Ann Math Artif Intell
DOI 10.1007/s10472-014-9420-z

Inductive Venn Prediction

Antonis Lambrou · Ilia Nouretdinov · Harris Papadopoulos

Abstract Venn Predictors (VPs) are machine learning algorithms that can provide well calibrated multiprobability outputs for their predictions. An important drawback of Venn Predictors is their computational inefficiency, especially in the case of large datasets. In this work, we investigate and propose Inductive Venn Predictors (IVPs), which can overcome the computational inefficiency problem of the original Transductive Venn Prediction framework. We develop an IVP algorithm and perform a detailed comparison of its time efficiency, accuracy, and quality of probabilistic outputs with those of the original Transductive Venn Predictor (TVP). The results demonstrate that our method provides well calibrated results while maintaining high accuracy. The IVP outperforms the original TVP method in terms of time efficiency, while also providing well-calibrated probabilistic estimates. Another observation is that the probability intervals of the IVP are tighter than those of the TVP.

Keywords Well calibrated probabilities · Large datasets · Inductive Venn Predictor · Machine learning

Mathematics Subject Classifications (2010) 62-07 · 68T05 · 68Q32

A. Lambrou (✉) · I. Nouretdinov
Computer Learning Research Centre, Computer Science Department, Royal Holloway, University of London, London, England, UK
e-mail: a.lambrou@cs.rhul.ac.uk

I. Nouretdinov
e-mail: I.Nouretdinov@cs.rhul.ac.uk

H. Papadopoulos
Computer Science and Engineering Department, Frederick University, Nicosia, Cyprus
e-mail: H.Papadopoulos@frederick.ac.cy

A. Lambrou · H. Papadopoulos
Frederick Research Center, Nicosia, Cyprus

Published online: 24 June 2014

1 Introduction

Machine Learning algorithms are used widely in several applications for classification or regression. Nevertheless, most algorithms do not provide probability estimates for their predictions. In [13], it is shown that algorithms that provide probabilistic outputs do not always guarantee that their probability estimates will be well-calibrated. Venn Prediction is a novel machine learning framework that can be combined with conventional classifiers for producing well calibrated multiprobability predictions under the assumption that the data in question are identically and independently distributed (i.i.d.). In particular, multiprobability predictions are a set of probability distributions for the true classification of a new example (with unknown classification). In effect this set defines lower and upper bounds for the conditional probability of the new example belonging to each one of the possible classes. These bounds are guaranteed (up to statistical fluctuations) to contain the corresponding true conditional probabilities. In [30], the Venn Prediction framework is described thoroughly and a proof of the validity of its probabilistic outputs is given.

In order to overcome the computational inefficiency problem of the original Transductive Venn Prediction (TVP) approach, which renders it not suitable for application to large datasets, we propose an Inductive Venn Predictor (IVP) based on the idea of Inductive Conformal Prediction (ICP). As it was shown in many studies, see e.g. [14, 19, 21], ICPs are as computationally efficient as the conventional algorithms they are based on. This paper extends the work presented in [13] were the Inductive Venn Prediction approach was proposed, by performing a thorough comparison between the proposed IVP and its TVP counterpart. This comparison examines the validity and effectiveness of the probabilistic outputs produced by each method as well as their accuracy and computational efficiency. Additionally, we examine the effect that the size of the dataset has on the difference between the results of the IVP and the TVP.

Our experiments were performed on four classification datasets, the Car Evaluation [2], the Wine Quality [3], the Spambase, and the MiniBooNE [8] datasets, which are all freely available at the University of California, Irvine (UCI) machine learning repository [8]. The obtained results show that the IVP gives well-calibrated probabilities under the i.i.d. assumption. Moreover, the proposed method surpasses the TVP in terms of time efficiency while retaining similar accuracy. In terms of the probability intervals produced by the IVP and TVP approaches, our results show that the intervals produced by the IVP are tighter than those of the TVP, while still being well calibrated.

The rest of the paper is structured as follows. In Section 2, we provide related work. In Section 3, we describe the Venn Prediction framework, and propose the Inductive version of the framework. In Section 4, we detail our experimental settings and the obtained results. Finally, in Section 5, we give our conclusions and future plans.

2 Related work

There are a number of methods for providing probabilistic outputs. In Section 2.1 we describe three popular methods found in the literature: Binning, Isotonic Regression, and Platt's method. In particular, these three methods have been implemented using Support Vector Machines (SVMs), and convert the unthresholded output $f(x_i)$ of the SVM decision rule into a probability estimate. Hereon, $f(x_i)$ will also be referred as the SVM score of an example x_i. In Section 2.2 we outline background work on Venn Predictors.

2.1 Other methods that provide probabilistic outputs

The binning method [7] sorts the training examples according to their SVM scores, and then divides them into b equal sized sets, or bins, each having an upper and lower bound. Given a test example x_i, it is placed in a bin according to its classifier score. The corresponding probability $P(Y_j = 1|x_i)$ is the fraction of positive training examples that fall within that bin. There is no imposed lower or upper bound on SVM scores. Therefore, when using this method it is possible for some scores from the test examples to fall below or above the low and high scores, respectively, of the training examples. If this happens the corresponding probability $P(Y_j = 1|x_i)$ is that of the nearest bin to the score of x_i.

Isotonic regression has been used in order to map the SVM scores into probability estimates in [31]. An isotonic function $g(i)$ has a monotonically increasing trend, which means that for all i, j:

$$i > j \implies g(i) > g(j) \; and \; i < j \implies g(i) < g(j). \tag{1}$$

If the scores of the SVM are ranked correctly, we can assume that the probability $P(Y_j = 1|x_i)$ will be increasing as the SVM scores increase. Therefore, we can use isotonic regression to map SVM scores into probability estimates. The most common algorithm used for isotonic regression is the Pair-Adjacent-Violators (PAV) algorithm. The algorithm learns the probability estimate $g(x_i)$ for each ranked example x_i. First, we set $g(x_i) = 1$ if x_i is a positive example, and $g(x_i) = 0$ otherwise. If g is already isotonic the function has been learned. Otherwise, there must be an example where $g(x_{i-1}) > g(x_i)$. The two examples x_{i-1} and x_i are called pair-adjacent violators, because they violate the isotonic assumption. The values of $g(x_{i-1})$ and $g(x_i)$ are then replaced by their average, so that their values no longer violate the isotonic assumption. This process is repeated until an isotonic set of values is obtained. In the end, we have a list of probability estimates together with the adjacent SVM scores of the training examples. When a new example arrives, we assign the mapped probability estimate based on the score that x_i has obtained from the SVM decision rule. Normally, there will be intervals of scores with the same probability estimates. Since there are no imposed boundaries on the SVM scores, the lowest interval begins from $-\infty$ and the highest interval ends at $+\infty$.

Platt introduced a method in [24] to estimate posterior probabilities based on the decision function f by fitting a sigmoid:

$$P(Y_j = 1|f(x_i)) = \frac{1}{1 + \exp(Af(x_i) + B)}, \tag{2}$$

where $Y_j \in \{-1, 1\}$. The best parameters A and B are determined so that they minimise the negative log-likelihood of the training data. Platt uses a Levenberg-Marquardt (LM) optimisation algorithm to solve this. As indicated in [24], any method for optimisation can be used.

2.2 Venn Prediction

The Venn Prediction (VP) framework is based on the Conformal Prediction (CP) framework. CP is a novel technique for obtaining reliable confidence measures. The technique was proposed in [10] and later improved in [27] and [29]. CPs are built using classical machine learning algorithms, called underlying algorithms. CPs complement the predictions of the underlying algorithms with measures of confidence. Many CPs have been built to date, based on various algorithms such as Support Vector Machines [27], k-Nearest Neighbours

for classification [25] and for regression [20], Random Forests [6], and Genetic Algorithms [11]. The computational efficiency of CPs has also been greatly improved using Inductive Conformal Prediction (ICP) [14], as demonstrated in applications to Ridge Regression [19], k-Nearest Neighbours [21], and more recently in applications to Neural Networks [22]. The CP framework has been successfully applied to medical problems, such as evaluation of the risk of stroke [12], breast cancer diagnosis [9], classification of leukaemia subtypes [1], and acute abdominal pain diagnosis [17]. Additionally, CPs have been applied to other problems such as Software Effort Estimation in [18].

Venn Prediction has been introduced in [28] where the interested reader can find a detailed description of the framework. Since then, VPs have been developed based on k-Nearest Neighbours [5], Nearest Centroid [4] and Neural Networks [15, 16]. Furthermore, VPs based on SVMs have been developed in [13, 32], and have been compared with Platt's method [24], Binning [7] and Isotonic Regression [31]. As it is shown in [13], the latter three methods do not guarantee that the probabilistic outputs will be well-calibrated.

3 Venn Prediction

In this section, we describe the Venn Prediction framework. Typically, we have a training set[1] of the form $\{z_1, \ldots, z_l\}$, where each $z_i \in Z$ is a pair (x_i, y_i) consisting of the object x_i and its classification y_i. For a new object x_{l+1} we intend to estimate the probability of $y_{l+1} = Y_j$ for all possible classifications $Y_j \in \{Y_1, \ldots, Y_c\}$. The main idea behind Venn prediction is to divide all examples into a number of categories and calculate the probability of x_{l+1} belonging to each class $Y_j \in \{Y_1, \ldots, Y_c\}$ as the frequency of Y_j in the category that contains it. However, as we don't know the true class of x_{l+1}, we assign each one of the possible classes to it in turn and for each assigned classification Y_k we calculate an empirical probability distribution for the true class of x_{l+1} based on the examples

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

The Venn Prediction framework assigns each one of the possible classifications Y_j to x_{l+1} and divides all examples $\{(x_1, y_1), \ldots, (x_{l+1}, Y_j)\}$ into a number of categories based on what is called a *Venn taxonomy*. For $n \in \mathbb{N}$, an n-taxonomy is a measurable function $K : Z^n \times Z \to \mathbf{K}$, where $\mathbf{K}$ is a measurable space, that is equivariant with respect to permutations in the sense of

$$i = \pi(i) \implies K((z_1, \ldots, z_n), z_i) = K((z_{\pi(1)}, \ldots, z_{\pi(n)}), z_{\pi(i)}), \tag{4}$$

for all $i = 1, \ldots, n$ and any permutation π of $(1, \ldots, n)$. The set $\mathbf{K}$ is usually finite; we will refer to its elements as categories. Every taxonomy defines a different VP. Typically each taxonomy is based on a traditional machine learning algorithm, called the *underlying algorithm* of the Venn predictor. The output of this algorithm for each attribute vector $x_i, i = 1, \ldots, l + 1$ after being trained on the set (3), is used to assign (x_i, y_i) to one of a predefined set of categories $\kappa_i \in \mathbf{K}$. For example, a Venn taxonomy that can be used with every traditional algorithm puts in the same category all examples that are assigned the same classification by the underlying algorithm. In Section 3.2, we define a taxonomy based on the output of the Support Vector Machine (SVM) classifier.

[1] The training set is in fact a multiset, as it can contain some examples more than once.

After assigning the category $\kappa_i^{Y_j} = K((z_1, \ldots, z_l, (x_{l+1}, Y_j)), z_i)$ to each example in the extended set (3), the empirical probability of each classification Y_k in $\kappa_{l+1}^{Y_j}$ will be

$$p^{Y_j}(Y_k) = \frac{\left|\{i = 1, \ldots, l+1 | \kappa_i^{Y_j} = \kappa_{l+1}^{Y_j} \,\&\, y_i = Y_k\}\right|}{\left|\{i = 1, \ldots, l+1 | \kappa_i^{Y_j} = \kappa_{l+1}^{Y_j}\}\right|} \tag{5}$$

This is an empirical probability distribution for the true class of x_{l+1}. So after assigning all possible classifications to x_{l+1} we get a set of probability distributions $P_{l+1} = \{p^{Y_j} : Y_j \in \{Y_1, \ldots, Y_c\}\}$ that compose the multi-probability prediction of the VP. As proved in [30] the predictions produced by any Venn predictor are automatically valid multiprobability predictions. This is true regardless of the taxonomy of the Venn predictor. Of course the taxonomy used is still very important as it determines how efficient, or informative, the resulting predictions are. We want the diameter of multiprobability predictions and therefore their uncertainty to be small and we also want the predictions to be as close as possible to zero or one.

The maximum and minimum probabilities obtained for each label Y_k amongst all distributions $\{p^{Y_j} : Y_j \in \{Y_1, \ldots, Y_c\}\}$, define the interval for the probability of the new example belonging to Y_k. We denote these probabilities as $U(Y_k)$ and $L(Y_k)$, respectively. The VP outputs the prediction $\hat{y}_{l+1} = Y_{k_{best}}$, where

$$k_{best} = \arg\max_{k=1,\ldots,c} \overline{p(k)}, \tag{6}$$

and $\overline{p(k)}$ is the mean of the probabilities obtained for label Y_k amongst all probability distributions. The probability interval for this prediction is $[L(Y_k), U(Y_k)]$.

3.1 Inductive Venn Prediction

Here, we describe the proposed Inductive Venn Predictor (IVP) method. The transductive nature of the original Venn Prediction framework is computationally inefficient, since it requires training the underlying algorithm for every possible class of each new test example. To address this problem we follow the idea of the Inductive Conformal Prediction, and propose an efficient Inductive Venn Predictor (IVP). Our approach splits the available training examples into two parts, the proper training set with q examples and the calibration set with the remaining $r = l - q$ examples. We then use the proper training set to train the underlying algorithm and the calibration set to calculate the set of probability distributions for each new example. The main advantage of the IVP method is that the underlying algorithm is trained only once on the training set, and the probability distributions are calculated from the calibration set for every class of the test example. There is no more the requirement to re-train the algorithm for every possible class of the test example. The original taxonomy function K is transformed to another taxonomy $K' : Z^{r+1} \times Z \to \mathbf{K}$ such that

$$\begin{aligned} &K'_{r+1}((z_{q+1}, \ldots, z_{l+1}), z_i) = \\ &K_q((z_1, \ldots, z_q), z_i), i = q+1, \ldots, l+1. \end{aligned} \tag{7}$$

In this definition we assume that the proper training set $\{z_1, \ldots, z_q\}$ is a fixed part of K' and therefore K' is a valid Venn taxonomy.

After assigning the category $\kappa_i^{Y_j} = K'((z_{q+1}, \ldots, z_l, (x_{l+1}, Y_j)), z_i)$ to each example in the calibration set $i = q+1, \ldots, l+1$, the empirical probability of each classification Y_k in $\kappa_{l+1}^{Y_j}$ will be

$$p^{Y_j}(Y_k) = \frac{\left|\{i = q+1, \ldots, l+1 | \kappa_i^{Y_j} = \kappa_{l+1}^{Y_j} \,\&\, y_i = Y_k\}\right|}{\left|\{i = q+1, \ldots, l+1 | \kappa_i^{Y_j} = \kappa_{l+1}^{Y_j}\}\right|} \tag{8}$$

Online mode In the online mode there is no fixed training set. On each step of the algorithm, a new example is predicted and then it is added to the calibration set. Therefore, as the IVP makes predictions, new examples are considered for calibrating the probabilistic outputs. After a number of m predictions, we remove $m - s$ of the examples from the calibration set and we add them to the training set (where s is chosen such that both the training set and calibration set grow with equal rate on each update step). The algorithm is then re-trained on the training set and proceeds on predicting new examples. In contrast with the TVP, the IVP is re-trained only once every m steps, while the TVP is retrained on every step of the algorithm for every possible class of the new example. In the online mode, we are able to test the probabilistic outputs of the algorithm and examine whether the actual accuracy falls near the probability estimates. We must mention here that the validity of the IVP method is not entirely retained, since the training set and therefore a part of the Venn taxonomy changes every m steps. Nevertheless, as we will show in our experiments, validity is not affected in practice. There would have been provable validity had there been no retraining. Within each update interval the results are valid.

3.2 Taxonomy

We define a taxonomy based on the output o_i (given test example x_i) of the multiclass underlying algorithm. As explained in Section 3, the validity of a Transductive VP is guaranteed under the i.i.d. assumption, regardless of the taxonomy used. For instance, a taxonomy that puts all examples in one single category would still give a valid predictor. Nevertheless, the performance of each VP is highly affected by the information provided from the categories defined in a taxonomy.

In this work, our taxonomy is simply based on the classification output o_i of a conventional classification algorithm. Therefore, $\kappa_i^{Y_j} = f(x_i)$, where $f(x_i)$ is the classification output of the underlying algorithm after being trained on $z_1, \ldots, z_q$ where each $z_i = (x_i, y_i)$. This taxonomy will give c categories. If the classifier is performing well, then each category should contain sufficient information for the VP to perform well in terms of accuracy. The IVP algorithm is presented in Algorithm 1. In our implementation, we have used the SVM classifier with Sequential Minimal Optimisation (SMO) as our underlying algorithm [23].

3.3 Time efficiency

The nature of the Transductive Venn Predictor algorithm makes it inefficient in the case of large datasets. The algorithm has a training phase (learning phase) for every new example and every possible class of the example. This time inefficiency problem is removed from the Inductive Venn Predictor algorithm, because of the use of the calibration set. The training phase of the algorithm needs to be performed only once, and then for every new example the

Algorithm 1 IVP algorithm

Input: proper training set $\{(x_1, y_1), \ldots, (x_q, y_q)\}$, calibration set $\{(x_{q+1}, y_{q+1}), \ldots, (x_l, y_l)\}$, new example x_{l+1}, possible classes $\{Y_1, \ldots, Y_c\}$.

Train the multiclass underlying algorithm on the proper training set $\{(x_1, y_1), \ldots, (x_q, y_q)\}$;

Supply the input patterns $x_{q+1}, \ldots, x_{l+1}$ to the trained underlying algorithm to obtain the outputs $o_{q+1}, \ldots, o_{l+1}$;

for $i = q + 1$ **to** $l + 1$ **do**

Assign κ_i to (x_i, y_i) according to the underlying algorithm classification output o_i;

end

for $j = 1$ **to** c **do**

Assume classification Y_j for x_{l+1}.

for $k = 1$ **to** c **do**

$$p^{Y_j}(Y_k) = \frac{\left|\left\{i=1,\ldots,n+1 | \kappa_i^{Y_j} = \kappa_{l+1}^{Y_j} \ \&\ y_i \in Y_k\right\}\right|}{\left|\left\{i=1,\ldots,l+1 | \kappa_i^{Y_j} = \kappa_{l+1}^{Y_j}\right\}\right|};$$

end

end

for $k = 1$ **to** c **do**

$$\overline{p(Y_k)} := \frac{1}{c}\sum_{j=1}^{c} p^{Y_j}(Y_k);$$

end

Output:

Prediction $k_{best} = \arg\max_{k=1,\ldots,c} \overline{p(Y_k)}$;

$\hat{Y} = k_{best}$;

The probability interval for $\hat{Y}$: $\left[\min_{k=1,\ldots,c} p^{Y_k}(\hat{Y}), \max_{k=1,\ldots,c} p^{Y_k}(\hat{Y})\right]$.

calibration set is being used to calculate the probabilistic outputs. This modification of the algorithm not only removes computationally expensive calculations, but also maintains the property that the probabilistic outputs will be well-calibrated, under the i.i.d. assumption. The time efficiency of the IVP becomes more important when the underlying algorithm is expensive in terms of time efficiency. For example if the underlying algorithm requires $O(n)$ time, the TVP method will reuse the underlying algorithm $n * c$ times, which makes the time requirement of the TVP to $O(n * n * c) = O(n^2)$ for small numbers of c. The IVP method, will only use the algorithm once, thus leaving the time requirements to $O(n)$. As we will see in our experimental results in Section 4, the difference between TVP and IVP has great impact in practice.

4 Experiments and results

We have conducted experiments with the proposed IVP algorithm in order to compare the results with its transductive counterpart. In the following subsections, we describe the

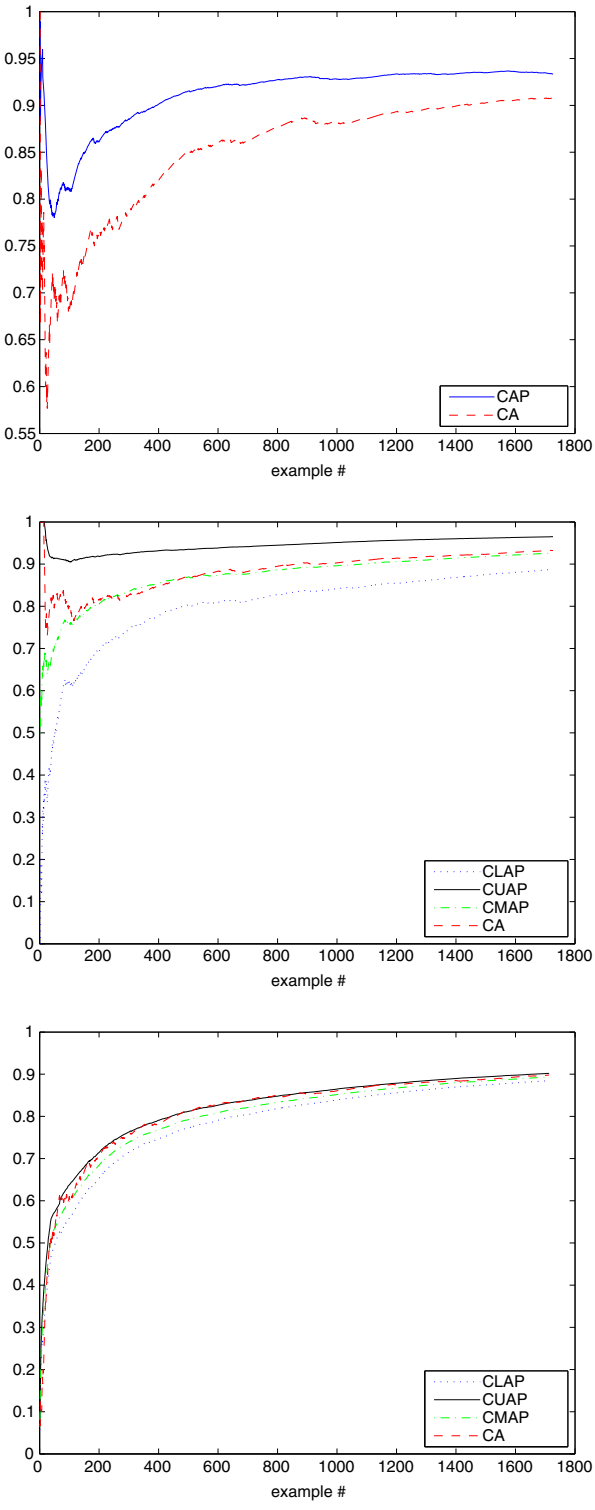

Fig. 1 Online experiments with SVM-LR (1st), TVP (2nd), and IVP (3rd) on the Car evaluation dataset

datasets used, the online mode experiments, and the offline mode (10-fold cross validation) experiments.

4.1 Datasets

Car evaluation dataset The Car Evaluation dataset was derived from hierarchical decision model [2] and is available at [8]. The dataset contains 1728 examples with 6 features for each example. There are 4 classes for this dataset which describe the car acceptability based on features that represent the price, technology, and comfort of a car.

Red Wine quality dataset The Red Wine quality dataset contains 1599 examples of physiochemical features of red variants of the "Vinho Verde" wine [3]. Each example has a quality score from 1 to 10. In this work, we have used the scores as 10 different classes from 1 to 10. This dataset is particularly difficult and requires some pre-processing to remove redundant features, or even reduce the number of classes. In our experiments, we have intentionally left the dataset to its original state in order to demonstrate the reliability of our probability estimates on difficult problems. The Red Wine quality dataset was used in the online experiments for its complexity. Nevertheless, it was not used in the offline experiments, since the large number of classes was prohibitive (in terms of time efficiency) for the evaluation of the TVP method.

Spambase dataset The Spambase dataset which is available at [8], contains 4601 examples of email messages. There are 57 attributes which describe the content of each email. The emails can be classified into two classes: spam or non-spam.

MiniBooNE dataset The MiniBooNE particle identification dataset (Booster Neutrino Experiment) [8] contains 130065 instances of electron neutrinos and muon neutrinos. Each instance contains 50 real valued attributes which describe signal events. This dataset was used only with the IVP online method, in order to demonstrate its ability to handle large datasets.

4.2 Online experiments

In order to show the validity of the probability estimates, we conduct experiments in the on-line mode. Initially all examples are test examples and they are added to the training set one by one after a prediction for each one is made. We calculate the cumulative average accuracy of the predictors, and the cumulative average probability. The cumulative average accuracy is calculated as the total accuracy of all tested examples, divided by the total number of tested examples. In the same way we calculate the cumulative average probability. If the methods provide well calibrated probability estimates, the cumulative average accuracy should be near the cumulative average probability. We compare the online results of three algorithms, namely SVM with Logistic Regression (SVM-LR) which is Platt's

Table 1 Comparison of online results on the Car evaluation dataset

Algorithm	Accuracy	Duration (hh:mm:ss)	Probabilities
SVM-LR	90.63 %	00:24:03	93.21 %
TVP	93.11 %	00:52:38	88.61 %–96.39 %
IVP	89.70 %	00:03:37	88.36 %–90.10 %

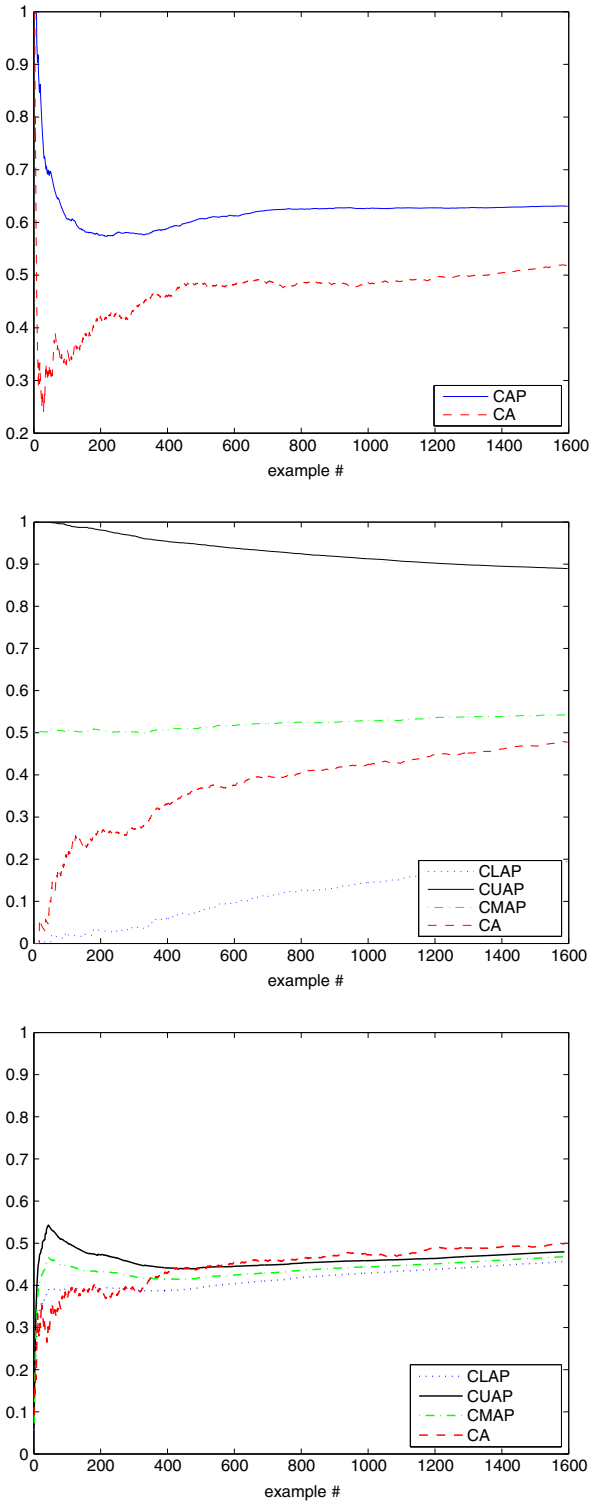

Fig. 2 Online experiments with SVM-LR (1st), TVP (2nd), and IVP (3rd) on the Wine Quality dataset

method as described in Section 2.1, SVM Transductive Venn Predictor (TVP) and SVM Inductive Venn Predictor (IVP). For the IVP, we chose the number $s = 12$, such that 40 % of the observed examples are assigned to the calibration set on each update. SVM-LR has been added in the comparison in order to compare the validity of its probabilistic outputs.

For the VPs, we graph the Cumulative Lower Accuracy Probability (CLAP), the Cumulative Upper Accuracy Probability (CUAP), and the Cumulative Accuracy (CA) curves:

$$CLAP(t) = \frac{1}{t}\sum_{i=1}^{t} U_i(Y_{k_{best}}), \tag{9}$$

$$CUAP(t) = \frac{1}{t}\sum_{i=1}^{t} L_i(Y_{k_{best}}), \tag{10}$$

$$CA(t) = \frac{1}{t}\sum_{i=1}^{t} Acc_i, \tag{11}$$

where t is the number of test examples that have been added to the training set, and $Acc_i = 1$ when the prediction for example x_i is correct and 0 otherwise. We also plot the Cumulative Mean Accuracy Probability (CMAP) curve, which is the mean of the CLAP and CUAP curves. Since VPs provide well calibrated probabilistic outputs, it is expected that the CA curve will fall within or near the CLAP and CUAP curves. For classical probabilistic predictors (the SVM-LR algorithm) we plot the CA and the Cumulative Accuracy Probability (CAP) curves. The CAP curve is similarly calculated as the CA curve:

$$CAP(t) = \frac{1}{t}\sum_{i=1}^{t} \max_{j=1}^{c} f(x_{ij}), \tag{12}$$

where $f(x_{ij})$ is the probability estimate given for a prediction. Here, the CA curve should fall near the CAP curve, if the SVM-LR algorithm provides well-calibrated probabilities.

In Fig. 1 we show the plots of the three algorithms on the Car Evaluation dataset. The plot for the SVM-LR algorithm (top) contains the CA and CAP curves, while the rest of the plots contain the CLAP, CUAP, CMAP, and CA curves. From the results we see that the IVP provides the tightest probability estimates. In Table 1 we show the final results of the three algorithms on the Car Evaluation dataset. For the SVM-LR algorithm there is about 3 % difference for the estimated probability and accuracy, while the TVP provides well calibrated results with an interval of about 8 %. The IVP provides well calibrated results with a much better interval of about 2–3 %. The accuracy of the IVP remains at the same level as with the SVM-LR method, although the TVP performs better in terms of accuracy with a final 93.11 %. The IVP accuracy is expected to be lower than the TVP accuracy, since there is a number of examples removed from the training set and are used as the calibration set.

Table 2 Comparison of online results on the Wine Quality dataset

Algorithm	Accuracy	Duration (hh:mm:ss)	Probabilities
SVM-LR	51.84 %	01:26:42	62.99 %
TVP	47.78 %	10:04:10	19.34 %–88.85 %
IVP	48.59 %	00:56:26	48.81 %–50.11 %

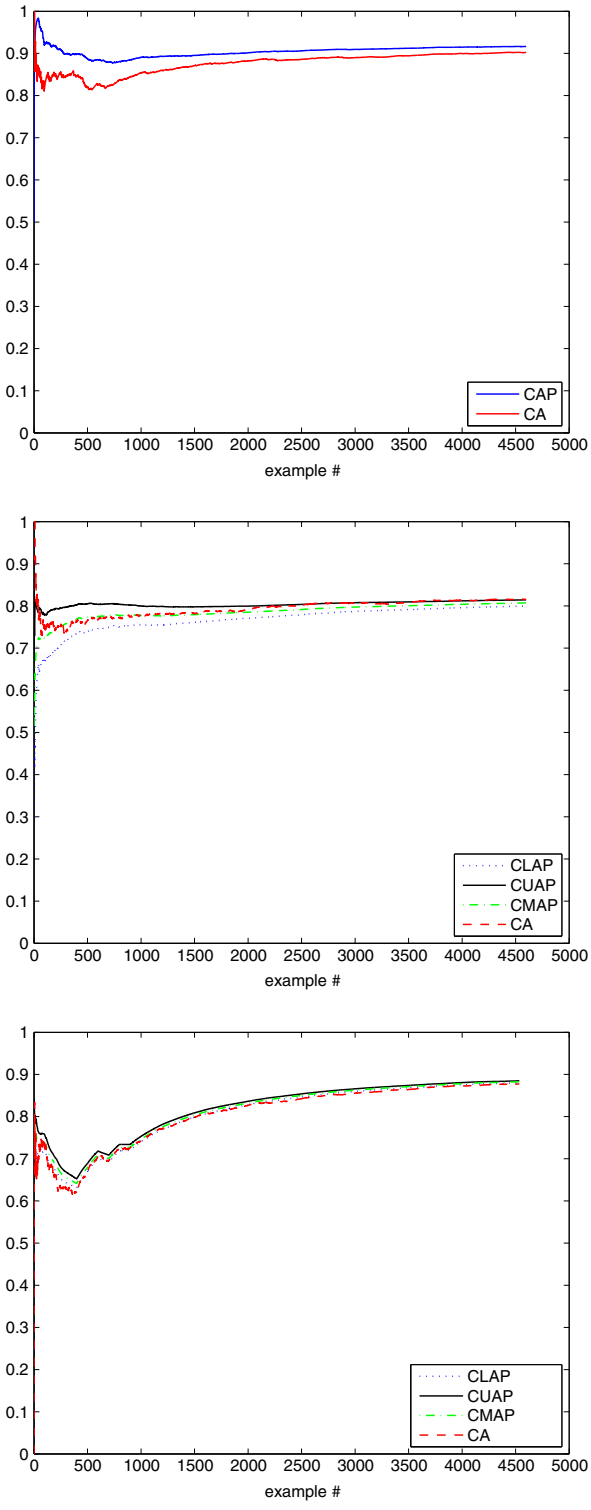

Fig. 3 Online experiments with SVM-LR (1st), TVP (2nd), and IVP (3rd) on the Spambase dataset

Table 3 Comparison of online results on the Spambase dataset

Algorithm	Accuracy	Duration (hh:mm:ss)	Probabilities
SVM-LR	87.97 %	04:22:43	89.99 %
TVP	81.54 %	07:53:20	80.00 %–81.44 %
IVP	86.50 %	00:26:57	86.75 %–87.22 %

The great advantage of the IVP method is the time efficiency, which is compared with the rest of the methods in Table 1. As we can see the SVM-LR algorithm required about 24 min and the TVP 52 min to finish the experiment, while the IVP required only 3 min to finish. The IVP method is much faster since the training of the underlying algorithm is required once every $m = 20$ steps in our experiments. The SVM-LR, and TVP use the training set at each update, while the IVP uses the calibration set at each update. Moreover, the training set of the IVP method is slightly smaller, since there is a percentage used for the calibration set.

Figure 2 shows the online results of the three algorithms on the Wine Quality dataset. Here, we highlight another advantage of the IVP over the rest of the methods. The probability estimates of the IVP are well calibrated and effective. The lower and upper probability interval is very tight in the case of the IVP, while the TVP provides very wide probability estimates. The SVM-LR method provides misleading probability estimates, since there is a discrepancy of almost 10 % between the average probability and average accuracy. It is surprising how the IVP provides such tight probabilistic outputs, even when the TVP does not perform so well. A possible explanation for this result, is that the IVP method calculates the probabilities using only the calibration set and the underlying algorithm is trained only once. The only thing that changes during each test is the assumed label of the test example. The change of the assumed label does not affect the outputs of the algorithm on the training set, and the examples in the calibration set remain in the same category. Therefore, we should not expect a lot of difference in the probabilities calculated. In contrast, the TVP

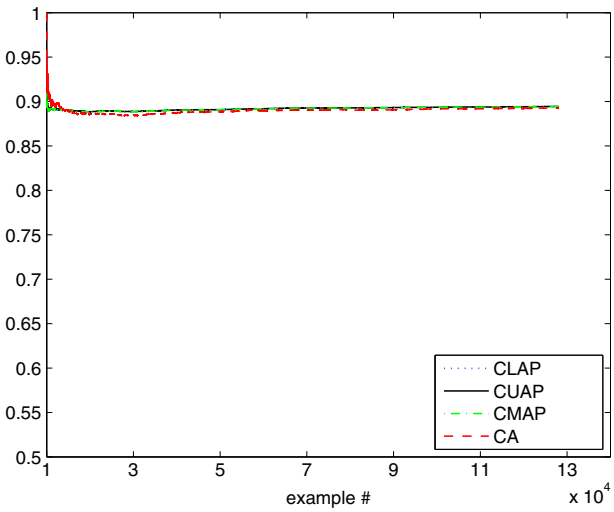

Fig. 4 Online experiment with IVP on the MiniBooNE dataset

method re-trains the training set for every assumed label of the test example, and therefore the categorization for calculating the probabilities might change drastically.

In Table 2 we show the final results and durations of all algorithms. Again, the IVP is faster compared with SVM-LR and TVP. Similar results are provided by our algorithm on the Spambase dataset shown in Fig. 3 and Table 3. In the case of the Spambase dataset, the IVP algorithm provides better accuracy than its transductive counterpart. Since the dataset is larger here, we notice that when using the calibration set for estimating probabilities, we actually get more accurate results. The number of steps before a training update for the IVP on the Spambase dataset was chosen to be $m = 100$.

We have additionally performed an experiment on the IVP method on a larger scale. We have used the MiniBooNE particle identification dataset [8], which contains 130065 particle instances. The dataset was not possible to be tested with the TVP method due to the time inefficiency problem of the method. Since in this experiment we do not compare the IVP with other methods, we have used the C4.5 decision tree classifier [26] as the underlying algorithm of the IVP, which runs faster. As it is shown in Fig. 4, the IVP method has provided well-calibrated and accurate results. The number of steps before each training update was chosen to $m = 10000$ which allowed us to overcome the time inefficiency problem.

4.3 Offline experiments

We have performed 10-fold cross validation experiments with the Car evaluation and Spambase datasets in order to evaluate the results of the IVP method and compare it with the TVP. Our intention here is to compare the probabilistic intervals that the two algorithms give, based on the training set size. Therefore, the SVM-LR algorithm has not been used

Table 4 IVP 10-fold cross validation results on the Car evaluation dataset

Total # of examples	Accuracy	Lower Prob.	Upper Prob.	BS
100	0.8110	0.8028	0.8898	0.2805
200	0.8710	0.8552	0.9104	0.2055
300	0.8787	0.8626	0.9058	0.1964
400	0.8980	0.8891	0.9199	0.1595
500	0.9088	0.9043	0.9280	0.1524
600	0.9152	0.9043	0.9257	0.1420
700	0.9230	0.9120	0.9309	0.1361
800	0.9365	0.9227	0.9390	0.1145
900	0.9366	0.9350	0.9498	0.1136
1000	0.9452	0.9384	0.9515	0.1002
1100	0.9476	0.9397	0.9518	0.0962
1200	0.9366	0.9350	0.9498	0.1136
1300	0.9538	0.9439	0.9546	0.0867
1400	0.9555	0.9461	0.9559	0.0824
1500	0.9561	0.9495	0.9586	0.0825
1600	0.9591	0.9533	0.9617	0.0775
1728	0.9637	0.9569	0.9648	0.0699

here. Moreover, we have not used the Wine Quality dataset in the offline experiments, due to the large number of classes of the dataset, which made the TVP method prohibitively slow. The Spambase dataset has only two classes, which allowed us to evaluate the TVP and IVP methods together in the offline mode. For the IVP method, we have used 30 % of the training set as calibration set. We compare the two methods using different sizes of the datesets, in order to evaluate which method performs better on various data sizes. Since the IVP uses a percentage from the training set as the calibration set, we expect the IVP to give lower accuracy when the dataset is small, and as the data size increases, we expect the accuracy of the IVP to match the accuracy of the TVP.

In Table 4 and Fig. 5 (top), we show the average results of the IVP on the Car Evaluation dataset. In each row of the table we show the results with a different number of examples in

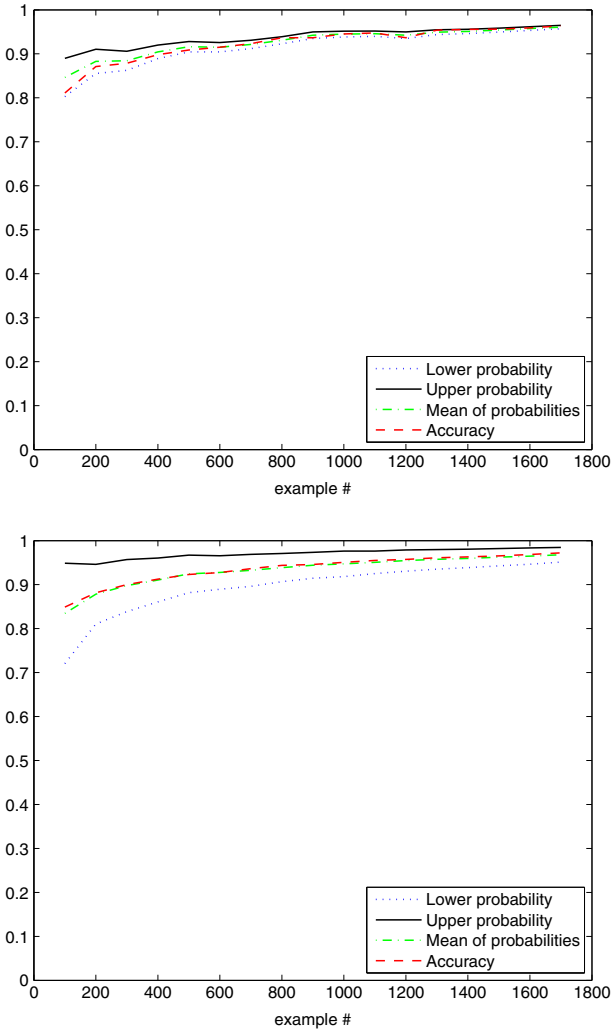

Fig. 5 IVP (*top*) and TVP (*bottom*) 10-fold cross validation results on the Car evaluation dataset

Table 5 TVP 10-fold cross validation results on the Car evaluation dataset

Total # of examples	Accuracy	Lower Prob.	Upper Prob.	BS
100	0.8490	0.7205	0.9488	0.2243
200	0.8815	0.8104	0.9462	0.1743
300	0.9000	0.8387	0.9570	0.1500
400	0.9125	0.8607	0.9604	0.1327
500	0.9230	0.8818	0.9671	0.1139
600	0.9273	0.8895	0.9657	0.1103
700	0.9362	0.8961	0.9689	0.0989
800	0.9437	0.9067	0.9708	0.0887
900	0.9457	0.9146	0.9735	0.0846
1000	0.9506	0.9181	0.9762	0.0759
1100	0.9550	0.9252	0.9764	0.0724
1200	0.9574	0.9306	0.9789	0.0671
1300	0.9610	0.9353	0.9798	0.0629
1400	0.9632	0.9385	0.9807	0.0596
1500	0.9653	0.9427	0.9821	0.0571
1600	0.9686	0.9464	0.9835	0.0526
1728	0.9723	0.9512	0.9845	0.0468

the training set. The results in each row are the averages of ten 10-fold cross validation runs. We have started the experiments with only 100 examples and increased the number of examples in each experiment. As expected, the IVP always provides well calibrated probabilistic

Table 6 IVP 10-fold cross validation results on the Spambase dataset

Total # of examples	Accuracy	Lower Prob.	Upper Prob.	BS
100	0.7170	0.6586	0.7162	0.3867
200	0.7345	0.6984	0.7289	0.3667
300	0.7680	0.7520	0.7742	0.3249
400	0.7960	0.7748	0.7910	0.3018
500	0.8158	0.8018	0.8149	0.2822
600	0.8302	0.8275	0.8384	0.2554
700	0.8483	0.8520	0.8614	0.2327
800	0.8557	0.8478	0.8560	0.2310
900	0.8552	0.8448	0.8521	0.2315
1000	0.8655	0.8546	0.8612	0.2162
1500	0.8947	0.8852	0.8896	0.1864
2000	0.9041	0.8956	0.8989	0.1728
2500	0.9072	0.9025	0.9052	0.1675
3000	0.9097	0.9056	0.9078	0.1631
3500	0.9129	0.9108	0.9127	0.1579
4000	0.9177	0.9175	0.9191	0.1511
4601	0.9216	0.9222	0.9236	0.1455

outputs, regardless of the number of examples in the training set. Nevertheless, as the training set grows, the accuracy increases and the interval of the probabilities becomes tighter. We can also see clearly that the accuracy always falls within the probability estimates. We have also calculated the Brier Score (BS) in each experiment, which indicates the quality of the probability estimates. The BS is calculated as

$$BS = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{c}(f(x_{ij}) - o_{ij})^2, \tag{13}$$

where $f(x_{ij})$ is the mean of the probabilities obtained for class j. The value o_{ij} is set to 1 if example x_i belongs to class j, and 0 otherwise. The constant c is the number of classes

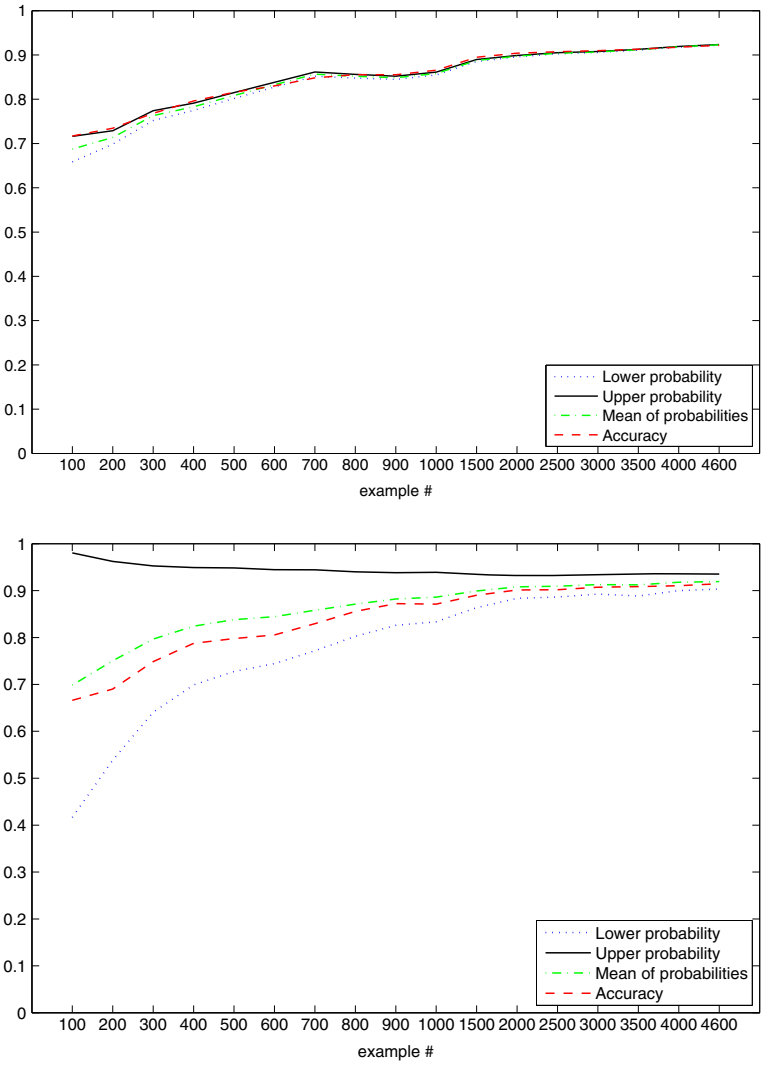

Fig. 6 IVP (*top*) and TVP (*bottom*) 10-fold cross validation results on the Spambase dataset

Table 7 TVP 10-fold cross validation results on the Spambase dataset

Total # of examples	Accuracy	Lower Prob.	Upper Prob.	BS
100	0.6660	0.4164	0.9803	0.3820
200	0.6900	0.5385	0.9622	0.3389
300	0.7487	0.6406	0.9526	0.2888
400	0.7875	0.6990	0.9492	0.2525
500	0.7978	0.7276	0.9483	0.2392
600	0.8055	0.7444	0.9445	0.2353
700	0.8296	0.7719	0.9443	0.2087
800	0.8558	0.8025	0.9400	0.1885
900	0.8723	0.8260	0.9381	0.1792
1000	0.8712	0.8334	0.9391	0.1819
1500	0.8900	0.8636	0.9348	0.1644
2000	0.9015	0.8835	0.9325	0.1565
2500	0.9019	0.8864	0.9324	0.1540
3000	0.9073	0.8924	0.9334	0.1487
3500	0.9086	0.8886	0.9361	0.1444
4000	0.9105	0.9001	0.9358	0.1411
4601	0.9146	0.9031	0.9352	0.1395

and N is the number of examples. As shown in the results, the BS decreases as the training set grows. A smaller BS indicates better quality of the probability estimates. In Table 5 and Fig. 5 (bottom), we show the results of the TVP method on the same dataset. Comparing the results of the TVP with the IVP method, we can see that the TVP method provides slightly higher accuracy whether the data size is small or large (about 1 % difference) and slightly better BS. Nonetheless, although the TVP provides well-calibrated probabilities, it gives intervals that are much wider than those of the IVP method, especially when the training set is small. On the Car Evaluation dataset with 100 examples, the IVP probability interval has 8.7 % width, while the TVP interval has 22.83 % width. On the same dataset with all examples (1728), the IVP probability interval has 0.79 % width and the TVP interval has 3.33 % width. In Table 6 and Fig. 6 (top) we show the results of the IVP method on the Spambase dataset. Again, the probability estimates interval of the IVP method is very tight regardless of the size of the training set, while the probability estimates of the TVP method, shown in Table 7 and Fig. 6 (bottom), are generally wider. On the Spambase dataset with 100 examples, the IVP diameter is 5.76 % and the TVP diameter is 56.39 %. On the same dataset with all examples (4601), the IVP diameter is 0.14 % and the TVP diameter is 3.21 %. From these results, we see how the IVP outperforms the TVP method

Table 8 Comparison of offline results on the Car evaluation dataset

Algorithm	Accuracy	Duration (hh:mm:ss)	Probabilities
SVM-LR	95.54 %	00:00:11	93.95 %
TVP	93.11 %	02:17:47	88.61 %–96.39 %
IVP	89.70 %	00:05:49	88.36 %–90.10 %

Table 9 Comparison of offline results on the Spambase dataset

Algorithm	Accuracy	Duration (hh:mm:ss)	Probabilities
SVM-LR	91.81 %	00:10:11	88.33 %
TVP	91.46 %	72h+	90.31 %–93.52 %
IVP	92.16 %	04:50:22	92.22 %–92.36 %

in terms of tighter probability outputs.The accuracy of the IVP matches the accuracy of the TVP method (in fact, in some cases the IVP provides slightly higher accuracy). Therefore, accuracy is retained, while more effective probabilistic outputs are provided by the IVP.

In Tables 8 and 9, we present the final results of the 10-fold cross validation experiments of the SVM-LR, TVP, and IVP algorithms on the Car evaluation and Spambase datasets. Here, we also compare the time duration of each experiment. From the results, we notice that SVM-LR provides faster results. Nevertheless, our intention here is to compare our proposed IVP method with its transductive counterpart method which provides well-calibrated results (as it is shown in the online experiments). We can clearly see that, while the accuracy of the two methods on both datasets is around the same level, the probabilistic outputs of the IVP method are tighter than those of the TVP method and are well-calibrated. Moreover, the most important advantage of the IVP method against its transductive counterpart is shown again in the duration of each experiment, where the IVP outperforms the TVP in terms of time efficiency.

5 Conclusion

We have developed the Inductive version of the Venn Prediction framework which can provide well calibrated probabilistic outputs based on the only assumption that the data used are identically and independently distributed. We have performed extensive experiments with an SVM Inductive Venn Predictor on four datasets and we have compared its probabilistic outputs and computational efficiency to those of the SVM with Logistic Regression and an SVM Transductive Venn Predictor on three of the four datasets. In the comparison, it is shown that our IVP outperforms both SVM-LR and TVP in terms of the reliability and effectiveness of the probability outputs. Moreover, we have compared the time efficiency of the algorithms, and the proposed IVP has performed much better than the corresponding TVP method.

The probabilistic outputs of the IVP are near the actual accuracy as expected. In the future, we wish to experiment with more datasets, and use different values of steps for each re-training of the IVP algorithm, in order to understand if these parameters affect the results of the IVP method. Moreover, we aim to try several taxonomies for the developed IVP, and further we wish to investigate other algorithms that can be used to build effective IVPs. The particular IVP proposed here is based on the SVM classifier, but we expect that our conclusions will be true regardless of the underlying algorithm used.

Acknowledgments This work was co-funded by the European Regional Development Fund and the Cyprus Government through the Cyprus Research Promotion Foundation "DESMI 2009-2010", project TPE/ORIZO/0609(BIE)/24 ("Development of New Venn Prediction Methods for Osteoporosis Risk Assessment"). The authors are also grateful to Professor Vladimir Vovk for his help and useful discussions.

References

1. Bellotti, T., Luo, Z., Gammerman, A.: Reliable classification of childhood acute leukaemia from gene expression data using confidence machines. In: Proceedings of IEEE International Conference on Granular Computing (GRC '06), pp. 148–153 (2006)
2. Bohanec, M., Rajkovič, V.: Knowledge acquisition and explanation for multi-attribute decision making. In: 8th International Workshop "Expert Systems and Their Applications" (1988)
3. Cortez, P., Cerdeira, A., Almeida, F., Matos, T., Reis, J.: Modeling wine preferences by data mining from physicochemical properties. Decis. Support. Syst. **47**(4), 547–553 (2009)
4. Dashevskiy, M., Luo, Z.: Reliable probabilistic classification and its application to internet traffic. In: Advanced Intelligent Computing Theories and Applications. With Aspects of Theoretical and Methodological Issues, volume 5226 of LNCS, pp. 380–388. Springer (2008)
5. Dashevskiy, M., Luo, Z.: Predictions with confidence in applications. In: Perner, P. (ed.) Machine Learning and Data Mining in Pattern Recognition, volume 5632 of LNCS, pp. 775–786. Springer (2009)
6. Dmitry, D., Ilia, N.: Prediction with confidence based on a random forest classifier. In: Papadopoulos, H., Andreou, A., Bramer, M. (eds.) Artificial Intelligence Applications and Innovations, vol. 339, pp. 37–44. Springer, Boston (2010)
7. Drish, J.: Obtaining calibrated probability estimates from Support Vector Machines (1998)
8. Frank, A., Asuncion, A.: UCI machine learning repository (2010)
9. Gammerman, A., Nouretdinov, I., Burford, B., Chervonenkis, A., Vovk, V., Luo, Z.: Clinical mass spectrometry proteomic diagnosis by conformal predictors. Stat. Appl. Genet. Mol. Biol. **7**(2) (2008)
10. Gammerman, A., Vovk, V., Vapnik, V.: Learning by transduction. In: Uncertainty in Artificial Intelligence, pp. 148–155. Morgan Kaufmann (1998)
11. Lambrou, A., Papadopoulos, H., Gammerman, A.: Reliable confidence measures for medical diagnosis with evolutionary algorithms. IEEE Trans. Inf. Technol. Biomed. **15**(1), 93–99 (2011)
12. Lambrou, A., Papadopoulos, H., Kyriacou, E., Pattichis, C.S., Pattichis, M.S., Gammerman, A., Nicolaides, A.: Evaluation of the risk of stroke with confidence predictions based on ultrasound carotid image analysis. Int. J. Artif. Intell. Tools **21**(04), 1240016 (2012)
13. Lambrou, A., Papadopoulos, H., Nouretdinov, I., Gammerman, A.: Reliable probability estimates based on support vector machines for large multiclass datasets. In: Iliadis, L., Maglogiannis, I., Papadopoulos, H., Karatzas, K., Sioutas, S. (eds.) Artificial Intelligence Applications and Innovations, volume 382 of IFIP Advances in Information and Communication Technology, pp. 182–191. Springer, Berlin Heidelberg (2012)
14. Papadopoulos, H.: Inductive conformal prediction: theory and application to neural networks. In: Fritzsche, P. (ed.) Tools in Artificial Intelligence, chapter 18, pp. 315–330. InTech, Vienna (2008)
15. Papadopoulos, H.: Reliable probabilistic prediction for medical decision support. In: Proceedings of the 7th IFIP International Conference on Artificial Intelligence Appications and Innovations (AIAI 2011), volume 364 of IFIP AICT, pp. 265–274. Springer (2011)
16. Papadopoulos, H.: Reliable probabilistic classification with neural networks. Neurocomputing **107**, 59–68 (2013)
17. Papadopoulos, H., Gammerman, A., Vovk, V.: Reliable diagnosis of acute abdominal pain with conformal prediction. Int. J. Eng. Intell. Syst. Electr. Eng. Commun. **17**(2–3), 127–137 (2009)
18. Papadopoulos, H., Papatheocharous, E., Andreou, A.S.: Reliable confidence intervals for software effort estimation. In: Proceedings of the 2nd Workshop on Artificial Intelligence Techniques in Software Engineering (AISEW 2009), volume 475 of CEUR Workshop Proceedings, pp. 211–220. CEUR WS.org (2009)
19. Papadopoulos, H., Proedrou, K., Vovk, V., Gammerman, A.: Inductive confidence machines for regression. In: Proceedings of the 13th European Conference on Machine Learning (ECML'02), volume 2430 of LNCS, pp. 345–356. Springer (2002)
20. Papadopoulos, H., Vovk, V., Gammerman, A.: Regression conformal prediction with nearest neighbours. J. Artif. Intell. Res. **40**, 815–840 (2011)
21. Papadopoulos, H., Vovk, V., Gammerman, A.: Qualified predictions for large data sets in the case of pattern recognition. In: Proceedings of the 2002 International Conference on Machine Learning and Applications (ICMLA'02), pp. 159–163. CSREA Press (2002)
22. Papadopoulos, H., Vovk, V., Gammerman, A.: Conformal prediction with neural networks. In: Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'07), vol. 2, pp. 388–395. IEEE Computer Society (2007)

23. Platt, J.C.: Fast training of support vector machines using sequential minimal optimization. In: Schölkopf, B., Burges, C.J.C., Smola, A.J. (eds.) Advances in kernel methods, pp. 185–208. Cambridge (1999)
24. Platt, J.C.: Probabilistic outputs for Support Vector Machines and comparisons to regularized likelihood methods. In: Advances in large margin classifiers, pp. 61–74. MIT Press (1999)
25. Proedrou, K., Nouretdinov, I., Vovk, V., Gammerman, A.: Transductive confidence machines for pattern recognition. In: Proceedings of the 13th European Conference on Machine Learning (ECML'02), volume 2430 of Lecture Notes in Computer Science, pp. 381–390. Springer (2002)
26. Ross Quinlan, J.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco (1993)
27. Saunders, C., Gammerman, A., Vovk, V.: Transduction with confidence and credibility. In: Proceedings of the 16th International Joint Conference on Artificial Intelligence, vol. 2, pp. 722–726. Morgan Kaufmann, Los Altos (1999)
28. Vovk, V., Shafer, G., Nouretdinov, I.: Self-calibrating probability forecasting. In: Thrun, S., Saul, L.K., Schölkopf, B. (eds.) Advances in Neural Information Processing Systems 16, pp. 1133–1140. MIT Press, Cambridge (2004)
29. Vovk, V., Gammerman, A., Saunders, C.: Machine-learning applications of algorithmic randomness. In: Proceedings of the Sixteenth International Conference on Machine Learning, pp. 444–453. Morgan Kaufmann (1999)
30. Vovk, V., Gammerman, A., Shafer, G.: Algorithmic Learning in a Random World. Springer, New York (2005)
31. Zadrozny, B., Elkan, C.: Transforming classifier scores into accurate multiclass probability estimates. In: Proceedings of the 8th ACM International Conference on Knowledge Discovery and Data Mining, pp. 694–699 (2002)
32. Zhou, C., Nouretdinov, I., Luo, Z., Adamskiy, D., Randell, L., Coldham, N., Gammerman, A.: A comparison of Venn Machine with Platt's method in probabilistic outputs. In: Proceedings of the 7th IFIP International Conference on Artificial Intelligence Appications and Innovations (AIAI 2011), volume 364 of IFIP AICT, pp. 483–490. Springer (2011)