Provably Improving Expert Predictions with Conformal Prediction

Eleni Straitouri¹, Lequn Wang², Nastaran Okati¹, and Manuel Gomez Rodriguez¹

¹Max Planck Institute for Software Systems, {estraitouri, nastaran, manuelgr}@mpi-sws.org
²Cornell University, lw633@cornell.edu

Abstract

Automated decision support systems promise to help human experts solve tasks more efficiently and accurately. However, existing systems typically require experts to understand when to cede agency to the system or when to exercise their own agency. Moreover, if the experts develop a misplaced trust in the system, their performance may worsen. In this work, we lift the above requirement and develop automated decision support systems that, by design, do not require experts to understand when to trust them to provably improve their performance. To this end, we focus on multiclass classification tasks and consider automated decision support systems that, for each data sample, use a classifier to recommend a subset of labels to a human expert. We first show that, by looking at the design of such systems from the perspective of conformal prediction, we can ensure that the probability that the recommended subset of labels contains the true label matches almost exactly a target probability value. Then, we identify the set of target probability values under which the human expert is provably better off predicting a label among those in the recommended subset and develop an efficient practical method to find a near-optimal target probability value. Experiments on synthetic and real data demonstrate that our system can help the experts make more accurate predictions and is robust to the accuracy of the classifier it relies on.

1 Introduction

In recent years, there has been an increasing interest in developing automated decision support systems to help human experts solve tasks in a wide range of critical domains, from medicine [1] and drug discovery [2] to criminal justice processes [3]. Among them, one of the main focus has been classification tasks, where a decision support system uses a classifier to make label predictions, and the experts decide when to follow or not follow the predictions made by the classifier [4, 5, 6].

However, these systems typically require the human experts to understand when to trust a prediction made by the classifier. Otherwise, the experts may be better off solving the classification tasks on their own [7]. This follows from the fact that, in general, the accuracy of a classifier differs across data samples [8]. Unfortunately, it is not yet clear how to make sure that the experts do not develop a misplaced trust that decreases their performance [9, 10, 11]. In this work, our goal is to develop decision support systems that, by design, do not require experts to understand when to trust the system and provably improve their performance.

Our contributions. We consider multiclass classification tasks and decision support systems that, for each data sample, use a classifier to recommend a subset of labels to a human expert. We view this type of decision support systems as more natural since, given a set of alternatives, human experts tend to narrow down their options to a subset of them before making their final decision [12, 13, 14]. In a way, our support systems help the experts by automatically narrowing down their options for them. In this context, a recent empirical study has also concluded that, in terms of overall accuracy, it may be more beneficial to recommend a subset of options than a final decision [15].

Here, one could still argue that the experts need to understand when to trust the system—when to predict one of the labels in the subset recommended by the system—as noted by Levy et al. [15]. This is largely due to the fact that the accuracy of the classifier differs across different data samples. Moreover, when the
size of the recommended subset is always the same across samples, it is difficult to bound the probability that the true label belongs to the subsets without making distributional assumptions about the data and the classifier. To circumvent this, in our work, we allow the recommended subset to be of different size across samples. Then, using the theory of conformal prediction [16, 17], we can guarantee that the probability that the true label belongs to the subset always matches almost exactly a given target probability value, without making any distributional assumptions about the data or the classifier.

In addition, we identify the set of target probability values under which human experts are guaranteed to achieve better average accuracy by predicting a label from the recommended subset, rather than the entire set of labels. Further, we develop an efficient method to find a near-optimal target probability value given an estimator of the expert’s success probability for any recommended subset. In this context, we also propose a practical method to obtain such estimator using the confusion matrix of the expert predictions in the original classification task as well as a given discrete choice model.

Finally, we experiment with synthetic and real data comprising of 511,400 expert predictions over 10,000 natural images. The results demonstrate that our decision support system is robust to the performance of the classifier it relies on—the competitive advantage it provides improves with the accuracy of the classifier, and the human experts do not decrease their performance by using the system even if the classifier is very inaccurate. Additionally, the results also show that, even if the classifiers that our system relies on have high accuracy, an expert using our system may achieve significantly higher accuracy than the classifiers on their own—in our experiments with real data, the relative reduction in misclassification probability is over 58%.

Further related work. Our work builds upon further related work on distribution-free uncertainty quantification and learning under algorithmic triage.

There exist three fundamental notions of distribution-free uncertainty quantification in the literature: calibration, confidence intervals, and prediction sets [17, 18, 19, 16]. Our work is most closely related to the rapidly increasing literature on prediction sets [20, 21, 22, 23], however, to the best of our knowledge, prediction sets have not been optimized to serve automated decision support systems such as ours—existing works do not consider there is a human expert, with a given prediction accuracy, who predicts a label among those in the constructed prediction sets.

Learning under algorithmic triage seeks the development of machine learning models that operate under different automation levels—models that take decisions for a given fraction of instances and leave the remaining ones to human experts [8, 24, 25, 26, 27]. This line of work has predominantly focused on supervised learning settings with a few very recent notable exceptions [28, 29]. However, in this line of work, each sample is either predicted by the model or by the human expert. In contrast, in our work, the model helps the human predict each sample.

2 Problem Formulation

We consider a multiclass classification task where a human expert observes a feature vector \( x \in \mathcal{X} \), with \( x \sim P(X) \), and needs to predict a label \( y \in \mathcal{Y} = \{1, \ldots, n\} \), with \( y \sim P(Y|X) \). Then, our goal is to design an automated decision support system \( C \) that, given a feature vector \( x \in \mathcal{X} \), helps the expert by automatically narrowing down the set of potential labels to a subset of them \( C(x) \subseteq \mathcal{Y} \) using a trained classifier \( \hat{f}(x) \in [0,1]^n \) that outputs softmax scores for each class. Refer to Figure 1 for an illustration of the automated decision support system we consider.

Here, we would like that, by design, the expert is always better-off by using the automated decision support system \( C \) to make predictions, i.e.,

\[
P[\hat{Y} = Y | C(X)] > P[\hat{Y} = Y | \mathcal{Y}],
\]

where \( P[\hat{Y} = Y | C(X)] \) denotes the expert’s probability of success, i.e., the probability that the label \( \hat{Y} \) predicted by the human expert matches the true label \( Y \), if she predicts a label among those in the subset \( C(X) \).

1To facilitate research in this area, we will release an open-source implementation of our system with the final version of the paper. We have submitted an anonymized version of the code with the supplementary material.

2We denote random variables with capital letters and realizations of random variables with lower case letters.

2
Figure 1: Our automated decision support system $C$. Given a sample with a feature vector $x$, our system $C$ narrows down the set of potential labels $y \in \mathcal{Y}$ to a subset of them $C(x)$ using the softmax scores $\hat{f}_y(x)$ provided by a classifier $\hat{f}$ for each class $y$. The human expert receives the recommended subset $C(x)$, together with the sample, and predicts a label $\hat{y}$ from $C(x)$.

However, not all automated decision support systems fulfilling the above requirement will be equally useful—some will help experts increase their success probability more than others. Therefore, among those systems satisfying Eq. 1, we would ideally like to find the system $C^*$ that would help the expert achieve the highest success probability, i.e.,

$$C^* = \arg\max_C \mathbb{P}[\hat{Y} = Y | C(X)].$$

To address the design of such a system, we will look at the problem from the perspective of conformal prediction [16].

3 Subset Selection using Conformal Prediction

In general, if the trained classifier $\hat{f}$ we use to build $C(X)$ is not perfect, the true label $Y$ may or may not be included in $C(X)$. Therefore, we start by rewriting the success probability of human experts using a decision support system $C$ as follows:

$$\mathbb{P}[\hat{Y} = Y | C(X)] = \mathbb{P}[Y \in C(X)] \mathbb{P}[\hat{Y} = Y | C(X), Y \in C(X)] + \mathbb{P}[Y \notin C(X)] \mathbb{P}[\hat{Y} = Y | C(X), Y \notin C(X)]$$

$$= \mathbb{P}[Y \in C(X)] \mathbb{P}[\hat{Y} = Y | C(X), Y \in C(X)],$$

where we have used the fact that, if $Y \notin C(X)$, the human expert will always misclassify the true label $Y$, i.e., $\mathbb{P}[\hat{Y} = Y | C(X), Y \notin C(X)] = 0$.

Now, we will construct the subsets $C(X)$ using the theory of conformal prediction. This will allow our system to be robust to the accuracy of the classifier $\hat{f}$ it uses—the probability $\mathbb{P}[Y \in C(X)]$ that the true label $Y$ belongs to the subset $C(X) = C_\alpha(X)$ will match almost exactly a given target probability value $1 - \alpha$, without making distributional assumptions about the data distribution $P(X)P(Y | X)$ nor the classifier $\hat{f}$.

Let $D_{\text{cal}} = \{(x_i, y_i)\}_{i=1}^m$ with $(x_i, y_i) \sim P(X)P(Y | X)$ be a calibration set, $s(x_i, y_i) = 1 - \hat{f}_{y_i}(x_i)$ be the conformal score $s$ (i.e., if the classifier is catastrophically wrong, the conformal score will be large), and $\hat{q}$ be the $\lceil (m+1)\alpha \rceil$ empirical quantile of the conformal scores $s(x_1, y_1), \ldots, s(x_m, y_m)$. Then, if we use the quantile $\hat{q}$ to construct the subsets $C_\alpha(X)$ for new data samples as follows:

$$C_\alpha(X) = \{ y \mid s(X, y) \leq \hat{q} \},$$

3In general, the conformal score $s(x, y)$ can be any function of $x$ and $y$ measuring the similarity between samples.
it holds that the probability that the true label $Y$ belongs to the subset $C_\alpha(X)$ is almost exactly $1 - \alpha$. More formally, we have the following well-known conformal calibration coverage guarantee (refer to Appendix A in Angelopoulos and Bates [16] for a proof):

**Theorem 1** Consider an automated decision support system $C_\alpha$ that constructs the subsets $C_\alpha(X)$ using Eq. 4. Then, it holds that

$$1 - \alpha \leq \mathbb{P}(Y \in C_\alpha(X)) \leq 1 - \alpha + \frac{1}{m + 1},$$

where the probability is over the randomness in the samples it helps predicting and the calibration data used to compute the empirical quantile $\hat{q}$.

Further, if we combine the above Theorem and Eq. 5, we can readily bound the expert’s success probability $\mathbb{P}(\hat{Y} = Y | C_\alpha(X))$ from above and below, i.e.,

$$(1 - \alpha)\mathbb{P}(\hat{Y} = Y | C_\alpha(X), Y \in C_\alpha(X)) \leq \mathbb{P}(\hat{Y} = Y | C_\alpha(X)) \leq (1 - \alpha + \frac{1}{m + 1})\mathbb{P}(\hat{Y} = Y | C_\alpha(X), Y \in C_\alpha(X)).$$

Now, building on the lower bound above, we can characterize the set of target probability values $1 - \alpha$ under which the human expert will provably benefit from using the automated decision support system to make predictions.

**Proposition 1** Consider an automated decision support system $C_\alpha$ that constructs the subsets $C_\alpha(X)$ using Eq. 4. If

$$(1 - \alpha)\mathbb{P}(\hat{Y} = Y | C_\alpha(X), Y \in C_\alpha(X)) > \mathbb{P}(\hat{Y} = Y | Y),$$

then $\mathbb{P}(\hat{Y} = Y | C_\alpha(X)) > \mathbb{P}(\hat{Y} = Y | Y)$.

The above result tells us that, for the system to improve the expert’s performance, the probability that the expert predicts the true label from the recommended set $C_\alpha(X)$ given that the true label is within the subset should be strictly larger than $(1 - \alpha)^{-1}$ times the success probability when she predicts a label without using the system. However, we would like to emphasize that not all automated decision support systems fulfilling the above requirement are equally useful—some target probability values $1 - \alpha$ will lead to larger performance gains. Next, we will investigate how to find the optimal target probability value $1 - \alpha^*$.

### 4 Optimizing Conformal Prediction

In this section, we start by realizing that, to identify the optimal value of $\alpha^*$ given a calibration set $D_{cal}$, we only need to consider a finite countable set $\mathcal{A} = \{\alpha_i\}_{i \in [nm]}$. This is because, for each sample $(x_i, y_i) \in D_{cal}$, the subset $C_\alpha(x_i)$ will only change whenever $\hat{q}$ crosses each score value $s(x_i, y)$ for each different potential label value $y \in Y$.

As a result, we can conclude that, to recover the optimal value $\alpha^*$ for systems that construct the recommended subsets using Eq. 4, we need to solve the following maximization problem:

$$\alpha^* = \arg\max_{\alpha \in \mathcal{A}} \mathbb{P}(\hat{Y} = Y | C_\alpha(X)).$$

(6)

To find a near optimal solution $\hat{\alpha}$ to the above problem, we leverage the lower bound of the expert’s success probability $\mathbb{P}(\hat{Y} = Y | C_\alpha(X))$ given by Eq. 5. However, to use the lower bound, we need to estimate the expert’s conditional success probability $\mathbb{P}(\hat{Y} = Y | C_\alpha(X), Y \in C_\alpha(X))$. Assume for now that we have access to an estimator $\hat{\mu}_\alpha$ of the expert’s conditional success probability such that with probability at least $1 - \delta$, where $\delta \in (0, 1)$, it holds that $|\hat{\mu}_\alpha - \mathbb{P}(\hat{Y} = Y | C_\alpha(X), Y \in C_\alpha(X))| \leq \epsilon_\alpha$ for all $\alpha \in \mathcal{A}$. Then, we can use the following Proposition to find a near optimal solution $\hat{\alpha}$ to Eq. 6 with high probability:

---

4 All proofs are included in the Appendix.
Algorithm 1 Finding a near-optimal target probability $1 - \hat{\alpha}$

1: Input: $f, D_{est}, D_{cal}$
2: Initialize: $A = \emptyset, \hat{\alpha} \leftarrow 0, t \leftarrow 0$
3: for $x \in D_{cal}$ do
4:     for $y \in \mathcal{Y}$ do
5:         $\alpha \leftarrow 1 - \min\{i \mid s(x,y_i) \leq s(x,y_i),(x,y_i) \in D_{cal}\}$
6:     $A \leftarrow A \cup \{\alpha\}$
7: end for
8: end for
9: for $\alpha \in A$ do
10:    $\hat{\mu}_\alpha, \epsilon_\alpha \leftarrow \text{Estimate}(\alpha, D_{est})$ \{It uses Eq. 10\}
11:    if $t \leq (1 - \alpha)(\hat{\mu}_\alpha - \epsilon_\alpha)$ then
12:        $t \leftarrow (1 - \alpha)(\hat{\mu}_\alpha - \epsilon_\alpha)$
13:        $\hat{\alpha} \leftarrow \alpha$
14:    end if
15: end for
16: return $\hat{\alpha}$

Proposition 2 Consider an automated decision support system $C_{\hat{\alpha}}$ with

$$\hat{\alpha} = \arg\max_{\alpha \in A} (1 - \alpha)(\hat{\mu}_\alpha - \epsilon_\alpha).$$

Then, with probability at least $1 - \delta$, it holds that

$$P[\hat{Y} = Y | C_{\alpha}(X)] \geq P[\hat{Y} = Y | C_{\alpha}(X)] - 2(1 - \alpha)\epsilon_\alpha - \frac{1}{m + 1}$$

for all $\alpha \in A$.

More specifically, the above result directly implies that, with probability at least $1 - \delta$, it holds that:

$$P[\hat{Y} = Y | C_{\alpha^*}(X)] - P[\hat{Y} = Y | C_{\hat{\alpha}}(X)] \leq 2(1 - \alpha^*)\epsilon_{\alpha^*} + \frac{1}{m + 1}.$$  \hspace{1cm} (8)

Algorithm 1 summarizes the resulting procedure, which has an overall complexity of $O(nm \log m)$.

Estimating expert’s conditional success probability. We would first like to emphasize that our results so far, including Algorithm 1, are agnostic to the method used to estimate the expert’s conditional success probability $P[\hat{Y} = y | C_\alpha(x), y \in C_\alpha(x)]$. In what follows, we propose a practical estimation method that builds upon the multinomial logit model (MNL), one of the most popular models in the vast literature on discrete choice models [30].

More specifically, we assume that we have access to (an estimation of) the confusion matrix $C$ for the expert predictions in the (original) multiclass classification task, similarly as in Kerrigan et al. [31], i.e.,

$$C = [C_{yy'}]_{y,y' \in \mathcal{Y}}, \text{ where } C_{yy'} = P[\hat{Y} = y' | \mathcal{Y}, Y = y].$$

Moreover, given a sample $(x,y)$, we also assume that the expert’s conditional success probability for the subset $C_\alpha(x)$ follows a multinomial logit model (MNL) [30], i.e.,

$$P[\hat{Y} = y | C_\alpha(x), y \in C_\alpha(x)] = \frac{e^{u_{yy}}}{\sum_{y' \in C(x)} e^{u_{yy'}}},$$  \hspace{1cm} (9)
Then, it holds that:

$\hat{\mu}_\alpha = \frac{1}{k_\alpha} \sum_{i \in [m]} \mathbb{P}[Y = y_i \mid \mathcal{C}_\alpha(x_i), y_i \in \mathcal{C}_\alpha(x_i)],$  \hspace{1cm} (10)

where $k_\alpha = \sum_{i \in [m]} \mathbb{I}\{y_i \in \mathcal{C}_\alpha(x_i)\}$. Here, using Hoeffding’s inequality\(^5\) we can conclude that, with probability at least $1 - \delta$, it holds that (refer to Appendix C):

$$|\hat{\mu}_\alpha - \mathbb{P}[Y = Y \mid \mathcal{C}_\alpha(X), Y \in \mathcal{C}_\alpha(X)]| \leq \sqrt{\frac{\log \frac{mn}{\delta}}{2k_\alpha}} := \epsilon_\alpha.$$  

As a consequence, as $m \to \infty$, the right hand side of Eq. 8 converges to zero almost surely, since $\lim_{m \to \infty} \frac{k_\alpha}{m} = (1 - \alpha)$ almost surely by the law of large numbers.

5 Beyond Standard Conformal Prediction

Until now, we have used standard conformal prediction \([16]\) to construct the recommended subsets $\mathcal{C}(X)$—we have constructed the subsets $\mathcal{C}_\alpha(X)$ by comparing the conformal scores $s(X, y)$ to a single threshold $\hat{q}$, as shown in Eq. 4. Here, we will show that we can sometimes improve the performance of our system by constructing $\mathcal{C}_\alpha(X)$ using two thresholds $\hat{q}_1$ and $\hat{q}_2$. By doing so, the recommended subsets will include label values whose corresponding conformal scores are neither unreasonably large, as in standard conformal prediction, nor unreasonably low, in comparison with the conformal scores of the samples in the calibration set $\mathcal{D}_{\text{cal}}$. This will be particularly useful whenever the classifier our system relies on has low accuracy.

More specifically, let $\alpha_1, \alpha_2 \in [0, 1]$, with $\alpha_1 < \alpha_2$, and $\hat{q}_1$ and $\hat{q}_2$ be the $\left\lfloor \frac{m+1(1-\alpha_1)}{m} \right\rfloor$ and $\left\lfloor \frac{m+1(1-\alpha_2)}{m} \right\rfloor$ empirical quantiles of the conformal scores $s(x_1, y_1), \ldots, s(x_m, y_m)$ of the samples in the calibration set $\mathcal{D}_{\text{cal}}$. Then, if we use the quantiles $\hat{q}_1$ and $\hat{q}_2$ to construct the subsets $\mathcal{C}_{\alpha_1,\alpha_2}(X)$ for new data samples as follows:

$$\mathcal{C}_{\alpha_1,\alpha_2}(X) = \{y \mid \hat{q}_2 \leq s(X, y) \leq \hat{q}_1\},$$  \hspace{1cm} (11)

it holds that the probability that the true label $Y$ belongs to the subset $\mathcal{C}_{\alpha_1,\alpha_2}(X)$ is almost exactly $\alpha_2 - \alpha_1$. More formally, we have the following conformal calibration coverage guarantee, which is the counterpart of Theorem 1.

Theorem 2 Consider an automated decision support system $\mathcal{C}_{\alpha_1,\alpha_2}$ that constructs $\mathcal{C}_{\alpha_1,\alpha_2}(X)$ using Eq. (11). Then, it holds that:

$$\alpha_2 - \alpha_1 - \frac{1}{m+1} \leq \mathbb{P}[Y \in \mathcal{C}_{\alpha_1,\alpha_2}(X)] \leq \alpha_2 - \alpha_1 + \frac{1}{m+1}.$$  

Perhaps surprisingly, building upon the above Theorem, we can identify a condition under which a system $\mathcal{C}_{\alpha_1,\alpha_2}$ that constructs $\mathcal{C}_{\alpha_1,\alpha_2}(X)$ using Eq. (11) is provably better than a system $\mathcal{C}_{\alpha_1}$ that constructs $\mathcal{C}_{\alpha_1}(X)$ using Eq. 4.

Proposition 3 Consider two automated decision support systems $\mathcal{C}_{\alpha_1}$ and $\mathcal{C}_{\alpha_1,\alpha_2}$ that construct the subsets $\mathcal{C}_{\alpha_1}(X)$ and $\mathcal{C}_{\alpha_1,\alpha_2}(X)$ using Eq. 4 and (11), respectively. If

$$\left(\alpha_2 - \alpha_1 - \frac{1}{m+1}\right) \mathbb{P}[\hat{Y} = Y \mid \mathcal{C}_{\alpha_1,\alpha_2}(X), Y \in \mathcal{C}_{\alpha_1,\alpha_2}(X)] > \left(1 - \alpha_1 + \frac{1}{m+1}\right) \mathbb{P}[\hat{Y} = Y \mid \mathcal{C}_{\alpha_1}(X), Y \in \mathcal{C}_{\alpha_1}(X)],$$

then $\mathbb{P}[\hat{Y} = Y \mid \mathcal{C}_{\alpha_1,\alpha_2}(X)] > \mathbb{P}[\hat{Y} = Y \mid \mathcal{C}_{\alpha_1}(X)].$
Moreover, let \( \hat{\mu}_{\alpha_1, \alpha_2} \) be an estimator of the expert’s conditional success probability such that, with probability at least \( 1 - \delta \), where \( \delta \in (0, 1) \), it holds that \(|\hat{\mu}_{\alpha_1, \alpha_2} - \mathbb{P}[\hat{Y} = Y | C_{\alpha_1, \alpha_2}(X), Y \in C_{\alpha_1, \alpha_2}(X)]| \leq \epsilon_{\alpha_1, \alpha_2} \). Then, we can proceed similarly as in standard conformal prediction to find the near optimal \( \hat{\alpha}_1, \hat{\alpha}_2 \in A \) with high probability:

**Proposition 4** Consider an automated decision support system \( C_{\hat{\alpha}_1, \hat{\alpha}_2} \) with

\[
\hat{\alpha}_1, \hat{\alpha}_2 = \arg\max_{\alpha_1, \alpha_2 \in A | \alpha_1 < \alpha_2} \left( \alpha_2 - \alpha_1 - \frac{1}{m+1} \right) \cdot (\hat{\mu}_{\alpha_1, \alpha_2} - \epsilon_{\alpha_1, \alpha_2}),
\]

where \( \delta \in [0, 1] \). Then, with probability at least \( 1 - \delta \), it holds that

\[
\mathbb{P}[\hat{Y} = Y | C_{\hat{\alpha}_1, \hat{\alpha}_2}(X)] \geq \mathbb{P}[\hat{Y} = Y | C_{\alpha_1, \alpha_2}(X)] - \frac{2}{m+1} - 2 \left( \alpha_2 - \alpha_1 - \frac{1}{m+1} \right) \epsilon_{\alpha_1, \alpha_2}
\]

for all \( \alpha_1, \alpha_2 \in A \) such that \( \alpha_1 < \alpha_2 \).

The resulting procedure is a straightforward extension of Algorithm 1. Here, it is worth pointing out that, in contrast with the case of standard conformal prediction, the complexity of finding the near optimal \( \hat{\alpha}_1 \) and \( \hat{\alpha}_2 \) is \( O(n^2m^2\log m) \). However, we can still rely on the practical method to estimate the expert’s conditional success probability introduced in Section 4 since the dependence of \( \epsilon_{\alpha_1, \alpha_2} \) with respect to the number of \( \alpha_1 \) and \( \alpha_2 \) values in \( A \) is logarithmic.

### 6 Experiments on Synthetic Data

In this section, we quantify the influence that the accuracy of the expert and the classifier, the size of the calibration and estimation sets, as well as the number of label values have on the performance of our system using synthetic data.

**Experimental setup.** We create a variety of synthetic prediction tasks, each with 20 features per sample and a varying number of label values \( n \) and difficulty. Refer to Appendix G for more details about the prediction tasks. For each prediction task, we generate 10,000 samples, pick 20% of these samples at random as test set and split also at random the remaining 80% into three disjoint subsets for training, calibration and estimation, whose sizes we vary across experiments. In each experiment, we specify the number of samples in the calibration and estimation sets—the remaining samples are used for training.

---

7 All algorithms ran on a Debian machine equipped with Intel Xeon E5-2667 v4 @ 3.2 GHz, 32GB memory and two M40 Nvidia Tesla GPU cards. See Appendix for further details.
The performance of our system under $\hat{\alpha}$ found by Algorithm 1 and under $\alpha^*$ is very similar. Given three prediction tasks where the expert and the classifier achieve different success probabilities $P[\hat{Y} = Y | X]$ and $P[Y' = Y]$, we compare the performance of our system under the near optimal $\hat{\alpha}$ found by Algorithm 1 and under all other possible $\alpha \in A$ values. Figure 2 summarizes the results, which suggest that: (i) the performance under $\hat{\alpha}$ is very close to that under $\alpha^*$, as suggested by Proposition 2, and, (ii) as long as $\alpha \leq \alpha^*$, the performance of our system increases monotonically with respect to $\alpha$, however, once $\alpha > \alpha^*$, the performance deteriorates as we increase $\alpha$. (iii) the higher the expert’s success probability $P[\hat{Y} = Y | X]$, the smaller the optimal $\alpha^*$ and thus the greater the average size of the subsets $C_{\alpha^*}(X)$. We found qualitatively
We consider prediction tasks with a varying number of label values, from $n$. The greater the number of label values, the more an expert benefits from using our system.

The size of the calibration and estimation set is $m$. Table 2: Average success probabilities $P[Y' = Y]$ and $P[\hat{Y} = Y | C_{\alpha}(X)]$ achieved by three popular deep neural network classifiers and by an expert using our system with each of the classifiers on the CIFAR-10H dataset. The size of the calibration and estimation set is $m = 1,500$ and the expert’s average success probability at solving the (original) multiclass task is $P[\hat{Y} = Y | \mathcal{Y}] = 0.947$. For readability, each cell shows only the average over 10 random data splits since the standard errors are all below $10^{-2}$.

| Classifiers     | $P[Y' = Y]$ | $P[\hat{Y} = Y | C_{\alpha}(X)]$ |
|-----------------|-------------|----------------------------------|
| ResNet-110      | 0.928       | 0.985                            |
| PreResNet-110   | 0.944       | 0.984                            |
| DenseNet        | 0.964       | 0.985                            |

Table 2: Average success probabilities $P[Y' = Y]$ and $P[\hat{Y} = Y | C_{\alpha}(X)]$ achieved by three popular deep neural network classifiers and by an expert using our system with each of the classifiers on the CIFAR-10H dataset. The size of the calibration and estimation set is $m = 1,500$ and the expert’s average success probability at solving the (original) multiclass task is $P[\hat{Y} = Y | \mathcal{Y}] = 0.947$. For readability, each cell shows only the average over 10 random data splits since the standard errors are all below $10^{-2}$.

similar results using other expert-classifier pairs with different success probabilities.

**Our system needs a relatively small amount of calibration and estimation data.** We vary the amount of calibration and estimation data $m$ we feed into Algorithm 1 and, each time, estimate the expert’s success probability $P[\hat{Y} = Y | C_{\alpha}(X)]$. Across prediction tasks, we consistently find that our system needs a relatively small amount of calibration and estimation data to perform well. For example, for all prediction tasks with $n = 10$ label values and varying level of difficulty, if we increase the amount of calibration and estimation data from $m = 160$ to $m = 1200$, the relative gain in success probability $P[\hat{Y} = Y | C_{\alpha}(X)]$ with respect to $P[\hat{Y} = Y | \mathcal{Y}]$, averaged across experts with $P[\hat{Y} = Y | \mathcal{Y}] \in \{0.3, 0.5, 0.7, 0.9\}$, goes from 47.51 ± 6.62% to 48.67 ± 6.65%.

The greater the number of label values, the more an expert benefits from using our system. We consider prediction tasks with a varying number of label values, from $n = 10$ to $n = 100$, and estimate the expert’s success probability $P[\hat{Y} = Y | C_{\alpha}(X)]$ for each task. Our results suggest that the relative gain in success probability $P[\hat{Y} = Y | C_{\alpha}(X)]$ with respect to $P[\hat{Y} = Y | \mathcal{Y}]$, averaged across experts with $P[\hat{Y} = Y | \mathcal{Y}] \in \{0.3, 0.5, 0.7, 0.9\}$, increases with the number of label values. For example, for $m = 400$, it goes from 46.64 ± 6.42% for $n = 10$ to 71.19 ± 8.96% for $n = 100$. For other $m$ values, we found a similar trend.

### 7 Experiments on Real Data

In this section, we evaluate the performance of our system using a dataset with real expert predictions over natural images and several popular and highly accurate deep neural network classifiers. Here, we focus on systems $C_{\alpha}, C_{\alpha,1}$ that construct $C_{\alpha}(X)$ using Eq. 1 because, whenever the classifiers are highly accurate, systems $C_{\alpha_1, \alpha_2}$ with $\alpha_2 \neq 1$ do not offer a competitive advantage.

**Data description and experimental setup.** We experiment with the dataset CIFAR-10H [32], which contains 10,000 natural images taken from the test set of the standard CIFAR-10 [33]. Each of these images belongs to one of $n = 10$ classes and contains approximately 50 expert predictions $Y, \{} \hat{Y} \}. Here, we randomly split the dataset into three disjoint subsets for calibration, estimation and test, whose sizes we vary across experiments. In each experiment, we specify the number of samples in the calibration and estimation sets—the remaining samples are used for testing.

Rather than training a classifier, we use three popular and highly accurate deep neural network classifiers trained on CIFAR-10, namely ResNet-110 [34], PreResNet-110 [35] and DenseNet [36]. Moreover, we use the human predictions $\hat{Y}$ to estimate the confusion matrix $C$ for the expert predictions in the (original) multiclass classification task [31] and then sample the expert’s prediction $\hat{Y}$ from the multinomial logit model.

---

8The dataset CIFAR-10H is among the only publicly available datasets that we found containing multiple expert predictions per sample, necessary to estimate the confusion matrix $C$, a relatively large number of samples, and more than two classes. However, since our theoretical guarantees are rather general, our system may help improving expert predictions in other applications.
defined by Eq. 9 to both estimate the expert’s conditional success probability using Eq. 10, which in turn we use during calibration, and estimate the expert’s success probability during test.

**Results.** We start by estimating the average success probability $P[\hat{Y} = Y | \mathcal{C}_\alpha(X)]$ during test for an expert using our system with each of the above mentioned classifiers. Table 2 summarizes the results, where we compare the success probability achieved by our system and by the corresponding classifier on its own and we report the expert’s success probability at solving the (original) multiclass task in the caption. We find very encouraging that, despite the classifiers are highly accurate, our results suggest that an expert using our system can solve the prediction task significantly more accurately than the classifiers. More specifically, the relative reduction in misclassification probability goes from 58\% (DenseNet) to a striking 79.2\% (ResNet-110). Moreover, it is also worth noting that, by using our system, our results suggest that the (average) expert would reduce their misclassification probability by $\sim$70%.

Next, for each choice of classifier, we compare the performance of our system under the near optimal $\hat{\alpha}$ found by Algorithm 1 and under all other possible $\alpha$ values, including the optimal $\alpha^*$. Figure 3 summarizes the results, which suggest that, similarly as in the experiments on synthetic data, the performance of our system under $\hat{\alpha}$ and $\alpha^*$ is very similar. However, since the classifiers are all highly accurate, the average size of the recommended subsets under $\hat{\alpha}$ and $\alpha^*$ is quite close to one even though $\hat{\alpha}$ is much smaller than in the experiments in synthetic data.

Finally, we also investigate to what extent the amount of calibration and estimation data $m$ we feed into Algorithm 1 influences the expert’s success probability $P[\hat{Y} = Y | \mathcal{C}_\alpha(X)]$ under our system. In contrast with the experiments on synthetic data, we do find that our system needs larger amounts of calibration and estimation data to realize its full potential. For example, while the relative gain in success probability $P[\hat{Y} = Y | \mathcal{C}_{\alpha}(X)]$ with respect to $P[\hat{Y} = Y | \mathcal{C}_{\alpha'}(X)]$ is just 0.26 ± 0.22\% under $m = 200$, it raises to 3.18 ± 0.12\% under $m = 1,500$. However, this is not surprising since the classifiers are very accurate and thus the optimal $\alpha^*$ is very small. Therefore, we need larger calibration sets to have enough granularity to distinguish among small $\alpha$ values.

### 8 Conclusions

In this work, we have initiated the development of automated decision support systems that, by design, do not require human experts to understand when to trust them to provably improve their performance. In particular, we have focused on multiclass classification tasks and build upon the theory of conformal prediction to design a system that, for each data sample, recommends a subset of labels to the experts using a classifier. Moreover, we have shown both theoretically and empirically that our system can help experts make
predictions more accurately and is robust to the performance of the classifier it relies on—the competitive advantage it provides improves with the accuracy of the classifier and the experts do not decrease their performance by using the system even if the classifier is inaccurate.

Our work opens up many interesting avenues for future work. For example, we have considered a simple conformal score function from the literature. It would be interesting to develop score functions especially designed to improve the performance of our system. Moreover, to estimate the expert’s success probability, we have introduced a simple procedure that builds upon the multinomial logit model (MNL), a classical model from the discrete choice model literature. However, more research is needed to accurately estimate the expert’s success probability, e.g., using more sophisticated discrete choice models. In this context, it would be also interesting to develop systems that perform online estimation of the expert’s conditional success probability. Finally, it would be important to deploy and evaluate our system on a real-world application with human experts.

Acknowledgements. Gomez-Rodriguez acknowledges support from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No. 945719).

References

[1] Wei Jiao, Gurnit Atwal, Paz Polak, Rosa Karlic, Edwin Cuppen, Alexandra Danyi, Jeroen de Ridder, Carla van Herpen, Martijn P. Lolkema, Neeltje Steeghs, et al. A deep learning system accurately classifies primary and metastatic cancers using passenger mutation patterns. Nature communications, 11(1):1–12, 2020.

[2] Ruishan Liu, Shenra Rizzo, Samuel Whipple, Navdeep Pal, Arturo Lopez Pineda, Michael Lu, Brandon Arnieri, Ying Lu, William Capra, Ryan Copping, et al. Evaluating eligibility criteria of oncology trials using real-world data and ai. Nature, 592(7855):629–633, 2021.

[3] Nina Grgić-Hlača, Christoph Engel, and Krishna P Gummadi. Human decision making with machine assistance: An experiment on bailing and jailing. Proceedings of the ACM on Human-Computer Interaction, 3(CSCW):1–25, 2019.

[4] Gagan Bansal, Besmira Nushi, Ece Kamar, Walter S. Lasecki, Daniel S. Weld, and Eric Horvitz. Beyond accuracy: The role of mental models in human-ai team performance. Proceedings of the AAAI Conference on Human Computation and Crowdsourcing, 7(1):2–11, Oct. 2019.

[5] Brian Lubars and Chenhao Tan. Ask not what ai can do, but what ai should do: Towards a framework of task delegability. Advances in Neural Information Processing Systems, 32:57–67, 2019.

[6] Sebastian Bordt and Ulrike von Luxburg. When humans and machines make joint decisions: A non-symmetric bandit model. arXiv preprint arXiv:2007.04800, 2020.

[7] Harini Suresh, Natalie Lao, and Ilaria Liccardi. Misplaced trust: Measuring the interference of machine learning in human decision-making. In 12th ACM Conference on Web Science, pages 315–324, 2020.

[8] Maithra Raghu, Katy Blumer, Greg Corrado, Jon Kleinberg, Ziad Obermeyer, and Sendhil Mullainathan. The algorithmic automation problem: Prediction, triage, and human effort. arXiv preprint arXiv:1903.12220, 2019.

[9] Mahsan Nourani, Joanie T. King, and Eric D. Ragan. The role of domain expertise in user trust and the impact of first impressions with intelligent systems. ArXiv, abs/2008.09100, 2020.

[10] Ming Yin, Jennifer Wortman Vaughan, and Hanna Wallach. Understanding the effect of accuracy on trust in machine learning models. In Proceedings of the 2019 chi conference on human factors in computing systems, pages 1–12, 2019.
[11] Yunfeng Zhang, Q Vera Liao, and Rachel KE Bellamy. Effect of confidence and explanation on accuracy and trust calibration in ai-assisted decision making. In Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, pages 295–305, 2020.

[12] Peter Wright and Fredrick Barbour. Phased decision strategies: Sequels to an initial screening. Graduate School of Business, Stanford University, 1977.

[13] Lee Roy Beach. Broadening the definition of decision making: The role of prechoice screening of options. Psychological Science, 4(4):215–220, 1993.

[14] Moshe Ben-Akiva and Bruno Boccara. Discrete choice models with latent choice sets. International Journal of Research in Marketing, 12(1):9–24, 1995.

[15] Ariel Levy, Monica Agrawal, Arvind Satyanarayan, and David Sontag. Assessing the impact of automated suggestions on decision making: Domain experts mediate model errors but take less initiative. In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, 2021.

[16] Anastasios N. Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. arXiv preprint arXiv:2107.07511, 2021.

[17] Vladimir Vovk, Alex Gammerman, and Glenn Shafer. Algorithmic Learning in a Random World. Springer-Verlag, Berlin, Heidelberg, 2005.

[18] Vineeth Balasubramanian, Shen-Shyang Ho, and Vladimir Vovk. Conformal prediction for reliable machine learning: theory, adaptations and applications. Newnes, 2014.

[19] Chirag Gupta, Aleksandr Podkopaev, and Aaditya Ramdas. Distribution-free binary classification: prediction sets, confidence intervals and calibration. In Advances in Neural Information Processing Systems, 2020.

[20] Yaniv Romano, Evan Patterson, and Emmanuel Candes. Conformalized quantile regression. Advances in Neural Information Processing Systems, 32:3543–3553, 2019.

[21] Yaniv Romano, Matteo Sesia, and Emmanuel Candes. Classification with valid and adaptive coverage. Advances in Neural Information Processing Systems, 33:3581–3591, 2020.

[22] Anastasios Angelopoulos, Stephen Bates, Jitendra Malik, and Michael I Jordan. Uncertainty sets for image classifiers using conformal prediction. In International Conference on Learning Representations, 2021.

[23] Aleksandr Podkopaev and Aaditya Ramdas. Distribution-free uncertainty quantification for classification under label shift. In Proceedings of the 37th Conference on Uncertainty in Artificial Intelligence, 2021.

[24] Abir De, Paramita Koley, Niloy Ganguly, and Manuel Gomez-Rodriguez. Regression under human assistance. In Proceedings of the AAAI Conference on Artificial Intelligence, 2020.

[25] Abir De, Nastaran Okati, Ali Zarezade, and Manuel Gomez-Rodriguez. Classification under human assistance. In Proceedings of the AAAI Conference on Artificial Intelligence, 2021.

[26] Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert. In International Conference on Machine Learning, pages 7076–7087, 2020.

[27] Nastaran Okati, Abir De, and Manuel Gomez-Rodriguez. Differentiable learning under triage. In Advances in Neural Information Processing Systems, 2021.

[28] Vahid Balazadeh Meresht, Abir De, Adish Singla, and Manuel Gomez-Rodriguez. Learning to switch between machines and humans. arXiv preprint arXiv:2002.04258, 2020.
[29] Eleni Straitouri, Adish Singla, Vahid Balazadeh Meresht, and Manuel Gomez-Rodriguez. Reinforcement learning under algorithmic triage. arXiv preprint arXiv:2109.11328, 2021.

[30] Florian Heiss. Discrete choice methods with simulation. Taylor & Francis, 2016.

[31] Gavin Kerrigan, Padhraic Smyth, and Mark Steyvers. Combining human predictions with model probabilities via confusion matrices and calibration. arXiv preprint arXiv:2109.14591, 2021.

[32] Joshua C. Peterson, Ruairidh M. Battleday, Thomas L. Griffiths, and Olga Russakovsky. Human uncertainty makes classification more robust. arXiv preprint arXiv:1908.07086, 2019.

[33] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[34] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[35] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In European conference on computer vision, pages 630–645. Springer, 2016.

[36] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2261–2269, 2017.
A Proof of Proposition 1

Based on Eq. 5, we have that:
\[ P[\hat{Y} = Y \mid \mathcal{C}_\alpha(X)] \geq (1 - \alpha)(P[\hat{Y} = Y \mid \mathcal{C}_\alpha(X), Y \in \mathcal{C}_\alpha(X)]), \]
and hence whenever:
\[ (1 - \alpha)(P[\hat{Y} = Y \mid \mathcal{C}_\alpha(X), Y \in \mathcal{C}_\alpha(X)]) > P[\hat{Y} = Y \mid Y], \]
we will have that:
\[ P[\hat{Y} = Y \mid \mathcal{C}_\alpha(X)] \geq (1 - \alpha)(P[\hat{Y} = Y \mid \mathcal{C}_\alpha(X), Y \in \mathcal{C}_\alpha(X)]) > P[\hat{Y} = Y \mid Y], \]
and hence:
\[ P[\hat{Y} = Y \mid \mathcal{C}_\alpha(X)] > P[\hat{Y} = Y \mid Y]. \]

B Proof of Proposition 2

Given the estimators of \( P[\hat{Y} = Y \mid \mathcal{C}_\alpha(X), Y \in \mathcal{C}_\alpha(X)] \) we have that \( \forall \alpha \in \mathcal{A} \) with probability at least 1 - \( \delta \)
\[ |\hat{\mu}_\alpha - P[\hat{Y} = Y \mid \mathcal{C}_\alpha(X), Y \in \mathcal{C}_\alpha(X)]| \leq \epsilon_\alpha. \] (13)

From Eq. 5 it holds that
\[ P[\hat{Y} = Y \mid \mathcal{C}_\alpha(X)] \geq (1 - \alpha)P[\hat{Y} = Y \mid \mathcal{C}_\alpha(X), Y \in \mathcal{C}_\alpha(X)]. \]

From Eq. 13 with probability at least 1 - \( \delta \) the above becomes:
\[ P[\hat{Y} = Y \mid \mathcal{C}_\alpha(X)] \geq (1 - \alpha)P[\hat{Y} = Y \mid \mathcal{C}_\alpha(X), Y \in \mathcal{C}_\alpha(X)] \]
\[ \geq (1 - \alpha)(\hat{\mu}_\alpha - \epsilon_\alpha). \] (14)

Let \( \hat{\alpha} = \arg\max_{\alpha \in \mathcal{A}} \{(1 - \alpha)(\hat{\mu}_\alpha - \epsilon_\alpha)\} \). For \( \hat{\alpha} \) Eq. 14 becomes
\[ P[\hat{Y} = Y \mid \mathcal{C}_\alpha(X)] \geq (1 - \hat{\alpha})(\hat{\mu}_{\hat{\alpha}} - \epsilon_{\hat{\alpha}}) \]
\[ \geq (1 - \hat{\alpha})(\hat{\mu}_{\hat{\alpha}} - \epsilon_{\hat{\alpha}}) \]
\[ = (1 - \hat{\alpha})(\hat{\mu}_{\hat{\alpha}} - \epsilon_{\hat{\alpha}}) + 2(1 - \hat{\alpha})\epsilon_\alpha - 2(1 - \hat{\alpha})\epsilon_\alpha \]
\[ = (1 - \hat{\alpha})(\hat{\mu}_{\hat{\alpha}} + \epsilon_{\hat{\alpha}}) - 2(1 - \hat{\alpha})\epsilon_\alpha, \forall \alpha \in \mathcal{A}. \]*

To proceed we are going to use Eq. 13 Since the above holds for \( \forall \alpha \in \mathcal{A} \) we need Eq. 13 also to hold \( \forall \alpha \in \mathcal{A} \),
which happens with probability at least 1 - |\( \mathcal{A} \)| \cdot \( \delta = 1 - m \cdot n \cdot \delta \), from union bound. Therefore, we shall choose \( \delta = \frac{\delta'}{m} \) so that the below holds with probability at least 1 - \( \delta' \), \( \forall \alpha \in \mathcal{A} \):
\[ P[\hat{Y} = Y \mid \mathcal{C}_\alpha(X)] \geq (1 - \alpha)(\hat{\mu}_\alpha + \epsilon_\alpha) - 2(1 - \alpha)\epsilon_\alpha \]
\[ \geq (1 - \alpha)P[\hat{Y} = Y \mid \mathcal{C}_\alpha(X), Y \in \mathcal{C}_\alpha(X)] - 2(1 - \alpha)\epsilon_\alpha \]
\[ = \left(1 - \alpha + \frac{1}{m + 1}\right)P[\hat{Y} = Y \mid \mathcal{C}_\alpha(X), Y \in \mathcal{C}_\alpha(X)] \]
\[ - \frac{1}{m + 1}P[\hat{Y} = Y \mid \mathcal{C}_\alpha(X), Y \in \mathcal{C}_\alpha(X)] - 2(1 - \alpha)\epsilon_\alpha \]
\[ \geq \left(1 - \alpha + \frac{1}{m + 1}\right)P[\hat{Y} = Y \mid \mathcal{C}_\alpha(X), Y \in \mathcal{C}_\alpha(X)] - \frac{1}{m + 1} - 2(1 - \alpha)\epsilon_\alpha, \]

because \( P[\hat{Y} = Y \mid \mathcal{C}_\alpha(X), Y \in \mathcal{C}_\alpha(X)] \in [0, 1] \). Finally from Eq. 5
\[ P[\hat{Y} = Y \mid \mathcal{C}_\alpha(X)] \geq P[\hat{Y} = Y \mid \mathcal{C}_\alpha(X)] - 2(1 - \alpha)\epsilon_\alpha - \frac{1}{m + 1}. \]
C Derivation of error expression for Hoeffding’s inequality

From Hoeffding’s inequality we have that:

**Theorem 3** Let $Z_1, ..., Z_k$ be i.i.d., with $Z_i \in [a, b], i = 1, ..., k, a < b$ and $\hat{\mu}$ be the empirical estimate $\hat{\mu} = \frac{\sum_{i=1}^{k} Z_i}{k}$ of $E[Z] = E[Z_i]$. Then:

$$P \left| \hat{\mu} - E[Z] \right| \geq \epsilon \leq \exp \left( -\frac{2k\epsilon^2}{(b-a)^2} \right)$$

(15)

and

$$P \left| \hat{\mu} - E[Z] \right| \leq -\epsilon \leq \exp \left( -\frac{2k\epsilon^2}{(b-a)^2} \right)$$

(16)

hold for all $\epsilon \geq 0$.

In our case we have $k = k_\alpha$, $Z_i = P[\hat{Y} = y_i | C_\alpha(x_i), y_i \in C_\alpha(x_i)]$, $P[\hat{Y} = y_i | C_\alpha(x_i), y_i \in C_\alpha(x_i)] \in (0, 1)$, with $y_i \in C_\alpha(x_i)$, for $i = 1, ..., k_\alpha$. Hence, for the empirical estimate $\hat{\mu} = \hat{\mu}_\alpha$ of $P[\hat{Y} = Y | C_\alpha(X), Y \in C_\alpha(X)]$ and its error $\epsilon = \epsilon_\alpha$:

$$P \left[ \hat{\mu}_\alpha - P[\hat{Y} = Y | C_\alpha(X), Y \in C_\alpha(X)] \geq \epsilon_\alpha \right] \leq \exp \left( -\frac{2k_\alpha\epsilon_\alpha^2}{(1-0)^2} \right)$$

(17)

and

$$P \left[ \hat{\mu}_\alpha - P[\hat{Y} = Y | C_\alpha(X), Y \in C_\alpha(X)] \leq -\epsilon_\alpha \right] \leq \exp \left( -\frac{2k_\alpha\epsilon_\alpha^2}{(1-0)^2} \right)$$

(18)

hold. Further, if we set

$$\delta = \exp \left( -2k_\alpha\epsilon_\alpha^2 \right),$$

(19)

then

$$1 - P \left[ \hat{\mu}_\alpha - P[\hat{Y} = Y | C_\alpha(X), Y \in C_\alpha(X)] \leq \epsilon_\alpha \right] \leq \delta \Rightarrow P \left[ \hat{\mu}_\alpha - P[\hat{Y} = Y | C_\alpha(X), Y \in C_\alpha(X)] \leq \epsilon_\alpha \right] \geq 1 - \delta$$

(20)

and

$$1 - P \left[ \hat{\mu}_\alpha - P[\hat{Y} = Y | C_\alpha(X), Y \in C_\alpha(X)] \geq -\epsilon_\alpha \right] \leq \delta \Rightarrow P \left[ \hat{\mu}_\alpha - P[\hat{Y} = Y | C_\alpha(X), Y \in C_\alpha(X)] \geq -\epsilon_\alpha \right] \geq 1 - \delta$$

(21)

hold for any $\epsilon_\alpha \geq 0$. As follows, based on Eq. (19)

$$\delta = \exp \left( -2k_\alpha\epsilon_\alpha^2 \right) \Rightarrow \log \frac{1}{\delta} = 2k_\alpha\epsilon_\alpha^2 \Rightarrow \epsilon_\alpha^2 = \frac{\log \frac{1}{\delta}}{2k_\alpha} \Rightarrow \epsilon_\alpha = \sqrt{\frac{\log \frac{1}{\delta}}{2k_\alpha}}.$$

We choose $\delta = \frac{\delta'}{mn}$ so that we can find $\hat{\alpha}$ that, with probability at least $1 - \delta'$, maximizes the success probability, as shown in the proof of Proposition 2. As a result:

$$\epsilon_\alpha = \sqrt{\frac{\log \frac{1}{\delta'}}{2k_\alpha}}.$$
D Proof of Theorem 2

Let \( C_{\alpha_1}(X), C_{\alpha_2}(X) \) be a prediction set constructed with standard conformal prediction with coverage \( 1 - \alpha_1 \) and \( 1 - \alpha_2 \) respectively. Based on Theorem 1 it holds that:

\[
1 - \alpha_1 \leq \mathbb{P}[Y \in C_{\alpha_1}(X)] \leq 1 - \alpha_1 + \frac{1}{m+1} \tag{22}
\]

and

\[
1 - \alpha_2 \leq \mathbb{P}[Y \in C_{\alpha_2}(X)] \leq 1 - \alpha_2 + \frac{1}{m+1}. \tag{23}
\]

Since \( \alpha_1 < \alpha_2 \) we have that \( \hat{q}_2 = \left\lfloor \frac{(m+1)(1-\alpha_2)}{m} \right\rfloor < \left\lfloor \frac{(m+1)(1-\alpha_1)}{m} \right\rfloor = \hat{q}_1 \). We know that \( C_{\alpha_2}(X) = \{ y : s(X, y) < \hat{q}_2 \} \) and \( C_{\alpha_1}(X) = \{ y : s(X, y) < \hat{q}_1 \} \). So for any \( X \), it holds that \( C_{\alpha_2}(X) \subset C_{\alpha_1}(X) \). Based on the definition of \( C_{\alpha_1,\alpha_2}(X) \) in [1] we have that:

\[
C_{\alpha_1,\alpha_2}(X) = C_{\alpha_1}(X) \setminus C_{\alpha_2}(X) = \{ y : \hat{q}_2 \leq s(X, y) < \hat{q}_1 \},
\]

and therefore

\[
\mathbb{P}[Y \in C_{\alpha_1,\alpha_2}(X)] = \mathbb{P}[C_{\alpha_1}(X)] - \mathbb{P}[C_{\alpha_2}(X)]. \tag{24}
\]

Taking the upper-bound in Eq. 22 and the lower-bound in Eq. 23

\[
\mathbb{P}[Y \in C_{\alpha_1}(X)] - \mathbb{P}[Y \in C_{\alpha_2}(X)] \leq \alpha_2 - \alpha_1 + \frac{1}{m+1}. \tag{25}
\]

Similarly, taking the lower-bound in Eq. 22 and the upper-bound in Eq. 23

\[
\alpha_2 - \alpha_1 - \frac{1}{m+1} \leq \mathbb{P}[Y \in C_{\alpha_1,\alpha_2}(X)] \leq \alpha_2 - \alpha_1 + \frac{1}{m+1}. \tag{26}
\]

From Eqs. 24, 25 and 26 we have that:

\[
\alpha_2 - \alpha_1 - \frac{1}{m+1} \leq \mathbb{P}[Y \in C_{\alpha_1,\alpha_2}(X)] \leq \alpha_2 - \alpha_1 + \frac{1}{m+1}.
\]

E Proof of Proposition 3

From

\[
1 - \alpha_1 \leq \mathbb{P}[Y \in C_{\alpha_1}(X)] \leq 1 - \alpha_1 + \frac{1}{1+m}, \tag{27}
\]

we have that

\[
\mathbb{P}[\hat{Y} = Y|C_{\alpha_1}(X)] = \mathbb{P}[\hat{Y} = Y|C_{\alpha_1}(X), Y \notin C_{\alpha_1}(X)]\mathbb{P}[Y \notin C_{\alpha_1}(X)]
\]

\[
+ \mathbb{P}[\hat{Y} = Y|C_{\alpha_1}(X), Y \in C_{\alpha_1}(X)]\mathbb{P}[Y \in C_{\alpha_1}(X)]
\]

\[
\leq (1 - \alpha_1 + \frac{1}{1+m})\mathbb{P}[\hat{Y} = Y|C_{\alpha_1}(X), Y \in C_{\alpha_1}(X))],
\]

since \( \mathbb{P}[\hat{Y} = Y|C_{\alpha_1}(X), Y \notin C_{\alpha_1}(X)] = 0 \). For \( C_{\alpha_1,\alpha_2} \), it also holds that

\[
\alpha_2 - \alpha_1 - \frac{1}{m+1} \leq \mathbb{P}[Y \in C_{\alpha_1,\alpha_2}(X)] \leq \alpha_2 - \alpha_1 + \frac{1}{m+1}. \tag{28}
\]
and, as a direct consequence, we have that

$$
P[\hat{Y} = Y | \mathcal{C}_{\alpha_1, \alpha_2}(X)] = P[\hat{Y} = Y | \mathcal{C}_{\alpha_1, \alpha_2}(X), Y \notin \mathcal{C}_{\alpha_1, \alpha_2}(X)]P[Y \notin \mathcal{C}_{\alpha_1, \alpha_2}(X)] + P[\hat{Y} = Y | \mathcal{C}_{\alpha_1, \alpha_2}(X), Y \in \mathcal{C}_{\alpha_1, \alpha_2}(X)]P[Y \in \mathcal{C}_{\alpha_1, \alpha_2}(X)] \geq (\alpha_2 - \alpha_1 - \frac{1}{m+1})P[\hat{Y} = Y | \mathcal{C}_{\alpha_1, \alpha_2}(X), Y \in \mathcal{C}_{\alpha_1, \alpha_2}(X)],
$$

where we have also used that $P[\hat{Y} = Y | \mathcal{C}_{\alpha_1, \alpha_2}(X), Y \notin \mathcal{C}_{\alpha_1, \alpha_2}(X)] = 0$. Then, it holds that

$$
P[\hat{Y} = Y | \mathcal{C}_{\alpha_1, \alpha_2}(X)] > P[\hat{Y} = Y | \mathcal{C}_{\alpha_1}(X)], \quad \text{(29)}
$$
as long as

$$
(\alpha_2 - \alpha_1 - \frac{1}{m+1})P[\hat{Y} = Y | \mathcal{C}_{\alpha_1, \alpha_2}(X), Y \in \mathcal{C}_{\alpha_1, \alpha_2}(X)] > (1 - \alpha_1 + \frac{1}{1 + m})P[\hat{Y} = Y | \mathcal{C}_{\alpha_1}(X), Y \in \mathcal{C}_{\alpha_1}(X)].
$$

\section{Proof of Proposition 4}

Given the estimators of $P[\hat{Y} = Y | \mathcal{C}_{\alpha_1, \alpha_2}(X), Y \in \mathcal{C}_{\alpha_1, \alpha_2}(X)]$ we have that $\forall \alpha_1, \alpha_2 \in \mathcal{A}, \alpha_1 < \alpha_2$ with probability at least $1 - \delta$

$$
|\hat{\mu}_{\alpha_1, \alpha_2} - P[\hat{Y} = Y | \mathcal{C}_{\alpha_1, \alpha_2}(X), Y \in \mathcal{C}_{\alpha_1, \alpha_2}(X)]| \leq \epsilon_{\alpha_1, \alpha_2}.
$$

From Eq. (3) and Theorem 2 it holds that

$$
\left(\alpha_2 - \alpha_1 - \frac{1}{m+1}\right)P[\hat{Y} = Y | \mathcal{C}_{\alpha_1, \alpha_2}(X), Y \in \mathcal{C}_{\alpha_1, \alpha_2}(X)] \leq P[\hat{Y} = Y | \mathcal{C}_{\alpha_1, \alpha_2}(X)] \leq \left(\alpha_2 - \alpha_1 + \frac{1}{m+1}\right)P[\hat{Y} = Y | \mathcal{C}_{\alpha_1, \alpha_2}(X), Y \in \mathcal{C}_{\alpha_1, \alpha_2}(X)]. \quad \text{(31)}
$$

From the left-hand side of the above and Eq. (30) we have that

$$
P[\hat{Y} = Y | \mathcal{C}_{\alpha_1, \alpha_2}(X)] \geq \left(\alpha_2 - \alpha_1 - \frac{1}{m+1}\right)P[\hat{Y} = Y | \mathcal{C}_{\alpha_1, \alpha_2}(X), Y \in \mathcal{C}_{\alpha_1, \alpha_2}(X)] \geq \left(\alpha_2 - \alpha_1 - \frac{1}{m+1}\right)(\hat{\mu}_{\alpha_1, \alpha_2} - \epsilon_{\alpha_1, \alpha_2}).
$$

(32)

Let $\hat{\alpha}_1, \hat{\alpha}_2 = \arg\max_{\alpha_1, \alpha_2 \in \mathcal{A} | \alpha_1 < \alpha_2} \left\{\left(\alpha_2 - \alpha_1 - \frac{1}{m+1}\right)(\hat{\mu}_{\alpha_1, \alpha_2} - \epsilon_{\alpha_1, \alpha_2})\right\}$. For $\hat{\alpha}_1, \hat{\alpha}_2$, Eq. (32) becomes

$$
P[\hat{Y} = Y | \mathcal{C}_{\hat{\alpha}_1, \hat{\alpha}_2}(X)] \geq \left(\hat{\alpha}_2 - \hat{\alpha}_1 - \frac{1}{m+1}\right)(\hat{\mu}_{\hat{\alpha}_1, \hat{\alpha}_2} - \epsilon_{\hat{\alpha}_1, \hat{\alpha}_2}) \geq \left(\alpha_2 - \alpha_1 - \frac{1}{m+1}\right)(\hat{\mu}_{\alpha_1, \alpha_2} - \epsilon_{\alpha_1, \alpha_2}) = \left(\alpha_2 - \alpha_1 - \frac{1}{m+1}\right)(\hat{\mu}_{\alpha_1, \alpha_2} - \epsilon_{\alpha_1, \alpha_2}) + 2 \left(\alpha_2 - \alpha_1 - \frac{1}{m+1}\right)\epsilon_{\alpha_1, \alpha_2} - 2 \left(\alpha_2 - \alpha_1 - \frac{1}{m+1}\right)\epsilon_{\alpha_1, \alpha_2} = \left(\alpha_2 - \alpha_1 - \frac{1}{m+1}\right)(\hat{\mu}_{\alpha_1, \alpha_2} + \epsilon_{\alpha_1, \alpha_2}) - 2 \left(\alpha_2 - \alpha_1 - \frac{1}{m+1}\right)\epsilon_{\alpha_1, \alpha_2}, \forall \alpha \in \mathcal{A}
$$
To proceed further, we are going to use Eq. 13. Since the above holds for \( \forall \alpha \in \mathcal{A} \), we need Eq. 13 also to hold \( \forall \alpha \in \mathcal{A} \), which happens with probability at least \( 1 - \left| \{ \alpha_1, \alpha_2 \in \mathcal{A} | \alpha_1 < \alpha_2 \} \right| \cdot \delta \) from a union bound. Therefore, we shall choose \( \delta = \frac{\delta'}{|\{ \alpha_1, \alpha_2 \in \mathcal{A} | \alpha_1 < \alpha_2 \}|} \) so that the inequality below holds with probability at least \( 1 - \delta' \), \( \forall \alpha_1, \alpha_2 \in \mathcal{A}, \alpha_1 < \alpha_2 \):

\[
P[\hat{Y} = Y | C_{\hat{\alpha}_1, \hat{\alpha}_2}(X)] \geq \left( \alpha_2 - \alpha_1 - \frac{1}{m + 1} \right) (\hat{\mu}_{\alpha_1, \alpha_2} + \epsilon_{\alpha_1, \alpha_2}) - 2 \left( \alpha_2 - \alpha_1 - \frac{1}{m + 1} \right) \epsilon_{\alpha_1, \alpha_2}
\]

\[
\geq \left( \alpha_2 - \alpha_1 - \frac{1}{m + 1} \right) P[\hat{Y} = Y | C_{\alpha_1, \alpha_2}(X), Y \in C_{\alpha_1, \alpha_2}(X)] - 2 \left( \alpha_2 - \alpha_1 - \frac{1}{m + 1} \right) \epsilon_{\alpha_1, \alpha_2}
\]

\[
= \left( \alpha_2 - \alpha_1 + \frac{1}{m + 1} \right) P[\hat{Y} = Y | C_{\alpha_1, \alpha_2}(X)]
\]

\[
- \frac{2}{m + 1} P[\hat{Y} = Y | C_{\alpha_1, \alpha_2}(X), Y \in C_{\alpha_1, \alpha_2}(X)] - 2 \left( \alpha_2 - \alpha_1 - \frac{1}{m + 1} \right) \epsilon_{\alpha_1, \alpha_2}
\]

\[
\geq \left( \alpha_2 - \alpha_1 + \frac{1}{m + 1} \right) P[\hat{Y} = Y | C_{\alpha_1, \alpha_2}(X)]
\]

\[
- \frac{2}{m + 1} - 2 \left( \alpha_2 - \alpha_1 - \frac{1}{m + 1} \right) \epsilon_{\alpha_1, \alpha_2},
\]

because \( P[\hat{Y} = Y | C_{\alpha_1, \alpha_2}(X), Y \in C_{\alpha_1, \alpha_2}(X)] \in [0, 1] \). Finally, from Eq. 31, we can conclude that

\[
P[\hat{Y} = Y | C_{\hat{\alpha}_1, \hat{\alpha}_2}(X)] \geq P[\hat{Y} = Y | C_{\alpha_1, \alpha_2}(X)] - \frac{2}{m + 1} - 2 \left( \alpha_2 - \alpha_1 - \frac{1}{m + 1} \right) \epsilon_{\alpha_1, \alpha_2}.
\]
G Implementation Details

To implement our algorithms and run all the experiments on synthetic and real data, we used PyTorch 1.8.1, NumPy 1.20.1 and Scikit-learn 1.0.2 on Python 3.7.3. For reproducibility, we use a fixed random seed in all random procedures. Moreover, we set $\delta = 0.1$ everywhere.

**Synthetic prediction tasks.** We create $4 \times 3 = 12$ different prediction tasks, where we vary the number of labels $n \in \{10, 50, 100\}$ and the level of difficulty of the task. More specifically, for each value of $n$, we create four different tasks of increasing difficulty where the success probability of the logistic regression classifier is $\mathbb{P}[Y' = Y] = 0.9, 0.7, 0.5$ and 0.3, respectively.

To create each task, we use the function `make_classification` of the Scikit-learn library. This function allows the creation of data for synthetic prediction tasks with very particular user-defined characteristics, through the generation of clusters of normally distributed points on the vertices of a multidimensional hypercube. The number of the dimensions of the hypercube indicates the number of informative features of each sample, which in our case we set at 15 for all prediction tasks. Linear combinations of points, i.e., the informative features, are used to create redundant features, the number of which we set at 5. The difficulty of the prediction task is controlled through the size of the hypercube, with a multiplicative factor, namely `clas_sep`, which we tuned accordingly for each value $n$ so that the success probability of the logistic regression classifier above spans a wide range of values across tasks. All the selected values of this parameter can be found in the configuration file `config.py` in the code. Finally, we set the proportion of the samples assigned to each label, i.e., the function parameter `weights`, using a Dirichlet distribution of order $n$ with parameters $\alpha_1 = ... = \alpha_n = 1$. 


Additional Experiments on Synthetic Data

To complement the results in Table 1 in the main paper, we experiment with additional prediction tasks with different number of labels $n$ and amount of calibration and estimation data $m$. For each value of $n$ and $m$, we estimate the average success probability $\mathbb{P}[\hat{Y} = Y | C_{\hat{\alpha}_1, \hat{\alpha}_2}(X)]$ during test for four different experts using our system, each with a different success probability $\mathbb{P}[\hat{Y} = Y | Y]$, on four prediction tasks where the classifier achieves a different success probability $\mathbb{P}[Y' = Y]$. Figure 4 summarizes the results.

| $\mathbb{P}[\hat{Y} = Y | Y]$ | $\mathbb{P}[Y' = Y]$ | | $\mathbb{P}[\hat{Y} = Y | Y]$ | $\mathbb{P}[Y' = Y]$ |
|-----------------------------|---------------------|-----------------------------|---------------------|
| 0.3 | 0.56 | 0.72 | 0.85 | 0.93 | | 0.3 | 0.63 | 0.76 | 0.88 | 0.95 |
| 0.5 | 0.67 | 0.80 | 0.89 | 0.96 | | 0.5 | 0.73 | 0.84 | 0.91 | 0.97 |
| 0.7 | 0.79 | 0.87 | 0.93 | 0.97 | | 0.7 | 0.82 | 0.90 | 0.95 | 0.98 |
| 0.9 | 0.91 | 0.95 | 0.97 | 0.99 | | 0.9 | 0.93 | 0.95 | 0.98 | 0.99 |

(a) $n = 50, m = 1,200$

| $\mathbb{P}[\hat{Y} = Y | Y]$ | $\mathbb{P}[Y' = Y]$ | | $\mathbb{P}[\hat{Y} = Y | Y]$ | $\mathbb{P}[Y' = Y]$ |
|-----------------------------|---------------------|-----------------------------|---------------------|
| 0.3 | 0.40 | 0.57 | 0.75 | 0.88 | | 0.3 | 0.56 | 0.72 | 0.84 | 0.92 |
| 0.5 | 0.55 | 0.68 | 0.80 | 0.90 | | 0.5 | 0.67 | 0.80 | 0.89 | 0.94 |
| 0.7 | 0.72 | 0.79 | 0.88 | 0.93 | | 0.7 | 0.79 | 0.88 | 0.94 | 0.97 |
| 0.9 | 0.90 | 0.92 | 0.95 | 0.97 | | 0.9 | 0.92 | 0.95 | 0.97 | 0.99 |

(c) $n = 10, m = 400$

| $\mathbb{P}[\hat{Y} = Y | Y]$ | $\mathbb{P}[Y' = Y]$ | | $\mathbb{P}[\hat{Y} = Y | Y]$ | $\mathbb{P}[Y' = Y]$ |
|-----------------------------|---------------------|-----------------------------|---------------------|
| 0.3 | 0.63 | 0.76 | 0.88 | 0.94 | | 0.3 | 0.76 | 0.88 | 0.94 | 0.97 |
| 0.5 | 0.73 | 0.84 | 0.92 | 0.96 | | 0.5 | 0.84 | 0.92 | 0.96 | 0.99 |
| 0.7 | 0.83 | 0.90 | 0.95 | 0.92 | | 0.7 | 0.90 | 0.95 | 0.96 | 0.98 |
| 0.9 | 0.93 | 0.96 | 0.98 | 0.96 | | 0.9 | 0.96 | 0.98 | 0.96 | 0.99 |

(e) $n = 100, m = 400$

Figure 4: Average success probability $\mathbb{P}[\hat{Y} = Y | C_{\hat{\alpha}_1, \hat{\alpha}_2}(X)]$ during test for four different experts using our system, each with a different success probability $\mathbb{P}[\hat{Y} = Y | Y]$, on four prediction tasks where the classifier achieves a different success probability $\mathbb{P}[Y' = Y]$. Each table corresponds to a different number of label values $n$ and calibration and estimation set size $m$. For readability, each cell shows only the average since the standard errors are all below $10^{-2}$. 