A positive answer can be obtained by augmenting CV$_{\text{loo}}$ stability with stability of the expected error and stability of the empirical error to define a new notion of stability, CVEEE$_{\text{loo}}$ stability, as follows: the learning map L is distribution-independent, CVEEE$_{\text{loo}}$ stable if for all probability distributions μ: (1) is CV$_{\text{loo}}$ stable; (2) $\lim\limits_{n\to\infty} \sup\limits_{i\epsilon\{1,\ldots,n\}} |I[f_S] - I[f_{S^i}]| = 0$ in probability; (3)$\lim\limits_{n\to\infty} \sup\limits_{i\epsilon\{1,\ldots,n\}} |I_S[f_S] - I_{S^i}[f_{S^i}]| = 0$ in probability.

Properties (2) and (3) are weak and satisfied by most 'reasonable' learning algorithms. They are not sufficient for generalization. In the case of ERM CV$_{\text{loo}}$ stability is the key property since both conditions (2) and (3) are implied by consistency of ERM[16]. Unlike CV$_{\text{loo}}$ stability, CVEEE$_{\text{loo}}$ stability is sufficient for generalization for any learning algorithm[16]:

Theorem B. If a learning algorithm is CVEEE$_{\text{loo}}$ stable and the loss function is bounded, then f_S generalizes, that is uniformly for all μ $\lim\limits_{n\to\infty} |I[f_S] - I_S[f_S]| = 0$ in probability.

Notice that theorem B provides a general condition for generalization but not for consistency, which is not too surprising because the class of non-ERM algorithms is very large. In summary, CVEEE$_{\text{loo}}$ stability is sufficient for generalization for any algorithm and necessary and sufficient for generalization and consistency of ERM.

Good generalization ability means that the performance of the algorithm on the training set will accurately reflect its future performance on the test set. Therefore an algorithm that guarantees good generalization will predict well if its empirical error on the training set is small. Conversely, notice that it is also possible for an algorithm to generalize well but predict poorly (when both empirical and expected performances are poor). Crucially, therefore, one can empirically determine the predictive performance by looking at the error on the training set.

Box 3

Uniform stability

There have been many different notions of stability going back of Tikhonov[26], Devroye and Wagner[27] and Kearns and Ron[28]. A significant step was taken recently by Bousquet and Elisseeff[15], who defined uniform stability. The learning map L is uniformly stable if

$$\forall S \epsilon Z^n, \quad \forall i \epsilon \{1,\ldots,n\} \quad \sup_{z\epsilon Z} |V(f_S, z) - V(f_{S^i}, z)| \leq \beta^{(n)}$$

and $\beta^{(n)} = K/n$, where K is a constant. Uniform stability is a natural measure of continuity of L because the supremum measures the largest change at any point z. Uniform stability implies good generalization[15], and Tikhonov regularization algorithms (including support vector machines[29]) are uniformly stable[15].

Unfortunately, uniform stability is a very strong requirement because the change in error when a point is removed must be small for any $x,y \in Z$ and any training set. Most algorithms are not uniformly stable. For example, ERM with a hypothesis space of only two functions is not guaranteed to be uniformly stable[17].

The search remained open for a notion of stability that is sufficient for generalization, and necessary and sufficient for ERM. A partial answer was provided by Kutin and Niyogi[17] when they introduced the notion of cross validation (CV) stability and showed that this (with an additional weaker condition) was adequate in the classical Probably Approximately Correct (PAC) setting[30]. A general answer was not found.

In this Letter, we show that a leave-one-out version of CV stability (CV$_{\text{loo}}$) along with some more technical conditions provides the answer.

Learning techniques are similar to fitting a multivariate function to measurement data, a classical problem in nonparametric statistics[10,18,19]. The key point is that the fitting should be predictive. In this sense 'learning from examples' can also be considered as a stylized model for the scientific process of developing predictive theories from empirical data[2]. The classical conditions for generalization of ERM can then be regarded as the formalization of a 'folk theorem', which says that simple theories should be preferred among all of those that fit the data. Our stability conditions would instead correspond to the statement that the process of research should—most of the time—only incrementally change existing scientific theories as new data become available.

It is intellectually pleasing that the concept of stability, which cuts across so many different areas of mathematics, physics and engineering, turns out to play such a key role in learning theory. It is somewhat intuitive that stable solutions are predictive, but it is especially surprising that our specific definition of CVEEE$_{\text{loo}}$ stability fully subsumes the classical necessary and sufficient conditions on $\mathcal{H}$ for consistency of ERM.

In this Letter we have stated the main properties in an asymptotic form. The detailed statements[16] of the two stability theorems sketched here, however, provide bounds on the difference between empirical and expected error for any finite size of the training set.

Our observations on stability suggest immediately several other questions. The first challenge is to bridge learning theory and a quite different and broad research area—the study of inverse problems in applied mathematics and engineering[20]—since stability is a key condition for the solution of inverse problems (our stability condition can be seen as an extension[16] to the general learning problem of the classical notion of condition number that characterize stability of linear systems). In fact, while predictivity is at the core of classical learning theory, another motivation drove the development of several of the best existing algorithms (such as regularization algorithms of which SVMs are a special case): well-

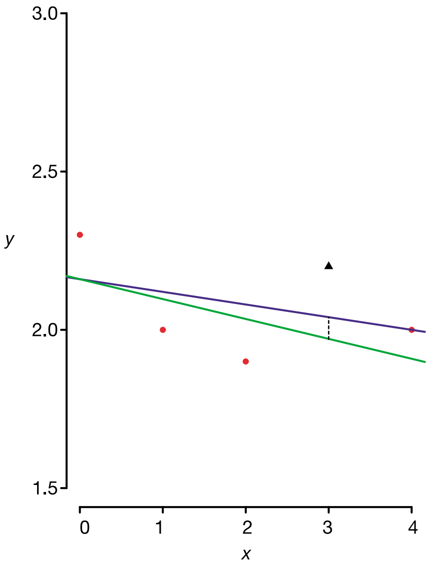

Figure 2 Measuring CV$_{100}$ stability in a simple case. The blue line was obtained by a linear regression (for example, ERM with square loss on a hypothesis space of linear functions) on all five training points ($n = 5$). The green line was obtained in the same way by 'leaving out' the black triangle from the training set. In this case, CV_{loo} stability requires that when a single point is removed from a data set, the change in error at the removed point (here indicated by the black dashed line) is small and decreases to zero in probability for n increasing to infinity.

posedness and, specifically, stability of the solution. These two requirements—consistency and stability—have been treated so far as 'defacto' separate, and in fact there was no a priori reason to believe that they are related[6]. Our new result shows these two apparently different motivations are actually completely equivalent for ERM.

The most immediate implication of CVEEE$_{\text{loo}}$ stability and its properties has to do with developing learning theory beyond the ERM approach. In particular, CVEEE$_{\text{loo}}$ stability can provide generalization bounds for algorithms other than ERM. For some of them a 'VC-style' analysis (see Box 2) in terms of complexity of the hypothesis space can still be used; for others, such as k-nearest neighbour, such an analysis is impossible because the hypothesis space has unbounded complexity or is not even defined, whereas CVEEE$_{\text{loo}}$ stability can still be used. Though a detailed analysis for specific algorithms needs to be done, some interesting observations in terms of stability can be inferred easily from existing analyses. For instance, the results of ref. 15 imply that regularization and SVMs are CVEEE$_{\text{loo}}$ stable; a version of bagging with the number k of regressors increasing with n is CVEEE$_{\text{loo}}$ stable[21]; k-nearest neighbour with $k \rightarrow \infty$ and $k/n \rightarrow 0$ and kernel rules with the width $h_n \rightarrow 0$ and $h_n n \rightarrow \infty$ are also CVEEE$_{\text{loo}}$ stable. Thus because of theorem B, all these algorithms have the generalization property (and some are also universally consistent).

More importantly, CVEEE$_{\text{loo}}$ stability may also suggest new algorithms that, unlike ERM, enforce stability directly. Similarly, it may be possible to gain a better understanding of existing statistical tests and develop new ones based on the definition of CVEEE$_{\text{loo}}$ stability and extensions of it. Furthermore, the search for equivalent and possibly 'simpler' conditions than CVEEE$_{\text{loo}}$ stability is open.

A theory of learning based on stability may have more direct connections with cognitive properties of the brain's mechanisms for learning (needless to say, learning is much more than memory). Stability is a condition on the learning machinery (or algorithms), as the classical conditions, such as the uGC property of $\mathcal{H}$, constrain the domain of learning—the hypothesis space. It is interesting that neural circuits approximating algorithms such as radial basis functions have been proposed as learning modules in the brain[22,23]. Such algorithms are CVEEE$_{\text{loo}}$ stable because they follow from regularization. There are certainly other learning algorithms that are biologically plausible, do not follow from regularization or ERM, but may be analysable in terms of CVEEE$_{\text{loo}}$ stability.

Consider for example, the problem of learning a language. The language learning algorithm $\mathcal{A}_L$ is a map from linguistic data (sentences produced by people) to computable functions (grammars) that are learned from those data. Corresponding to the learning algorithm $\mathcal{A}_L$ there exists a class $H_{\mathcal{A}_L}$ which is the class of all learnable grammars. Thus $H_{\mathcal{A}_L}$ is the hypothesis class corresponding to the language learning algorithm. In the tradition of generative linguistics identified most strongly with Chomsky, the class of possible natural language grammars $H_{\mathcal{A}_L}$ is called 'universal grammar', and different linguistic theories attempt to characterize the nature of this class. For example, the principles and parameters approach[24] tries to describe this class in a parametric fashion with a finite number of boolean parameters.

Although this tradition in generative linguistics is a meaningful one, it reflects an emphasis on the hypothesis class. In this sense, it is in the same philosophical spirit as the classical approach to learning theory of Vapnik and Červonenkis where conditions on the hypothesis space (expressed in terms of the so called VC dimension) are outlined for learnability. While $\mathcal{A}$ and $\mathcal{H}_{\mathcal{A}}$ are related, it is possible that in many cases, $\mathcal{A}$ may admit an easier mathematical characterization than $\mathcal{H}_{\mathcal{A}}$. Thus, for example, it may be possible that the language learning algorithm may be easy to describe mathematically while the class of possible natural language grammars may be difficult to describe. In that case, the shift in focus to the algorithm rather than the hypothesis class would correspond to a shift to a stability rather than a VC point of view.

It is often the case in the natural world that multiple representations of the same underlying phenomena are formally equivalent but some representations are more insightful than others. In the case of natural language, only time will tell whether $\mathcal{A}$ or $\mathcal{H}_{\mathcal{A}}$ will prove to be the easier and more insightful object. At present, learning theory in the context of language focuses heavily on the latter, and many 'learnable classes' are studied and theories are developed about them. We hope that focus on the learning algorithm may stimulate a new kind of language learning theory and practice. □

Received 3 November 2003; accepted 16 January 2004; doi:10.1038/nature02341.

1. Vapnik, V. & Chervonenkis, A. Y. The necessary and sufficient conditions for consistency in the empirical risk minimization method. *Pattern Recognit. Image Anal.* **1,** 283–305 (1991).
2. Vapnik, V. N. *Statistical Learning Theory* (Wiley, New York, 1998).
3. Alon, N., Ben-David, S., Cesa-Bianchi, N. & Haussler, D. Scale-sensitive dimensions, uniform convergence, and learnability. *J. Assoc. Comp. Mach.* **44,** 615–631 (1997).
4. Dudley, R. M. *Uniform Central Limit Theorems* (Cambridge studies in advanced mathematics, Cambridge Univ. Press, 1999).
5. Dudley, R., Gine, E. & Zinn, J. Uniform and universal Glivenko-Cantelli classes. *J. Theor. Prob.* **4,** 485–510 (1991).
6. Poggio, T. & Smale, S. The mathematics of learning: Dealing with data. *Not. Am. Math. Soc.* **50,** 537–544 (2003).
7. Cucker, F. & Smale, S. On the mathematical foundations of learning. *Bull. Am. Math. Soc.* **39,** 1–49 (2001).
8. Wahba, G. *Spline Models for Observational Data* (Series in Applied Mathematics, Vol. 59, SIAM, Philadelphia, 1990).
9. Breiman, L. Bagging predictors. *Machine Learn.* **24,** 123–140 (1996).
10. Hastie, T., Tibshirani, R. & Friedman, J. *The Elements of Statistical Learning* (Springer series in statistics, Springer, Basel, 2001).
11. Freund, Y. & Schapire, R. A decision-theoretic generalization of online learning and an application to boosting. *J. Comput. Syst. Sci.* **55,** 119–139 (1997).
12. Fix, E. & Hodges, J. Discriminatory analysis, nonparametric discrimination: consistency properties? (Techn. rep. 4, Project no. 21-49-004, USAF School of Aviation Medicine, Randolph Field, TX, 1951).
13. Bottou, L. & Vapnik, V. Local learning algorithms. *Neural Comput.* **4**(6), 888–900 (1992).
14. Devroye, L. & Wagner, T. Distribution-free performance bounds for potential function rules. *IEEE Trans. Inform. Theory* **25,** 601–604 (1979).
15. Bousquet, O. & Elisseeff, A. Stability and generalization. *J Machine Learn. Res.* **2,** 499–526 (2001).
16. Mukherjee, S., Niyogi, P., Poggio, T. & Rifkin, R. Statistical learning: stability is sufficient for generalization and necessary and sufficient for consistency of empirical risk minimization (CBCL Paper 223, Massachusetts Institute of Technology, 2002, revised 2003).
17. Kutin, S. & Niyogi, P. in *Proceedings of Uncertainty in AI* (eds Daruich, A. & Friedman, N.) (Morgan Kaufmann, Univ. Alberta, Edmonton, 2002).
18. Stone, C. The dimensionality reduction principle for generalized additive models. *Ann. Stat.* **14,** 590–606 (1986).
19. Donocho, D. & Johnstone, I. Projection-based approximation and a duality with kernel methods. *Ann. Stat.* **17,** 58–106 (1989).
20. Engl, H., Hanke, M. & Neubauer, A. *Regularization of Inverse Problems* (Kluwer Academic, Dordrecht, 1996).
21. Evgeniou, T., Pontil, M. & Elisseeff, A. Leave one out error, stability, and generalization of voting combinations of classifiers. *Machine Learn.* (in the press).
22. Pouget, A. & Sejnowski, T. J. Spatial transformations in the parietal cortex using basis functions. *J. Cogn. Neurosci.* **9,** 222–237 (1997).
23. Poggio, T. A theory of how the brain might work. *Cold Spring Harbor Symp. Quant. Biol.* **55,** 899–910 (1990).
24. Chomsky, N. *Lectures on Government and Binding* (Foris, Dordrecht, 1995).
25. Zhou, D. The covering number in learning theory. *J. Complex.* **18,** 739–767 (2002).
26. Tikhonov, A. N. & Arsenin, V. Y. *Solutions of Ill-posed Problems* (Winston, Washington DC, 1977).
27. Devroye, L. & Wagner, T. Distribution-free performance bounds for potential function rules. *IEEE Trans. Inform. Theory* **25,** 601–604 (1979).
28. Kearns, M. & Ron, D. Algorithmic stability and sanity-check bounds for leave-one-out cross-validation. *Neural Comput.* **11,** 1427–1453 (1999).
29. Evgeniou, T., Pontil, M. & Poggio, T. Regularization networks and support vector machines. *Adv. Comput. Math.* **13,** 1–50 (2000).
30. Valiant, L. A theory of the learnable. *Commun. Assoc. Comp. Mach.* **27,** 1134–1142 (1984).

Acknowledgements We thank D. Panchenko, R. Dudley, S. Mendelson, A. Rakhlin, F. Cucker, D. Zhou, A. Verri, T. Evgeniou, M. Pontil, P. Tamayo, M. Poggio, M. Calder, C. Koch, N. Cesa-Bianchi, A. Elisseeff, G. Lugosi and especially S. Smale for several insightful and helpful comments. This research was sponsored by grants from the Office of Naval Research, DARPA and National Science Foundation. Additional support was provided by the Eastman Kodak Company, Daimler-Chrysler, Honda Research Institute, NEC Fund, NTT, Siemens Corporate Research, Toyota, Sony and the McDermott chair (T.P.).

Competing interests statement The authors declare that they have no competing financial interests.

Correspondence and requests for materials should be addressed to T.P. (tp@ai.mit.edu).