Improving Expert Predictions with Prediction Sets

Eleni Straitouri\(^1\), Lequn Wang\(^2\), Nastaran Okati\(^1\), and Manuel Gomez Rodriguez\(^1\)

\(^1\)Max Planck Institute for Software Systems, \{estraitouri, nastaran, manuelgr\}@mpi-sws.org
\(^2\)Cornell 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 each of their recommendations is accurate to improve their performance. To this end, we focus on multiclass classification tasks and consider an automated decision support system that, for each data sample, uses a classifier to recommend a subset of labels to a human expert. We first show that, by looking at the design of such a system 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 with high probability. Then, we develop an efficient and near-optimal search method to find the target probability value under which the expert benefits the most from using our system. 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 (Jiao et al., 2020) and drug discovery (Liu et al., 2021) to candidate screening (Wang et al., 2022) and criminal justice (Grgić-Haća et al., 2019), to name a few. 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 (Bansal et al., 2019; Lubars and Tan, 2019; Bordt and von Luxburg, 2020).

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 (Suresh et al., 2020). This follows from the fact that, in general, the accuracy of a classifier differs across data samples (Raghu et al., 2019). In this context, several recent studies have analyzed how factors such as model confidence, model explanations and overall model calibration modulate trust (Papenmeier et al., 2019; Wang and Yin, 2021; Vodrahalli et al., 2022). Unfortunately, it is not yet clear how to make sure that the experts do not develop a misplaced trust that decreases their performance (Nourani et al., 2020; Yin et al., 2019; Zhang et al., 2020). In this work, our goal is to develop decision support systems that, by design, do not require experts to understand when the systems’ recommendations are accurate to 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 (Wright and Barbour, 1977; Beach, 2006).
In a way, such support systems would help experts by automatically narrowing down their options for them, decreasing their cognitive load and allowing them to focus their attention where it is most needed. This could be particularly useful when the task is tedious or requires domain knowledge since it is difficult to outsource the task and domain experts are often a scarce resource. In the context of clinical text annotation\(^1\), a recent empirical study has also concluded that, in terms of the overall accuracy, it may be more beneficial to recommend a subset of options than a single option (Levy et al., 2021).

Here, one could still argue that the expert needs 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. (2021). This is largely due to the fact that the accuracy of the classifier differs across different data samples. To circumvent this, we use the theory of conformal prediction (Vovk et al., 2005; Angelopoulos and Bates, 2021) to construct trustworthy subsets where the probability that the true label belongs to a recommended subset always matches almost exactly a given target probability value, without making any distributional assumptions about the data or the classifier. In addition, given an estimator of the expert’s success probability for any of the recommended subsets, we develop an efficient and near-optimal search method to find the target probability value under which the expert is guaranteed to achieve the greatest accuracy with high probability. In this context, we also propose a practical method to obtain such an 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 72%. Finally, by using our system, our results suggest that the (average) expert would reduce their misclassification probability by \(\sim 80\%\). \(^2\)

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

There exist three fundamental notions of distribution-free uncertainty quantification in the literature: calibration, confidence intervals, and prediction sets (Vovk et al., 2005; Balasubramanian et al., 2014; Gupta et al., 2020; Angelopoulos and Bates, 2021). Our work is most closely related to the rapidly increasing literature on prediction sets (Romano et al., 2019, 2020; Angelopoulos et al., 2021; Podkopaev and Ramdas, 2021), however, to the best of our knowledge, prediction sets have not been optimized to serve automated decision support systems such as ours. In this context, we acknowledge that Babbar et al. (2022) have also proposed using prediction sets in decision support systems, however, this work is contemporary to ours and has been carried out independently. Moreover, in contrast to our work, for each data sample, they allow the expert to predict label values outside the recommended subset and do not optimize the probability that the true label belongs to the subset. As a result, their method is not directly comparable to ours.

There is an extensive line of work on reliable or cautious classification (Del Coz et al., 2009; Yang et al., 2017; Mortier et al., 2021; Ma and Denoeux, 2021; Nguyen and Hüllermeier, 2021; Liu et al., 2014). Reliable classification aims to develop models that can provide set-valued predictions to account for the prediction uncertainty of a classifier. However, in this line of work, there are no human experts who make the final predictions given the set-valued predictions, in contrast with our work. Moreover, the provided set-valued predictions typically lack distribution-free guarantees.

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

---

\(^1\)Clinical text annotation is a task where medical experts aim to identify clinical concepts in medical notes and map them to labels in a large ontology.

\(^2\)An open-source implementation of our system is available at [https://github.com/Networks-Learning/improve-expert-predictions-conformal-prediction](https://github.com/Networks-Learning/improve-expert-predictions-conformal-prediction).
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 $y \in Y$ to a subset of them $C(x)$ using the 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 $C(x)$ according to a policy $\pi(x, C(x))$.

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 $y \sim P(Y \mid X)$. Then, our goal is to design an automated decision support system $C : \mathcal{X} \rightarrow 2^\mathcal{Y}$ 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 scores for each class (e.g., softmax scores). The higher the score $\hat{f}_y(x)$, the more the classifier believes the true label $Y = y$. Here, we assume that, for each $x \sim P(X)$, the human expert predicts a label $\hat{y}$ among those in the subset $C(x)$ according to an unknown policy $\pi(x, C(x))$. More formally, $\hat{y} \sim \pi(x, C(x))$, where $\pi : \mathcal{X} \times 2^\mathcal{Y} \rightarrow \Delta(\mathcal{Y})$ and $\Delta(\mathcal{Y})$ denotes the probability simplex over the set of labels $\mathcal{Y}$, and $\pi_y(x, C(x)) = 0$ if $y \notin C(x)$. Refer to Figure 1 for an illustration of the automated decision support system we consider.

Ideally, we would like that, by design, the expert can only benefit from using the automated decision support system $C$, i.e.,

$$
P[\hat{Y} = Y \mid C] \geq P[\hat{Y} = Y \mid \mathcal{Y}],$$

where $P[\hat{Y} = Y \mid C]$ denotes the expert’s success probability if, for each $x \sim P(X)$, the human expert predicts a label $\hat{Y}$ among those in the subset $C(x)$. 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. For example, a system that always recommends $C(x) = \mathcal{Y}$ for all $x \in \mathcal{X}$ satisfies Eq. 1. However, it is useless to the experts. Therefore, among those systems satisfying Eq. 1, we would like to find the system $C^*$

---

3We denote random variables with capital letters and realizations of random variables with lower case letters.
4The assumption that $\hat{f}(x) \in [0, 1]^n$ is without loss of generality.
that helps the experts achieve the highest success probability\footnote{Note that maximizing the expert’s success probability $\mathbb{P}[\hat{Y} = Y ; C]$ is equivalent to minimizing the expected 0-1 loss $\mathbb{E}[(\hat{Y} \neq Y) ; C]$. Considering other types of losses is left as an interesting venue for future work.} \emph{i.e.},
\[
C^* = \arg \max_C \mathbb{P}[\hat{Y} = Y ; C].
\] (2)

To address the design of such a system, we will look at the problem from the perspective of conformal prediction \cite{Vovk2005,Angelopoulos_2021}.

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 $C(X)$. In what follows, 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 $P[Y \in C(X)]$ that the true label $Y$ belongs to the subset $C(X) = C_\alpha(X)$ will match almost exactly a given target probability $1 - \alpha$ with high probability, without making any distributional assumptions about the data distribution $P(X)\, | \, Y$ or the classifier $\hat{f}$.

Let $\mathcal{D}_{cal} = \{(x_i, y_i)\}_{i=1}^m$ be a calibration set, where $(x_i, y_i) \sim P(X)\, P(Y \mid X)$, $s(x_i, y_i) = 1 - \hat{f}_{q_i}(x_i)$ be the \emph{conformal score} \footnote{In general, the conformal score $s(x, y)$ can be any function of $x$ and $y$ measuring the similarity between samples.} \emph{i.e.}, if the classifier is catastrophically wrong, the conformal score will be close to one, and $\hat{q}_\alpha$ be the $\frac{(m+1)(1-\alpha)}{m}$ empirical quantile of the conformal scores $s(x_1, y_1), \ldots, s(x_m, y_m)$. Moreover, use the quantile $\hat{q}_\alpha$ to construct the subsets $C_\alpha(X)$ for new data samples as follows:
\[
C_\alpha(X) = \{y \mid s(X, y) \leq \hat{q}_\alpha\}. \tag{3}
\]

Then, as long as the size $m$ of the calibration set is sufficiently large, the probability that the true label $Y$ belongs to the subset $C_\alpha(X)$ conditionally on the calibration set $\mathcal{D}_{cal}$ is almost exactly $1 - \alpha$ with high probability. More specifically, we first note that the coverage probability is a random quantity whose distribution is given by the following proposition (refer to Appendix A.5 in \cite{Hulsmann2022} for a proof):

\begin{proposition}
For a decision support system $C_\alpha$ that constructs the subsets $C_\alpha(X)$ using Eq. 3 it holds that
\[
\mathbb{P}[Y \in C_\alpha(X) \mid \mathcal{D}_{cal}] \sim \text{Beta}(\lceil(m + 1)(1 - \alpha)\rceil, \lceil(m + 1)\alpha\rceil)
\] (4)
\end{proposition}

as long as the conformal scores $s(x_i, y_i)$ for all $(x_i, y_i) \in \mathcal{D}_{cal}$ are almost surely distinct.

Then, using the definition of the beta distribution, we have that
\[
1 - \alpha \leq E[\mathbb{P}[Y \in C_\alpha(X) \mid \mathcal{D}_{cal}]] = 1 - \frac{(m + 1)\alpha}{m + 1} \leq 1 - \alpha + \frac{1}{m + 1}.
\]

As an immediate consequence, given a target probability $1 - \alpha$ and tolerance values $\delta, \epsilon \in (0, 1)$, we can compute the minimum size $m$ of the calibration set $\mathcal{D}_{cal}$ such that $C_\alpha$ enjoys Probably Approximately Correct (PAC) coverage guarantees, \emph{i.e.}, with probability $1 - \delta$, it holds that
\[
1 - \alpha - \epsilon \leq \mathbb{P}[Y \in C_\alpha(X) \mid \mathcal{D}_{cal}] \leq 1 - \alpha + \epsilon.
\]

While the above coverage guarantee is valid for any choice of $\alpha$ value, we would like to emphasize that there may be some $\alpha$ values that will lead to larger gains in terms of success probability $\mathbb{P}[\hat{Y} = Y ; C_\alpha]$ than others. Therefore, in what follows, our goal is to find the optimal $\alpha^*$ that maximizes the expert’s success probability given a calibration set $\mathcal{D}_{cal}$.

\textbf{Remark.} Most of the literature on conformal prediction focuses on the following conformal calibration guarantee (refer to Appendix D in \cite{Angelopoulos_2021} for a proof):
Theorem 1 For an automated decision support system $C_\alpha$ that constructs the subsets $C_\alpha(X)$ using Eq. 3, it holds that

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

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

However, to afford the above marginal guarantee in our work, we would be unable to optimize the $\alpha$ value to maximize the expert’s success probability given a calibration set $D_{cal}$. This is because the guarantee requires that $\alpha$ and $D_{cal}$ are independent.

4 Optimizing Across Conformal Predictors

We start by realizing that, given a calibration set $D = \{ (x_i, y_i) \}_{i=1}^m$, there only exist $m$ different conformal predictors. This is because the empirical quantile $\hat{q}_\alpha$, which the subsets $C_\alpha(x_i)$ depend on, can only take $m$ different values. As a result, to find the optimal conformal predictor that maximizes the expert’s success probability, we need to solve the following maximization problem:

$$\alpha^* = \arg\max_{\alpha \in A} P[\hat{Y} = Y; C_\alpha],$$

(5)

where $A = \{ \alpha_i \}_{i \in [m]}$, with $\alpha_i = 1 - i/(m + 1)$, and the probability is only over the randomness in the samples the system helps predicting.

However, to find a near optimal solution $\hat{\alpha}$ to the above problem, we need to estimate the expert’s success probability $P[\hat{Y} = Y; C_\alpha]$. Assume for now that, for each $\alpha \in A$, we have access to an estimator $\hat{\mu}_\alpha$ of the expert’s success probability such that, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$, it holds that $|\hat{\mu}_\alpha - P[\hat{Y} = Y; C_\alpha]| \leq \epsilon_{\alpha, \delta}$. Then, we can use the following proposition to find a near optimal solution $\hat{\alpha}$ to Eq. 5 with high probability:

Proposition 2 For any $\delta \in (0, 1)$, consider an automated decision support system $C_\alpha$ with

$$\hat{\alpha} = \arg\max_{\alpha \in A} \hat{\mu}_\alpha - \epsilon_{\alpha, \delta/m}. $$

(6)

With probability at least $1 - \delta$, it holds that $P[\hat{Y} = Y; C_\alpha] \geq P[\hat{Y} = Y; C_\alpha'] - 2\epsilon_{\alpha, \delta/m} \forall \alpha \in A$ simultaneously.

More specifically, the above result directly implies that for any $\delta \in (0, 1)$, with probability at least $1 - \delta$, it holds that:

$$P[\hat{Y} = Y; C_\alpha^*] - P[\hat{Y} = Y; C_\alpha] \leq 2\epsilon_{\alpha, \delta/m}. $$

(7)

Here, note that the above guarantees do not make use of the marginal guarantees afforded by conformal prediction—the guarantees hold for any parameterized set-value predictor.

In what follows, we propose a practical method to estimate the expert’s success probability $P[\hat{Y} = Y; C_\alpha]$ that builds upon the multinomial logit model (MNL), one of the most popular models in the vast literature on discrete choice models (Heiss 2016). More specifically, given a sample $(x, y)$, we assume that the expert’s conditional success probability for the subset $C_\alpha(x)$ is given by:

$$P[\hat{Y} = y; C_\alpha | y \in C_\alpha(x)] = \frac{e^{u_{yy}}}{\sum_{y' \in C_\alpha(x)} e^{u_{yy'}}}, $$

(8)

where $u_{yy'}$ denotes the expert’s preference for label value $y' \in Y$ whenever the true label is $y$. In the language of discrete choice models, one can view the true label $y$ as the context in which the expert chooses among alternatives (Tversky and Simonson 1993). In Appendix H, we consider and experiment with a more
Algorithm 1 Finding a near-optimal $\hat{\alpha}$

1: Input: $\hat{f}$, $\mathcal{D}_{\text{est}}$, $\mathcal{D}_{\text{cal}}$, $\delta$, $m$
2: Initialize: $\mathcal{A} = \{\}$, $\hat{\alpha} \leftarrow 0$, $t \leftarrow 0$
3: for $i = 1, \ldots, m$ do
4: $\alpha \leftarrow 1 - \frac{i}{m}$
5: $\mathcal{A} \leftarrow \mathcal{A} \cup \{\alpha\}$
6: end for
7: for $\alpha \in \mathcal{A}$ do
8: $\hat{\mu}_\alpha, \epsilon_{\alpha,\delta}/m \leftarrow \text{ESTIMATE}(\alpha, \delta, \mathcal{D}_{\text{est}}, \mathcal{D}_{\text{cal}}, \hat{f})$
9: if $t \leq \hat{\mu}_\alpha - \epsilon_{\alpha,\delta}/m$ then
10: $t \leftarrow \hat{\mu}_\alpha - \epsilon_{\alpha,\delta}/m$
11: $\hat{\alpha} \leftarrow \alpha$
12: end if
13: end for
14: return $\hat{\alpha}$

expressive context that, in addition to the true label, distinguishes between different levels of difficulty across data samples.

Further, to estimate the parameters $u_{yy'}$, we assume 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. (2021), i.e.,

$$C = [C_{yy'}]_{y,y' \in \mathcal{Y}},$$

where $C_{yy'} = \mathbb{P}[\hat{Y} = y' ; \mathcal{Y} | Y = y]$, and naturally set $u_{yy'} = \log C_{yy'}$. Then, we can compute a Monte-Carlo estimator $\hat{\mu}_\alpha$ of the expert’s success probability $\mathbb{P}[\hat{Y} = Y ; C_\alpha]$ using the above conditional success probability $\mathbb{P}[\hat{Y} = Y ; C_\alpha | y \in C_\alpha(x)]$ and an estimation set $\mathcal{D}_{\text{est}} = \{(x_i, y_i)\}_{i \in [m]}$, i.e.,

$$\hat{\mu}_\alpha = \frac{1}{m} \sum_{i \in [m] \mid y_i \in C_\alpha(x_i)} \mathbb{P}[\hat{Y} = y_i ; C_\alpha | y_i \in C_\alpha(x_i)].$$  \hspace{1cm} (9)

Finally, for each $\alpha \in \mathcal{A}$, using Hoeffding’s inequality\footnote{The number of samples in $\mathcal{D}_{\text{cal}}$ and $\mathcal{D}_{\text{est}}$ can differ. For simplicity, we assume both sets contain $m$ samples.} we can conclude that, with probability at least $1 - \delta$, it holds that (refer to Appendix A.2):

$$|\hat{\mu}_\alpha - \mathbb{P}[\hat{Y} = Y ; C_\alpha]| \leq \sqrt{\frac{\log \frac{1}{\delta}}{2m}} = \epsilon_{\alpha,\delta}. \hspace{1cm} (10)$$

As a consequence, as $m \to \infty$, $\epsilon_{\alpha,\delta}$ converges to zero. This directly implies that the near-optimal $\hat{\alpha}$ converges to the true optimal $\alpha^*$ and that, with probability at least $1 - \delta$, our system $C_{\alpha^*}$ satisfies Eq. (11) asymptotically with respect to the number of samples $m$ in the estimation set.

Algorithm\footnote{By using Hoeffding’s inequality, we derive a fairly conservative constant error bound for all $\alpha$ values, however, we have experimentally found that, even with a relatively small amount of estimation and calibration data, our algorithm identifies near-optimal $\hat{\alpha}$ values providing large gains as shown in Figures 2(a) and 3(a) and Tables 1 and 2. That being said, one could use tighter concentration inequalities such as Hoeffding–Bentkus and Waudby-Smith–Ramdas (Bates et al. 2021).} summarizes the overall search method, where the function $\text{ESTIMATE}(\cdot)$ uses Eqs. (9) and (10).

The algorithm first builds $\mathcal{A}$ and then finds the near optimal $\hat{\alpha}$ in $\mathcal{A}$. To build $\mathcal{A}$, it needs $\mathcal{O}(m)$ steps. To find the near-optimal $\hat{\alpha}$, for each value $\alpha \in \mathcal{A}$ and each sample $(x_i, y_i) \in \mathcal{D}_{\text{est}}$, it needs to compute a subset $C_\alpha(x)$. This is achieved by sorting the conformal scores and reusing computations across $\alpha$ values, which takes $\mathcal{O}(m \log m + mn \log n)$ steps. Therefore, the overall time complexity is $\mathcal{O}(m \log m + mn \log n)$.

Remark. By using the MNL, we implicitly assume the independence of irrelevant alternatives (IIA)\footnote{By using Hoeffding’s inequality, we derive a fairly conservative constant error bound for all $\alpha$ values, however, we have experimentally found that, even with a relatively small amount of estimation and calibration data, our algorithm identifies near-optimal $\hat{\alpha}$ values providing large gains as shown in Figures 2(a) and 3(a) and Tables 1 and 2. That being said, one could use tighter concentration inequalities such as Hoeffding–Bentkus and Waudby-Smith–Ramdas (Bates et al. 2021).} [Luce 1959], an axiom that states that the expert’s relative preference between two alternatives remains the same
over all possible subsets containing these alternatives. While IIA is one of the most widely used axioms in the literature on discrete choice models, there is also a large body of experimental literature claiming to document real-world settings where IIA fails to hold (Simonson [1989], Tversky [1972], Huber et al. [1982]). In Appendix E, we study the robustness of our results to violations of the IIA assumption in the estimator of the expert’s success probability. Looking into the future, it would be very interesting to use more sophisticated discrete choice models that do not rely on the IIA assumption to accurately estimate the expert’s success probability.

5 Beyond Standard Conformal Prediction

Until now, we have used standard conformal prediction (Angelopoulos and Bates, 2021) to construct the recommended subsets \( C(X) \)—we have constructed \( C(X) \) by comparing the conformal scores \( s(X, y) \) to a single threshold \( \hat{q} \), as shown in Eq. [3]. Here, we will show that we can sometimes improve the performance of our system by constructing \( C(X) \) using two thresholds \( \hat{q}_{\alpha_1} \) and \( \hat{q}_{\alpha_2} \). 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 \( D_{\text{cal}} \). This will be useful in scenarios where the classifier underpinning our system has not particularly high average accuracy.

More specifically, given a calibration set \( D_{\text{cal}} = \{ (x_i, s_i) \}_{i=1}^m \), let \( \alpha_1, \alpha_2 \in [0, 1] \), with \( \alpha_1 < \alpha_2 \), and \( \hat{q}_{\alpha_1} \) and \( \hat{q}_{\alpha_2} \) be the \( \lfloor (m+1)(1-\alpha_1) \rfloor \) and \( \lceil (m+1)(1-\alpha_2) \rceil \) empirical quantiles of the conformal scores \( s(x_1, y_1), \ldots, s(x_m, y_m) \). Moreover, use the quantiles \( \hat{q}_{\alpha_1} \) and \( \hat{q}_{\alpha_2} \) to construct the subsets \( C_{\alpha_1,\alpha_2}(X) \) for new data samples as follows:

\[
C_{\alpha_1,\alpha_2}(X) = \{ y \mid \hat{q}_{\alpha_1} < s(X, y) \leq \hat{q}_{\alpha_2} \}.
\]

(11)

Then, as long as the size \( m \) of the calibration set is sufficiently large, the probability that the true label \( Y \) belongs to the subset \( C_{\alpha_1,\alpha_2}(X) \) conditionally on the calibration set \( D_{\text{cal}} \) is almost exactly \( \alpha_2 - \alpha_1 \) with high probability. More specifically, we first note that the coverage probability is a random quantity whose distribution is given by the following proposition, which is the counterpart of Proposition 1.

**Proposition 3** For a decision support system \( C_{\alpha_1,\alpha_2} \) that constructs \( C_{\alpha_1,\alpha_2}(X) \) using Eq. [11] as long as the conformal scores \( s(x_i, y_i) \) for all \( (x_i, y_i) \in D_{\text{cal}} \) are almost surely distinct, it holds that:

\[
P[Y \in C_{\alpha_1,\alpha_2}(X) \mid D_{\text{cal}}] \sim \text{Beta}(l, m - l + 1),
\]

(12)

where \( l = \lfloor (m+1)(1-\alpha_1) \rfloor - \lceil (m+1)(1-\alpha_2) \rceil \).

Then, using the definition of the beta distribution, we have that

\[
\alpha_2 - \alpha_1 \leq E[P[Y \in C_{\alpha_1,\alpha_2}(X) \mid D_{\text{cal}}]] = \alpha_2 - \alpha_1 + \frac{c_1 - c_2}{m+1} \leq \alpha_2 - \alpha_1 + \frac{1}{m+1},
\]

where \( c_1, c_2 \in [0, 1] \). As an immediate consequence, given a target probability \( \alpha_2 - \alpha_1 \) and tolerance values \( \delta, \epsilon \in (0, 1) \), we can compute the minimum size \( m \) of the calibration set \( D_{\text{cal}} \) such that \( C_{\alpha_1,\alpha_2} \) enjoys Probably Approximately Correct (PAC) coverage guarantees, i.e., with probability \( 1 - \delta \), it holds that

\[
\alpha_2 - \alpha_1 - \epsilon \leq P[Y \in C_{\alpha}(X) \mid D_{\text{cal}}] \leq \alpha_2 - \alpha_1 + \epsilon.
\]

Finally, given an estimator of the expert’s success probability \( \hat{\mu}_{\alpha_1,\alpha_2} \) such that for each \( \alpha_1 < \alpha_2 \) and \( \delta \in (0, 1) \), with probability at least \( 1 - \delta \), it holds that \( |\hat{\mu}_{\alpha_1,\alpha_2} - P[Y = Y; C_{\alpha_1,\alpha_2}]| \leq \epsilon_{\alpha_1,\alpha_2,\delta} \), we can proceed similarly as in standard conformal prediction to find the near optimal \( \hat{\alpha}_1, \hat{\alpha}_2 \in \mathcal{A} \) that maximizes the expert’s success probability with high probability, by using \( \hat{\mu}_{\alpha_1,\alpha_2} \) and \( \epsilon_{\alpha_1,\alpha_2,2\delta/(m(m-1))} \). Here, it is worth pointing out that,\footnote{In such scenarios, the conformal scores of the samples in the calibration set can occasionally have low values—otherwise, the classifier would be highly accurate—and thus it is beneficial to exclude label values with (very) low conformal scores from the recommended subsets—those label values the classifier is confidently wrong about.}
Table 1: Empirical success probability achieved by four different experts using our system during test, each with a different success probability $P[\hat{Y} = Y; Y]$, on four prediction tasks where the classifier achieves a different success probability $P[Y' = Y]$. Each column corresponds to a prediction task and each row to an expert. In each task, the number of label values $n = 10$ and the size of the calibration and estimation sets is $m = 1,200$. Each cell shows only the average since the standard errors are all below $10^{-2}$.

| $\mathbb{P}[\hat{Y} = Y; Y]$ | $\mathbb{P}[Y' = Y]$ |
|-----------------------------|---------------------|
| 0.3                         | 0.3                 |
| 0.5                         | 0.5                 |
| 0.7                         | 0.7                 |
| 0.9                         | 0.9                 |
| 0.41                        | 0.58                |
| 0.68                        | 0.80                |
| 0.72                        | 0.79                |
| 0.90                        | 0.91                |

in contrast with the case of standard conformal prediction, the time complexity of finding the near optimal $\hat{\alpha}_1$ and $\hat{\alpha}_2$ is $O(m \log m + mn \log n + mn^2)$. Moreover, we can still rely on the practical method to estimate the expert’s conditional success probability introduced in Section 4.

Remarks. Conformal prediction is one of many possible ways to construct set-valued predictors (Chzhen et al., 2021), i.e., predictors that, for each sample $x \in \mathcal{X}$, output a set of label candidates $\mathcal{C}(x)$. In our work, we favor conformal predictors over alternatives because they provably output trustworthy sets $\mathcal{C}_d(x)$ without making any assumption about the data distribution nor the classifier they rely upon. In fact, we can use conformal predictors with any off-the-shelf classifier. However, we would like to emphasize that our efficient search method (Algorithm 1) is rather generic and, together with an estimator of the expert’s success probability, may be used to find a near-optimal set-valued predictor within a discrete set of set-valued predictors that maximizes the expert’s success probability. This is because our near-optimal guarantees in Proposition 2 do not make use of the marginal guarantees afforded by conformal prediction, as discussed previously. We hope our work will encourage others to develop set-valued predictors specifically designed to serve decision support systems.

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. Appendices D, E and F contain additional experiments where we analyze the sensitivity of our system to the choice of the calibration set, analyze the robustness of our results to violations of the IIA assumption in the estimator of the expert’s success probability and benchmark our system against a top-k set-valued predictor which, for each data sample, returns the $k$ label values with the highest score.

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 B 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, which we use to estimate the performance of our system, 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.

For each prediction task, we train a logistic regression model $P_\theta(Y' | X)$, which depending on the difficulty of the prediction task, achieves different success probability values $P[Y' = Y]$. Moreover, we sample the expert’s predictions $\hat{Y}$ from the multinomial logit model defined by Eq. 8, with $C_{yy} = \frac{\pi}{n} \pm \gamma \epsilon$, and $C_{yy'} = \frac{1-\epsilon}{n} \pm \beta$, where $\pi$ is a parameter that controls the expert’s success probability $P[\hat{Y} = Y; Y]$, $\epsilon \sim U(0, \min(1 - \frac{\pi}{n}, \frac{\pi}{n}))$.

10All 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 B for further details.
Figure 2: Empirical success probability achieved by two different experts using our system, each with a different success probability $P[\hat{Y} = Y; Y]$ and average size of the recommended sets during test for each $\alpha \in A$ on two synthetic prediction tasks where the classifier achieves a different success probability $P[Y' = Y]$. Here, note that the empirical average set size only depends on the classifier’s success probability $P[Y' = Y]$, not the expert, and thus we only need two lines. In all experiments, the number of label values $n = 10$ and the size of the calibration and estimation sets is $m = 1,200$. Each marker corresponds to a different $\alpha$ value and the darker points correspond to $\hat{\alpha}$. The coloring of the darker points for each prediction task is the same in both panels.

Experts always benefit from our system even if the classifier has low accuracy. We estimate the success probability $P[\hat{Y} = Y; C_{\hat{\alpha}_1,\hat{\alpha}_2}]$ achieved by four different experts, each with a different success probability $P[\hat{Y} = Y; Y]$, on four prediction tasks where the classifier achieves a different success probability $P[Y' = Y]$. Table I summarizes the results, where each column corresponds to a different prediction task and each row corresponds to a different expert. We find that, using our system, the expert solves the prediction task significantly more accurately than the expert or the classifier on their own. Moreover, it is rather remarkable that, even if the classifier has low accuracy, the expert always benefits from using our system—in other words, our system is robust to the performance of the classifier it relies on. We found qualitatively similar results for prediction tasks with other values of $n$ and $m$, which we report in Appendix C. Since we found that, in the majority of repetitions of each experiment, the near-optimal $\hat{\alpha}_2 = 1$, in what follows, we only experiment with systems $C_{\hat{\alpha}} = C_{\hat{\alpha}_1}$ that construct $C_{\hat{\alpha}}(X)$ using Eq. 3.

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 $P[\hat{Y} = Y; Y]$ 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 $P[\hat{Y} = Y; Y]$, the smaller the near optimal $\hat{\alpha}$ and thus the greater the average size of the subsets $C_{\hat{\alpha}}(X)$. Moreover, in Appendix C we also show that, the smaller the near optimal $\hat{\alpha}$, the greater the spread of the empirical distribution of the size of the subsets $C_{\hat{\alpha}}(X)$. We found qualitatively 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
Table 2: Empirical success probabilities achieved by three popular deep neural network classifiers and by an expert using our system with these classifiers during test on the CIFAR-10H dataset. The size of the calibration and estimation sets is $m = 1,500$ and the expert’s empirical success probability at solving the (original) multiclass task is $P[\hat{Y} = Y; Y] \approx 0.947$. Each cell shows only the average since the standard errors are all below $10^{-2}$.

| Classifier          | Expert using $C_{\alpha}$ |
|---------------------|-----------------------------|
| ResNet-110          | 0.928                       |
| PreResNet-110       | 0.944                       |
| DenseNet            | 0.964                       |

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 $P[\hat{Y} = Y; C_{\alpha}]$ for each task. Our results suggest that the relative gain in success probability, averaged across experts with $P[\hat{Y} = Y; Y] \in \{0.3, 0.5, 0.7, 0.9\}$, increases with the number of label values. For example, for $m = 400$, it goes from $48.36 \pm 4.50\%$ for $n = 10$ to $69.44 \pm 5.20\%$ 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. 3 because, whenever the classifiers are highly accurate, systems $C_{\alpha,\alpha_2}$ with $\alpha_2 \neq 1$ do not offer a competitive advantage. Similarly as in experiments on synthetic
data, we analyze the robustness of our results to violations of the IIA assumption in the estimator of the expert’s success probability and benchmark our system against a top-$k$ set-valued predictor in Appendices D and E, respectively. Appendices F and H contain additional experiments where we analyze the sensitivity of our system to the choice of the calibration set and we experiment with a more sophisticated estimator of the expert’s success probability that accounts for different levels of difficulty among data samples.

**Data description and experimental setup.** We experiment with the dataset CIFAR-10H [Peterson et al., 2019], which contains 10,000 natural images taken from the test set of the standard CIFAR-10 [Krizhevsky et al., 2009]. Each of these images belongs to one of $n = 10$ classes and contains approximately 50 expert predictions $\hat{Y}_i^{11}$. 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 use the test set to estimate the performance of our system and 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 (He et al., 2016a), PreResNet-110 (He et al., 2016b) and DenseNet (Huang et al., 2017). Moreover, we use the human predictions $\hat{Y}$ to estimate the confusion matrix $C$ for the expert predictions in the (original) multiclass classification task (Kerrigan et al., 2021) and then sample the expert’s prediction $\hat{Y}$ from the multinomial logit model defined by Eq. 8 to both estimate the expert’s conditional success probabilities in Eq. 9 in Algorithm 1 and estimate the expert’s success probability during testing.

**Results.** We start by estimating the success probability $P[\hat{Y} = Y ; C_{\alpha}]$ achieved by 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 an expert using our system and by the corresponding classifier on its own. In the caption, we also report the (empirical) success probability achieved by an expert solving the (original) multiclass task in their own. We find it 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 72.2% (DenseNet) to 81.9% (ResNet-110). Moreover, by using our system, our results suggest that the (average) expert would reduce their misclassification probability by $\sim$80%.

Next, for each choice of classifier, we compare the performance of our system under the near optimal $\alpha$ 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 $\alpha$ and $\alpha^*$ is very similar. However, since the classifiers are all highly accurate, the average size of the recommended subsets under $\alpha$ and $\alpha^*$ is quite close to one even though $\alpha$ 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 $P[\hat{Y} = Y ; C_{\alpha}]$ under our system. Similarly as in experiments on synthetic data, we find that our system can perform well with a relatively small amount of calibration and estimation data. For example, the relative gain in empirical success probability achieved by an expert using our system with respect to an expert on their own just raises from 3.38 ± 0.05% under $m = 200$, to 3.80 ± 0.03% under $m = 1,500$.

**8 Conclusions**

We have initiated the development of automated decision support systems that, by design, do not require human experts to understand when each of their recommendations is accurate to improve their performance with high probability. In particular, we have focused on multiclass classification tasks and designed a system that, for each data sample, recommends a subset of labels to the experts using a classifier. Moreover, we have shown that our system can help experts make predictions more accurately and is robust to the performance

---

11The 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 methodology is rather general, our system may help improving expert predictions in other applications.
of the classifier it relies on.

Our work opens up many interesting avenues for future work. For example, it would be valuable to
develop score functions especially designed to improve the performance of our system. Moreover, it would be
interesting to develop systems that perform online estimation of the expert’s conditional success probability.
In addition, it would be important to investigate the ethical impact of our decision support system, including
human trust and bias, and consider alternative performance metrics such as expert prediction time. 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

Angelopoulos, A., Bates, S., Malik, J., and Jordan, M. I. (2021). Uncertainty sets for image classifiers using
conformal prediction. In International Conference on Learning Representations.

Angelopoulos, A. N. and Bates, S. (2021). A gentle introduction to conformal prediction and distribution-free
uncertainty quantification. arXiv preprint arXiv:2107.07511.

Babbar, V., Bhatt, U., and Weller, A. (2022). On the utility of prediction sets in human-ai teams. arXiv
preprint arXiv:2205.01411.

Balasubramanian, V., Ho, S.-S., and Vovk, V. (2014). Conformal prediction for reliable machine learning:
three adaptation and applications. Newnes.

Bansal, G., Nushi, B., Kamar, E., Lasecki, W. S., Weld, D. S., and Horvitz, E. (2019). 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.

Bates, S., Angelopoulos, A., Lei, L., Malik, J., and Jordan, M. (2021). Distribution-free, risk-controlling
prediction sets. J. ACM, 68(6).

Beach, L. R. (1993). Broadening the definition of decision making: The role of prechoice screening of options.
Psychological Science, 4(4):215–220.

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

Bordt, S. and von Luxburg, U. (2020). When humans and machines make joint decisions: A non-symmetric
bandit model. arXiv preprint arXiv:2007.04800.

Chzhen, E., Denis, C., Hebiri, M., and Lorieul, T. (2021). Set-valued classification—overview via a unified
framework. arXiv preprint arXiv:2102.12318.

De, A., Koley, P., Ganguly, N., and Gomez-Rodriguez, M. (2020). Regression under human assistance. In
Proceedings of the AAAI Conference on Artificial Intelligence.

De, A., Okati, N., Zarezade, A., and Gomez-Rodriguez, M. (2021). Classification under human assistance. In
Proceedings of the AAAI Conference on Artificial Intelligence.

Del Coz, J. J., Díez, J., and Bahamonde, A. (2009). Learning nondeterministic classifiers. Journal of Machine
Learning Research, 10(10).

Grgić-Hlača, N., Engel, C., and Gummadri, K. P. (2019). Human decision making with machine assistance: An
experiment on bailing and jailing. Proceedings of the ACM on Human-Computer Interaction, 3(CSCW):1–25.
Gupta, C., Podkopaev, A., and Ramdas, A. (2020). Distribution-free binary classification: prediction sets, confidence intervals and calibration. In *Advances in Neural Information Processing Systems*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016a). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

He, K., Zhang, X., Ren, S., and Sun, J. (2016b). Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer.

Heiss, F. (2016). *Discrete choice methods with simulation*. Taylor & Francis.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269.

Huber, J., Payne, J. W., and Puto, C. (1982). Adding asymmetrically dominated alternatives: Violations of regularity and the similarity hypothesis. *Journal of consumer research*, 9(1):90–98.

Hulsman, R. (2022). Distribution-free finite-sample guarantees and split conformal prediction. *arXiv preprint arXiv:2210.14735*.

Jiao, W., Atwal, G., Polak, P., Karlic, R., Cuppen, E., Danyi, A., de Ridder, J., van Herpen, C., Lolkema, M. P., Steeghs, N., et al. (2020). A deep learning system accurately classifies primary and metastatic cancers using passenger mutation patterns. *Nature communications*, 11(1):1–12.

Kerrigan, G., Smyth, P., and Steyvers, M. (2021). Combining human predictions with model probabilities via confusion matrices and calibration. *arXiv preprint arXiv:2109.14591*.

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images. *Citeseer*.

Levy, A., Agrawal, M., Satyanarayan, A., and Sontag, D. (2021). 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*.

Liu, R., Rizzo, S., Whipple, S., Pal, N., Pineda, A. L., Lu, M., Arnieri, B., Lu, Y., Capra, W., Copping, R., et al. (2021). Evaluating eligibility criteria of oncology trials using real-world data and ai. *Nature*, 592(7855):629–633.

Liu, Z.-G., Pan, Q., Dezert, J., and Mercier, G. (2014). Credal classification rule for uncertain data based on belief functions. *Pattern Recognition*, 47(7):2532–2541.

Lubars, B. and Tan, C. (2019). 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.

Luce, R. D. (1959). On the possible psychophysical laws. *Psychological review*, 66(2):81.

Ma, L. and Denoeux, T. (2021). Partial classification in the belief function framework. *Knowledge-Based Systems*, 214:106742.

Meresht, V. B., De, A., Singla, A., and Gomez-Rodriquez, M. (2022). Learning to switch among agents in a team. *Transactions on Machine Learning Research*.

Mortier, T., Wydmuch, M., Dembczyński, K., Hüllermeier, E., and Waegeman, W. (2021). Efficient set-valued prediction in multi-class classification. *Data Mining and Knowledge Discovery*, 35(4):1435–1469.

Mozannar, H. and Sontag, D. (2020). Consistent estimators for learning to defer to an expert. In *International Conference on Machine Learning*, pages 7076–7087.
Nguyen, V.-L. and Hüllermeier, E. (2021). Multilabel classification with partial abstention: Bayes-optimal prediction under label independence. *Journal of Artificial Intelligence Research*, 72:613–665.

Nourani, M., King, J. T., and Ragan, E. D. (2020). The role of domain expertise in user trust and the impact of first impressions with intelligent systems. *ArXiv*, abs/2008.09100.

Okati, N., De, A., and Gomez-Rodriguez, M. (2021). Differentiable learning under triage. In *Advances in Neural Information Processing Systems*.

Papenmeier, A., Englebienne, G., and Seifert, C. (2019). How model accuracy and explanation fidelity influence user trust. *arXiv preprint arXiv:1907.12652*.

Peterson, J. C., Battleday, R. M., Griffiths, T. L., and Russakovsky, O. (2019). Human uncertainty makes classification more robust. *arXiv preprint arXiv:1908.07086*.

Podkopaev, A. and Ramdas, A. (2021). Distribution-free uncertainty quantification for classification under label shift. In *Proceedings of the 37th Conference on Uncertainty in Artificial Intelligence*.

Raghu, M., Blumer, K., Corrado, G., Kleinberg, J., Obermeyer, Z., and Mullainathan, S. (2019). The algorithmic automation problem: Prediction, triage, and human effort. *arXiv preprint arXiv:1903.12220*.

Romano, Y., Patterson, E., and Candes, E. (2019). Conformalized quantile regression. *Advances in Neural Information Processing Systems*, 32:3543–3553.

Romano, Y., Sesia, M., and Candes, E. (2020). Classification with valid and adaptive coverage. *Advances in Neural Information Processing Systems*, 33:3581–3591.

Simonson, I. (1989). Choice based on reasons: The case of attraction and compromise effects. *Journal of consumer research*, 16(2):158–174.

Straitouri, E., Singla, A., Meresht, V. B., and Gomez-Rodriguez, M. (2021). Reinforcement learning under algorithmic triage. *arXiv preprint arXiv:2109.11328*.

Suresh, H., Lao, N., and Liccardi, I. (2020). Misplaced trust: Measuring the interference of machine learning in human decision-making. In *12th ACM Conference on Web Science*, pages 315–324.

Tversky, A. (1972). Elimination by aspects: A theory of choice. *Psychological review*, 79(4):281.

Tversky, A. and Simonson, I. (1993). Context-dependent preferences. *Management Science*, 39(10):1179–1189.

Vodrahalli, K., Gerstenberg, T., and Zou, J. (2022). Uncalibrated models can improve human-ai collaboration. In *Advances in Neural Information Processing Systems*.

Vovk, V., Gammerman, A., and Shafer, G. (2005). *Algorithmic Learning in a Random World*. Springer-Verlag, Berlin, Heidelberg.

Wang, L., Joachims, T., and Gomez-Rodriguez, M. (2022). Improving screening processes via calibrated subset selection. In *Proceedings of the 39th International Conference on Machine Learning*.

Wang, X. and Yin, M. (2021). Are explanations helpful? a comparative study of the effects of explanations in ai-assisted decision-making. In *26th International Conference on Intelligent User Interfaces*, pages 318–328.

Wright, P. and Barbour, F. (1977). *Phased decision strategies: Sequels to an initial screening*. Graduate School of Business, Stanford University.

Yang, G., Destercke, S., and Masson, M.-H. (2017). Cautious classification with nested dichotomies and imprecise probabilities. *Soft Computing*, 21(24):7447–7462.
Yin, M., Wortman Vaughan, J., and Wallach, H. (2019). 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.

Zhang, Y., Liao, Q. V., and Bellamy, R. K. (2020). 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.
A Proofs

A.1 Proof of Proposition 2

Given the estimators $\hat{\mu}_\alpha$ of $\mathbb{P}[\hat{Y} = Y; C_\alpha]$, we have that, for each $\alpha \in \mathcal{A}$, it holds that

$$\left| \hat{\mu}_\alpha - \mathbb{P}[\hat{Y} = Y; C_\alpha] \right| \leq \epsilon_{\alpha,\delta/m}$$

(13)

with probability at least $1 - \delta/m$. By applying the union bound, we know that the above events hold simultaneously for all $\alpha \in \mathcal{A}$ with probability at least $1 - \delta$. Moreover, by rearranging, the above expression can be rewritten as

$$\hat{\mu}_\alpha - \epsilon_{\alpha,\delta/m} \leq \mathbb{P}[\hat{Y} = Y; C_\alpha] \leq \hat{\mu}_\alpha + \epsilon_{\alpha,\delta/m}.$$  \hspace{1cm} (14)

Let $\hat{\alpha} = \arg\max_{\alpha \in \mathcal{A}} \{ (\hat{\mu}_\alpha - \epsilon_{\alpha,\delta/m}) \}$. For $\hat{\alpha}$, with probability $1 - \delta$, it holds that for all $\alpha \in \mathcal{A}$,

$$\mathbb{P}[\hat{Y} = Y; C_{\hat{\alpha}}] \geq \hat{\mu}_{\hat{\alpha}} - \epsilon_{\hat{\alpha},\delta/m} \geq \hat{\mu}_\alpha - \epsilon_{\alpha,\delta/m} = \hat{\mu}_\alpha - \epsilon_{\alpha,\delta/m} + 2\epsilon_{\alpha,\delta/m} - 2\epsilon_{\alpha,\delta/m} \geq \mathbb{P}[\hat{Y} = Y; C_\alpha] - 2\epsilon_{\alpha,\delta/m},$$

where the last inequality follows from Eq. 14.

A.2 Derivation of Error Expression for Hoeffding’s Inequality

From Hoeffding’s inequality we have that:

**Theorem 2** Let $Z_1, \ldots, Z_k$ be i.i.d., with $Z_i \in [a, b], i = 1, \ldots, k, a < b$ and $\hat{\mu}$ be the empirical estimate $\hat{\mu} = \frac{1}{k} \sum_{i=1}^{k} Z_i$ of $\mathbb{E}[Z] = \mathbb{E}[Z_i]$. Then:

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

(15)

and

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

(16)

hold for all $\epsilon \geq 0$.

In our case we have $k = m$, $Z_i = \mathbb{P}[\hat{Y} = Y_i | C_\alpha(X_i), Y_i] \in (0, 1)$. Hence, for the empirical estimate $\hat{\mu} = \hat{\mu}_\alpha$ of $\mathbb{P}[\hat{Y} = Y; C_\alpha]$ and its error $\epsilon = \epsilon_{\alpha,\delta}$:

$$\mathbb{P}\left[ \hat{\mu}_\alpha - \mathbb{P}[\hat{Y} = Y; C_\alpha] \geq \epsilon_{\alpha,\delta} \right] \leq \exp \left( -\frac{2m\epsilon_{\alpha,\delta}^2}{(1-0)^2} \right)$$

(17)

and

$$\mathbb{P}\left[ \hat{\mu}_\alpha - \mathbb{P}[\hat{Y} = Y; C_\alpha] \leq -\epsilon_{\alpha,\delta} \right] \leq \exp \left( -\frac{2m\epsilon_{\alpha,\delta}^2}{(1-0)^2} \right)$$

(18)

hold. Further, if we set

$$\delta = \exp \left( -2m\epsilon_{\alpha,\delta}^2 \right),$$

(19)
then
\[ 1 - \mathbb{P} \left[ \hat{\mu}_\alpha - \mathbb{P}[\hat{Y} = Y; C_\alpha] \leq \epsilon_{\alpha, \delta} \right] \leq \delta \Rightarrow \mathbb{P} \left[ \hat{\mu}_\alpha - \mathbb{P}[\hat{Y} = Y; C_\alpha] \leq \epsilon_{\alpha, \delta} \right] \geq 1 - \delta \] (20)

and
\[ 1 - \mathbb{P} \left[ \hat{\mu}_\alpha - \mathbb{P}[\hat{Y} = Y; C_\alpha] \geq -\epsilon_{\alpha, \delta} \right] \leq \delta \Rightarrow \mathbb{P} \left[ \hat{\mu}_\alpha - \mathbb{P}[\hat{Y} = Y; C_\alpha] \geq -\epsilon_{\alpha, \delta} \right] \geq 1 - \delta \] (21)

hold for any \( \epsilon_{\alpha, \delta} \geq 0 \). As follows, based on Eq. 19:

\[ \delta = \exp \left( -2m\epsilon^2_{\alpha, \delta} \right) \Rightarrow \log \frac{1}{\delta} = 2m\epsilon^2_{\alpha, \delta} \Rightarrow \epsilon_{\alpha, \delta} = \sqrt{\frac{\log \frac{1}{\delta}}{2m}}. \]

A.3 Proof of Proposition 3

We proceed similarly as in the Appendix A.5 in Hulsman (2022). First, note that, by definition, we have that

\[ \hat{q}_{\alpha_1} = s(\lceil(1-\alpha_1)(m+1)\rceil) \quad \text{and} \quad \hat{q}_{\alpha_2} = s(\lceil(1-\alpha_2)(m+1)\rceil), \]

where \( s(i) \) denotes the \( i \)-th smallest conformal score in the calibration set \( D_{\text{cal}} \). Then, as long as the conformal scores in the calibration set are almost surely distinct, it follows directly from Proposition 4 in Hulsman (2022) that

\[ \mathbb{P} \left[ \hat{q}_{\alpha_2} < s(X,Y) \leq \hat{q}_{\alpha_1} \mid D_{\text{cal}} \right] \sim \text{Beta}(l, m - l + 1), \] (22)

where \( l = \lceil(m+1)(1 - \alpha_1) \rceil - \lceil(m+1)(1 - \alpha_2) \rceil \). Moreover, for any \( (X,Y) \sim P(X)P(Y \mid X) \), we have that, by construction, \( Y \in C_{\alpha_1, \alpha_2}(X) \) if and only if \( s(X,Y) \in (\hat{q}_{\alpha_2}, \hat{q}_{\alpha_1}) \). Then, Eq. 12 follows directly from Eq. 22.
B Implementation Details

To implement our algorithms and run all the experiments on synthetic and real data, we used PyTorch 1.12.1, NumPy 1.20.1 and Scikit-learn 1.0.2 on Python 3.9.2. 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 $\alpha_1 = \ldots = \alpha_n = 1$. 

C Additional Synthetic Prediction Tasks, Number of Labels and Amount of Calibration and Estimation 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 success probability $P[Y = Y; Y]$ achieved by four different experts using our system, each with a different success probability $P[Y = Y; Y']$, on four prediction tasks where the classifier achieves a different success probability $P[Y' = Y]$. Figure 4 summarizes the results.

| $P[Y = Y; Y]$ | $P[Y' = Y]$ | $P[Y = Y]$ | $P[Y' = Y]$ |
|---------------|-------------|-------------|-------------|
| $0.3$         | $0.56$      | $0.72$      | $0.84$      | $0.94$      |
| $0.5$         | $0.68$      | $0.80$      | $0.89$      | $0.95$      |
| $0.7$         | $0.79$      | $0.87$      | $0.93$      | $0.97$      |
| $0.9$         | $0.92$      | $0.95$      | $0.97$      | $0.99$      |

(a) $n = 50$, $m = 1200$

| $P[Y = Y; Y]$ | $P[Y' = Y]$ | $P[Y = Y]$ | $P[Y' = Y]$ |
|---------------|-------------|-------------|-------------|
| $0.3$         | $0.42$      | $0.58$      | $0.75$      | $0.91$      |
| $0.5$         | $0.55$      | $0.66$      | $0.80$      | $0.93$      |
| $0.7$         | $0.72$      | $0.79$      | $0.87$      | $0.96$      |
| $0.9$         | $0.90$      | $0.92$      | $0.94$      | $0.98$      |

(b) $n = 100$, $m = 1200$

| $P[Y = Y; Y]$ | $P[Y' = Y]$ | $P[Y' = Y]$ |
|---------------|-------------|-------------|
| $0.3$         | $0.56$      | $0.73$      | $0.84$      | $0.94$      |
| $0.5$         | $0.67$      | $0.80$      | $0.88$      | $0.96$      |
| $0.7$         | $0.79$      | $0.88$      | $0.93$      | $0.98$      |
| $0.9$         | $0.92$      | $0.94$      | $0.97$      | $0.99$      |

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

| $P[Y = Y; Y]$ | $P[Y' = Y]$ | $P[Y = Y]$ | $P[Y' = Y]$ |
|---------------|-------------|-------------|-------------|
| $0.3$         | $0.62$      | $0.77$      | $0.87$      | $0.95$      |
| $0.5$         | $0.73$      | $0.83$      | $0.91$      | $0.97$      |
| $0.7$         | $0.83$      | $0.89$      | $0.95$      | $0.98$      |
| $0.9$         | $0.93$      | $0.96$      | $0.98$      | $0.99$      |

(d) $n = 50$, $m = 400$

| $P[Y = Y; Y]$ | $P[Y' = Y]$ |
|---------------|-------------|
| $0.3$         | $0.62$      |
| $0.5$         | $0.73$      |
| $0.7$         | $0.83$      |
| $0.9$         | $0.93$      |

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

Figure 4: Empirical success probability achieved by four different experts using our system during test, each with a different success probability $P[Y = Y; Y']$, on four prediction tasks where the classifier achieves a different success probability $P[Y' = Y]$. 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 $10^{-2}$. 

19
D  Sensitivity To The Choice of Calibration Set

In this section, we repeat the experiments on synthetic and real data using 100 independent realizations of the calibration, estimation and test sets. Then, for each data split, we compare the empirical coverage 
\[
\frac{1}{|D_{\text{test}}|} \sum_{(x,y) \in D_{\text{test}}} \mathbb{I}[y \in C_{\hat{\alpha}}(x)] := 1 - \hat{\alpha}_{\text{emp}}
\]
achieved by our system \(C_{\hat{\alpha}}\) on the test set \(D_{\text{test}}\) to the corresponding target coverage \(1 - \hat{\alpha}\).

Figure 5 summarizes the results for (a) one synthetic prediction task and one synthetic expert and (b) one popular deep neural network classifier on the CIFAR-10H dataset. We find that the value of the near-optimal \(\hat{\alpha}\) does not vary significantly across experiments (i.e., across calibration sets) and, for each experiment, the empirical coverage \(1 - \hat{\alpha}_{\text{emp}}\) is very close to and typically higher than the target coverage \(1 - \hat{\alpha}\). We found similar results for other expert-classifier pairs with different success probabilities.

Figure 5: Empirical test coverage \(1 - \alpha_{\text{emp}}\) and target coverage \(1 - \hat{\alpha}\) for 100 independent realizations of the calibration, estimation and test sets. In Panel (a), the synthetic task comprises a classifier with \(\mathbb{P}[Y' = Y] = 0.5\) and an expert with \(\mathbb{P}[\hat{Y} = Y; \mathcal{Y}] = 0.5\), the number of labels is \(n = 10\) and the size of the calibration and estimation sets is \(m = 1,200\). In Panel (b), the classifier is the popular DenseNet classifier and \(m = 1,500\).
E Robustness to Violations of the IIA Assumption

To study the robustness of our results to violations of the IIA assumption in the estimator of the expert’s success probability, we allow the expert’s preference $u_{yy'}$ for each label value $y' \neq y$ in Eq. 8 to depend on the corresponding prediction set $C_\alpha(x)$ at test time. More specifically, we set

$$u_{yy'} = \log \left( C_{y' y} + p \frac{\mathbb{I}[y' \neq y]}{|C_\alpha(x) \setminus \{y\}|} \sum_{y'' \in C_\alpha(x)} C_{y y''} \right),$$

where $p \in [0, 1]$ is a parameter that controls the severity of the violation of the IIA assumption at test time. Here, note that, if $p = 1$, the expert does not benefit from using our system as long as the prediction set $C_\alpha(x) \neq \{y\}$, i.e., the expert’s conditional success probability is given by $P[\hat{Y} = y; C_\alpha | y \in C_\alpha(x)] = P[Y = y; Y]$. Figures 6 and 7 summarize the results for different $p$ values on synthetic and real data, respectively. We find that our system is robust to (strong) violations of the IIA assumption in the estimator of the expert’s success probability. It is remarkable that, for accurate classifiers as those used in our experiments on real data, the expert benefits from using our system even when $p = 1$. This is because, for accurate classifiers, many of the prediction sets are just singletons containing the true label.

![Figure 6](image6.png)

Figure 6: Empirical success probability achieved by three different experts using our system during test, each with a different success probability $P[\hat{Y} = Y; Y]$, against severity $p$ of the violation of the IIA assumption on three prediction tasks where the classifier achieves a different success probability $P[Y' = Y]$. In each panel, the horizontal dashed line shows the empirical success probability achieved by the expert at solving the (original) multiclass task during test. The number of labels is $n = 10$ and the size of the calibration and estimation sets is $m = 1,200$. Shaded regions correspond to 95% confidence intervals.

![Figure 7](image7.png)

Figure 7: Empirical success probability achieved by an expert using our system with three different classifiers during test against severity $p$ of the violation of the IIA assumption on the CIFAR-10H dataset. The empirical success probability achieved by the expert at solving the (original) multiclass task during test is $P[\hat{Y} = Y; Y] \approx 0.947$. The size of the calibration and estimation sets is $m = 1,500$. Shaded regions correspond to 95% confidence intervals.
Comparison with Top-k Predictors

In this section, we benchmark our system ($\mathcal{C}_\hat{\alpha}$) against a top-$k$ set-valued predictor ($\mathcal{C}_k$) that returns the $k$ label values with the highest scores using both synthetic and real data. Figures 8 and 9 summarize the results for different $k$ values, which show that, by allowing for recommended subsets of varying size, our system is consistently superior to the top-$k$ set-valued predictor across configurations. Moreover, the results on synthetic data also show that, the higher the expert’s success probability $P[\hat{Y} = Y; \mathcal{Y}]$, the greater the optimal $k$ value (i.e., the greater the optimal size of the recommended subsets $\mathcal{C}_k(X)$). This latter observation is consistent with the behavior exhibited by our system, where the higher the expert’s success probability $P[\hat{Y} = Y; \mathcal{Y}]$, the lower the value of the near-optimal $\hat{\alpha}$ and thus the greater the average size of the recommended subsets $\mathcal{C}_\hat{\alpha}(X)$, as shown in Figure 2(b).

![Figure 8: Empirical success probability achieved by two different experts using the top-$k$ set-valued predictor ($\mathcal{C}_k$) during test, each with a different success probability $P[\hat{Y} = Y; \mathcal{Y}]$, on three prediction tasks where the classifier achieves a different success probability $P[Y' = Y]$. In each panel, the horizontal dashed line shows the empirical success probability achieved by the same experts using our system ($\mathcal{C}_\hat{\alpha}$) during test. In all panels, the number of labels is $n = 10$, the size of the calibration and estimation sets is $m = 1,200$ and the results for the optimal $k$ value during test are highlighted in orange.](image-url)
Figure 9: Empirical success probability achieved by an expert using three different top-$k$ predictors ($C_k$) during test, each with a different deep neural network classifier, on the CIFAR-10H dataset. In each panel, the horizontal dashed line shows an empirical success probability achieved by the same expert using our system ($C_\alpha$) during test. In all panels, the size of the calibration and estimation sets is $m = 1,500$ and the results for the optimal $k$ value during test are highlighted in orange.
G  Size Distribution of the Recommended Subsets

Figure 10 shows the empirical size distribution of the subsets $C_{\hat{\alpha}}(X)$ recommended by our system during test for different experts and prediction tasks on synthetic data. The results show that, as the expert’s success probability $P[\hat{Y} = Y; Y]$ increases and the near optimal $\hat{\alpha}$ decreases, the spread of the size distribution increases.

Figure 10: Empirical size distribution of the subsets $C_{\hat{\alpha}}(X)$ recommended by our system during test for different prediction tasks where the expert and the classifier achieve different success probabilities $P[\hat{Y} = Y; Y]$ and $P[Y' = Y]$, respectively. In all panels, the number of labels is $n = 10$ and the size of the calibration and estimation sets is $m = 1,200$. 
H Additional Experiments using an Estimator of the Expert’s Success Probability with a More Expressive Context

In this section, we repeat the experiments on the CIFAR-10H dataset using an alternative discrete choice model with a more expressive context which, additionally to the true label, distinguishes between different levels of difficulty across data samples. The goal here is to show that our results are not an artifact of the choice of context used in the main paper.

We consider three increasing levels of difficulty, denoted as $L_{\text{easy}}$, $L_{\text{medium}}$, $L_{\text{hard}}$. The difficulty levels correspond to the 50% and 25% quantiles of the experts’ fractions of correct predictions per sample in the (original) multiclass classification task. Samples with a fraction of correct predictions larger than the 50% quantile belong to $L_{\text{easy}}$, those with a fraction of correct predictions smaller than the 25% quantile belong to $L_{\text{hard}}$, and the remaining ones belong to $L_{\text{medium}}$. Then, given a sample $(x, y)$ of difficulty $L$, we assume that the expert’s conditional success probability for the subset $C_\alpha(x)$ is given by:

\[
P(\hat{Y} = y; C_\alpha \mid y \in C_\alpha(x), L) = \frac{e^{u_{yy}^L}}{\sum_{y' \in C_\alpha(x)} e^{u_{yy'}^L}},
\]

where $u_{yy'}^L$ denotes the expert preference for the label value $y' \in Y$ whenever the true label is $y$ and the difficulty level of the sample is $L$.

Further, to estimate the parameters $u_{yy'}^L$, we resort to the conditional confusion matrix for the expert predictions on samples of difficulty $L$, i.e., $C^L = [C_{yy'}^L]_{y, y' \in Y}$, where $C_{yy'}^L = P(\hat{Y} = y'; Y \mid Y = y, L)$, and set $u_{yy'}^L = \log C_{yy'}^L$. Finally, we compute a Monte-Carlo estimate $\hat{\mu}_\alpha$ of the expert’s success probability $P(\hat{Y} = Y; C_\alpha)$ required by Algorithm 1 using the above conditional success probability and an estimation set $D_{\text{est}} = \{(x_i, y_i)\}_{i \in [m]}$, i.e.,

\[
\hat{\mu}_\alpha = \frac{1}{m} \sum_{i \in [m]} P(\hat{Y} = y_i; C_\alpha \mid y_i \in C_\alpha(x_i), L(x_i)),
\]

where $L(x_i) \in \{L_{\text{easy}}, L_{\text{medium}}, L_{\text{hard}}\}$ denotes the difficulty level of $x_i$.

Table 3 summarizes the results, which suggest that, in agreement with the main paper, an expert using our system may solve the prediction task significantly more accurately than the expert or the classifier on their own.

| CLASSIFIER     | EXPERT USING $C_\alpha$ |
|----------------|-------------------------|
| ResNet-110     | 0.928                   |
| PreResNet-110  | 0.944                   |
| DenseNet       | 0.964                   |

Table 3: Empirical success probabilities achieved by three popular deep neural network classifiers and by an expert using our system with these classifiers during test on the CIFAR-10H dataset. Here, we assume the expert follows the alternative discrete choice model defined by Eq. 23. The size of the calibration and estimation sets is $m = 1,500$ and the expert’s empirical success probability at solving the (original) multiclass task is $P(\hat{Y} = Y; Y) \approx 0.947$. Each cell shows only the average since the standard errors are all below $10^{-3}$. 


Similarly as in the main paper, we also find that the performance of the system under $\hat{\alpha}$ is very similar to the performance under $\alpha^*$ and the average size of the recommended subsets under $\hat{\alpha}$ and $\alpha^*$ is close to one, as shown in Figure 11. Finally, we also found that our system performs well with a small amount of calibration and estimation data—the relative gain in empirical success probability achieved by an expert using our system with respect to the same expert on their own raises from $3.02 \pm 0.05\%$ under $m = 200$ to just $3.28 \pm 0.04\%$ under $m = 1,500$.

Figure 11: Empirical success probability achieved by an expert following the alternative discrete choice model defined by Eq. 23 and average size of the recommended sets during test for each $\alpha \in A$ and for three popular deep neural network classifiers on the CIFAR-10H dataset. The size of the calibration and estimation sets is $m = 1,500$. Each marker corresponds to a different $\alpha$ value and the darker points correspond to $\hat{\alpha}$ for each task.