18

Inductive Conformal Prediction: Theory and Application to Neural Networks

Harris Papadopoulos
Frederick University
Cyprus

1. Introduction

Traditional machine learning algorithms for pattern recognition just output *simple predictions*, without any associated *confidence values*. Confidence values are an indication of how likely each prediction is of being correct. In the ideal case, a confidence of 99% or higher for all examples in a set, means that the percentage of erroneous predictions in that set will not exceed 1%.

Knowing the likelihood of each prediction enables us to assess the extent to which we can rely on it. For this reason, predictions that are associated with some kind of confidence values are highly desirable in many risk-sensitive applications, such as those used for medical diagnosis or financial analysis. In fact, such information can benefit any application that requires human-computer interaction, as confidence values can be used to determine the way in which each prediction will be treated. For instance, a filtering mechanism can be employed so that only predictions which satisfy a certain level of confidence will be taken into account, while the rest can be discarded or passed on to a human for judgement.

There are two main areas in mainstream machine learning that can be used in order to obtain some kind of confidence values; the Bayesian framework and the theory of Probably Approximately Correct learning (PAC theory). Quite often the Bayesian framework is used for producing algorithms that complement individual predictions with probabilistic measures of their quality. On the other hand, PAC theory can be used for producing upper bounds on the probability of error for a given algorithm with respect to some confidence level $1 - \delta$. Both of these approaches however, have their drawbacks.

In order to apply the Bayesian framework one is required to have some prior knowledge about the distribution that generates the data. When the correct prior is known, Bayesian methods provide optimal decisions. For real world data sets though, as the required knowledge is not available, one has to assume the existence of an arbitrarily chosen prior. In this case, if the assumed prior is incorrect, the resulting confidence levels may also be "incorrect"; for example the predictive regions output for the 95% confidence level may contain the true label in much less than 95% of the cases. This signifies a major failure, as we would expect confidence levels to bound the percentage of expected errors. An experimental demonstration of how misleading Bayesian methods can become when their assumptions are violated can be found in (Melluish et al., 2001).

PAC theory on the contrary, only assumes that the data are generated by some completely unknown i.i.d. distribution. There are some PAC methods that are capable of establishing non-trivial bounds that might be interesting in practice. In order for them to do so though, the data set should be particularly clean. If this is not the case, which is not for the majority of data sets, the bounds obtained from these methods are very loose and as such they are not very useful in practice. A demonstration of the crudeness of PAC bounds can be found in (Nouretdinov et al., 2001a), where there is an example of Littlestone and Warmuth's bound (found in (Cristianini & Shawe-Taylor, 2000), Theorems 4.25 and 6.8) applied to the USPS data set. In addition, PAC theory has two other drawbacks: (a) the majority of relevant results either involve large explicit constants or do not specify the relevant constants at all; (b) the bounds obtained by PAC theory are for the overall error and not for individual test examples.

A new approach to obtaining confidence values was suggested in (Saunders et al., 1999) and (Vovk et al., 1999), where what we call in this chapter "Conformal Prediction" (CP) was proposed. Conformal Predictors are built on top of traditional algorithms, called underlying algorithms, but unlike the latter they complement each of their predictions with a measure of confidence; they also produce a measure of "credibility", which serves as an indicator of how suitable the training data are for classifying the example.

In contrast to Bayesian methods, CPs give probabilistically valid results, as they are only based on the general i.i.d. assumption; a comparison between Bayesian methods and CPs can be found in (Melluish et al., 2001). Furthermore, unlike PAC theory, they produce confidence measures that are useful in practice and are associated with individual test examples. Different variants of CPs are described in the papers (Saunders et al., 1999; Nouretdinov et al., 2001b; Proedrou et al., 2002; Papadopoulos et al., 2008). The results reported in these papers show that not only the confidence values output by CPs are useful in practice, but also that their accuracy is comparable to, and sometimes even better than, that of traditional machine learning algorithms.

The only disadvantage of CPs is their relative computational inefficiency. This is due to the use of transductive inference, since all computations have to start from scratch for every test example. Unfortunately, this computational inefficiency problem renders the application CP highly unsuitable for any approach that requires long training times such as Neural Networks. For this reason, a modification of the original CP approach called "Inductive Conformal Prediction" (ICP) was proposed in (Papadopoulos et al., 2002a) for regression and in (Papadopoulos et al., 2002b) for pattern recognition. As suggested by its name, ICP replaces the transductive inference used in the original approach with inductive inference. As a result, ICPs are almost as computationally efficient as their underlying algorithms.

This chapter gives a detailed study of ICP and describes its application to Neural Networks, which is one of the most widely used approaches for solving machine learning problems. The next Section summarises the general idea of Conformal Prediction, while Section 3 details the way CPs and ICPs work and analyses the impact that the choice between trunsductive and inductive inference has on the performance of Conformal Prediction. Note that in order to differentiate clearly between the original CP and ICP approaches, the former will be mostly called Transductive Conformal Prediction (TCP). Section 4 explains the application of ICP to Neural Networks, first presented in (Papadopoulos et al., 2007), and Section 5 lists some experimental results obtained by testing the Neural Networks ICP on benchmark data sets. Finally, Section 6 presents the conclusions of the chapter.

2. Conformal prediction

This Section gives an outline of the main idea behind conformal prediction; for more details see (Vovk et al., 2005). We are given a training set (z_1, …, z_l) of examples, where each $z_i \in Z$ is a pair (x_i, y_i); $x_i \in \mathbb{R}^d$ is the vector of attributes for example i and y_i is the classification for that example. We are also given a new unclassified example x_{l+1} and our task is to predict the classification y_{l+1} of this example. We know a priori the set of all possible labels Y_1, …, Y_c and our only assumption, as with all the problems we are interested in, is the general i.i.d. model (z_1, z_2,… are independent and identically distributed). Now suppose we can measure how likely it is that a given sequence of classified examples were drawn independently from the same probability distribution; in other words how typical the sequence is wrt the i.i.d. model. Then by measuring the typicalness of the extended sequence

$$((x_1, y_1),...,(x_l, y_l),(x_{l+1}, Y_j)) \tag{1}$$

we would in effect be measuring the likelihood of the label Y_j being the true label of our new example x_{l+1}, since this is the only component of our sequence that was not given to us. In the spirit of (Martin-Löf, 1966), a function p: $Z^* \to [0,1]$ is a test for randomness wrt the i.i.d. model if

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

$$P^n\{z \in Z^n : p(z) \le \delta\} \le \delta; \tag{2}$$

- p is semi-computable from above.

We will use the term *p-value function* to refer to such a function, since this definition is practically equivalent to the notion of p-values used in traditional statistics. In effect, the second requirement, that p is semi-computable from above, is completely irrelevant from the practical point of view, since the p-value functions of any interest in applications of statistics are always computable.

Therefore, we can obtain the typicalness of a sequence of examples by using a computable function p: $Z^* \to [0,1]$ which satisfies (2). We will call the output of this function for the sequence

$$((x_1, y_1),...,(x_l, y_l),(x_{l+1}, Y_j)) \tag{3}$$

(where Y_j is one of the c possible labels of our new example) the p-value of Y_j and denote it by $p(Y_j)$. If the p-value of a given label is under some very low threshold, say 0.05, this would mean that this label is highly unlikely, since such sequences will only be generated at most 5% of the time by any i.i.d. process.

A p-value function can be constructed by considering how different each example in our sequence is from all other examples. In order to formalize the fact that the order in which examples appear should not matter we use the concept of a bag (also called a multiset); we write $\llbracket z_1,...,z_n \rrbracket$ to denote the bag consisting of the elements $z_1,...,z_n$. We use a family of functions $A_n : Z^{(n-1)} \times Z \to \mathbb{R}$, n = 1, 2, …, which assign a numerical score

$$\alpha_i = A_n(\llbracket z_1,...,z_{i-1},z_{i+1},...,z_n \rrbracket, z_i), \tag{4}$$

to each example z_i, indicating how different it is from the examples in the bag $\llbracket z_1,...,z_{i-1},z_{i+1},...,z_n \rrbracket$; such families of functions are called *nonconformity measures*. The nonconformity scores of all examples can now be used for computing the p-value of our sequence with the function

$$p(z_1,...,z_n) = \frac{\#\{i = 1,...,n : \alpha_i \geq \alpha_n\}}{n}. \tag{5}$$

This function satisfies (2); a proof can be found in (Nouretdinov et al., 2001a).

2.1 Measuring nonconformity

We can measure the nonconformity α_i of each example z_i in a bag $\llbracket z_1,...,z_n \rrbracket$ with the aid of some traditional machine learning method, which we call the *underlying algorithm* of the CP. Given a bag of examples $\llbracket z_1,...,z_n \rrbracket$ as training set, each such method creates a prediction rule $D_{\llbracket z_1,...,z_n \rrbracket}$, which maps any unclassified example x to a label $\hat{y}$. As this prediction rule is based on the examples in the bag, the deviation of the predicted label

$$\hat{y}_i = D_{\llbracket z_1,...,z_n \rrbracket}(x_i) \tag{6}$$

from the actual label y_i of the example z_i tells us how different z_i is from the rest of the examples in the bag. Therefore, this deviation gives us a measure of the nonconformity of example z_i.

Alternatively, we can create the prediction rule $D_{\llbracket z_1,...,z_{i-1},z_{i+1},...,z_n \rrbracket}$ using all the examples in the bag except z_i, and measure the deviation of

$$\hat{y}_i = D_{\llbracket z_1,...,z_{i-1},z_{i+1},...,z_n \rrbracket}(x_i) \tag{7}$$

from y_i.

3. Transductive and inductive conformal predictors

This Section gives a general description of the way Transductive and Inductive Conformal Predictors work and analyses their differences. Before focusing on Conformal Predictors, it first looks at the main concepts of Transductive and Inductive inference. It then details the steps that the two approaches follow and explains the meaning of the confidence and credibility measures they produce. This is followed by the presentation of an alternative mode in which CPs can be used and a discussion about the difference between the validity and the usefulness of their results. Finally, it compares the two approaches both in terms of computational efficiency and accuracy.

3.1 Transductive and inductive inference

The difference between transductive and inductive inference is very informal and far from being clear-cut. In this subsection we describe the main idea behind each one, so that the difference between Transductive and Inductive Conformal Prediction becomes clearer.

In inductive inference, we use our training set to generate a more or less general "rule" (or "model", or "theory") about the data, which we then apply to each test pattern to obtain our

predictions. Hence all the information we need from the training set are incorporated into our general "rule" and we do not make any direct use of the training examples in order to produce each prediction. In Transductive inference on the other hand, the first step, generating a general "rule", is skipped. Thus no processing is applied to the training set beforehand and all our computations are based on each individual test example, using the actual training set for deriving our prediction.

3.2 Transductive conformal prediction

The general steps the Transductive Conformal Prediction approach follows for a given input vector x_{l+g} are:

- Consider all possible classifications $Y_1, \ldots, Y_c$ and apply the underlying algorithm to every one of the possible completions

$$\begin{matrix} (x_1, y_1),\ldots,(x_l, y_l),(x_{l+g}, Y_1) \\ \vdots \\ (x_1, y_1),\ldots,(x_l, y_l),(x_{l+g}, Y_c) \end{matrix} \tag{8}$$

- For every possible completion Y_j, assign a nonconformity score to each training example $(x_1, y_1), \ldots, (x_l, y_l)$ and to the pair (x_{l+g}, Y_j). This process will result in the sequences

$$\begin{matrix} \alpha_1^{(Y_1)},\ldots,\alpha_l^{(Y_1)},\alpha_{l+g}^{(Y_1)} \\ \vdots \\ \alpha_1^{(Y_c)},\ldots,\alpha_l^{(Y_c)},\alpha_{l+g}^{(Y_c)} \end{matrix} \tag{9}$$

- Compute the p-value for x_{l+g} being classified as each possible label Y_j by applying (5) to the corresponding sequence

$$\alpha_1^{(Y_j)},\ldots,\alpha_l^{(Y_j)},\alpha_{l+g}^{(Y_j)}. \tag{10}$$

  So that

$$p(Y_j) = \frac{\#\{i = 1,\ldots,l,l+g : \alpha_i^{(Y_j)} \geq \alpha_{l+g}^{(Y_j)}\}}{l+1}. \tag{11}$$

- Predict the classification with the largest p-value.
- Output as confidence to this prediction one minus the second largest p-value, and as credibility the p-value of the output prediction, i.e. the largest p-value.

Note that this process, which includes the application of the CPs underlying algorithm c times, is repeated for each input vector $x_{l+1}, \ldots, x_{l+r}$. This is the cause of the computational inefficiency of the original CP approach.

3.3 Inductive conformal prediction

Inductive Conformal Prediction splits the training set into two parts, the *proper training set* and the *calibration set*. It then applies the underlying algorithm to the proper training set and uses the examples in the calibration set together with the new example to compute the

p-value for each possible classification. As a result, it only needs to apply the underlying algorithm once. The general steps inductive conformal prediction follows are:

- Split the training set into two smaller sets, the proper training set with $m := l - q$ examples and the calibration set with q examples; where q is a parameter of the algorithm.
- Use the proper training set $(z_1,\ldots,z_m)$ to generate a general rule $D_{[\![z_1,\ldots,z_m]\!]}$ for classifying new examples; this general rule is created by the underlying algorithm.
- Assign a nonconformity score to each one of the examples in the calibration set. This will result in the sequence $\alpha_{m+1},\ldots,\alpha_{m+q}$.
- For each input vector $x_{l+g} : g = 1,\ldots,r$,
  - consider each possible classification $Y_j : j = 1,\ldots,c$ and compute the nonconformity score $\alpha_{l+g}^{(Y_j)}$ of each pair (x_{l+g}, Y_j).
  - Compute the p-value for x_{l+g} being classified as each possible label Y_j by applying (5) to the nonconformity scores of the calibration examples together with $\alpha_{l+g}^{(Y_j)}$:

$$\alpha_{m+1},\ldots,\alpha_{m+q},\alpha_{l+g}^{(Y_j)}. \tag{12}$$

So that

$$p(Y_j) = \frac{\#\{i = m+1,\ldots,m+q,l+g : \alpha_i \geq \alpha_{l+g}^{(Y_j)}\}}{q+1}. \tag{13}$$

  - Predict the classification with the largest p-value.
  - Output as confidence to this prediction one minus the second largest p-value, and as credibility the p-value of the output prediction, i.e. the largest p-value.

The parameter q in the above scheme determines the number of training examples that will be allocated to the calibration set and the nonconformity scores of which will be used by the ICP to calculate its p-values. These examples should only take up a small portion of the training set, so that their removal will not dramatically reduce the predictive ability of the underlying algorithm. As we are mainly interested in the confidence levels of 99% and 95%, the calibration sizes we use are of the form $q = 100n - 1$, where $n \in \mathbb{N}$; according to (13), in order for a prediction to have a confidence level of 99%, the new example should be among the $\left\lfloor \frac{q+1}{100} \right\rfloor$ strangest examples when assigned all other possible classifications (so that the p-value of these classifications will be 0.01).

3.4 Confidence and credibility measures

Confidence gives us a measure of how likely our prediction is compared to all other possible classifications, according to the training set. To give a more detailed explanation of the confidence measure we need to recall the basic property of valid p-values; for any i.i.d. distribution P and for every significance level δ,

$$P\{p(Y_j) \leq \delta\} \leq \delta. \tag{14}$$

This means that if the p-value of a label Y_j is smaller than or equal to a significance level δ, then either Y_j is not the true label or an event of at most δ probability occurred. Now suppose that δ is equal to the second largest p-value for a given input vector x_{l+g} with a set of possible labels $\{Y_1, \ldots, Y_c\}$. This would mean that the probability of any label other than Y_j, where $p(Y_j)$ is the largest p-value, being the true label is at most δ. Consequently, our confidence value for each prediction is one minus the probability of any one of the other labels being the true label; the higher the confidence value for a prediction the less likely for it not being the true label. Note that this is an informal argument, as it uses a δ dependent on the observed examples.

The credibility measure is equal to the highest p-value of any one of the possible classifications being the true label according to the training set. As such it gives us an indication of how good the training set is for classifying the current example i.e. if the credibility of a prediction is very low this means that either the training set is not random or the new example is not representative of the training set.

3.5 The two modes of conformal prediction

The p-values obtained by (11) and (13) for each possible classification, can be used in two different modes:

- For each test example output the predicted classification together with a confidence and credibility measure for that classification.
- Given a confidence level $1 - \delta$, where $\delta > 0$ is the significance level (typically a small constant), output the appropriate set of classifications such that one can be $1 - \delta$ confident that the true label will be included in that set.

The first case corresponds to the algorithms detailed in Sections 3.2 and 3.3. In the second case the CP outputs the set

$$\{Y_j : p(Y_j) > \delta\}, \tag{15}$$

where $j = 1, \ldots, c$ (c is the number of possible classifications). In other words, it outputs the set consisting of all the classifications that have a greater than δ p-value of being the true label.

3.6 Validity and usefulness

The p-values computed by the functions (11) and (13) are valid in the sense of satisfying (2), provided that the data in question are drawn independently from the same probability distribution. Thus, the particular nonconformity measure definition and underlying algorithm used by a given conformal predictor do not influence the validity of its p-values in any way; i.e. if the model generating the data is i.i.d., the results produced by any ICP or TCP will be valid. In fact, any function can be used as a nonconformity measure definition, even if its output has nothing to do with how nonconforming the input example is. The use of such a measure will not have an impact on the validity of the results produced by the CP, but it will, on the other hand, affect their usefulness.

To demonstrate the influence of an inadequate nonconformity measure definition on the results of a CP, let us consider the case of a trivial definition that always returns the value of 1 for any given example. This will make the p-values of all possible labels equal to 1 and will result in a randomly chosen prediction with a confidence of 0%, which although is valid, does not provide us with any information. Therefore, if the nonconformity measure

definition or the underlying algorithm of a CP are not suitable for the data in question, this will be reflected in the usefulness of the resulting confidence measures.

3.7 Comparison

Although the main motivation behind the introduction of transductive inference, by Vapnik (Vapnik, 1998), was the creation of more computationally efficient versions of learning algorithms, this does not seem to be the case in the theory of conformal prediction. The Transductive CP starts all computations from scratch each time a new unclassified example arrives. This turns out to be very computationally inefficient, since it means that for every test example it has to apply the underlying algorithm and compute all the nonconformity scores of the examples c times, one for each possible classification. On the other hand, the Inductive CP carries out all these computations in advance, which makes it much faster. It only needs to calculate the nonconformity scores of the new example being assigned each one of the possible classifications, using the already generated general rule.

In order to analyse further the computational efficiency difference between ICP and TCP, let us consider the computational complexity of each method with respect to the complexity of its underlying algorithm U. The complexity of U when applied to a data set with l training examples and r test examples will be

$$\Theta(U_{\text{train}}(l) + rU_{\text{apply}}), \tag{16}$$

where $U_{\text{train}}(l)$ is the time required by U to generate its general rule and U_{apply} is the time needed to apply this general rule to a new example. Note that although the complexity of any algorithm also depends on the number of attributes d that describe each example, this was not included in our notation for simplicity reasons. The corresponding complexity of the TCP will be

$$\Theta(rc(U_{\text{train}}(l+1) + (l+1)U_{\text{apply}})), \tag{17}$$

where c is the number of possible labels of the task; we assume that the computation of the nonconformity scores and p-values for each possible label is relatively fast compared to the time it takes to train and apply the underlying algorithm. Analogously, the complexity of the ICP will be

$$\Theta(U_{\text{train}}(l-q) + (q+r)U_{\text{apply}}), \tag{18}$$

where q is the size of the calibration set. Notice that the ICP takes less time than the original method to generate the prediction rule, since

$$U_{\text{train}}(l-q) < U_{\text{train}}(l), \tag{19}$$

while it then repeats U_{apply} a somewhat larger amount of times. The time needed for applying the prediction rule of most inductive algorithms, however, is insignificant compared to the amount of time spent for generating it. Consequently, the ICP will in most cases be slightly faster than the original method, as it spends less time during the most complex part of its underlying algorithm's computations. On the contrary, the corresponding TCP repeats a slightly bigger number of computations than the total computations of its underlying algorithm for rc times, thing that makes it much slower than both the original method and the ICP.

The only drawback of the ICP is a small loss in terms of accuracy. This is due to the fact that the TCP uses all the training examples for the training of its underlying algorithm, whereas the ICP uses only the examples in the proper training set. Furthermore, the TCP uses a richer set of nonconformity scores, computed from all the training examples, when calculating the p-values for each possible classification, as opposed to the small part of training examples, the calibration set, the ICP uses for the same purpose. As the experimental results in Section 5 and in (Papadopoulos et al., 2002a; Papadopoulos et al., 2002b; Papadopoulos et al., 2007) show, this loss of accuracy is negligible while the improvement in computational efficiency is massive.

4. Neural networks ICP

In this Section we analyse the Neural Networks ICP. This method can be implemented in conjunction with any Neural Network for pattern recognition as long as it uses the 1-of-*n* output encoding, which is the typical encoding used for such networks. We first give a detailed description of this encoding and then move on to the definition of two nonconformity measures for Neural Networks. Finally, we detail the Neural Networks ICP algorithm.

4.1 Output encoding

Typically the output layer of a classification Neural Network consists of c units, each representing one of the c possible classifications of the problem at hand; thus each label is encoded into c target outputs. To explicitly describe this encoding consider the label, $y_i = Y_u$ of a training example i, where $Y_u \in \{Y_1,...,Y_c\}$ is one of the c possible classifications. The resulting target outputs for y_i will be

$$t_1^i,...,t_c^i \tag{20}$$

where

$$t_j^i = \begin{cases} 1, \text{ if } j = u, \\ 0, \text{ otherwise} \end{cases} \tag{21}$$

for j = 1,..., c. Here we assumed that the Neural Network in question has a softmax output layer, as this was the case for the networks used in our experiments. The values 0 and 1 can be adjusted accordingly depending on the range of the output activation functions of the network being used.

As a result of this encoding, the prediction $\hat{y}_g$ of the network, for a test pattern g, will be the label corresponding to its highest output value.

4.2 Nonconformity measures

According to the above encoding, for an example i with true classification Y_u, the higher the output o_u^i (which corresponds to that classification) the more conforming the example, and the higher the other outputs the less conforming the example. In fact, the most important of all other outputs is the one with the maximum value $\max_{j=1,...,c: j \neq u} o_j^i$, since that is the one which might be very near or even higher than o_u^i.

So a natural nonconformity measure for an example $z_i = (x_i, y_i)$ where $y_i = Y_u$ would be defined as

$$\alpha_i = \max_{j=1,\ldots,c: j \neq u} o_j^i - o_u^i, \tag{22}$$

or as

$$\alpha_i = \frac{\max_{j=1,\ldots,c: j \neq u} o_j^i}{o_u^i + \gamma}, \tag{23}$$

where the parameter $\gamma \geq 0$ in the second definition enables us to adjust the sensitivity of our measure to small changes of o_u^i depending on the data in question. We added this parameter in order to gain control over which category of outputs will be more important in determining the resulting nonconformity scores; by increasing γ one reduces the importance of o_u^i and consequently increases the importance of all other outputs.

4.3 The algorithm

We can now follow the general ICP algorithm detailed in Section 3.3 together with nonconformity measure (22) or (23) to produce the Neural Network ICP. More specifically, the exact steps the Neural Network ICP follows are:

- Split the training set into two smaller sets, the proper training set with $m := l - q$ examples and the calibration set with q examples; where q is a parameter of the algorithm.
- Use the proper training set to train the neural network.
- For each example $z_{m+t} = (x_{m+t}, y_{m+t}) : t = 1,\ldots,q$ in the calibration set,
  - supply the input pattern x_{m+t} to the trained network to obtain the output values $o_1^{m+t},\ldots,o_c^{m+t}$ and
  - calculate the nonconformity score α_{m+t} of the pair (x_{m+t}, y_{m+t}) by applying (22) or (23) to these values.
- For each test pattern $x_{l+g} : g = 1,\ldots,r$,
  - supply the input pattern x_{l+g} to the trained network to obtain the output values $o_1^{l+g},\ldots,o_c^{l+g}$,
  - consider each possible classification $Y_u : u = 1,\ldots,c$ and
    - compute the nonconformity score $\alpha_{l+g}^{(Y_u)}$ of the pair (x_{l+g}, Y_u) by applying (22) or (23) to the outputs of the network,
    - calculate the p-value $p(Y_u)$ for x_{l+g} being classified as Y_u by applying (5) to the nonconformity scores of the calibration examples and $\alpha_{l+g}^{(Y_u)}$:

$$p(Y_u) = \frac{\#\{i = m+1,\ldots,m+q,l+g : \alpha_i \geq \alpha_{l+g}^{(Y_u)}\}}{q+1}, \tag{24}$$

  - predict the classification with the smallest nonconformity score,
  - output as confidence to this prediction one minus the second largest p-value, and as credibility the p-value of the output prediction, i.e. the largest p-value.

5. Experimental results of the neural networks ICP

Here we detail the experimental results of the Neural Networks ICP on the Satellite, Shuttle and Segment data sets, which were used in the experiments of the Statlog Project (King et al., 1995); see also (Michie et al., 1994).

The Satellite data set consists of 6435 satellite images, split into 4435 training examples and 2000 test examples, described by 36 attributes. The classification task is to distinguish among 6 different soil conditions that are represented in the images. The calibration set was formed from 199 of the 4435 training examples.

The Shuttle data set consists of 43500 training examples and 14500 test examples with 9 attributes each, describing the conditions in a space shuttle. The classification task is to choose which one of the 7 different sets of actions should be taken according to the conditions. In this case we used 999 of the training examples to form the calibration set.

The Segment data set consists of 2310 outdoor images described by 18 attributes each. The classification task is to choose between: brick-face, sky, foliage, cement, window, path, grass. For our experiments on this data set we used 10 fold cross-validation, as this was the testing procedure followed in the Statlog project. The set was divided into 10 equally sized parts and our tests were repeated 10 times, each time using one of the 10 parts as the test set and the remaining 9 as the training set. Consequently, the resulting training and test sets consisted of 2079 and 231 examples respectively. Of the 2079 training examples 199 were used to form the calibration set.

Our experiments on these data sets were performed using 2 layer fully connected networks, with sigmoid hidden units and softmax output units. The number of their input and output units were determined by the format of each data set; equal to the number of attributes and possible classifications of the examples respectively. These networks were trained with the backpropagation algorithm minimizing a cross-entropy loss function. The number of hidden units and the learning and momentum rates used for each data set are reported in table 1. It is worth to note that the same parameters were used both for the ICP and its underlying algorithm.

Here we report the error percentages of the Neural Network ICP and compare them to the ones of its underlying algorithm as well as to those of some other traditional methods. In addition, we check the quality of its p-values by analysing the results obtained from its second mode, described in Section 3.5, for the 99%, 95% and 90% confidence levels. For the purpose of reporting the results of this mode we separate its outputs into three categories:

- A set with more than one labels
- A set with only one label
- The empty set

Our main concern here will be the number of outputs that belong to the first category; we want this number to be relatively small, since these are the examples for which the ICP is not certain in only one label at the required confidence level $1 - \delta$. In addition to the percentage of examples in each category, we also report the number of errors made by the ICP in this mode. This is the number of examples for which the true label was not included in the set output by the ICP; including all cases where the set output by the ICP was empty. Over many runs on different sets (both training and test) generated from the same i.i.d. distribution, the percentage of these errors will be close to the corresponding significance level δ; an experimental demonstration of this can be found in (Vovk, 2002). Finally, we

examine the computational efficiency of the method by comparing its processing times with those of its underlying algorithm.

	Satellite	Shuttle	Segment
Hidden Units	23	12	11
Hidden Learning Rate	0.002	0.002	0.002
Output Learning Rate	0.001	0.001	0.001
Momentum Rate	0.1	0	0.1

Table 1. The parameters used in our experiments for each data set.

Learning Algorithm	Percentage of error (%)		
	Satellite	Shuttle	Segment
Neural Networks ICP	10.40	0.0414	3.46
Backpropagation	10.24	0.0414	3.20
k-Nearest Neighbours	9.45	0.12	3.68
C4.5	15.00	0.10	4.00
CART	13.80	0.08	4.00
Naïve Bayes	28.70	4.50	26.50
CASTLE	19.40	3.80	11.20
Linear Discriminant	17.10	4.83	11.60

Table 2. Error rate comparison of the Neural Networks ICP with traditional algorithms.

In table 2 we compare the performance of the ICP on the three statlog project data sets with that of its underlying algorithm (we denote this as backpropagation) and that of 6 other traditional methods. These are the *k*-Nearest Neighbours algorithm, two decision tree algorithms, namely C4.5 and CART, the Naïve Bayes classifier, a Bayesian network algorithm called CASTLE and a linear discriminant algorithm. The results of the *k*-Nearest Neighbours algorithm were produced by the author, while those of all other methods were reported in (King et al., 1995) and (Michie et al., 1994). Note that the aim of the CP is not to outperform other algorithms but to produce more information with each prediction. So in comparing these error percentages we want to show that the accuracy of this method is not inferior to that of traditional algorithms.

We did not perform the same experiments with the corresponding original CP algorithm, due to the huge amount of time that would have been needed for doing so. However, its results in terms of error percentages would not have been significantly different from those of its underlying algorithm (backpropagation). So the first two rows of table 2 also serve as a performance comparison between ICP and TCP.

Table 2 clearly shows that the accuracy of the Neural Networks ICP is comparable to that of traditional methods. Of course, our main comparison here is with the performance of its underlying algorithm, since that is where ICPs base their predictions and since that is also the performance of the corresponding TCP. So by comparing its results to those of the backpropagation method, we can see that although in most cases the ICP suffers a small loss of accuracy, this loss is negligible. Moreover, we observe that as the data set gets bigger the difference between the error percentage of the ICP and that of its underlying algorithm

becomes smaller. In fact, for the shuttle data set, which is the biggest, the ICP gives exactly the same results with its underlying network.

Tables 3 to 5 detail the performance of the second mode of the ICP on each of the three data sets. Here we can see that the percentage of examples for which it needs to output more than one label is relatively small even for a confidence level as high as 99%, having in mind the difficulty of each task and the performance of its underlying algorithm on each data set. This reflects the quality of the p-values calculated by this method and consequently the usefulness of its confidence measures.

Nonconformity Measure	Confidence Level	Only one Label (%)	More than one label (%)	No Label (%)	Errors (%)
(4)	99%	60.72	39.28	0.00	1.11
	95%	84.42	15.58	0.00	4.67
	90%	96.16	3.02	0.82	9.59
(5)	99%	61.69	38.31	0.00	1.10
	95%	85.70	14.30	0.00	4.86
	90%	96.11	3.10	0.79	9.43

Table 3. Results of the second mode of the Neural Networks ICP for the Satellite data set.

Nonconformity Measure	Confidence Level	Only one Label (%)	More than one label (%)	No Label (%)	Errors (%)
(4)	99%	99.23	0.00	0.77	0.77
	95%	93.52	0.00	6.48	6.48
	90%	89.08	0.00	10.92	10.92
(5)	99%	99.30	0.00	0.70	0.70
	95%	93.86	0.00	6.14	6.14
	90%	88.72	0.00	11.28	11.28

Table 4. Results of the second mode of the Neural Networks ICP for the Shuttle data set.

Nonconformity Measure	Confidence Level	Only one Label (%)	More than one label (%)	No Label (%)	Errors (%)
(4)	99%	90.69	9.31	0.00	0.95
	95%	97.71	1.25	1.04	3.68
	90%	94.68	0.00	5.32	6.71
(5)	99%	91.73	8.27	0.00	1.04
	95%	97.79	1.21	1.00	3.55
	90%	94.76	0.00	5.24	6.67

Table 5. Results of the second mode of the Neural Networks ICP for the Segment data set.

Finally, table 6 lists the processing times of the Neural Network ICP together with those of its underlying algorithm. In the case of the Segment data set the times listed are for the total duration of the experiments on all 10 splits. As mentioned in the computational complexity comparison of Section 3.7, in most cases the ICP is faster than its underlying algorithm because it uses less training examples. In the case of Neural Networks, this reduction in training examples reduces slightly the training time per epoch and, for more or less the

same number of epochs, it results in a shorter total training time. This was the case for the Satellite and Shuttle data sets. However, for the Segment data set the number of epochs increased and this resulted in a slightly bigger total training time for the ICP.
Based on our computational complexity analysis of Section 3.7, if we were to perform the same experiments using the original CP method coupled with Neural Networks it would have taken approximately 183 days for the Satellite data set, 53 years for the Shuttle data set and 93 days for the Segment data set. This shows the huge computational efficiency improvement of ICP in the case of Neural Networks. In fact, it shows that ICP is the only conformal prediction method that can be used with this approach.

Learning Algorithm	Time (in seconds)		
	Satellite	Shuttle	Segment
Neural Networks ICP	1077	11418	5322
Backpropagation	1321	16569	4982

Table 6. The processing times of the Neural Networks ICP and its underlying algorithm.

6. Conclusion

This chapter presented the Inductive Conformal Prediction (ICP) approach for producing confidence measures with predictions and described its application to Neural Networks. ICPs accompany each of their predictions with probabilistically valid measures of confidence. Furthermore, they do not need the relatively large amount of processing time spend by Transductive Conformal Predictors (TCPs) to perform their computations. In fact their computational efficiency is virtually the same with that of their underlying algorithms.
The experimental results detailed in Section 5 and in (Papadopoulos et al., 2002a; Papadopoulos et al., 2002b; Papadopoulos et al., 2007) show that the accuracy of ICPs is comparable to that of traditional methods, while the confidence measures they produce are useful in practice. Of course, as a result of removing some examples from the training set to form the calibration set, they sometimes suffer a small, but usually negligible, loss of accuracy from their underlying algorithm. This is not the case, however, for large data sets, which contain enough training examples so that the removal of the calibration examples does not make any difference to the training of the algorithm.

7. Acknowledgements

This work was supported by the Cyprus Research Promotion Foundation through research contract PLHRO/0506/22 (“Development of New Conformal Prediction Methods with Applications in Medical Diagnosis”).

8. References

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

King, R.; Feng, C. & Sutherland, A. (1995). Statlog: Comparison of classification algorithms on large real-world problems, *Applied Artificial Intelligence*, Vol. 9, No. 3, pp. 259–287.

Martin-Löf, P. (1966). The definition of random sequences. *Information and Control*, Vol. 9, pp. 602–619.

Melluish, T.; Saunders, C.; Nouretdinov, I. & Vovk, V. (2001). Comparing the Bayes and Typicalness frameworks, *Proceedings of the 12th European Conference on Machine Learning (ECML'01)*, Vol. 2167 of *Lecture Notes in Computer Science*, pp. 360–371, Springer.

Michie, D.; Spiegelhalter, D. & Taylor, C. (Ed.) (1994). *Machine Learning, Neural and Statistical Classification*, Ellis Horwood.

Nouretdinov, I.; Vovk, V.; Vyugin, M. & Gammerman, A. (2001a). Pattern recognition and density estimation under the general iid assumption, *Proceedings of the 14th Annual Conference on Computational Learning Theory (COLT'01) and 5th European Conference on Computational Learning Theory (EuroCOLT'01)*, Vol. 2111 of *Lecture Notes in Computer Science*, pp. 337–353, Springer.

Nouretdinov, I.; Melluish, T. & Vovk. V. (2001b). Ridge regression confidence machine, *Proceedings of the 18th International Conference on Machine Learning (ICML'01)*, pp. 385–392, Morgan Kaufmann, San Francisco, CA.

Papadopoulos, H.; Proedrou, K.; Vovk, V. & Gammerman, A. (2002a). Inductive confidence machines for regression, *Proceedings of the 13th European Conference on Machine Learning (ECML'02)*, Vol. 2430 of *Lecture Notes in Computer Science*, pp. 345–356, Springer.

Papadopoulos, H.; Vovk, V. & Gammerman, A. (2002b). Qualified predictions for large data sets in the case of pattern recognition, *Proceedings of the 2002 International Conference on Machine Learning and Applications (ICMLA'02)*, pp. 159–163, CSREA Press.

Papadopoulos, H.; Vovk, V. & Gammerman, A. (2007). Conformal prediction with neural networks, *Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'07)*, Vol. 2, pp. 388-395, Patras, Greece, October 2007, IEEE Computer Society, Los Alamitos, CA.

Papadopoulos, H.; Gammerman, A. & Vovk, V. (2008). Normalized nonconformity measures for regression conformal prediction, Proceedings of the IASTED International Conference on Artificial Intelligence and Applications (AIA 2008), pp. 64-69, Innsbruck, Austria, February 2008, ACTA Press.

Proedrou, K.; Nouretdinov, I.; Vovk, V. & Gammerman, A. (2002). Transductive confidence machines for pattern recognition, *Proceedings of the 13th European Conference on Machine Learning (ECML'02)*, Vol. 2430 of *Lecture Notes in Computer Science*, pp. 381–390. Springer.

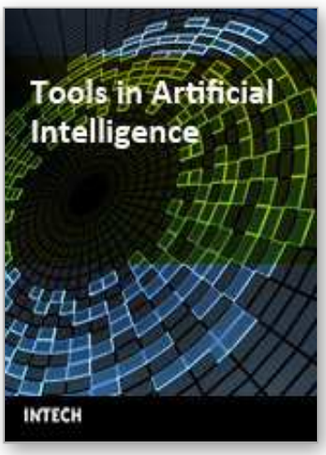
Saunders, C.; Gammerman, A. & Vovk, V. (1999). Transduction with confidence and credibility, *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, Vol. 2, pp. 722–726, Morgan Kaufmann, Los Altos, CA.

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

Vovk, V.; Gammerman, A. & Saunders, C. (1999). Machine learning applications of algorithmic randomness, *Proceedings of the 16th International Conference on Machine Learning (ICML'99)*, pp. 444–453, Morgan Kaufmann, San Francisco, CA.

Vovk, V. (2002). On-line confidence machines are well-calibrated, *Proceedings of the 43rd Annual Symposium on Foundations of Computer Science (FOCS'02)*, pp. 187–196, IEEE Computer Society, Los Alamitos, CA.

Vovk, V.; Gammerman, A. & Shafer, G. (2005). *Algorithmic Learning in a Random World*, Springer, New York.



Tools in Artificial Intelligence

Edited by Paula Fritzsche

ISBN 978-953-7619-03-9
Hard cover, 488 pages
Publisher InTech
Published online 01, August, 2008
Published in print edition August, 2008

This book offers in 27 chapters a collection of all the technical aspects of specifying, developing, and evaluating the theoretical underpinnings and applied mechanisms of AI tools. Topics covered include neural networks, fuzzy controls, decision trees, rule-based systems, data mining, genetic algorithm and agent systems, among many others. The goal of this book is to show some potential applications and give a partial picture of the current state-of-the-art of AI. Also, it is useful to inspire some future research ideas by identifying potential research directions. It is dedicated to students, researchers and practitioners in this area or in related fields.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen