Uncertainty sets for Image classification using Conformal Prediction - RAPS Algorithm

2nd April 2025

1 Introduction

Q. Why do we need CP when we already have high accuracy CNN models?

Ans.
(1) Say you are a doctor and based on colonoscopy images the CNN model predicts that there are ulcers with 90% probability. But this is not enough to ensure the health of the patient, we also need to rule out colon cancer for instance! Thus we would want our classifier to additionally output actionable "uncertainty quantification", such as a Prediction set - set of predictions that is guaranteed to cover the true diagnosis with a high probability.
(2) More formally, for a discrete output space Y = $\{1, 2, ..., K\}$, and feature space X $\subseteq \mathbb{R}^d$, we want an "uncertainty set constructor" function C(X) such that the coverage property holds, i.e.

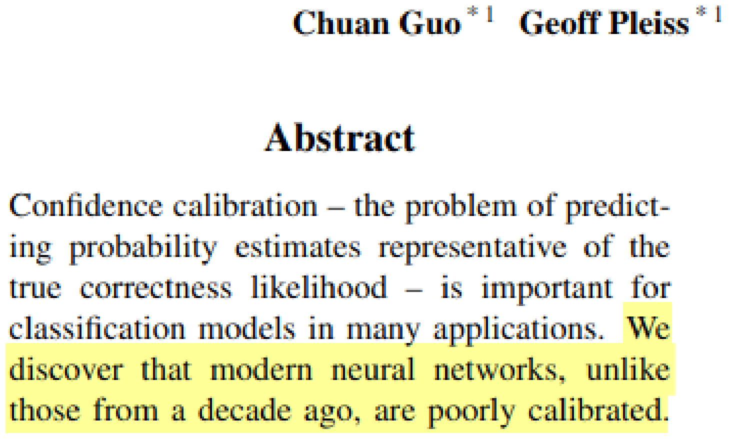
$$\mathcal{P}(Y \in C(X)) \geq 1 - \alpha$$

for user-defined confidence α.
Note: The guarantee in (2) above is marginal over X and Y, i.e. it holds on average, not for a particular X.

2 Naive approach to get Prediction set

Sort the classes as per the highest to lowest probability and then include in the prediction set starting from the most likely class until cumulative sum exceeds $1 - \alpha$. There are two major problems with this though,

1. CNNs are known to be "overconfident" in their predictions (refer Fig 1, Guo et al., 2017) or more generally have poor calibration. This means they are not very trustworthy and may not achieve coverage (no guarantee).
   1. ex. Consider following example - the true probability classes for {ulcer, cancer, none} is 50%, 40%, 10% but the CNN being overconfident predicts 80%, 15%, 5%. In this case if we use $\alpha = 0.2$ and if the true class is cancer, then the prediction set includes only ulcer, leaving out cancer. For a dataset of test samples where the true label is frequently the second-most probable class as in example given above, the probability of failure remains high across many test points, causing a systematic drop in marginal coverage.
   Note: Calibration is the degree to which the probabilities predicted for each class match the accuracy of the classifier on that prediction. Suppose we gather 1000 emails where the model predicts 70% probability of spam. If the model is well-calibrated, we expect that around 700 of these emails (70% of 1000) are truly spam. However, if only 500 emails (50%) turn out to be spam, the model is overconfident (it predicted 70% probability, but reality was only 50%).Conversely, if 850 emails (85%) are actually spam, the model is underconfident (it predicted 70%, but reality was higher).



Chuan Guo [*1] Geoff Pleiss [*1]

Abstract

Confidence calibration – the problem of predicting probability estimates representative of the true correctness likelihood – is important for classification models in many applications. We discover that modern neural networks, unlike those from a decade ago, are poorly calibrated.

Figure 1: From Guo et al., 2017

2. As a consequence of poor calibration, the model assigns (generally) higher probabilities to the topmost class, in turn assigning lower than deserved probabilities for the tail classes. This can cause the prediction sets to become larger than expected, making conformal prediction virtually useless.
   2. ex. Consider the case where true probabilities of ulcer, cancer, perforation, none are 50%, 40%, 8%, 2% and the CNN being poorly calibrated outputs 70%, 19%, 7%, 4%. Say we use $\alpha = 0.1$. Then even though originally our prediction set would be {ulcer, cancer}, the CNN's poor calibration leads us to a larger prediction set {ulcer, cancer, perforation}.

   To solve the first problem, we have used the concept of a "calibration set" that helps us adaptively (adapting to the models uncertainty in performance) select a threshold based on a conformal score function, as described in Angelopoulos et al., 2022 under "Adaptive Prediction Sets" and is the subject of Romano et al., 2020. The proof of coverage is based on exchangeability of calibration set data and approximating a perfect oracle classifier, further details of which can be found in resources listed above. Intuition is explained below,
   ex. The threshold in the naive approach is based on the cumulative probabilities exceeding $1-\alpha$. In APS procedure instead of taking the softmax scores at face value, we learn a new threshold using the calibration dataset. For example, with $\alpha = 0.1$, if we have empirically determined that choosing sets that contain 93% **estimated** (by model) probability achieve 90% coverage on the calibration set, we use the 93% cutoff instead.

However, APS has a problem in practice: the average set size is quite large, i.e problem (2) still persists. Deep learning classifiers suffer from a permutation problem: the scores of the **less confident classes** (e.g. classes 10 through 1000) are not reliable probability estimates. The ordering of these classes is primarily determined by "noise", so APS has to take very large sets for some difficult images. We solve this second problem by introducing regularization in the APS procedure, giving us RAPS.

3 Nested Conformal Prediction

Before we delve into the RAPS algorithm, we need understand a slightly modified but equivalent version of the conformal prediction known as "nested conformal prediction".

Q. How is Nested CP different from CP?

Ans.
Nested CP starts with a sequence of all possible prediction sets $\{\mathcal{F}_t(x)\}_{t\in\mathcal{T}}$ for some ordered set $\mathcal{T}$. The sequence $\mathcal{F}$ is basically a design choice just as the conformity scores were in CP. These prediction

set sequences have the following properties;

1. They are **"nested"** in the sense that for every $t_1 \leq t_2 \in \mathcal{T}$ we have $\mathcal{F}_{t_1} \subseteq \mathcal{F}_{t_2}$.

2. $\mathcal{F}_{inf\ \mathcal{T}} \ = \ \phi$ and $\mathcal{F}_{sup\ \mathcal{T}} \ = \ \mathcal{Y}$ where $\mathcal{Y}$ is the output class of the black-box model.

Then we try to find the smallest $t \in \mathcal{T}$ s.t.

$$\mathbb{P}(Y \in \mathcal{F}_t(X)) \geq 1 - \alpha$$

where α is user defined confidence level.

Following is an example of nested CP in a regression setting,

ex. We are given dataset $D_n \equiv \{(X_i, Y_i)\}_{i=1}^n$ drawn i.i.d. from $P_{XY} = P_X \times P_{Y|X}$ on $\mathcal{X} \times \mathcal{Y}$. First, split D_n into a training set $D_1 \equiv \{(X_i, Y_i)\}_{1 \leq i \leq m}$ and a calibration set $D_2 \equiv \{(X_i, Y_i)\}_{m < i \leq n}$. Using D_1, construct an estimate $\widehat{\mu}(\cdot)$ of the conditional mean of Y given X. Then construct the nonconformity score as the residuals of $\widehat{\mu}$ on D_2: $r_i := |Y_i - \widehat{\mu}(X_i)|$, for $i \in D_2$. Finally, define

$$C(X_{n+1}) = \Big\{ y \in \mathbb{R} : \ |y - \widehat{\mu}(X_{n+1})| < Q_{1-\alpha}(\{r_i\}_{i \in D_2}) \Big\},$$

where $Q_{1-\alpha}(A)$ for a finite set A represents the $(1-\alpha)$-th quantile of elements in A. Due to the exchangeability of the instances in calibration set, $C(\cdot)$ can be shown to be marginally valid (refer Angelopoulos et al., 2022 for proof).

We now give an alternate derivation of the above set using nested conformal:

1. After learning $\widehat{\mu}$ using D_1 (as done before), construct a sequence of nested prediction sets corresponding to symmetric intervals around $\widehat{\mu}(\cdot)$:

   $$\{\mathcal{F}_t(\cdot)\}_{t \geq 0} := \{[\widehat{\mu}(\cdot) - t, \widehat{\mu}(\cdot) + t] : t \geq 0\}.$$

   Note that $\mathcal{F}_t(\cdot)$ is a random set since it is based on $\widehat{\mu}(\cdot)$ which is random through D_1. It is clear that regardless of $\widehat{\mu}$, for any distribution of (X, Y), and any $\alpha \in [0,1]$, there exists a (minimal) $t = t(\alpha)$ such that $\mathbb{P}(Y \in \mathcal{F}_t(X)) \geq 1 - \alpha$. Hence we can rewrite our nested family as

   $$\Big\{[\widehat{\mu}(\cdot) - t, \widehat{\mu}(\cdot) + t] : t \geq 0\Big\} = \Big\{[\widehat{\mu}(\cdot) - t(\alpha), \widehat{\mu}(\cdot) + t(\alpha)] : \alpha \in [0,1]\Big\}.$$

2. The only issue now is that we do not know the map $\alpha \mapsto t(\alpha)$. Hence we use the calibration data to "estimate" the map $\alpha \to t(\alpha)$. For this we use conformal prediction as follows;
   i. define conformal score function $r(x, y) \ := \ \inf\{t \in \mathcal{T} : \ y \in \mathcal{F}_t(x)\}$.
   ii. Define the scores for the calibration data $\{r_i = r(X_i, Y_i)\}_{i \in D_2}$ and set

   $$Q_{1-\alpha}(r, D_2) \ := \ \lceil (1-\alpha)(1 + 1/|D_2|) \rceil\text{-th quantile of } \{r_i\}_{i \in \ D_2}.$$

   (that is, $Q_{1-\alpha}(r, D_2)$ is the $\lceil (1-\alpha)(1 + 1/|D_2|) \rceil$-th largest element of the set $\{r_i\}_{i \in D_2}$).
   iii. Get the prediction sets as $C(x) \ := \ \mathcal{F}_{Q_{1-\alpha}(r, D_2)}(x) = \{y \in \mathcal{Y} : \ r(x, y) \leq Q_{1-\alpha}(r, D_2)\}$. The coverage guarantee follows from marginal coverage property of CP.

Q. Above example shows us a way to construct conformity scores given a nested sequence $\{\mathcal{F}_t(\cdot)\}_t$. Can we do the vice versa to demonstrate equivalence between CP and nested CP?

Ans.
Yes! Given any nonconformity score r and an $x \in \mathcal{X}$, consider the family of nested sets $\{\mathcal{F}_t(x)\}_{t\in\mathbb{R}}$ defined as:

$$\mathcal{F}_t(x) := \{y \in \mathcal{Y} : r(x,y) \leq t\}.$$

Clearly, $y \in \mathcal{F}_t(x)$ if and only if $r(x,y) \leq t$. Hence,

$$\inf\{t \in \mathcal{T} : \, y \in \mathcal{F}_t(x)\} = \inf\{t \in \mathcal{T} : \, r(x,y) \leq t\} = r(x,y).$$

Thus, for any nonconformity score r, there exists a family of nested sets that recovers it.

For a more formal description refer Gupta et al., 2019. From now on, whenever we refer to CP, we mean "nested" CP.

4 Conformal Calibration - A more general setting of RAPS

Branching off from the discussion on nested conformal prediction, we move towards describing a general technique for producing valid output sets as follows,
Consider **ANY** procedure which outputs a prediction set given an input instance and is endowed with a 'tuning parameter' $\mathcal{T}$ that regulates the size of the sets (In APS $\mathcal{T}$ was cum sum of sorted softmax scores, and in RAPS as we will see it is cum sum of **penalized** softmax scores).
We now try to choose $\mathcal{T}$ s.t. the prediction sets give marginal coverage with the help of a calibration set. This process is described formally below;

Formally, let $(X_i, Y_i)_{i=1,\ldots,n}$ be an i.i.d. calibration set. Further, let $C(x, u, \mathcal{T}) : R^d \times [0,1] \times R \to 2^{\mathcal{Y}}$ be a 'set-predictor' function that takes a feature vector x to a subset of the possible labels. The construction of C for each given feature vector is basically what is outlined by the procedure. The second argument u is included to allow for randomized procedures, whose necessity is subject to a different section (refer Section 5 remark 2.). Suppose that the sets are indexed by $\mathcal{T}$ such that they are **nested**, meaning larger values of $\mathcal{T}$ lead to larger sets:

$$C(x,u,\mathcal{T}_1) \subseteq C(x,u,\mathcal{T}_2) \quad \text{if} \quad \mathcal{T}_1 \leq \mathcal{T}_2.$$

Our goal is to find a value of $\mathcal{T}$ that will achieve $1-\alpha$ coverage on test data. Consider following candidate;

$$\hat{\mathcal{T}}_{ccal} = \inf\left\{\mathcal{T} : \frac{|\{i : Y_i \in C(X_i, U_i, \mathcal{T})\}|}{n} \geq \frac{\lceil (n+1)(1-\alpha)\rceil}{n}\right\}.$$

The set function $C(x,u,\mathcal{T})$ with this $\mathcal{T}_{ccal}$ is guaranteed to have finite-sample coverage on a fresh test sampling, as stated formally next.

Thm 1. Suppose $(X_i, Y_i, U_i)_{i=1,\ldots,n}$ and $(X_{n+1}, Y_{n+1}, U_{n+1})$ are i.i.d. and let $C(x,u,\mathcal{T})$ be a 'set-predictor' function satisfying the nesting property. Suppose further that the sets $C(x,u,\mathcal{T})$ grow to include all labels for large enough $\mathcal{T}$ i.e. for all $x \in R^d$, $C(x,u,\mathcal{T}) = \mathcal{Y}$ for some $\mathcal{T}$. Then for $\hat{\mathcal{T}}_{ccal}$ defined above, we have the following,

$$P\Big(Y_{n+1} \in C(X_{n+1}, U_{n+1}, \hat{\mathcal{T}}_{ccal})\Big) \geq 1-\alpha.$$

Pf. It follows from proof of coverage for nested CP.

5 Regularized Adaptive Prediction Sets

We noted that conformal calibration was for ANY procedure that gave a prediction set with some input feature vector and had a tuning parameter. We shall now consider a special case;

Formally, let $\rho_x(y) = \sum_{y'=1}^{K} \hat{\pi}_x(y')\mathbb{I}_{\{\hat{\pi}_x(y') > \hat{\pi}_x(y)\}}$ be the total probability mass of the set of labels that are more likely than y. These are all the labels that will be included before y is included. In addition, let $o_x(y) = |\{y' \in \mathcal{Y} : \hat{\pi}_x(y') \geq \hat{\pi}_x(y)\}|$ be the ranking of y among the label based on the probabilities $\hat{\pi}$. We take

$$C^*(x, u, \mathcal{T}) := \Big\{ y \ : \ \rho_x(y) + \hat{\pi}_x(y) \cdot u + \underbrace{\lambda \cdot |(o_x(y) - k_{reg})|}_{\text{regularization term}} \leq \mathcal{T} \Big\}$$

where $\lambda,\ k_{reg} \geq 0$ are regularization hyperparameters that are helpful in preventing large prediction sets, selection of which are subject to future sections of the report (refer section 6).

Remarks:

1. $\rho_x(y)$ increases as y ranges from the most probable to least probable label, so our sets will prefer to include the y that are predicted to be the most probable by black-box model.

2. $\hat{\pi}_x(y) \cdot u$ is a randomized term to handle the fact that the value will jump discretely with the inclusion of each new y. The randomization term can never impact more than one value of y since there is at most one value of y such that $y \in C(x, 0, \mathcal{T})$ but $y \notin C(x, 1, \tau)$. The following example illustrates the need for randomized predictors,
   ex. Assume for a particular input image we expect a set of size k to have 91% coverage, and a set of size $k-1$ to have 89% coverage. In order to achieve our desired coverage of 90%, we randomly choose size k or $k-1$ with equal probability. In general, the probabilities will not be equal, but rather chosen so the weighted average of the two coverages is exactly 90%.

3. The regularization promotes small set sizes i.e. for values of y that occur farther down the ordered list of classes, the term $\lambda \cdot |(o_x(y) - k_{reg})|$ makes that value of y require a higher value of τ before it is included in the predictive set. For example, if $k_{reg} = 50$, then the 6^{th} most likely value of y has an extra penalty of size 44λ, so it will never be included until $\mathcal{T}$ exceeds $(\rho_x(y) + \hat{\pi}_x(y) \cdot u + 44\lambda)$, whereas it enters when $\mathcal{T}$ exceeds $(\rho_x(y) + \hat{\pi}_x(y) \cdot u)$ in the non-regularized version. Intuitively, a high λ value discourages sets large than k_{reg}

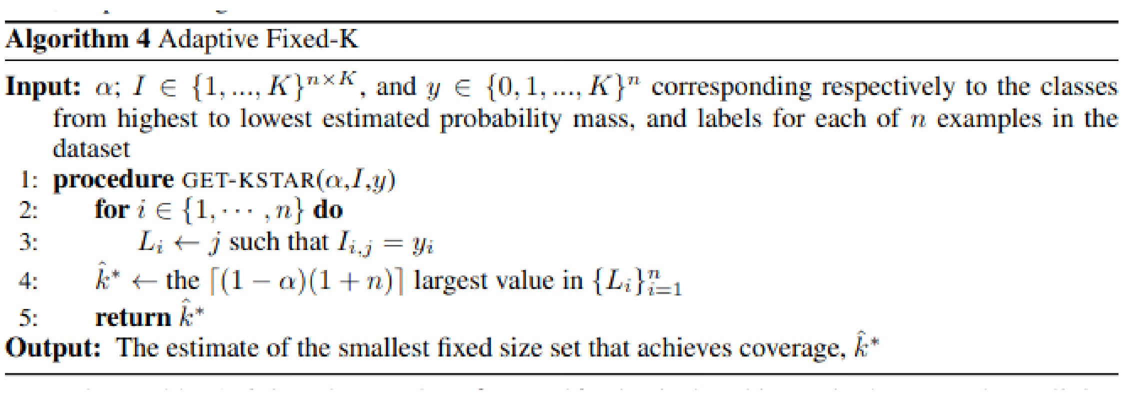
Following formally states RAPS coverage guarantee,
Thm 2. Suppose $(X_i, Y_i, U_i)_{i=1,\ldots,n}$ and $(X_{n+1}, Y_{n+1}, U_{n+1})$ are i.i.d. and let $C^*(x, u, \mathcal{T})$ be defined above. Suppose further that $\hat{\pi}_x(y) > 0$ for all x and y. Then for $\hat{\mathcal{T}}_{ccal}$ defined as in section 4, we have the following coverage guarantee,

$$1 - \alpha \leq P\Big(Y_{n+1} \in C^*(X_{n+1}, U_{n+1}, \hat{\mathcal{T}}_{ccal})\Big) \leq 1 - \alpha + \frac{1}{n+1}.$$

Proof. Note that the first inequality is a corollary of Thm 1. and for upper bound inequality we refer Appendix A of Angelopoulos et al., 2020.

6 RAPS hyperparameter selection

This requires an extra data splitting step, where a small amount of 'tuning data' $\{x_i, y_i\}_{i=1}^{m}$ is used to estimate k^*, and then k_{reg} is set to k^*. The above fig 2. shows the algorithm used for selecting k_{reg}. For λ a simple grid search followed by selection of the value such that achieves smallest size of prediction sets for given k^* on the holdout (tuning) set of size m suffices.



Algorithm 4 Adaptive Fixed-K

Input: α; $I \in \{1, ..., K\}^{n \times K}$, and $y \in \{0, 1, ..., K\}^n$ corresponding respectively to the classes from highest to lowest estimated probability mass, and labels for each of n examples in the dataset

```
procedure GET-KSTAR($\alpha$,$I$,$y$)
    for $i \in \{1, \cdots, n\}$ do
        $L_i \leftarrow j$ such that $I_{i,j} = y_i$
    $\hat{k}^* \leftarrow$ the $\lceil (1-\alpha)(1+n) \rceil$ largest value in $\{L_i\}_{i=1}^n$
    return $\hat{k}^*$
```

Output: The estimate of the smallest fixed size set that achieves coverage, $\hat{k}^*$

Figure 2: Algorithm to select k^*; Credit: Angelopoulos et al., 2020