CALIBRATING BLACK-BOX TRANSITION MODELS FOR MDP PLANNING USING CONFORMAL PREDICTION

Romeo Valentin, Hugo Buurmeijer

Department of Aeronautics & Astronautics
Stanford University

ABSTRACT

In this paper, we study the application of Conformal Prediction to the calibration of black-box transition models for planning in a Markov Decision Process setting. Successful deployment of this method frees us from having to make distributional assumptions on the underlying problem, which often leads to poor calibration, and can provide statistical guarantees, particularly relevant for safety-critical applications. Drawing inspiration from Composite Quantile Regression, we propose an alternative formulation of the Bellman equation, to be used with existing algorithms such as value iteration. Also, we show that we can tighten the approximation error arbitrarily tightly. Using our experimental setup, we find that policies based on our method have the ability to outperform other strategies, and the temperature scaling recalibration technique. Specifically, our method consistently achieves a higher discounted reward, and is more robust against a shift in the underlying distribution.

1. INTRODUCTION

Decision-making under uncertainty, including planning and control, has been studied for decades in a range of applications, including robotic navigation and control, medical decision making, financial services, and may even be applied to planning day-to-day tasks ("Life is a POMDP"). Commonly, a model of the environment and its interaction dynamics is constructed, which includes some measure of uncertainty. Then, the model can be used for probabilistic planning, with the aim of maximizing a reward function over a (possibly infinite) time horizon. Besides model uncertainty, the problem setup can also include uncertainty about the current state, in which case extra care has to be taken. Under some assumptions, the former scenario can be described and solved as a Markov-Decision Process (MDP), and the latter as a Partially-Observable MPD, or POMDP.

Both scenarios commonly require that the problem setup specifies state and model uncertainties using explicit distributions, e.g. Gaussian Distributions (continuous domain) or tabular values (discrete domain). These distributional assumptions that the practitioner must make may, however, be misspecified or poorly calibrated. For example, the Kalman Filter, which uses a POMDP formulation, makes the assumption of Gaussian uncertainties or errors, which may not properly represent the true uncertainty distribution; counterexamples include heavy tails, and skewed- or multi-modal distributions. Similarly, in a discrete MDP, the transition probabilities are typically represented by tabular values, which may be poorly calibrated.

In a separate line of research, the study of distribution-free predictive inference has seen a recent resurgence, with the aim of turning uncalibrated or misspecified predictive models into their calibrated counterparts, thereby overcoming the limits that distributional assumptions impose, and improving overall robustness of the models. Examples notably include Conformal Prediction [1], which can additionally provide finite-sample statistical guarantees for calibration, thereby aiding certifiability in safety-critical applications.

Despite recent advances in distribution free uncertainty quantification and Conformal Prediction (CP), only little research has been devoted to its intersection to decision making under uncertainty. In this project, we therefore aim to start bridging that gap, by applying CP techniques to the MDP setup, and exploring how existing algorithms need to be modified accordingly. Moreover, we provide an outlook on the applicability of the guarantees that come with CP, and showcase the method on a simple dataset.

Our code[1] is available at `https://github.com/RomeoV/AA228_FinalProject` and `https://github.com/RomeoV/DroneSurveillance.jl`.

2. MATHEMATICAL BACKGROUND

In this section, we will build up the mathematical background required to understand our methodology. We do this by first briefly discussing how MDP planning is currently performed. Subsequently, a measure of calibration error is

Correspondence: romeov@stanford.edu, hbuurmei@stanford.edu

[1] Refer to the AA228_FinalProject repository for GIF visualizations

presented, and recalibration is introduced. Lastly, we elaborate on Conformal Prediction and its statistical guarantees.

2.1. Conventional MDP Planning

To formulate a problem as an MDP, we need to define its four components, namely a state space $\mathcal{S}$, an action space $\mathcal{A}$, a reward function $R(s,a)$ and a state transition model $T(s'|s,a)$. Often, the state and action spaces follow directly from the problem, and the reward model is selected by the practitioner. The transition model, however, is often unknown in advance, and has to be estimated, potentially in a data-driven fashion. Once the MDP is constructed, a key equation used to find an optimal policy is the so-called Bellman equation

$$U(s) = \max_a \left(R(s,a) + \gamma \sum_{s'} T(s' \mid s,a) U(s') \right), \quad (1)$$

which denotes the necessary condition for the value function $U(s)$ to be optimal. Dynamic programming algorithms, such as policy iteration or value iteration, are subsequently used to recursively solve the Bellman optimality equation. Using the derived value function, one can plan by simply taking the action that maximizes the expected future reward.

2.2. Expected Calibration Error and Recalibration

Probabilistic forecasts of future events can be given by a predictive distribution function. Estimating the quality of a forecast is however not an easy endeavor, as a single or few realizations of the event provides little evidence of "correctness" in a probabilistic sense. However, from a frequentist perspective of probability, we can measure the consistency of a predictive distribution with the observations over many samples, which we call *calibration*. Different types of calibration have been proposed, most notably (i) probabilistic calibration, (ii) exceedance calibration, and (iii) marginal calibration [2]. We will restrict ourselves to types (i) and (iii), which informally state that, for a given probability p, realizations should fall into the cumulative density for p with frequency p, and that the predictive and observed distributions are indistinguishable in the limit of many datapoints. We may thus measure the expected calibration error (ECE) of a distribution D for a given probabililty p by computing

$$ECE_p = \left(\frac{1}{N} \sum_{i=1}^{N} \left[y_i \in cdf^{-1}(D,p) \right] \right) - p \quad (2)$$

and applying a suitable norm. Here, $[\cdot]$ denotes the Iverson-bracket.

If a model turns out to be poorly-calibrated, it is commonly *recalibrated*, typically using a held-out part of the data as a *calibration set*. Many recalibration techniques exist – for instance, for neural networks, it was shown that temperature scaling can be surprisingly effective in achieving good calibration, given its simplicity [3]. Temperature scaling uses a single scaling parameter $T > 0$ to "soften" the softmax function. The softmax function is used to predict the probabilities of a multinomial probability distribution from a given score, such as in the final layer of a neural network. The softmax function including the temperature parameter is given by

$$P_i = \frac{e^{\frac{y_i}{T}}}{\sum_{k=1}^{n} e^{\frac{y_k}{T}}} \quad (3)$$

which reduces to the original softmax expression if $T = 1$. A higher temperature results in a more uniform distribution, thereby signifying a higher uncertainty.

2.3. Conformal Prediction

Another way to recalibrate a model is given by Conformal Prediction (CP) [1, 4]. CP was first introduced in 2005 and recently repopularized by [5] for a wide variety of applications. In its basic form, CP can be used to turn probability distributions predicted by any black-box model into provably well-calibrated prediction sets, given a confidence level. In particular, consider a function $f : \mathcal{X} \to \mathcal{D}_\mathcal{Y}$, mapping an input $x \in \mathcal{X}$ to a (categorical or continuous) distribution $\mathcal{D}_\mathcal{Y}$ over values $y \in \mathcal{Y}$. Consider now the conformalization-operator $\mathcal{T}_\lambda$, which we apply to f, such that $[\mathcal{T}_\lambda \circ f] : \mathcal{X} \to \mathcal{P}_\mathcal{Y}$, where $\mathcal{P}$ denotes the power set. In other words, the conformalized model outputs a *prediction set* $\{y_{i_1}, y_{i_2}, \dots\}$ given the confidence level λ. Then, CP makes the following *marginal* guarantee:

$$\lambda \leq \mathbb{P}\left(y_{\text{test}} \in [\mathcal{T}_\lambda \circ f](x_{\text{test}})\right) \leq \lambda + \frac{1}{n+1}.$$

Notably, to construct $\mathcal{T}_\lambda$, a set of *calibration samples* is required, the number of which we denote as n. Finally, we note that this guarantee only holds *marginally*, i.e. averaged over all x in the data distribution.

3. CONSTRUCTING AND PLANNING WITH A CONFORMALIZED TRANSITION MODEL

In this section, we will first introduce how we approximate the transition function with a simple multinomial linear regression model, which we then subsequently "conformalize". Then, we will highlight one of the key difficulties when using CP for planning, which is that the Bellman equation has to be rewritten with a new way to compute the expected utility of the next state given a current state and action. We will provide several options to overcome this issue and mention theoretical motivations.

3.1. Approximating the transition function

We consider three types of transition model approximations: (i) a simple linear model, (ii) a recalibrated linear model, and (iii) a conformalized linear model.

Linear model. The linear model is constructed as follows: We first collect a dataset by initializing ten thousand random states, and observing a single transition according to the true transition function. Then, in order to reduce the complexity of the model, we reduce the input and prediction space by only considering the *position delta* between the drone and the agent, i.e. $\Delta s = (x_a - x_d, y_a - y_d)$. Then, we further augment the input by the action taken by the drone and a bias value, i.e. $x = (\Delta s_1, \Delta s_2, a_1, a_2, 1)$. Finally, we encode all possible state deltas for the next step as categorical variables, and add a final category which corresponds to transitioning to the terminal state.

Using this setup, we train a simple multinomial linear regression model on the constructed dataset, which outputs a probability distribution over all future states. Specifically, given model parameters $\theta \in \mathbb{R}^{n_{\text{states}} \times 5}$, we can compute the prediction over all future state deltas as $\sigma.(\theta x)$, where $\sigma_i(x) = \frac{\exp x_i}{\sum_j \exp x_j}$ denotes the softmax operator, and $\sigma.$ denotes the collection of softmax values over all i. We further prune states with a predicted probability smaller than $1e{-}4$.

Linear Recalibrated Model . In order to construct the recalibrated model using temperature scaling, we first create a linear model as described above, and then recalibrate using a calibration set of $n_{\text{calib}} = 100$, which may be sampled from a different transition model. Then, we select the optimal temperature by searching over a set of fixed temperatures ranging from 0.1 to 3, and selecting the temperature that minimizes the ECE (see Eq. (2)). The temperature is included in the softmax function as described in Eq. (3).

Conformalized Linear Model. The conformalized linear model is constructed again by first constructing a regular linear model as described above, and then computing a conformalizing map $\lambda \mapsto \hat{\lambda}$ as follows: First, a set $\Lambda = \{0.1, 0.2, \ldots, 0.9\} \cup \{0.99\}$ of coverage values is defined. Then, a calibration set is obtained similar to Section 3.1. Next, each $\lambda \in \Lambda$ is conformalized as described in [5]: the probability scores for each label in the calibration set is computed using the linear model, and then the conformalized coverage values $\hat{\lambda}$ are constructed by computing one minus the empirical p-quantile of the label probability scores, where $p = 1 - \lambda$.

Finally, when the conformalized model is evaluated, the prediction set is constructed as follows: Given s, a, and λ, the trained linear model is first evaluated using s and a, and λ is transformed by the conformalizing map (i.e. a fixed lookup table). Then, the prediction set is constructed by including all states for which the predicted probability is greater than $\hat{\lambda}$.

3.2. Computing a conformalized expectation

While CP promises some nice theoretical guarantees, unfortunately the distribution-free nature can make it more difficult to apply in practice. In particular, recall that distribution-based prediction function lend themselves for easy computation of statistical properties like the expectation

$$\mathbb{E}_{x \sim f_X} g(x) = \sum_{x \in \mathcal{X}} g(x) f(x).$$

For distribution-free prediction functions like CP, however, such an expectation is not easily computed. This become a problem when trying to use an algorithm like value iteration for planning: Recalling the normal Bellman equation, i.e. Eq. (1), and rewriting it slightly, indeed we find the need to compute the expectation over the utility of future states. Specifically, the transition function maps a state-action tuple to a distribution over next states, i.e. $T : \mathcal{S} \times \mathcal{A} \to \mathcal{D}(\mathcal{S})$. We may therefore rewrite Eq. (1) as

$$U(s) = \max_a \left(R(s,a) + \gamma \mathbb{E}_{s' \sim T(s'|s,a)} U(s') \right). \quad (4)$$

In order to proceed with our conformalized transition model, we will now consider how to replace T by

$$(\mathcal{C}_\lambda \circ T) : \mathcal{S} \times \mathcal{A} \times [0,1] \to \mathcal{P}(\mathcal{S}).$$

In particular, we would like to compute something that resembles the expectation

$$\mathbb{E}_{s' \sim [\mathcal{C}_\lambda \circ T](s,a), \lambda \in [0,1]} U(s'),$$

or, more generally,

$$\mathbb{E}_{x \sim [\mathcal{C}_\lambda \circ f_X](\cdot), \lambda \in [0,1]} g(x).$$

To this end, we propose a simple approximation inspired by Composite Quantile Regression [6]. In this approximation, we estimate the expectation by averaging over multiple choices for $\lambda \in \Lambda$. For notational convenience, we drop the $(\cdot)$ in e.g. $x \sim f_X(\cdot)$. Then we can write

$$\begin{aligned} \mathbb{E}_{x \sim f_X} g(x) &\approx \sum_{\lambda \in \Lambda} \left(w_\lambda \cdot \operatorname*{mean}_{x \in [\mathcal{C}_\lambda \circ f_X]} g(x) \right) \\ &= \sum_{\lambda \in \Lambda} w_\lambda \frac{1}{|[\mathcal{C}_\lambda \circ f_X]|} \sum_{x \in [\mathcal{C}_\lambda \circ f_X]} g(x). \end{aligned} \quad (5)$$

It is an interesting question how to choose the weights w_λ. For example, one could consider setting the weight proportional to the difference between two adjacent choices of λ. However, we note that in the literature around Quantile Regression, a similar problem has been discussed – in particular, *Composite Quantile Regression* (CQR) uses a set of quantile regressors with evenly distributed quantiles in $[0, 1]$

to estimate the parameters of a linear model. Each regressor can be used to compute an individual estimate, and the CQR method proposes simply averaging the estimates for different quantiles. The paper further provides proofs for error bounds and asymptotic efficiency. Inspired by the insights from CQR, we therefore propose setting all weights equally to $w_\lambda = \frac{1}{|\Lambda|}$.

3.3. Some theoretical considerations

We first highlight an underlying theoretical drawback of a conformalized model, which is that some state with non-zero probability mass may never show up in the prediction set. In particular, consider a conformalized model $[\mathcal{C}_\lambda \circ T]$ for a set of $\lambda < 1$. Let now s' be a state with probability mass $T(s' \mid s, a) < 1 - \lambda_{\max}$, but with value far away from the estimated expectation, e.g. $U(s') \ll 0$. Since in the conformalized expectation we may never consider the value of this state, we can see that our expected value can be arbitrarily wrong. We therefore suggest considering only problems where we can lower- and upper-bound the value. Let U_{lo} and U_{hi} denote the lower and upper bounds, respectively, let U^* be the true value, and let $\tilde{U}_{\lambda_{\max}}$ be the best approximation we could hope for by a conformalized prediction function. Then we can write

$$\lambda_{\max}\tilde{U}_{\lambda_{\max}} + (1 - \lambda_{\max})U_{lo} \leq U^* \leq \lambda_{\max}\tilde{U}_{\lambda_{\max}} + (1 - \lambda_{\max})U_{hi},$$

i.e. for sufficiently high $\lambda_{\max}$ we can tighten the approximation error arbitrarily tightly (at the cost of a larger calibration set).

4. EXPERIMENTS

In this section, we showcase the performance of our proposed algorithm on a specific MDP. In particular, we will compare the measured performance of a policy constructed using the conformalized transition model to several baselines, including planning using a perfect model, a random policy, and the basic and recalibrated linear models introduced in Section 3.1.

4.1. Drone Surveillance MDP

The specific problem instance chosen to demonstrate our method is the Drone Surveillance MDP, which is illustrated in Fig. 1. In this problem, a drone is tasked with surveying two regions indicated with green, while avoiding a ground agent indicated by a red eye. The blue region is the field of view, which is unlimited in our case, as we do not consider state uncertainty. Once the drone reaches the target, or is caught by the agent, it deterministically transitions to a terminal state.

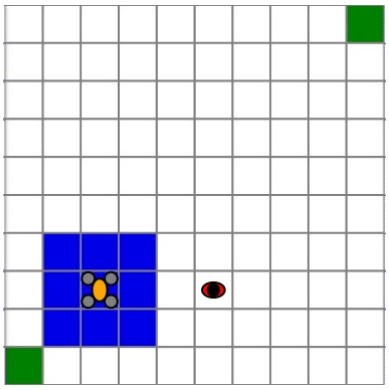

Fig. 1. The Drone Surveillance MDP. The starting and goal positions are in the lower left/top right, and the drone and agent are shown at positions (3, 3) and (5, 3).

Specifically, we consider a $n_x \times n_y = 10 \times 10$ grid, where the drone always starts in the lower left corner $(1, 1)$, and the agent always starts at $(6, 6)$. We use a discount factor of $\gamma = 0.99$, and we have a probability of $1/4$ that the agent fails to move in its intended direction. The reward model is such that the reaching the goal state at (n_x, n_y) has an associated reward of $r_{\text{win}} = 100$, and the reward for getting caught by the agent is $r_{\text{lose}} = -1000$. Finally, we introduce a parameter, defined on the unit interval, that dictates how aggressively the agent moves in the direction of the drone, which allows us to vary the "difficulty" of the problem. Note that for a fully aggressive agent it is usually impossible for the drone to reach the goal.

4.2. Comparing model performances

To show the efficacy of our method, we compare different approximations of the true transition function and use them to create a policy for the MDP. In particular, given a (approximated) transition function $\tilde{T}$, we first compute a value estimate for all possible states using a custom Value Iteration algorithm. Then, we use the value estimate to construct a simple greedy policy. Finally, we evaluate each policy using the Policy Iteration algorithm, which has access to the true transition function.

For the experiment, we consider the three different approximated transition functions $\tilde{T}$ introduced in Section 3.1, as well as the perfect transition function and a randomized policy. We compare all resulting models by varying the agent aggressiveness level from zero to one, and we run the evaluation multiple times.

Fig. 2 shows the results of our experiments. Each setup is run five times, and one and two standard deviations are visualized. First, we see that the random policy performs very poorly, as is expected. However, due to the discount factor, the expected future return is not equal to the final reward

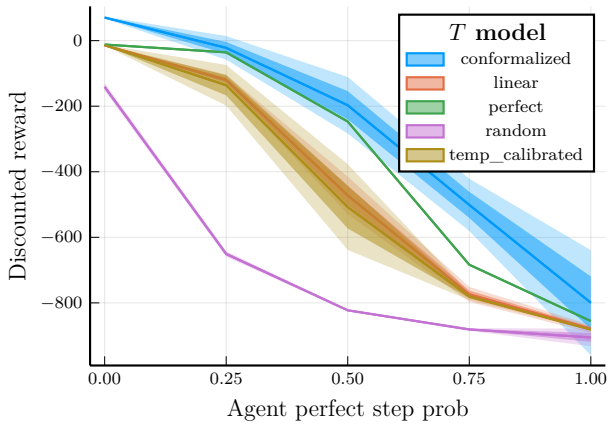

Fig. 2. Discounted reward vs agent agressiveness. Different model approximations are evaluated, and the discounted future return ($\gamma = 0.99$) is plotted against the "agent aggressiveness", i.e. the probability of the agent making a perfect step.

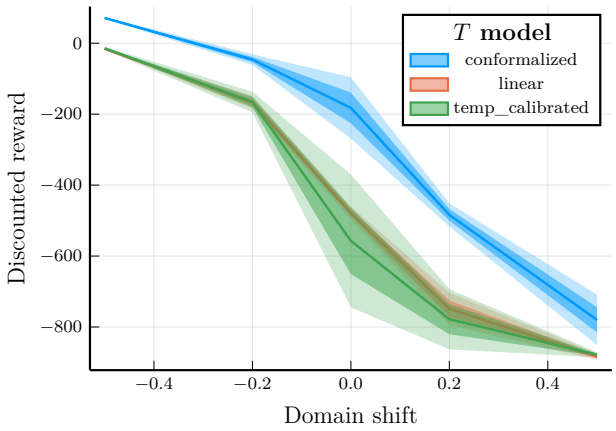

Fig. 3. Domain shift experiment. Models are trained on simulations with an "agent aggressiveness" of 0.5. Then, the aggressiveness is changed by the "domain shift", and the models are recalibrated and evaluated.

(-1000 for getting caught). Next, we notice that the linear and the recalibrated linear model performs almost exactly the same. We can therefore conclude that temperature scaling does not seem to be enough to overcome the limitations of the simple linear model. Finally, we notice that the policy created using the conformalized model performs really well – in fact, it outperforms the policy created using the true transition function. We rationalize this by noting that the conformalized model can behave overly cautious, which may ultimately lead to better results. Nonetheless, we stress that further investigations are necessary to understand the effect of conformalized models on the planned policies.

4.3. Domain shift experiment

Finally, we note that recalibration can not only be used to improve calibration on a given dataset, but also to improve performance on a new dataset, without retraining, and with only few samples. We therefore set up a domain shift experiment as follows: We initially consider an agent with aggressiveness $p_0 = 0.5$, and construct a linear model. Next, we vary the agent aggressiveness by a value Δp, and recalibrate the models with a small amount of data ($n_{\text{calib}} = 100$) from the new agent. We construct and evaluate the results as in the previous section.

Following, we construct and evaluate policies in a similar fashion to Section 4.2, however with the true model containing the domain shift. Fig. 3 presents the results of this experiment. Interestingly, we observe that despite the domain shift, the results look very similar to the previous experiment. This suggests that the recalibration had limited effect on the overall performance. Again, temperature scaling does not seem to improve the performance substantially, however the conformalized model performs significantly better than the linear model.

5. CONCLUSIONS

In this work, we have demonstrated recalibration of a multinomial linear regression transition model as part of an MDP formulation by employing Conformal Prediction. Specifically, the Bellman equation was modified to suit the use of Conformal Prediction, which can then be used with existing planning algorithms. This approach has the benefit that no distributional assumptions have to be made, with the additional advantage that statistical guarantees can be made.

The performance of this technique was benchmarked against policies generated using the original transition function, a perfect transition model, a random policy, and a recalibrated transition model by means of temperature scaling. We evaluated the discounted reward for different agent aggressiveness levels, as well as a shift in the underlying distribution. Interestingly, the results show that excellent performance can be achieved using our proposed method, both for recalibration on a given dataset, and recalibration on a limited new dataset. Further analysis of the results is highly recommended, but the preliminary results look promising.

Although in this paper only offline planning was considered, our method could easily be extended to online planning. Other future work could include applying our method to more complex transition function approximations, for instance a neural network. In addition, the investigated MDP setting could be scaled up, or a new problem could be explored, allowing for a more complete performance analy-

sis. One could also extend our work to POMDPs, which were mentioned in Section 1, by including state uncertainty, which are often a more realistic representation of the real world. Finally, the authors would like to do more research into the statistical guarantees that Conformal Prediction can provide.

6. CONTRIBUTIONS OF TEAM MEMBERS

The majority of this work has been done jointly in collaborative working sessions. This includes the project ideation, problem formulation, literature review, and the larger part of the coding. The statistical theory was mainly covered by Romeo. The last portion of coding, as well as the generation of the final results, were completed independently. Hugo wrote the abstract, part of Section 1, Section 2.1, Section 4.1, part of Section 4.2, Section 5, and Section 6. Romeo wrote part of Section 1, Section 2.2, Section 2.3, Section 3, part of Section 4.2, and Section 4.3.

7. REFERENCES

[1] Vladimir Vovk, A Gammerman, and Glenn Shafer, *Algorithmic learning in a random world*, Springer, New York, 2005, OCLC: 209818494.

[2] Tilmann Gneiting, Fadoua Balabdaoui, and Adrian E. Raftery, "Probabilistic forecasts, calibration and sharpness," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 69, no. 2, pp. 243–268, 2007, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-9868.2007.00587.x.

[3] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger, "On Calibration of Modern Neural Networks," in *Proceedings of the 34th International Conference on Machine Learning*. July 2017, pp. 1321–1330, PMLR, ISSN: 2640-3498.

[4] Glenn Shafer and Vladimir Vovk, "A tutorial on conformal prediction," June 2007, arXiv:0706.3188 [cs, stat].

[5] Anastasios N. Angelopoulos and Stephen Bates, "A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification," Dec. 2022, arXiv:2107.07511 [cs, math, stat].

[6] Hui Zou and Ming Yuan, "Composite quantile regression and the oracle Model Selection Theory," *The Annals of Statistics*, vol. 36, no. 3, June 2008, arXiv:0806.2905 [math, stat].