Simulation experiments for hide-and-seek with different seeker distribution update strategies

Tirtharaj Dash

January 18, 2020

1 Setup

We perform some simulation experiments for Hide-and-Seek with three different **seeker distribution update strategies**:

(1) **No update:** No update of the seeker distribution (leads to hide-and-seek with replacement results).

(2) **Uniform update:** Open a box, distribute its probability mass to every other unopened boxes, make its probability 0 (hide-and-seek without replacement).

(3) **Hot-cold update:** The seeker updates its probability distribution based on whether it opened a cold box or a hot box. A cold box is a box for which the performance of a box ($perf$) is less than the cold threshold (θ_c) and a hot box is a box with performance greater than a hot threshold (θ_h). We devise the following procedure for update:

1. Open a box i
2. If $perf(i) \geq \theta_h$: distribute its probability mass to all the unopened boxes in its neighbors.
3. If $perf(i) \leq \theta_c$: distribute its probability mass to all the unopened boxes except its neighbors.
4. If none of 2 or 3: distribute its probability mass to all the unopened boxes.
5. Repeat 1–4 until the hider is found.

The seeker update strategy (3) requires that the hider distribution falls into some continuity assumption. That is: the probability mass of a neighborhood of a box are in monotonic relationship to the probability of that box. This is a realistic demand and clauses which are related with each other are monotonic in their performance in some fashion. We construct such a hider distribution (H) with the following code[1].

2 Experiments

Parameter setting The experiments are performed for number of boxes $n = \{1000, 2000, 3000\}$. The maximum hiding trials is set at 1000. We call it a **failure**, if the hider is not found within n searches by using the seeker distribution. The neighborhood size (nbd) is varied as {1,2,3}. For all the experiments reported here, we define performance of a box by: $perf(i) = \frac{h_i}{\max(h_1,\ldots,h_n)}$, where $H = \{h_1, \ldots, h_n\}$ is the hider distribution. The thresholds are fixed at $\theta_h = 0.80$ and $\theta_c = 0.4$. The proportion of boxes that have high probability mass (spikes in H) are fixed at 50%. This says that 10% boxes are highly likely to contain the hiding locations.

[1] https://github.com/tirtharajdash/multimodalGaussianDistro

Results The mean and standard deviations of misses are calculated only for successful runs i.e. the hider was found by the seeker within maximum of n look-ups. Otherwise, it was treated as a failure and this result was not included for statistics. Below, we report results for each seeker update strategies.

choiceUpdS	SuccessRate	mean(misses)	sd(misses)
		$n = 1000$	
1	0.617	410.929	278.666
2	1.000	507.848	284.743
3 ($nbd = 1$)	1.000	470.455	269.662
3 ($nbd = 2$)	1.000	481.031	273.743
3 ($nbd = 3$)	1.000	496.920	281.998
		$n = 2000$	
1	0.636	825.852	570.932
2	1.000	1032.257	582.563
3 ($nbd = 1$)	1.000	941.797	559.658
3 ($nbd = 2$)	1.000	935.739	552.096
3 ($nbd = 3$)	1.000	993.009	565.132
		$n = 3000$	
1	0.646	1290.207	844.649
2	1.000	1489.962	867.514
3 ($nbd = 1$)	1.000	1366.866	800.812
3 ($nbd = 2$)	1.000	1430.131	820.963
3 ($nbd = 3$)	1.000	1429.897	840.993

Figure 1: Average number of misses for three seeker update strategies

Interpretation The results suggest that the seeker update strategy (3):

- reduces the average number of misses in comparison with search without replacement for which the expected number of misses is $\frac{n-1}{2}$ (i.e. choiceUpdS= 2 in tables).
- Increasing the *nbd* a box increases the average number of misses in almost all the cases.