Technion 236601 + 048719
Reliability in Machine Learning
Final Project

Roie Reshef
Adi Apotheker Tovi

July 2022

Abstract

This is our final project for the course ”Reliability in Machine Learning”. We have read the paper ”Uncertainty Sets for Image Classifiers using Conformal Prediction”, explained their work, and tried to improve their RAPS algorithm using our QRAPS algorithm.

1 Background and Problem Setup

Neural Networks had great success in many fields. One of the common ways to use them is as a classifier. The Network get an input $x \in \mathcal{X}$, usually a picture, and outputs a label $\hat{y} \in \mathcal{Y}$, which can be one of K classes. The goal of the Network is to classify the input correctly, meaning achieving $\hat{y} = y$, where y is the true label.

For each image x, the network outputs a probability vector $\hat{\pi}_x$, and we pick the class with the maximal probability: $\hat{y} = \underset{i \in \mathcal{Y}}{\operatorname{argmax}} \{\hat{\pi}_x(i)\}$.

However, by choosing just one class as the label, we are making a decision that we do not understand. We have no idea why this decision was made, or what are the guarantees that this decision is correct.

1.1 Conformal Prediction

Instead of choosing just one label $\hat{y} \in \mathcal{Y}$, we would rather choose a set $\mathcal{C}(x) \subseteq \mathcal{Y}$, that will cover the true label y with high probability. More specifically:

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

This is called $1-\alpha$ coverage, because the true label is covered by the set $1-\alpha$ of the times. To get this guarantee, we need two things: a calibration set and a score function.

The score function $S : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ gets an input (image) and a possible label, and outputs a score for the pair. The higher the score, it is less likely that the input has this label. The score function could be anything, but the better it is, the better the sets will be. It will usually be chosen after training a neural network.

The calibration set $\mathcal{D}_{cal} = \{x_i, y_i\}_{i=1}^n$ has n exchangeable labeled samples. The samples in the calibration set should also be exchangeable with the real unlabeled data, and should be independent with the score function (i.e. with the training data).

After we have these two we calculate the score of the calibration samples with their true labels: $s_i = s(x_i, y_i)$. We then pick the threshold to be the adjusted empirical quantile of the scores: $\hat{\tau} = \hat{Q}_{1-\alpha}(s_1, .., s_n)$, where α is our desired error rate, and the adjusted qunatile $\hat{Q}_{1-\alpha}$ is the $\lceil (1-\alpha)(n+1) \rceil_{th}$ order statistic of $\{s_i\}_{i=1}^n$. After getting the threshold, we have our set for a new sample x:

$$\mathcal{C}(x) = \{y : S(x, y) \leq \hat{\tau}\} \tag{2}$$

If the new sample is also exchangeable with the calibration set, we get that:

$$1 - \alpha \leq \mathbb{P}\left\{Y \in \mathcal{C}(X)\right\} \leq 1 - \alpha + \frac{1}{n+1} \tag{3}$$

Using conformal prediction guarantee coverage,but the better the score function, the better the sets are on other parameters. Specifically, we would like the sets to be as small as possible (while achieving coverage), and to be adaptive, meaning that they are small on easy samples and larger on hard samples. An example of adaptive sets can be seen in 1.

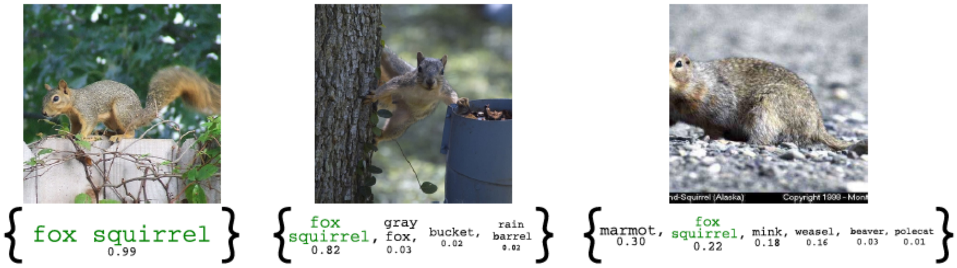

nd-Squirrel (Alaska)
Copyright 1999 - Mon
{ fox squirrel 0.99 }
{ fox squirrel, 0.82 gray fox, 0.03 bucket, 0.02 rain barrel 0.02 }
{ marmot, 0.30 fox squirrel, 0.22 mink, 0.18 weasel, 0.16 beaver, 0.03 polecat 0.01 }

Figure 1: Adaptive sets from [1]. The easier the image, the smaller the set.

2 The Chosen Paper

The paper[1] discusses three algorithms that create a set. The first one is a non-conformal naive algorithm with no guarantees, the second is a conformal algorithm, that is the base of their work, and the third one is the paper's contribution that improves the second algorithm to have smaller and more adaptive sets.

2.1 Naive

Since the network's outputs $\hat{\pi}_x(y)$ are approximating $\mathbb{P}\left\{Y = y|X = x\right\}$, we can use it. First, we sort the network's output in a decreasing order $\{I_1, ..., I_K\}$. We generate $U \sim Uniform(0, 1)$, for smoothing. We sum the estimations $\sum_{i=1}^{L} \hat{\pi}_x(I_i)$ until it is over $1 - \alpha$. If $\sum_{i=1}^{L-1} \hat{\pi}_x(I_i) + U\hat{\pi}_x(I_L) \geq 1 - \alpha$, we change $L \leftarrow L - 1$. Our set is $\mathcal{C}(x) = \{I_1, ..., I_L\}$.

This naive approach will work perfectly if we have the perfect oracle $\hat{\pi}_x(y) = \mathbb{P}\left\{Y = y|X = x\right\}$. In reality, however, it is never the case. This approach is not conformal, so it have no coverage guarantees.

2.2 Adaptive Predictive Sets

The problem with the naive approach is the it is not conformal. The APS algorithm works just like Naive, but pick the threshold in a conformal way. We will define $\rho_x(y) = \sum_{i=1}^{K} \hat{\pi}_x(i)\mathbb{1}\left\{\hat{\pi}_x(i) > \hat{\pi}_x(y)\right\}$, which is the sum of all estimated probabilities of x, that are better than the estimation for y. We will also have the the uniform random variable $U \sim Uniform(0, 1)$, for smoothing. Our score function will be:

$$S(x, y) = \rho_x(y) + U\hat{\pi}_x(y) \tag{4}$$

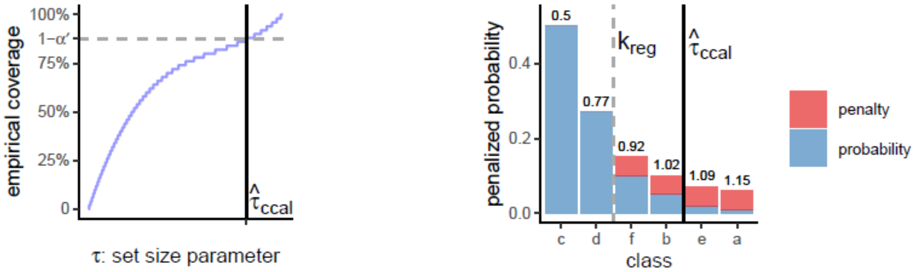
Now that we have the score function, we can use the calibration set to get the threshold $\hat{\tau}$ and choose the set for any new sample like in 2.

2.3 Regularized Adaptive Predictive Sets

APS is conformal, so it achieves coverage, as in 3, but the better the score function, the better the sets can be in term of size and adaptiveness. The paper[1] suggests to improve the sets, but adding a regularization to the score. The new score for the Regularized APS (RAPS):

$$S(x, y) = \rho_x(y) + U\hat{\pi}_x(y) + \lambda \left(o_x(y) - k_{reg}\right)_+ \tag{5}$$

The first two terms are the APS score, and the last term is the regularization term. First, we have $o_x(y) = |y : \hat{\pi}_x(i) > \hat{\pi}_x(y)|$, the number of classes that are more likely than y, according to the network. It can also be seen as the set size if y is the first class to not enter the set. Then, we have the regularization hyper-parameters $\lambda, k_{reg} \geq 0$, and $(\cdot)_+$ is the positive part, also known as ReLU. We can see that the regularization is 0 if $o_x(y) \leq k_{reg}$, and increase by λ each time we add a new label to our set when $o_x(y) > k_{reg}$. They call λ the penalty, because we can say we add it to $\hat{\pi}_y(x)$ if $o_x(y) > k_{reg}$ to calculate the score, as shown in 2. This regularization punishes set with set size larger than k_{reg}. We can see that if $\lambda = 0$ or $k_{reg} = K$, RAPS is equivalent to APS.



(a) Conformal calibration

(b) A RAPS prediction set

Figure 2: Visualization of Conformal Calibration on RAPS from [1]. The left graph shows how the threshold is chosen, and the right graph shows how the penalty looks like.

We can pick the hyper-parameters λ, k_{reg} ahead, or choose them automatically. To choose k_{reg} automatically, we split the calibration set and take out a small part of it (there is no need for the calibration data for k_{reg} to be large, because its value is stable). We calculate $o_x(y)$ on this small calibration set, which is the number of classes that our network favours over the true label of the image. We then pick k_{reg} to be the adjusted $1 - \alpha$ quantile of them: $k_{reg} = \hat{Q}_{1-\alpha}\left(o_{x_1}(y_1), ..., o_{x_n}(y_n)\right)$. Note that with this choice of k_{reg}, there will be no penalty on the true label with probability of at least $1 - \alpha$.

To pick λ automatically, we take out another part of the calibration set. We calculate the score on the calibration set with the chosen k_{reg} and different values of λ. We then pick the λ that gives us the best sets on average according to our criterion. The criterion can be the smallest set size, or the adaptiveness of the sets. We work with the size criterion, because it's easier to see the values of the set sizes than the adaptiveness of the sets.

The calibration points used to find the hyper-parameters cannot be used for finding the threshold on the score function, because the score function is dependant on them.

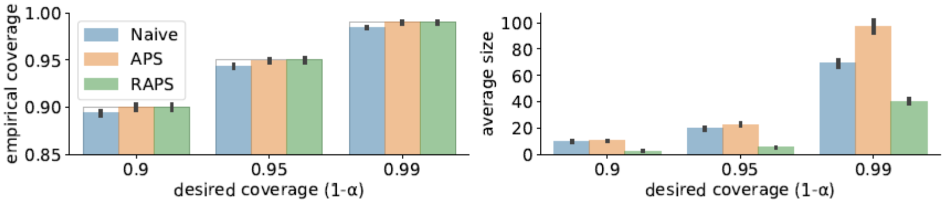


Figure 3: Comparison of the three methods from [1] in coverage and set size.

We can see on 3 that Naive does not achieve that desired coverage, APS achieve coverage and has slightly larger sets than Naive, and RAPS achieve the desired coverage with much smaller sets than the other two methods.

3 Creative Extension

We know from 2 that the choice of the score is what make the sets. Thus, we would like to further improve RAPS by enhancing the regularization. The regularization for RAPS gives a penalty of λ for each label beyond k_{reg}. In order to make the sets smaller, we can push for a regularization that increase the penalty for each label entering the set. We suggest the Quadratic Regularized Adaptive Predictive Sets (QRAPS) algorithm, that have a quadratic regularization beyond k_{reg}:

$$S(x,y) = \rho_x(y) + U\hat{\pi}_x(y) + \lambda \left(o_x(y) - k_{reg}\right)_+^2 \tag{6}$$

We can see that in this case, there is no regularization on the first k_{reg} classes, and a regularization of λ on the $k_{reg}+1$. The regularization increases beyond $k_{reg}+1$, but up to that, the regularization is the same for both QRAPS and RAPS, meaning we can see the difference between the two algorithms only for sets with size larger than $k_{reg}+1$. We pick the hyper-parameters automatically, as in RAPS. The increasing regularization should, in theory, discourage large sets.

4 Results

We compare the performance of QRAPS to that of RAPS on six different pretrained networks from the torchvision library in two variation:

- In the first variation, we let the algorithms pick the hyper-parameters automatically
- In the second variation, we pick $k_{reg} = 0$, and then pick λ automatically, meaning that the regularization affect the sets from the very beginning

The second variation is that we can better see the difference between QRAPS and RAPS, that differ from each other only from $k_{reg} + 1$. The first option of choosing k_{reg} is called True, and the second option is called False. We can see the results in the table1.

	Coverage				Size			
	QRAPS		RAPS		QRAPS		RAPS	
Model	False	True	False	True	False	True	False	True
resnet152	0.908	0.908	0.901	0.908	2.132	2.130	**2.031**	2.128
resnet18	0.902	0.903	0.900	0.903	4.804	**4.502**	4.697	4.687
inception_v3	0.902	0.902	0.901	0.903	5.463	5.474	**5.132**	5.433
resnet50	0.901	0.903	0.900	0.903	4.795	**4.492**	4.668	4.589
vgg16	0.899	0.904	0.900	0.904	**3.479**	3.568	3.569	3.578
densenet161	0.905	0.905	0.900	0.906	2.229	2.338	**2.132**	2.339

Table 1: Comapring QRAPS and RAPS

All results presented in the table1 were averaged on over 10 trails by the following listed model order (27,12,12,20,19,16). The reason for different and relative low trails number is due to resources limit we have. Of course it is better to increase the trails number for at least 100 trails like was done in the original paper[1].

In each trial, the calibration and validation sets were split by a different seed. It is highly important to mention that for each model and trial, the calibration sets for the four different algorithms (QRAPS False, QRAPS True, RAPS False, RAPS True) were identical because we enforce the same seed in a specific trial.

We can see that all options achieve coverage on all networks. It is no surprise, since all options are conformal, so we have the guarantee of 3. About the average set size, we can see that on all networks, the four options gives similar results, and there is no clear winner, since for each network there is a different option the gives the smallest average set size. We can note, however, that RAPS False has the best success, winning in three of the options, and RAPS True has the worst results, never being the best option.

Running more trails should give as a more concrete result, but we can see the QRAPS True beats RAPS True 4/6 of the times, RAPS False 3/6 of the times, and QRAPS False 3/6 of the times, making it a good option for choosing uncertainty sets.

5 Conclusion and Future Works

Predictive sets in many areas such as computer vision (from RAPS, QRAPS and other conformal methods) have many further uses, since they systematically identify hard test-time examples. Prediction sets are the most useful for problems with many classes.

We can see indicate QRAPS might offer a slight improvement over RAPS. Based on our experiments, the original RAPS algorithm results were graded last in all our tested models experiments. We can conclude that more specialized regularization might improve the sets, and that different networks might need different regularization.

Future Work:

- Run more trials, to get a better estimation of the average set size for each option.
- Check more models and more datasets.
- Check the adaptiveness of the QRAPS sets compared to RAPS.
- Think up new kinds of regularizations.
- Pick the best regularization option for our data, out of many options, using a calibration set.

This is our code, which is based on the code of [1].

References

[1] Anastasios Angelopoulos, Stephen Bates, Jitendra Malik, and Michael I. Jordan. Uncertainty sets for image classifiers using conformal prediction. *ICLR*, 2021.