Inductive Confidence Machines for Regression

Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman

Department of Computer Science, Royal Holloway, University of London
Egham, Surrey TW20 0EX, England
{harris,konstant,vovk,alex}@cs.rhul.ac.uk

Abstract. The existing methods of predicting with confidence give good accuracy and confidence values, but quite often are computationally inefficient. Some partial solutions have been suggested in the past. Both the original method and these solutions were based on transductive inference. In this paper we make a radical step of replacing transductive inference with inductive inference and define what we call the Inductive Confidence Machine (ICM); our main concern in this paper is the use of ICM in regression problems. The algorithm proposed in this paper is based on the Ridge Regression procedure (which is usually used for outputting bare predictions) and is much faster than the existing transductive techniques. The inductive approach described in this paper may be the only option available when dealing with large data sets.

1 Introduction

When presented with a test example, traditional machine learning algorithms only output a bare prediction, without any associated confidence values. For example, Support Vector Machine (Vapnik, 1998, Part II) outputs just one number (a bare prediction, as we will say), and one has to rely on the previous experience or relatively loose theoretical upper bounds on the probability of error to gauge the quality of the given prediction. This is also true for the more traditional Ridge Regression (RR) procedure as it is used in machine learning (see, e.g., Saunders, Gammerman, & Vovk, 1998).

Gammerman, Vapnik, and Vovk (1998) proposed what we call in this paper "Transductive Confidence Machine" (TCM), which complements the bare predictions with measures of confidence in those predictions. Both Transductive (see, e.g., Proedrou et al., 2001) and Inductive (proposed in this paper) Confidence Machines are currently built on top of the standard machine learning algorithms for outputting bare predictions; we will call the latter the *underlying algorithms*. TCM suggested in Gammerman et al. (1998) was greatly improved in (Saunders, Gammerman, & Vovk, 1999). Vovk, Gammerman, and Saunders (1999) introduced the universal confidence values: the best confidence values one can hope to obtain. The universal confidence values are defined using the algorithmic theory of randomness (or, in the simplest situations, Kolmogorov complexity; see Li and Vitányi, 1997) and are computable only in a very weak sense ("computable in the limit"). There are reasons to believe that the version

T. Elomaa et al. (Eds.): ECML, LNAI 2430, pp. 345–356, 2002.

of TCM defined in Saunders et al. (1999), when coupled with a good underlying algorithm, can give confidence values as good as the universal values provided by the algorithmic theory of randomness (Nouretdinov et al., 2001).

The main disadvantage of the existing variants of TCM is their relative computational inefficiency. An original motivation behind the idea of transductive inference (Vapnik, 1998) was to obtain more computationally efficient versions of learning algorithms. Whereas this remains an interesting long-term goal, so far in the theory of confident predictions a side-effect of using transduction has been computational inefficiency; for every test example, all computations need to be started from scratch. It was not clear, however, how prediction with confidence could be implemented without resorting to transduction.

Saunders, Gammerman, and Vovk (2001) proposed a much more efficient version of TCM; other efficient versions are described in Vovk and Gammerman (2001).

This paper makes a much more radical step introducing Inductive Confidence Machine, ICM. The computational efficiency of ICM is almost as good as that of the underlying algorithm. There is some loss in the quality of the confidence values output by the algorithm, but we show that this loss is not too serious. On the other hand, the improvement in the computational efficiency is massive.

ICM will be defined in Section 2. In the following section we will prove the validity of the predictive regions it outputs. Finally, in the last section we give some experimental results that measure the efficiency of our algorithm based on those criteria.

In the rest of this introductory section we will briefly describe the relevant literature.

Computing confidence values is, of course, an established area of statistics. In the non-parametric situations typically considered in machine learning the most relevant notion is that of tolerance regions (Fraser, 1957; Guttman, 1970). What we do in this paper is essentially finding tolerance regions without parametric assumptions, only assuming that the data is generated by some completely unknown i.i.d. distribution (we will call this the i.i.d. assumption). Traditional statistics, however, did not consider, in this context, the high-dimensional problems typical of machine learning, and no methods have been developed in statistics which could compete with TCM and ICM.

The two main areas in the mainstream machine learning which come close to providing confidence values similar to those output by TCM and ICM are the Bayesian methods and PAC theory. For detailed discussion, see (Melluish et al., 2001); here our discussion will be very brief.

Quite often Bayesian methods make it possible to complement bare predictions with probabilistic measures of their quality (theoretically this is always possible, but in practice there can be great computational difficulties); e.g., Ridge Regression can be obtained as a Bayesian prediction under specific assumptions and then it can be complemented by a measure of its accuracy (such as the variance of the *a posteriori* distribution). They require, however, strong extra assumptions, which the theory of TCM and ICM avoids. In fact, Bayesian

methods are only applicable if the stochastic mechanism generating the data is known in every detail; in practice, we will rarely be in such a happy situation. (Melluish et al., 2001) show how misleading Bayesian methods can become when their assumptions are violated and how robust TCM results are (ICM results are as robust).

PAC theory, in contrast, only makes the general i.i.d. assumption. There are some results, first of all those by Littlestone and Warmuth (1986; see also Cristianini & Shawe-Taylor, Theorem 4.25 and 6.8), which are capable of giving non-trivial confidence values for data sets that might be interesting in practice. However, in order for the PAC methods to give non-trivial results the data set should be particularly clean; they will fail in the vast majority of cases where TCM and ICM produce informative results (see Melluish et al., 2001). The majority of relevant results in the PAC theory are even less satisfactory in this respect: they either involve large explicit constants or do not specify the relevant constants at all (see, e.g., Cristianini and Shawe-Taylor, 2000, Section 4.5).

2 Inductive Confidence Machine

In this paper we are only interested in the problem of regression, with Ridge Regression as the underlying algorithm. In contrast to the original Ridge Regression method, every prediction output by ICM is not a single real value, but a set of possible values, called a *predictive region*.

We are given a training set $\{(x_1, y_1), \ldots, (x_l, y_l)\}$ of l examples, where $x_i \in \mathbb{R}^n$ are the attributes and $y_i \in \mathbb{R}$ are the labels, $i = 1, \ldots, l$, and the attributes of a new example $x_{l+1} \in \mathbb{R}^n$. When fed with a confidence level, such as 99%, ICM is required to find a predictive region such that one can be 99% confident that the label y_{l+1} of the new example will be covered by that predictive region.

The idea of ICM is as follows. We split the training set into two subsets:

- the *proper training set* $\{(x_1, y_1), \ldots, (x_m, y_m)\}$ with $m < l$ elements, and
- the *calibration set* $\{(x_{m+1}, y_{m+1}), \ldots, (x_l, y_l)\}$ with $k := l - m$ elements;

m and k are parameters of the algorithm. We apply the Ridge Regression method to the proper training set, and using the derived rule we associate a strangeness measure with every pair (x_i, y_i) in the calibration set. This measure can be defined as

$$\alpha_i := |y_{m+i} - \hat{y}_{m+i}|, \ i = 1, \ldots, k, \tag{1}$$

where $\hat{y}_{m+i}$ are the predictions given by the derived rule; later we will also consider other definitions. For every potential label y of the new unlabelled example x_{l+1} we can analogously define

$$\alpha_{k+1} := |y - \hat{y}_{l+1}|,$$

where $\hat{y}_{l+1}$ is the prediction for the new example given by the derived rule. Let us defined the *p-value* associated with the potential label y as

$$p(y) := \frac{\#\{i = 1, \ldots, k+1 : \alpha_i \geq \alpha_{k+1}\}}{k+1},$$

where $\#A$ stands for the number of elements in the set A; to emphasize the dependence on the training set and x_{l+1}, we will also write $p(x_1, y_1, \ldots, x_l, y_l, x_{l+1}, y)$ in place of $p(y)$. In Section 3 we will prove that $p(y)$ are indeed valid p-values.

Suppose we are given *a priori* some confidence level $1 - \delta$, where $\delta > 0$ is a small constant (typically one takes 1% or 5%); sometimes we will say that δ is the *significance level*. Given the significance level δ, the predictive region output by ICM is

$$\{y : p(y) > \delta\}\,. \tag{2}$$

In Section 4 we will see that this can be done efficiently.

3 Validity of the Predictive Regions

Recall that valid p-values $p(y)$ should satisfy, for any i.i.d. distribution $\mathbf{P}$ and for every significance level δ,

$$\mathbf{P}\{p(y) \le \delta\} \le \delta. \tag{3}$$

The next proposition shows that (2) defines valid p-values under the general i.i.d. assumption when the randomization is done over the training as well as over the new example (x_{l+1}, y_{l+1}).

Proposition 1. *For every probability distribution P in $\mathbb{R}^n \times \mathbb{R}$ and every significance level $\delta > 0$,*

$$P^{l+1}\Big\{(x_1, y_1, \ldots, x_l, y_l, x_{l+1}, y_{l+1}) : p(x_1, y_1, \ldots, x_l, y_l, x_{l+1}, y_{l+1}) \le \delta\Big\} \le \delta.$$

Proof. We will actually prove the stronger assertion that (3) is true if the randomization is done only over the calibration set and the new example. Let us fix the proper training set $x_1, y_1, \ldots, x_m, y_m$; our goal is to prove

$$P^{k+1}\Big\{(x_{m+1}, y_{m+1}, \ldots, x_{l+1}, y_{l+1}) : p(x_{m+1}, y_{m+1}, \ldots, x_{l+1}, y_{l+1}) \le \delta\Big\} \le \delta. \tag{4}$$

We can imagine that the sequence

$$(x_{m+1}, y_{m+1}), \ldots, (x_{l+1}, y_{l+1})$$

is generated in two stages:

– first the unordered set

$$\{x_{m+1}, y_{m+1}, \ldots, x_{l+1}, y_{l+1}\} \tag{5}$$

is generated;

– one of the $(k+1)!$ possible orderings $\{x_{\pi(m+1)}, y_{\pi(m+1)}, \ldots, x_{\pi(l+1)}, y_{\pi(l+1)}\}$ (where $\pi : \{m+1, \ldots, l+1\} \to \{m+1, \ldots, l+1\}$ is a permutation) of (5) is chosen (some of these orderings may lead to the same sequence if some example occurs twice in (5)).

Already the second stage will ensure (4): indeed, $p(y_{l+1}) \leq \delta$ if and only if α_{l+1} is among the $\lfloor \delta(k+1) \rfloor$ largest α_i; since all permutations π are equiprobable, the probability of this event will not exceed δ. □

This proof shows that the method of computing $\alpha_1, \ldots, \alpha_{k+1}$ should only satisfy the following condition in order for the computed p-values to be valid: every α_i, $i = 1, \ldots, k+1$, should be computed only from (x_{m+i}, y_{m+i}), the proper training set, and the unordered set $\{x_{m+1}, y_{m+1}, \ldots, x_{l+1}, y_{l+1}\}$, where y_{l+1} is understood to be the postulated label y of x_{l+1}. Definition (1) and definition (7) (see section 4) obviously satisfy this requirement.

Fix some significance level δ (small positive constant). Proposition 1 shows that ICM is valid in the following sense. Either the ICM prediction is correct (i.e., the prediction region contains the true label y_{l+1}) or an event of small (at most δ) probability occurred. If δ is chosen so that we are prepared to ignore events of probability δ, we can rely on the predictive region covering the true label.

4 Explicit ICM

In this section we will give a slightly more explicit representation of ICM.

Let us denote by

$$\alpha_{(1)}, \ldots, \alpha_{(k^*)}$$

the sequence of all α_i corresponding to the calibration set sorted in the descending order, with all repetitions deleted; let

$$j_s := \#\{\alpha_i : \alpha_i \geq \alpha_{(s)}\}, \ s = 1, \ldots, k^*,$$

be the number of αs at least as large as $\alpha_{(s)}$ (if all α_i are different, $j_1 = 1, j_2 = 2, \ldots$). Fix the confidence level $1 - \delta$. The "attainable" significance levels will be of the form $\frac{j_s}{k+1}$; decrease δ, if necessary, so that it is of this form: $\delta = \frac{j_s}{k+1}$ for some $s = 1, \ldots, k^*$.

It can be easily checked that the predictive region output by ICM can be represented as

$$(\hat{y}_{l+1} - \alpha_{(s)}, \hat{y}_{l+1} + \alpha_{(s)}), \tag{6}$$

provided the αs are computed according to (1).

Notice that the computational overhead of ICM is light; it is almost as efficient as the underlying algorithm. The decision rule is computed from the proper training set only once, and it is applied to the calibration set also only once. The value of s corresponding to the given significance level δ and the value $\alpha_{(s)}$ can be also computed in advance. For every test example we need to apply the decision rule to it to find its y_{l+1}; once this is done, computing the predictive region from (6) is trivial.

Another Way of Computing α_i

Definition (1) defines the strangeness of the new example as the error of the decision rule on it. A natural way to make this strangeness measure more precise is to take into account the predicted accuracy of the decision rule f found from the proper training set on a given unlabelled example from $\{x_{m+1}, \ldots, x_{l+1}\}$. Hopefully this should lead to smaller prediction regions.

Instead of using the strangeness measure $\alpha_i = |y_i - \hat{y}_i|$, we can use

$$\alpha_i := \left| \frac{y_i - \hat{y}_i}{\sigma_i} \right|, \tag{7}$$

where σ_i is an estimate of the accuracy of the decision rule f on x_i. More specifically, we take $\sigma_i := e^{\mu_i}$, where μ_i is the RR prediction of the value $\ln(|(y_i - f(x_i)|)$ for the example x_i. The use of the logarithmic scale instead of the direct one ensures that the estimate is always positive; besides, relatively more weight is given to examples with classifications close to f's predictions.

It is easy to see that when using α_i computed from (1) ICM will output predictive intervals of the same length for all test examples. This is not longer the case when (7) is used; the length of the predictive interval will be proportional to the predicted accuracy of f on the new example. What we are actually accomplishing by using (7) is that the predictive regions obtained will be smaller for points where the RR prediction is good and larger for points where it is bad.

Fixed Prediction Interval

There are two possible modes of using the p-values computed from (2):

1. For a given significance level δ, find a predictive region such that we can be $1 - \delta$ confident that it covers the true label.
2. Given a fixed predictive region, find the maximum level at which we can be confident that the true label will be covered.

The first mode corresponds to the regression ICM considered so far. The second mode is essentially what is usually done in classification problems, where a fixed predictive region may represent one of the possible classifications.

It is clear that the maximum confidence interval at which a given predictive interval $[a, b]$ is valid will be $1 - j_s/(k+1)$, where s is the maximum number such that

$$\alpha_{(s)} \geq \max\left(|\hat{y}_i - a|, |\hat{y}_i - b|\right).$$

5 Experimental Results

The first set of experiments check how reliable the obtained predictive regions are. We count the percentage of wrong predictive intervals; in other words, how many times the algorithm fails to give a predictive region that contains the real label of every test example. In effect this checks empirically the validity of our

Table 1. The average success of the predictions made, for different confidence levels using (1) as strangeness measure

Kernel Type	Empirical reliability		
	90%	95%	99%
Polynomial	93.6%	97.4%	99.3%
RBF	97.5%	98.6%	99.6%
ANOVA Splines	97.7%	97.2%	98.8%

Table 2. The average success of the predictions made, for different confidence levels using (7) as strangeness measure

Kernel Type	Empirical reliability		
	90%	95%	99%
Polynomial	95.2%	97.8%	99.1%
RBF	97.3%	98.8%	99.6%
ANOVA Splines	95%	97.6%	99.2%

algorithm, which was proven theoretically in Section 3. We expect that for a large number of examples the percentage of wrong predictions will not exceed (and perhaps will be close to) the specified significance level.

A second set of experiments checks the tightness of our predictive regions by calculating the median value of the lengths of all predictive regions obtained for a specific significance level. This gives us a measure of how efficient our algorithm is. We prefer using the median value instead of the mean, because it is more robust: if a few of the predictions are extreme (either very large or very small) due to noise or due to over-fitting, the average will be affected, while the median will remain unchanged.

The proposed algorithm has been tested on the Boston Housing data set, which gives the values of houses, ranging from 5K to 50K, depending on 13 attributes. In the experiments 100 splits of this data set have been used, with different examples for the proper training, calibration, and test sets each time. In every split the calibration set consisted of 99 examples, the test set of 25 examples, and the rest of 382 examples was used as the proper training set. In Tables 1 to 4 we give the widths of the predictive regions and the *empirical reliability* (i.e., the percentage of cases when the true label turned out to be outside the predictive region) of these bounds for specific significance levels (1%, 5%, and 10%) and for specific kernels (Polynomial, RBF, and ANOVA) used in conjunction with RR.

The results in Tables 1 and 2 confirm the validity of our algorithm: the rate of successful predictions is at least equal to the desired accuracy.

In Tables 3 and 4 we present results about tightness of our predictive regions for both variations of our algorithm. As we can see, in both cases the best results

Table 3. The median width of the predictive regions, for different confidence levels using (1) as strangeness measure

Kernel Type	Median width		
	90%	95%	99%
Polynomial	9.6	12.6	16.1
RBF	9.9	13.5	29.4
ANOVA Splines	9.5	12.2	15.4

Table 4. The median width of the predictions made, for different accuracy levels using (7) as strangeness measure

Kernel Type	Median width		
	90%	95%	99%
Polynomial	9.5	11.8	15.6
RBF	10	12.7	23.5
ANOVA Splines	9.7	11.7	15

Table 5. Comparison of the mean width of the predictive regions, for ICM and TCM Variation1

Algorithm	Mean width		
	90%	95%	99%
ICM	10.8	12.7	17.5
TCM Variant1	12.4	16.7	28.8

Table 6. Comparison of the median width of the predictive regions, for ICM and TCM Variation2

Algorithm	Median width		
	90%	95%	99%
ICM	9.5	11.8	15.6
TCM Variant2	7.5	9.3	18.8

were obtained when we used the ANOVA splines as our kernel function. By comparing the results for the two variations we notice that the method which uses (7) as strangeness value gives, on average, slightly better results. The difference is becoming relatively larger as we move toward higher confidence levels.

Figures 1 and 2 complement the information given in Tables 3 and 4 for ANOVA splines by also giving other characteristics of the distribution of the predictive interval widths. These figures show that the distribution of the method which uses (7) as strangeness measure is more spread out, as we would expect.

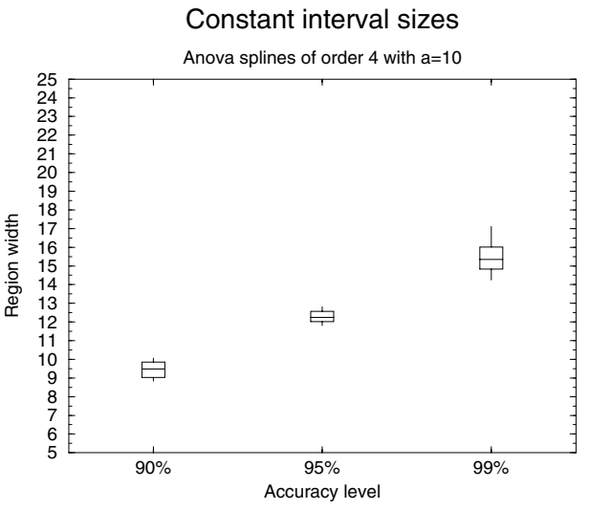

Fig. 1. Medians, upper and lower quartiles, and 10th and 90th percentile of the distributions of the predictive interval widths for the method using (1) as strangeness value

Finally, in Tables 5 and 6 we compare our algorithm with two variations of TCM which are described in (Melluish, Vovk, Gammerman, 1999) and (Nouretdinov, Melluish, Vovk, 2001), using the polynomial kernel[1]. It is obvious that ICM outperforms the first variation of the TCM in all hypothesis tests, while compared with the second variation the difference is small. Though the set of α values in TCM is richer than the one in ICM and the Ridge Regression rule is derived using less examples in the case of induction, this doesn't seem to worsen the performance of the latter significantly.

We also tested the algorithm on the Bank Rejection and the CPU Activity data sets both of which consist of 8192 examples split into 4096 training and 4096 test examples. This was done in order to demonstrate the algorithms ability to handle large sets. The Bank Rejection data set was generated from a simplistic simulator, which simulated the queues in a series of banks. Our task is to predict the rate of rejections(i.e., the fraction of customers that are turned away from the bank because all the open tellers have full queues) depending on 32 attributes.

The CPU Activity data set is a collection of a computer systems activity measures collected from a Sun Sparcstation 20/712 with 128 Mbytes of memory running in a multi-user university department. Users would typically be doing a large variety of tasks ranging from accessing the internet, editing files or running

[1] In Table 5 we compare the mean widths instead of the median as in (Melluish, Vovk, Gammerman, 1999) only mean widths are reported

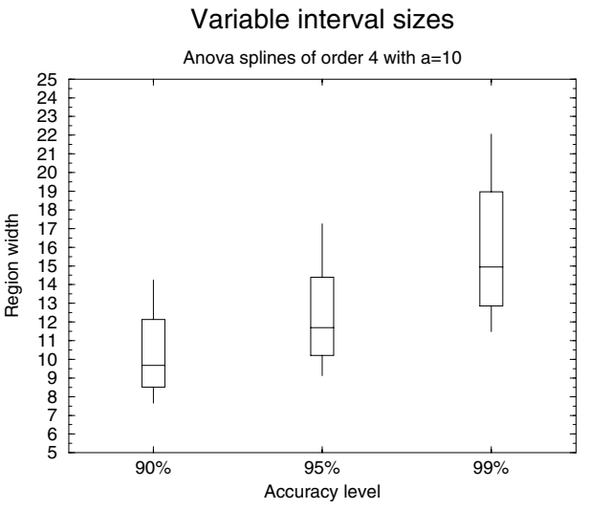

Fig. 2. Medians, upper and lower quartiles, and 10th and 90th percentiles of the distributions of the predictive interval widths for the method using (7) as strangeness value

Table 7. The median width of the predictive regions, for the Bank Rejection and the CPU Activity data sets

Data Set	Strangeness measure	Median width		
		90%	95%	99%
Bank	(1)	0.29	0.39	0.51
	(7)	0.24	0.28	0.40
CPU	(1)	8.89	11.81	16.79
	(7)	8.77	10.71	15.89

very cpu-bound programs. Our task is to predict the portion of time that the cpus run in tuser mode depending on 12 attributes.

The median widths of the predictive regions obtained by the Bank Rejection and the CPU Activity data sets are listed in Table 7. The rate of rejections in the Bank Rejection data set ranges from 0 to 0.7 and the portion of time that the cpus run in user mode in the CPU Activity data set ranges from 0 to 99. So even for a 99% confidence level the second variation of our algorithm gives a predictive region which covers only 57% and 17% of the whole range of labels for each set respectively.

6 Conclusions

We have defined ICM, a computationally efficient confidence machine for the regression problem based on inductive inference. In addition to the bare prediction ICM outputs a measure of its accuracy which has a clear probabilistic interpretation. The experimental results obtained give good empirical reliability that is constantly above the specified confidence level. This confirms that the algorithm can be used for obtaining reliable predictions. Furthermore, the width of our predictive regions, is almost as tight as that of the transductive version. The tightness of our predictive regions can be seen by the fact that our best result for the Boston Housing data set, which is given by the second variation of the algorithm (using (7) as strangeness measure), predicts a region that is only 33% of the whole range of house prices at the 99% confidence level.

Acknowledgements

We are grateful to David Surkov for useful discussions. This work was partially supported by EPSRC through grants GR/L35812 ("Support Vector and Bayesian learning algorithms"), GR/M14937 ("Predictive complexity: recursion-theoretic variants"), and GR/M16856 ("Comparison of Support Vector Machine and Minimum Message Length methods for induction and prediction").

References

1. Cristianini, N., & Shawe-Taylor, J. (2000). *Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge: Cambridge University Press.
2. Fraser, D. A. S. (1957). *Non-parametric Methods in Statistics*. New York: Wiley.
3. Gammerman, A., Vapnik, V., & Vovk, V. (1998). Learning by transduction. *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence* (pp. 148–156). San Francisco: Morgan Kaufmann.
4. Li, M., & Vitányi, P. (1997). *An Introduction to Kolmogorov Complexity and Its Applications*. Second edition. New York: Springer.
5. Melluish, T., Saunders, C., Nouretdinov, I., & Vovk, V. (2001). Comparing the Bayes and typicalness frameworks. *ECML'01*.
6. Melluish, T., Vovk, V., & Gammerman, A. (1999). Transduction for Regression Estimation with Confidence. *NIPS'99*.
7. Nouretdinov, I., Melluish, T., & Vovk, V. (1999). Ridge Regression Confidence Machine. *Proceedings of the 18th International Conference on Machine Learning*.
8. Nouretdinov, I., Vovk, V., V'yugin, V., & Gammerman, A. (2001). Transductive Confidence Machine is universal. Work in progress.
9. Proedrou, K., Nouretdinov, I., Vovk, V., & Gammerman, A. (2001). Transductive Confidence Machines for Pattern Recognition. *Proceedings of the 13th European Conference on Machine Learning*.
10. Saunders, C., Gammerman, A., & Vovk, V. (1999). Transduction with confidence and credibility. *Proceedings of the 16th International Joint Conference on Artificial Intelligence* (pp. 722–726).

11. Saunders, C., Gammerman, A., & Vovk, V. (2000). Computationally efficient transductive machines. *ALT'00 Proceedings*.
12. Vapnik, V. (1998). *Statistical Learning Theory*. New York: Wiley.
13. Vovk, V., Gammerman, A., & Saunders, C. (1999). Machine-learning applications of algorithmic randomness. *Proceedings of the 16th International Conference on Machine Learning* (pp. 444–453).
14. Vovk, V., & Gammerman, A. (2001). *Algorithmic Theory of Randomness and its Computer Applications*. Manuscript.
15. Vovk, V., and Gammerman, A. (1999). Statistical applications of algorithmic randomness. *Bulletin of the International Statistical Institute. The 52nd Session. Contributed Papers. Tome LVIII. Book 3* (pp. 469–470).