Transductive Confidence Machines for Pattern Recognition

Kostas Proedrou, Ilia Nouretdinov, Volodya Vovk and Alex Gammerman

Technical Report

CLRC-TR-01-02

June 18, 2001

Royal Holloway
University of London

Department of Computer Science
Egham, Surrey TW20 0EX, England

Abstract

We propose a new algorithm for pattern recognition that outputs some measures of "reliability" for every prediction made, in contrast to the current algorithms that output "bare" predictions only. Our algorithm is based on the well-known nearest neighbours algorithm and uses a similar rule to infer predictions.

1 Introduction

Current machine learning algorithms usually lack measures that can give an indication about how "good" the predictions are. Even when such measures are present they have certain disadvantages, such as:

- ⋄ They cannot be applied to individual test examples.
- ⋄ They often are not very useful in practice (PAC theory); see Nouretdinov, Vovk et al. (2001).
- ⋄ They often rely on strong underlying assumptions (Bayesian methods); see Melluish, Saunders et al. (2001).

In our case none of these disadvantages are present. Our only assumption is that data items are independent and produced by the same stochastic mechanism; our measures of confidence are applicable to individual examples, and they produce reasonable results for benchmark data sets (and so potentially are useful in practice).

Our prediction method is inspired by so called algorithmic theory of randomness. A short description of this theory is the subject of Section 2. Then, in Section 3, we describe our algorithm, and in the next section we give some experimental and comparison results for our algorithm as applied to the USPS data set.

As will be clear later, our algorithm follows the transductive approach; thus we call it transductive confidence machine (TCM-NN). This method is expected to outperform the inductive version at the cost of heavier computations.

2 Algorithmic theory of randomness

According to Martin-Löf's definition a function $t : Z^* \to [0,1]$ is a test for randomness (with respect to the class of all iid distributions in the set Z) if

- ⋄ for all $n \in \mathbb{N}$, for all $r \in [0,1]$ and for all probability distributions P in Z,

$$P^n\{z \in Z^n : t(z) \le r\} \le r \tag{1}$$

- ⋄ t is semi-computable from above

Here Z is a space that possesses some computability properties; in our application, Z is the set of all possible classified examples.

Martin-Löf proved that there exists a universal test for randomness smaller from all other tests up to a multiplicative constant. Such a test would be able to find all non-random patterns in a sequence of elements. Unfortunately, universal tests for randomness are not computable. Thus we have to approximate them using valid (in the sense of satisfying (1)) non-universal tests. The values taken by randomness tests will be called *p-values*.

3 Nearest Neighbours and Randomness

3.1 Formal setting of the problem

We have a training set $\{(x_1, y_1), \ldots, (x_m, y_m)\}$, of m elements, where $x_i = (x_i^1, \ldots, x_i^n)$ is the set of feature values for example i and y_i is the classification for example i, taking values from a finite set of possible classifications, which we identify as $\{1, 2, \ldots, c\}$. We also have a test set of s examples similar to the ones in the training set, only this time the actual classifications are withheld from us. Our goal is to assign to every test example one of the possible classifications. For every classification we also want to give some confidence measures, valid in the sense of (1), that will enable us to gain more insight in the predictions that we make.

3.2 Transductive confidence machine

Let us define the sorted sequence (in ascending order) of the distances of example i from the other examples with the same classification y as D_i^y, the jth shortest distance in this sequence as D_{ij}^y and the equivalent sequence for distances containing examples with classification different from y as D_i^{-y}. We assign to every example a measure called the individual strangeness measure. This measure defines the strangeness of the example in relation to the rest of the examples. In our case the individual strangeness measure for an example i with label y is defined as

$$\alpha_i = \frac{\sum_{j=1}^{k} D_{ij}^y}{\sum_{j=1}^{k} D_{ij}^{-y}}.$$

Thus, our measure for strangeness is the ratio of the sum of the k nearest distances from the same class to the sum of the k nearest distances from all other classes. This is a natural measure to use, as the strangeness of an example increases when the distance from the examples of the same class becomes bigger or when the distance from the other classes becomes smaller.

Now let us return to algorithmic randomness theory. In Vovk and Gammerman (1999) and Nouretdinov, Melluish and Vovk (2001) it is proved that the function

$$p := \frac{\#\{i : \alpha_i \geq \alpha_{\text{new}}\}}{m+1},$$

where α_{new} is the α for the test example (assuming there is only one test example, or that the test examples are processed one at a time), is a valid randomness test. Thus, the p-value for the sequence $\{\alpha_1, \ldots, \alpha_m, \alpha_{new}\}$, where $\{\alpha_1, \ldots, \alpha_m\}$ are the strangeness measures for the training examples and α_{new} is the strangeness measure of a test example with a possible classification assigned to it, is the value p. We can now give our algorithm.

TCM-NN Algorithm

---

for $i = 1$ to m do
  Find and store D_i^y and D_i^{-y}

end for
Calculate alpha values for all training examples
Calculate the dist vector as the distances of the new example from all training examples
for $j = 1$ to c do
  for every training example t classified as j do
    if $D^j_{tk} > \mathrm{dist}(t)$ recalculate the alpha value of example t
  end for
  for every training example t classified as non-j do
    if $D^{-j}_{tk} > \mathrm{dist}(t)$ recalculate the alpha value of example t
  end for
  Calculate alpha value for the new example classified as j
  Calculate p-value for the new example classified as j
end for
Predict the class with the largest p-value
Output as confidence one minus the 2nd largest p-value
Output as credibility the largest p-value

The first measure we are using gives us the confidence in our prediction. This is a measure that provides an indication of how improbable the classifications other than the predicted classification are. The second measure is the credibility of our prediction. This measure shows how suitable the training set is for the classification of the specific test example. For more details see Vovk & Gammerman (1999).

3.3 Inductive confidence machine

Also the Inductive Confidence Machine is used in this paper.

The difference between ICM and TCM is in the method of calculating p-values.

In ICM we fix some number $m' < m$. A similar formula is used for alphas:

$$\alpha_i = \frac{\sum_{j=1}^{k} D^y_{ij}}{\sum_{j=1}^{k} D^{-y}_{ij}}, m'+1 \leq i \leq m,$$

but we find nearest neighbours for the last $m - m'$ examples only amongst first m' examples; and p-values are computed by formula:

$$p := \frac{\#\{i : m'+1 \leq i \leq m, \alpha_i \geq \alpha_{\mathrm{new}}\}}{m - m' + 1},$$

This method also give valid p-values.

4 Experimental Results

We have tested our algorithm using a random split of the USPS data set that consists of handwritten digits from the set $\{0, 1, \ldots, 9\}$. The training set consists of 7291 examples and the test set of 2007 examples. Each example has

256 attributes (pixels) that describe the given digit. The standard comparison criterion in classification problems is the percentage of incorrectly classified examples. Here we shall also use a second one. We fix a specific significance level, say 1%, and we accept as possible classifications the ones whose p-value is above that level.

We compare our algorithm with the nearest neighbours confidence machine that uses induction instead of transduction (ICM-NN) in the 1 nearest neighbour case. The results are shown in the next tables.

Usual Euclidean distance is used, but all the data were pre-processed before applying the algorithm. As any image from the USPS data set was represented as 256 numbers $(x_1, ..., x_{256})$, we replaced it by $(y_1, ..., y_{256})$, where

$$y_i = \frac{x_i - S}{T}, S = \frac{\sum_{i=1}^{256} x_i}{256},$$

$$T = \sqrt{\frac{\sum_{i=1}^{256} (x_i - S)^2}{256}}$$

The aim of this preprocessing is to normalize the level of brightness. After it the mean-value of each image is 0 and the standard deviation is 1.

And the distance between $(y_1, ..., y_{256})$ and $(y'_1, ..., y'_{256})$ is calculated by formula:

$$d = \sqrt{\sum_{i=1}^{256} (y_i - y'_i)^{256}}.$$

The column "One class" gives the number of examples for which a confident prediction is made, the column "≥ 2 classes" gives the number of examples for which two or more possible classifications were not excluded at the given significance level, and the column "No class" gives the number of examples for which all possible classifications were excluded at the given significance level.

In the terms of confidence and credibility, the column "One class" corresponds to classifications with confidence more than 100%$-$Level and credibility more than Level; the column "≥ 2 classes" means that confidence is no greater than 100%$-$Level and the credibility is more than Level; and the column "No class" means that credibility is no greater than Level.

Table 1 Comparison of ICM-NN and TCM-NN.

Learning machine	Level	One class	≥ 2 classes	No class
TCM-NN	1%	94.77%	5.23%	0%
	5%	92.38%	0%	7.62%
ICM-NN	1%	94.47%	5.53%	0%
	5%	85.80%	0%	14.20%

In the table 2 we compare our results with Nearest Neighbours without confidence for this dataset and with Support Vector machines, described in Vapnik (1998).

Table 2 Error percentage of different mehods

Learning Algorithms	TCM-NN	ICM-NN	simple NN	SVM
% of error	4.28%	4.63%	4.29%	4.04%

Theoretically, we would expect the transductive algorithm to perform better, as it uses a much richer set of alpha values. The results on the overall error percentage and on the second criterion, though, indicate that both algorithms perform well, at least on the given data set.

5 Conclusion

Confidence machines seem to be very useful in making "qualitative" predictions for individual test examples. Their performance at predicting correctly, i.e., minimizing the error percentage, is heavily dependent upon their base algorithm (nearest neighbours in our case), as they use a very similar rule for deciding which classification to choose. It remains an open problem though whether one can find valid tests for randomness (under the general iid assumption) that are better approximations to the universal tests for randomness than the one used here.

References

Cristianini, N., Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and Other Kernel-based Methods.* Cambridge: Cambridge University Press.

Fraser, D. A. (1976). *Non-parametric methods in statistics.* New York: Wiley.

Li, M., Vitányi, P. (1997). *An introduction to Kolmogorov complexity and its applications* (2nd edn.). New York: Springer.

Melluish, T., Saunders, C., Nouretdinov, I., Vovk, V. (2001) *Comparing the Bayes and typicalness frameworks.* Accepted for ECML2001.

Nouretdinov, I., Melluish, T., Vovk V. (2001). *Ridge Regresson Confidence Machine.* Accepted for ICML2001.

Nouretdinov, I., Vovk, V., Vyugin, M., Gammerman, A. (2001). *Pattern recognition and density estimation under the general iid assumption.* Accepted for COLT2001/EUROCOLT2001.

Vapnik, V. (1998). *Statistical Learning Theory.* New York: Wiley.

Vovk, V., Gammerman, A. (1999). *Algorithmic Theory of Randomness and its Computer Applications.* Manuscript.