Evaluating Distributional Predictions of Search Time: Put Up or Shut Up Games

Sean Mariasin[1], Andrew Coles[2], Erez Karpas[3], Wheeler Ruml[4], Solomon Eyal Shimony[1], Shahaf Shperberg[1]

[1]Ben-Gurion University,
[2]King's College London,
[3]Technion,
[4]University of New Hampshire
seanmar@post.bgu.ac.il, andrew.coles@kcl.ac.uk, karpase@technion.ac.il
ruml@cs.unh.edu, shimony@cs.bgu.ac.il, shperbsh@bgu.ac.il

Introduction

In many real-world applications, heuristic search may take longer to complete than the remaining time before execution must begin. Recognizing such situations is important, as then fallbacks can be employed: opting for a suboptimal solution rather than an optimal one, beginning to execute a partially developed plan while continuing the search, or even declaring failure early, hoping to 'cut your losses'.

A metalevel controller seeking to make such decisions needs a reliable prediction of whether such search is taking too long. For example, situated planners (that plan 'online' while time is passing) need to assess the probability that a candidate partial plan will be executable at the time search terminates, thus requiring a distribution over remaining search time. Currently, basic distribution estimates are available, such as one based on one-step-error (Shperberg et al. 2021). While there is a body of work on attempting to predict the remaining amount of search effort (Thayer, Stern, and Lelis 2012; Sudry and Karpas 2022), typically these methods deliver a single number corresponding to the expected value, or other point estimate, of this quantity.

Developing distribution estimators raises the question of how to evaluate them. Testing in the context of metalevel control is unattractive, as most metareasoning control schemes are hard to analyze. But gauging the accuracy of the distribution estimate in isolation is also problematic, as a ground truth distribution is not available. The distribution of the remaining search time that we wish to model is over all possible problem instances consistent with the observations made so far, which is not realistically obtainable. Instead, we exploit the subjectivist Bayesian interpretation of probability, in which a rational agent that believes that some event e will occur with probability p must accept a bet where it pays less than p to gain a reward of 1 if e occurs.

Put Up or Shut Up Games

We propose 'put up or shut up' games as a measure of the quality of a subjective probability of remaining runtime. A given search algorithm (e.g. A^*, with a known heuristic h) runs on a given problem instance. An agent observes the search algorithm run up to a certain point and then, given a deadline, has to bet whether to 1) quit (shut up) or 2) pay a sum (put up) and collect a reward if the search ends successfully before the deadline.

Definition 1 (BPSG)**.** *Basic put-up or shut-up game: given a search algorithm A running on a problem instance I, observations O, a remaining time target t. Should we (shut up) stop the computation, avoiding any cost or gain, or (put up): paying an ante of θ, to get a known reward of R (thus net gain $R - \theta$) iff A solves instance I before t time passes?*

To play the game rationally, we must estimate the subjective probability $\hat{p} = P(t(A, I) \leq t|O)$, the probability that algorithm A will solve instance I in remaining time $t(A, I)$ that is less than t, given our observations O. The rational agent should not put up unless $\hat{p} \geq \frac{\theta}{R}$. The latter ratio is also called *betting odds* of θ to $R - \theta$.

This paper assumes an expansion-step-based search algorithm and that time is in units of expansion steps. Success at BPSG hinges on having a reliable error model; merely achieving a better prediction (such as a lower RMSE on a numerical estimate) is insufficient — a predictor more accurate in expected error may perform on average worse in BPSG than a predictor with a higher RMSE:

Example: Suppose that search algorithm A will run for another 100 or 101 expansions, each with probability 0.5. This is unknown to the agent, which relies on predictors to gauge the anticipated remaining search time, thereby informing its decision on whether to prolong or terminate the search, which must conclude within 100 more expansions to be usable. We have two predictors: predictor **a** is an accurate point estimator that has an unbiased error of at most 1, uniformly distributed. Importantly, the agent is unaware of the true underlying error model, and incorrectly assumes that predictor **a** delivers exactly the correct value. That is, when the true remaining number of expansions is 100, **a** will predict either 99, 100, or 101, each of these predictions having a probability of $\frac{1}{3}$ of being made, and the agent will believe that the predicted value is exact. Predictor **b** has an unbiased error of at most 2 expansions, uniformly distributed. Yet, unlike **a**, predictor **b** outputs a distribution. The returned distribution results from randomly drawing a value uniformly from within the range of ± 2 expansions from the true value, then constructing a uniform distribution within ± 2 around the predicted value. For instance, if the true expansion value

is 100, Predictor **b** would base its prediction around a value $v \in [98, 99, 100, 101, 102]$, each having a probability of being the base of $\frac{1}{5}$, and then **b** returns a uniform distribution over the 5 values $[v-2, v-1, v, v+1, v+2]$.

First, consider even betting odds, i.e. the ante is $\theta = 1$ and the reward is $R = 2$. Consider predictor **a** first. If the true value is 100 (probability 0.5), **a** will predict 99 or 100 with probability total $\frac{2}{3}$ in this case, and because it is sure its prediction is correct, the metareasoner will decide to ante up and collect the reward, gaining 1. When the true value is 101 (probability 0.5), **a** will predict 100 with probability $\frac{1}{3}$, and the metareasoner, believing that the prediction is correct, will ante up and lose. So at even odds, using predictor **a**, the metareasoner will gain $0.5(\frac{2}{3} - \frac{1}{3}) = \frac{1}{6}$.

Now consider predictor **b**. With a true value of 100, **b** returns a uniform distribution over $[v-2, v-1, v, v+1, v+2]$ (with $v \in [98, 99, 100, 101, 102]$ each with probability $\frac{1}{5}$). For $v = 98$, the probability of timely completion (≤ 100 more expansions) is 1. For $v = 99$, that probability is 0.8, etc. so all in all for $v \in [98, 99, 100, 101, 102]$ the corresponding probabilities of success are $[1, 0.8, 0.6, 0.4, 0.2]$. With the true number of expansions being 101, we have $v \in [99, 100, 101, 102, 103]$ and the corresponding probabilities of success will be $[0.8, 0.6, 0.4, 0.2, 0]$. The decision now depends on the betting odds. At even odds, the metareasoner using **b** will ante up and collect in 3 cases (total probability $\frac{3}{5}$) and will ante up and lose in 2 cases, for a total expected gain of $\frac{1}{10}$. As expected, at even odds using **a** the agent scores better on average than using **b**.

With different odds, say an ante $\theta = 5$ and reward $R = 6$ (again total gain 1), using **a** the agent makes the same decisions, now gaining 1 in 2 cases and losing 5 in one case; total expected **loss** is $\frac{1}{2}$. Using **b** the agent only antes up when its subjective probability of winning is better than $\frac{5}{6}$, so only antes if **b**'s prediction has $v = 98$, where it is certain that the true value is ≤ 100; thus its total expected **gain** is $\frac{1}{10}$. Conversely, with an ante of $\theta = 1$ and reward $R = 6$ (potential gain of 5), a similar calculation shows an expected gain using **a** of $\frac{9}{6}$, and with **b** we get $\frac{21}{10}$, again better than **a**.

Comparing Estimation Methods

In this work we compare one existing distribution predictor, several existing point estimators naively converted into distributions, and some new schemes we have developed.

The only scheme of which we are aware that does output a distribution is based on the *one-step error* (Dionne, Thayer, and Ruml 2011; Shperberg et al. 2021), which we consider as one type of baseline (os_{ed}).

Several point-based predictors exist. *Expansion delay* (Dionne, Thayer, and Ruml 2011) measures the time between when a node is expanded and when its parent was expanded. Multiplying by an estimate of the number of steps to the goal using h_{min} yields the **expansion-delay based** predictor d_{ed}. Another scheme (Hiraishi, Ohwada, and Mizoguchi 1998) measures *velocity*: rate (in units of time, or expansions) at which h_{min} decreases. Dividing h_{min} by the velocity yields a **velocity based** predictor d_v. The *NN* method estimates the current fraction of the of the search already performed with deep NN learning (Sudry and Karpas 2022). NN appears to be the state of the art w.r.t. RMSE. All these predictors can be naively converted into distributions by treating the prediction as absolutely certain.

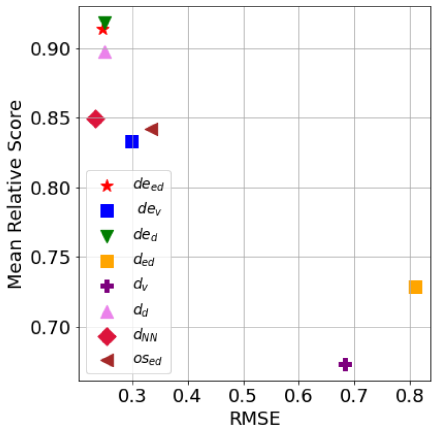

Figure 1: Mean Relative Score vs. progress pred. RMSE

Our novel contribution is to generate a relative-error distribution over instances of problems solved by A* for some point predictors; then use the error distribution to create a remaining A* runtime distribution from the point predictors.

All these methods play the BPSG over 20 odds ranging from 0.05 to 0.95, and average scores are computed over numerous problem instances, for sliding tile puzzle and pancake domains. Typical results are shown in Figure 1, for pancake using the Gap heuristic. Our new error-distribution corrected predictors d_{ed} and de_{ed} achieved the best BPSG scores. More important, however, is that in terms of RMSE the NN-based scheme (d_{NN}) was the winner, but still scored significantly worse than the (much simpler) distribution-based methods. This shows that apparent anomalous cases as shown in the above example actually occur in practice.

Acknowledgements

We are grateful for support from the US-Israel BSF (grant 2019730), US NSF (grant 2008594), EPSRC project CO-HERENT (EP/V062506/1), and Frankel Center at BGU-CS.

References

Dionne, A. J.; Thayer, J. T.; and Ruml, W. 2011. Deadline-Aware Search Using On-Line Measures of Behavior. In *SoCS*.

Hiraishi, H.; Ohwada, H.; and Mizoguchi, F. 1998. Time-Constrained Heuristic Search for Practical Route Finding. In *PRICAI*. Springer.

Shperberg, S. S.; Coles, A.; Karpas, E.; Ruml, W.; and Shimony, S. E. 2021. Situated Temporal Planning Using Deadline-aware Metareasoning. In *ICAPS*.

Sudry, M.; and Karpas, E. 2022. Learning to Estimate Search Progress Using Sequence of States. In *ICAPS*.

Thayer, J.; Stern, R.; and Lelis, L. 2012. Are We There Yet? — Estimating Search Progress. In *SoCS*.