Thesis Model

Jonas Nelle

July 2020

Contents

1 Simulation

This simulation allows for an arbitrary number N of actions in the environment. It currently assumes that the agents have access to the context-free values $\hat{V}$ for all actions. We might want to discuss how to remove this assumption, although my initial thinking is that doing so would be highly complex.

I will specify the decision-making process of the agent implemented at `https://github.com/jonasalexander/thesis`. Everything that I describe is repeated for each trial.

1.1 Generative Model

The context-free values $\forall i \in [1, N]$, i.i.d. $\hat{V}_i \sim N(\mu, \tau)$. The default values used unless specified otherwise is $\mu = 0$, $\tau = 1$. These $\hat{V}_i$ are re-sampled for each trial to avoid effects of the randomness of these N values (but the agent gets all $\hat{V}_i$ for free, so it's like each sample is an agent in equilibrium in an environment where they have a lot of experience and hence know all the $\hat{V}_i$). Given these $\hat{V}_i$, the generative model then samples the context-specific values: for each i, $V_i \sim N(\hat{V}_i, \sigma)$. The default value here is $\sigma = 1$ unless specified otherwise.

The environment also has an evaluation cost, which defaults to 0.2.

1.2 Fixed K Agents

These agents select the best k actions based on $\hat{V}_i$. They then achieve utility of the best action among these k, minus k times the cost of evaluation.

1.3 Dynamic Agents

As the fixed-k agents, the dynamic agent starts out with access to $\hat{V}_j$ for all actions $j \in [1, N]$.

The agent then iterates through some (or all) actions i in order of their $\hat{V}_i$. At each iteration, the agent keeps track of the value of the best action evaluated so far V^b and compares it to the expected value of continuing to evaluate more actions V^e. If V^b is higher, the agent stops evaluating and chooses the action with the best value so far (and achieves utility equal to the utility of that action, minus the total number of actions evaluated, times the cost of evaluation). In other words, the agent the recursively evaluates the following expressions, after having evaluated i actions (in order of their $\hat{V}$):

$$\begin{cases} V_i^b \\ \max(V_i^b, \max\limits_{i<j\leq N} V_j - (j-i-1)c) - c \end{cases} \tag{1}$$

where the second term in the max() expression ($\max\limits_{i<j\leq N}$...) is V^e.

Most of the complexity resides in the way in which the agent calculates V^e, the expected value of continuing to evaluate more actions. The agent does so by first sampling the posterior distribution of V_j, the normal distribution $V_j|\hat{V}_j \sim N(\hat{V}_j, \sigma)$ (for each action j not yet evaluated, i.e. $j > i$). By default the agent takes 1000 samples to create this empirical distribution. Then, the agent "floors" these distributions using the best value so far, V^b (this changes the variance by introducing a floor, i.e. chopping off the left part of the distribution). Finally, the agent subtracts the cost of evaluation from the distributions of the values of actions (more cost, the higher j is) - this just changes the mean of the distribution. Critically, the last step has to be done after flooring because the agent has to pay the entire cost of evaluation, even if they end up taking the first action they evaluated.

1.4 Heuristic Agents

Instead of taking 1000 samples to approximate the distributions of evaluating at least 1 more action (possibly many times, if the agent continues to evaluate and thus recurses to the same decision), what heuristics could the agent use?

As we discussed, it seems like UCB is not applicable because UCB works only if the sample was stochastic. In our current version, the agent gets the exact value of the action when they evaluate (and not a sample).

One of the nice thing with the simulations we have set up is that all we need to do is come up with ideas and we can run them and see how well they do.

1.4.1 Satisficing/Threshold

This model has some threshold level and settles for an action when it has found one that is above this threshold level.

The threshold level could be related to the median (past experiences), or a more complicated function of both historical mean and variance (conceptually we want this to be historical mean/variance in order for it to make sense that an agent would be able to learn this threshold, but in practice I assume we can just use true values, assuming the agent has a "warm start").

We may also want to modify the threshold so that our agent has an upper bound on how many actions they will evaluate, enabling them to move on in situations where none of the actions are above the threshold.

1.4.2 Value of Evaluating

This model tries to approximate some function $f(V^b, i)$ which takes as input the current best value V^b and the number of actions already evaluated i and returns an estimate of the expected value of evaluating another option. This value $f(V^b, i)$ could then be compared to the cost of evaluation, resulting in the agent taking the current best if $f(V^b, i)$ is less than the cost of evaluation.

We could try to use general smooth function learning to have the agent approximate $f(V^b, i)$ (though it's not clear that this is any computationally simpler than our initial dynamic agent). We might also just use interpolation and constant extrapolation using some set of simulated "past experiences" - i.e. for values of V^b between two values of V^b the agent has already experienced, for a given i, the agent assumes f will be in between the experienced values of evaluating another action.

2 Results

These charts are similar to the super-confusing mega-chart we had before, except that they separate out the different units and have separate pair-wise comparisons - I found that comparing the agents all at once was much less helpful.

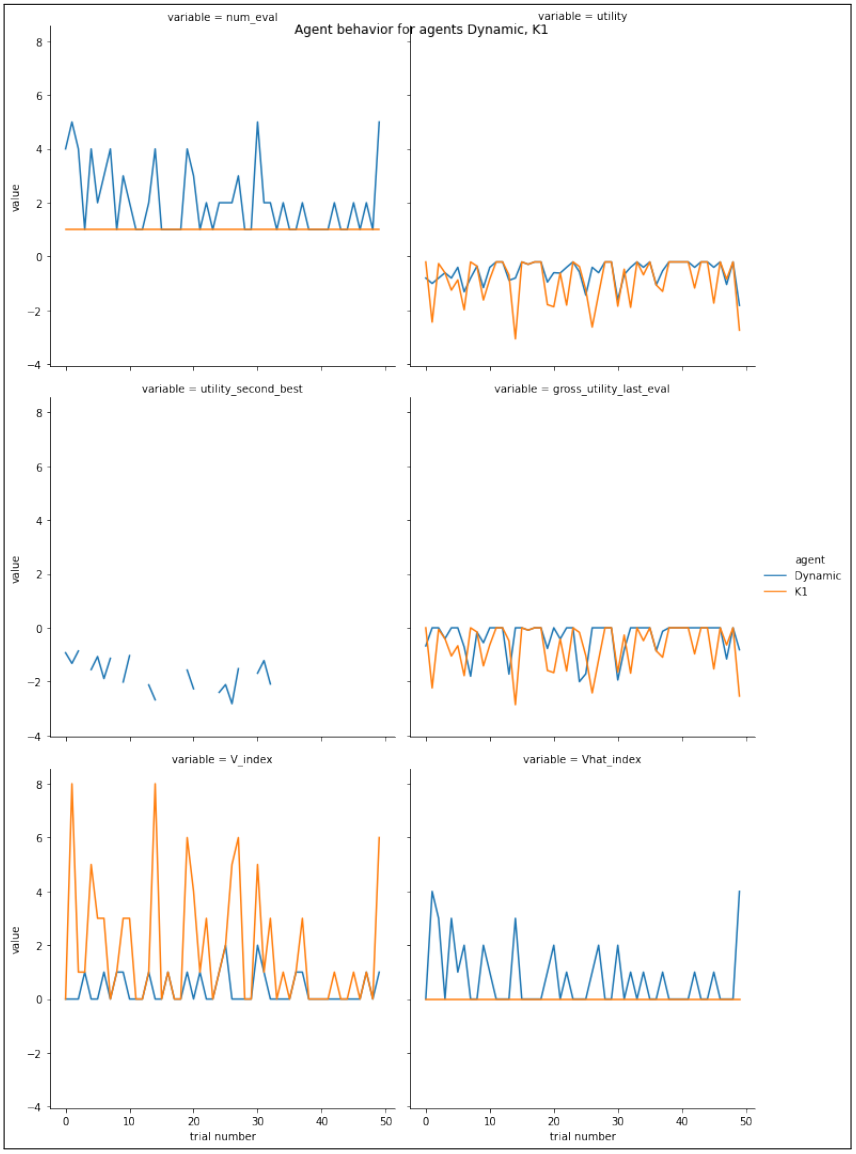

Figure 1: Trial-level data for dynamic vs fixed-K1 agent

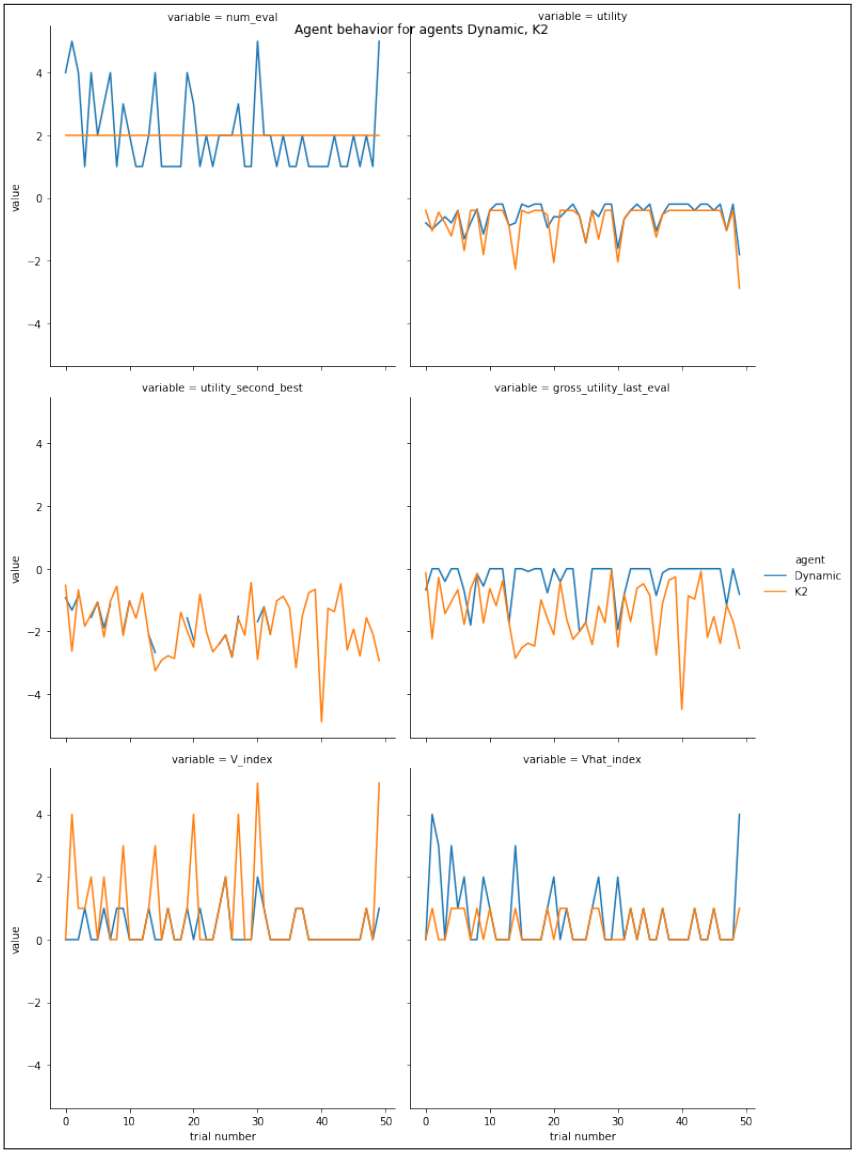

Figure 2: Trial-level data for dynamic vs fixed-K2 agent

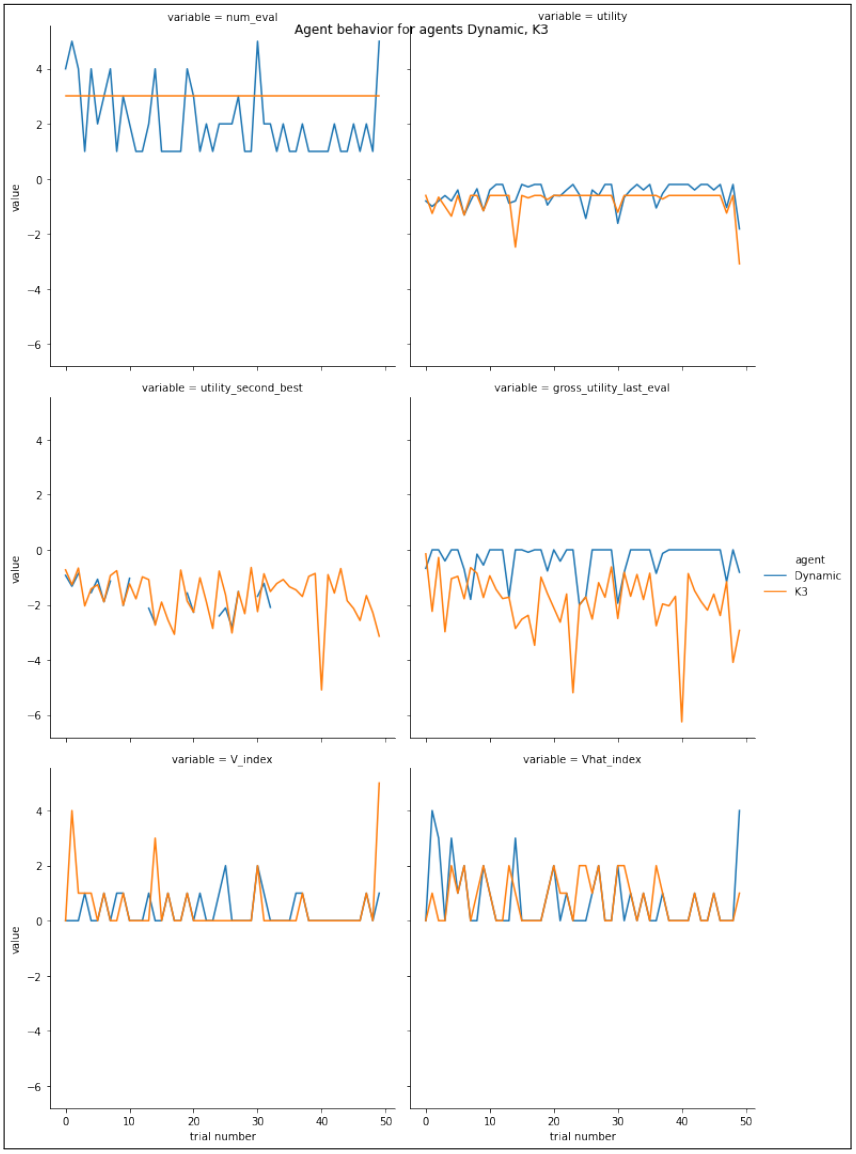

Figure 3: Trial-level data for dynamic vs fixed-K3 agent

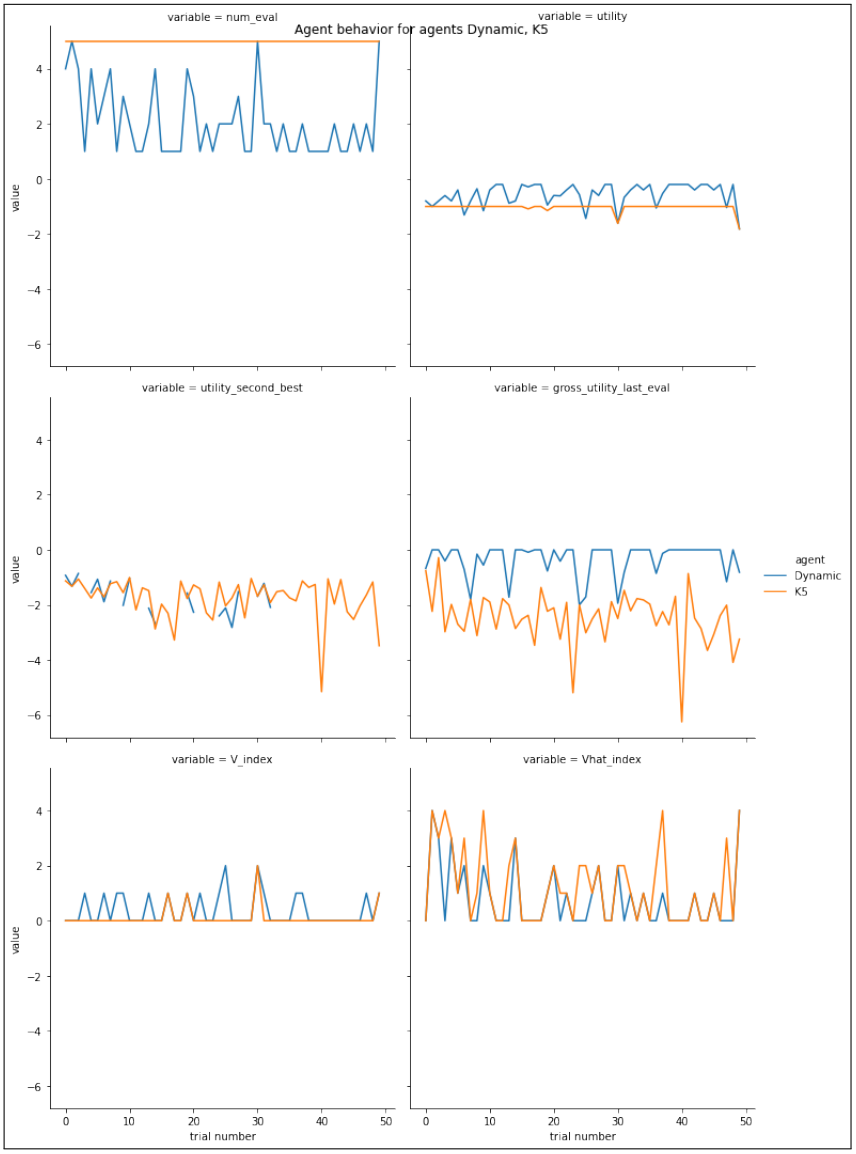

Figure 4: Trial-level data for dynamic vs fixed-K5 agent

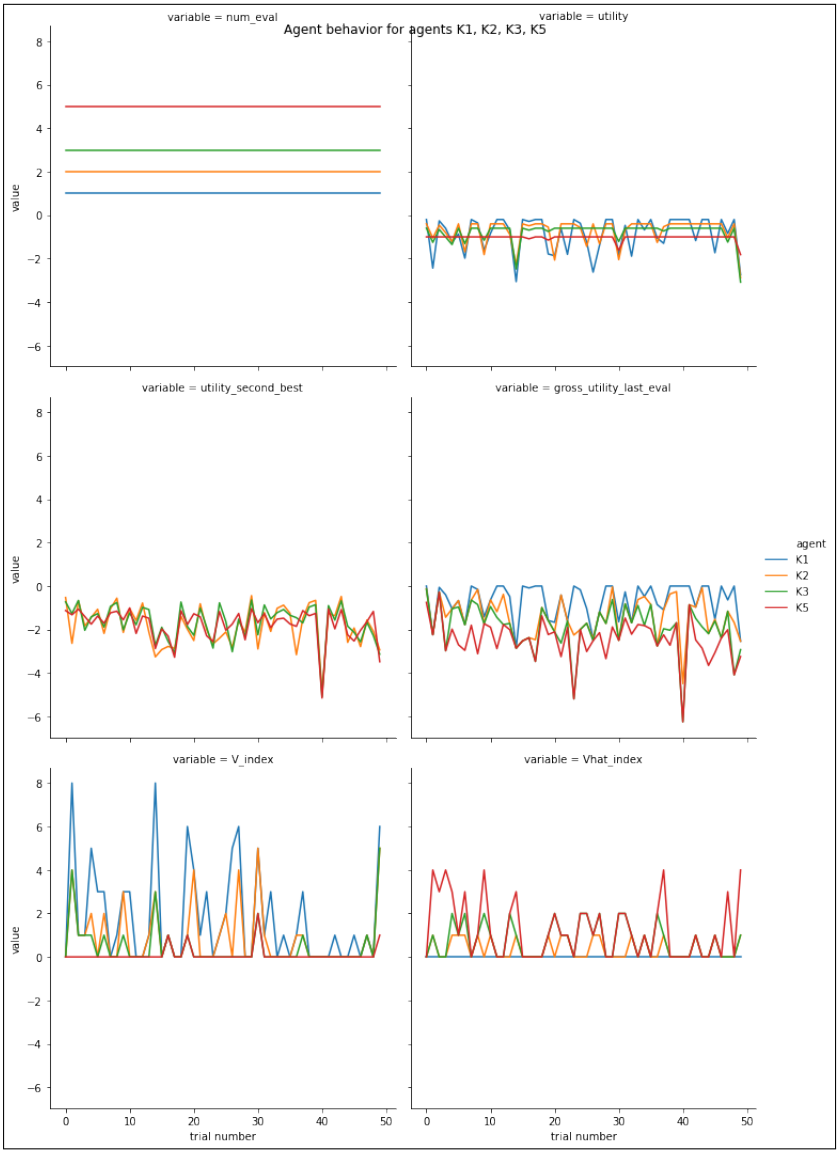

Figure 5: Trial-level for fixed-K agents (1, 2, 3, 5)

3 Empirical Predictions (With Simulation Results)

Unless otherwise noted, I ran the below experiments with 5000 samples.

3.1 With Order

3.1.1 Rank of Action Chosen

If we have access to the order in which items were evaluated then the dynamic model does indeed make a strong, unique prediction for the rank according to $\hat{V}$ of the action chosen. (I use "index" and "rank" interchangeably, because in an ordered list the index is the rank).

For both the dynamic and fixed-K agents, they are in general much more likely to pick actions earlier in the list of actions evaluated because actions are ordered by their context-free values $\hat{V}$ and, on average, the higher context-free value actions are also better in the specific context.

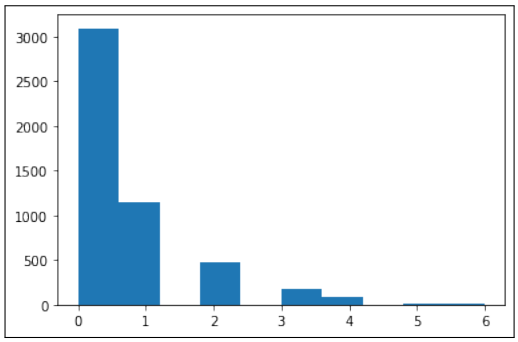

Figure 6: Distribution over rank of actions according to $\hat{V}$ for dynamic agent

The key here is that the fact that a dynamic agent has evaluated some number K of actions is itself information that is special. The dynamic agent only continues evaluating actions when the later actions are likely better. Thus, if we compare all the trials in which the dynamic agent ended up evaluating K trials to the fixed-K agent, we will see different profiles of the rank of the action chosen according to $\hat{V}$. Of course, in the specific trials where the dynamic agent and the fixed-K agent evaluate the same number of actions, they are completely indistinguishable (make the same decision, achieve the same utility, etc.). The difference lies in the other trials in which the dynamic agent does not evaluate K actions.

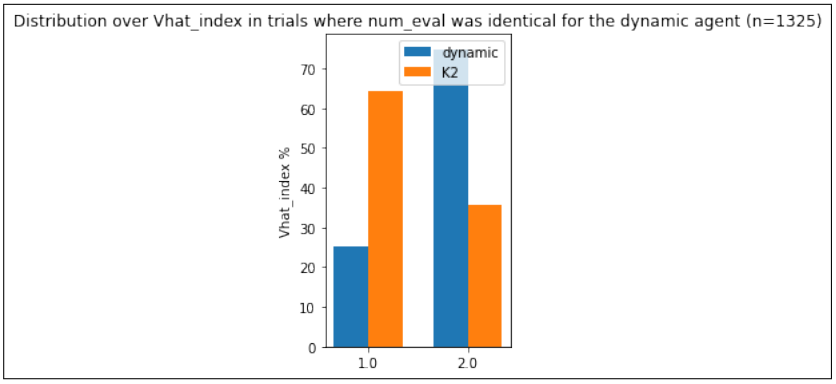

Figure 7: Distribution over rank of actions according to $\hat{V}$ for dynamic and K2 agents

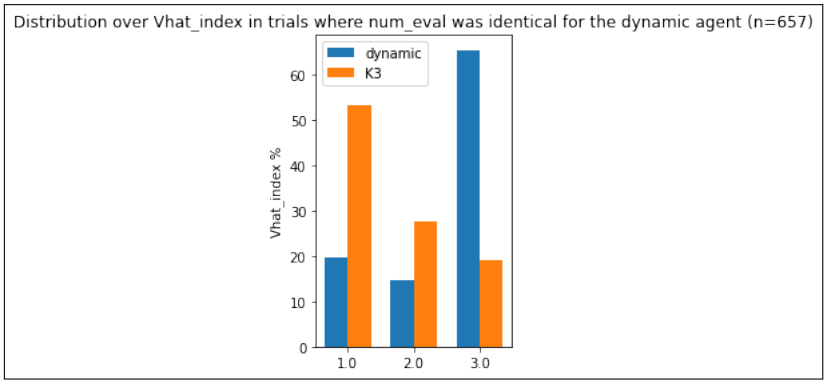

Figure 8: Distribution over rank of actions according to $\hat{V}$ for dynamic and K3 agents

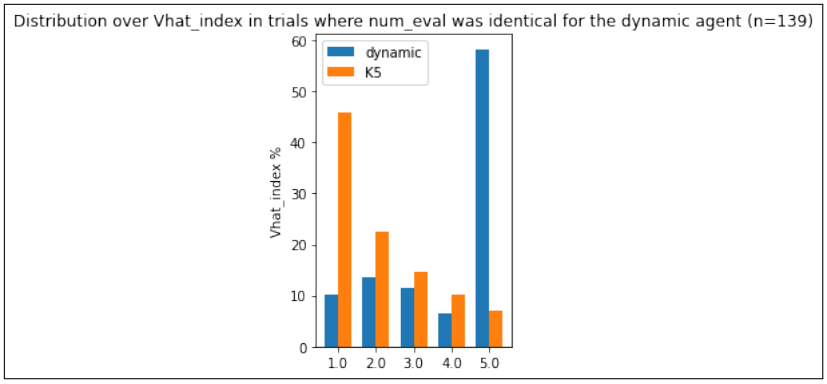

Figure 9: Distribution over rank of actions according to $\hat{V}$ for dynamic and K5 agents

Seeing the results for higher K makes is apparent that for the fixed-K agents, the rank according to $\hat{V}$ of the action they are likely to choose decreases and has a similar profile in shape as the chart above for the dynamic agent overall. However, the dynamic agent is most likely to choose the last one, and about equally likely to choose the others. The effect becomes more pronounced the higher K, it seems (n is relatively low because there aren't that many trials in which the dynamic agent evaluates 5 actions - 139 out of 5000 trials in this case, given the default parameter settings).

We can also compare the rank of actions chosen according to V. A rank of 1 means that the agent chose the best action. As we can see, The dynamic agent's performance is the same in all the pairwise comparisons below. The dynamic agent barely outperforms the K2 agents in choosing the best action, but is able to avoid the long tail of bad choices the K2 agent has. Compared to K3 and K5, the dynamic agent more frequently chooses the incorrect options - the point here is that the dynamic agent does better overall because in many trials these fixed-K agents evaluate far too many options. The times where the dynamic agent gets the first as opposed to the second or third best option are outweighed by the benefit it reaps from evaluating less (those may also be cases where the delta between the context-specific value of the first and second option is (in expectation) less than the cost of evaluation, in which even an omniscient agent wouldn't want to evaluate and take the best.

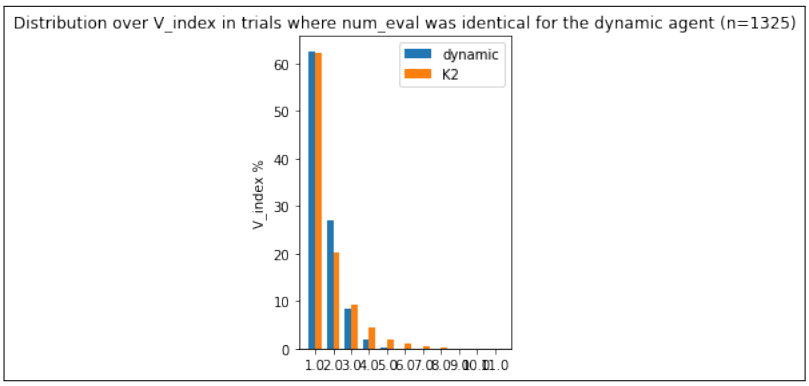

Figure 10: Distribution over rank of actions according to $\hat{V}$

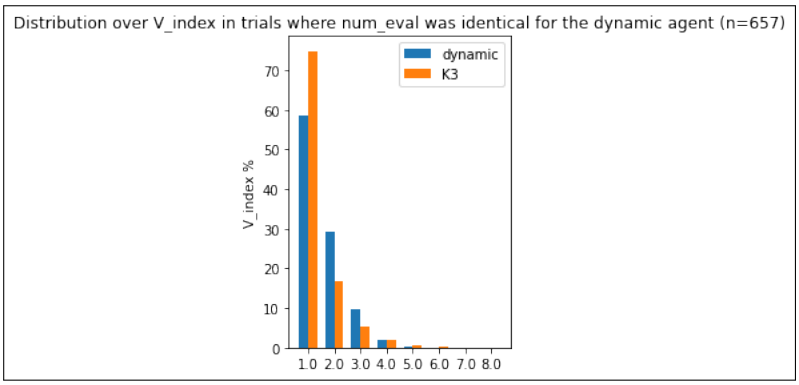

Figure 11: Distribution over rank of actions according to $\hat{V}$

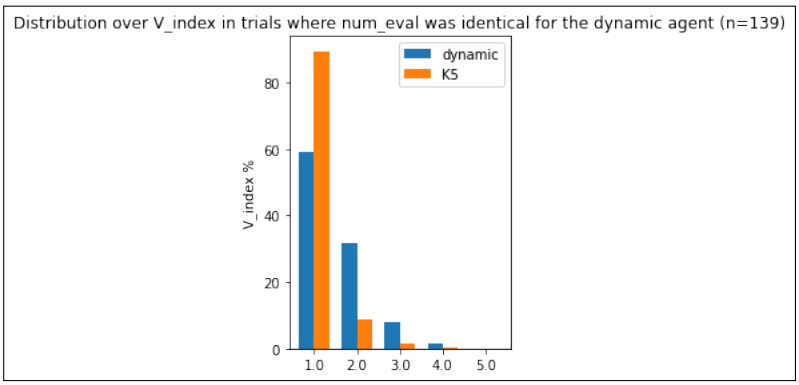

Figure 12: Distribution over rank of actions according to $\hat{V}$

3.1.2 Value of Last Action Considered

See the charts above under "Results" for the charts on a per-trial level. As noted in the header, note that these are gross utilities. I think we might actually want net utilities (?). Regardless, that only makes the advantage of the dynamic agent more pronounced. Other than "the dynamic agent does better", I'm not sure if there is a difference in kind here that we could use to distinguish between whether subjects act dynamically or not.

Here are plots of the full distribution of the gross (i.e. before subtracting the evaluation cost) value of the last action:

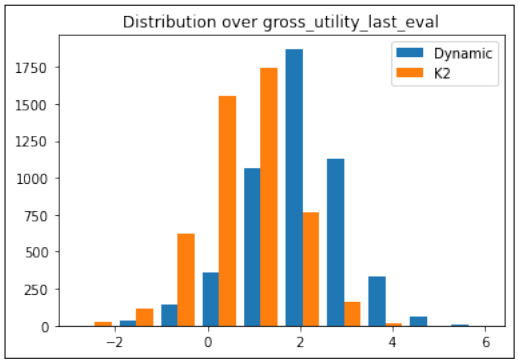

Figure 13: Distribution over the value of the last action considered

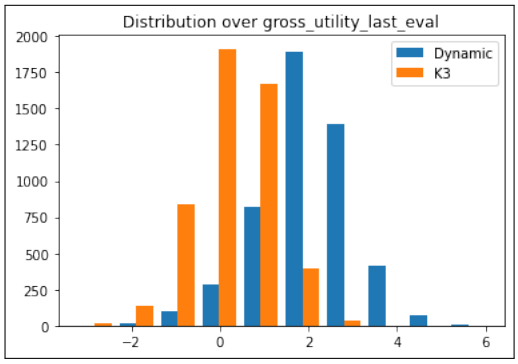

Figure 14: Distribution over the value of the last action considered

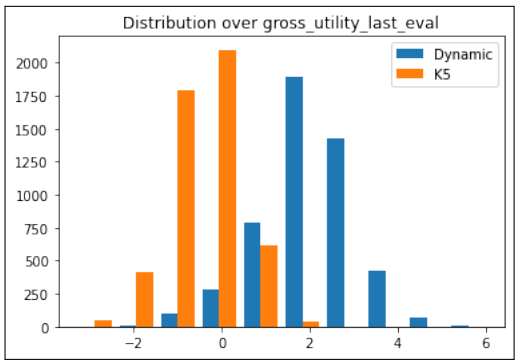

Figure 15: Distribution over the value of the last action considered

3.2 Without Order

In the dynamic model, when agents discover a good action, they will stop. This could lead to the prediction that the consideration sets constructed by dynamic agents are more likely to contain exactly one very good option. I tried to run simulations and measure the difference in context-specific value between best and second-best option evaluated (only in trials where more than one action is evaluated). I made the prediction that for the dynamic agent, the average will be greater than for fixed-K agents (this difference, between the context-specific value of the best and second-best option, should function as a measure of how much the best option is better than others).

Interestingly, I'm not even sure whether this holds, even after seeing the data. Looking forward to discussing.

Part of the complexity is that there are a bunch of different permutations on the exact metrics we look at. First, it is important to note that the K1 agent is irrelevant here because there is no value of the second best action considered. Likewise, we have to through out the trials of the dynamic agent in which the agent evaluates only one action.

The first question then becomes whether we only look at the fixed-K agents in those same trials, or still look at them across all trials. It turns out that the answer matters:

First, the results when we take all trials of the fixed-K agents:

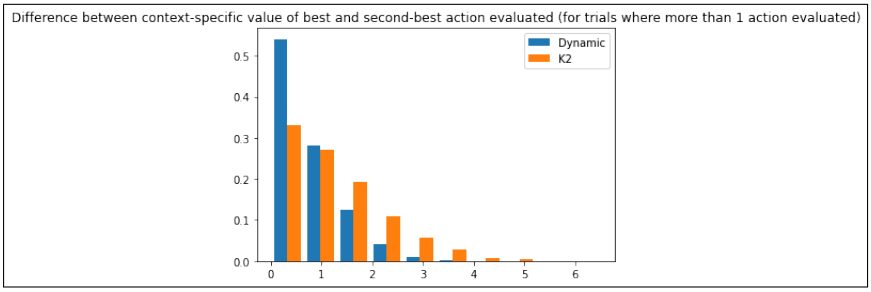

Figure 16: Distribution over the value of the last action considered

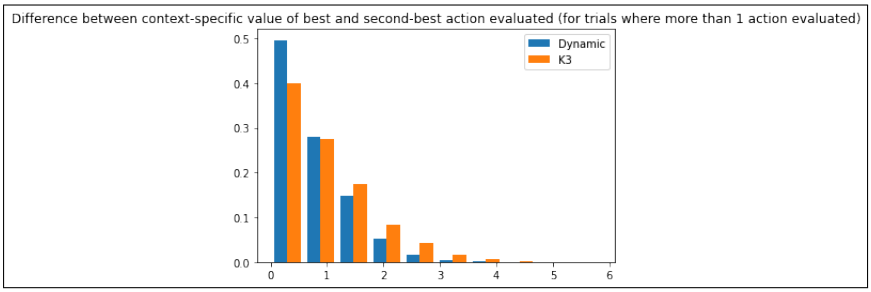

Figure 17: Distribution over the value of the last action considered

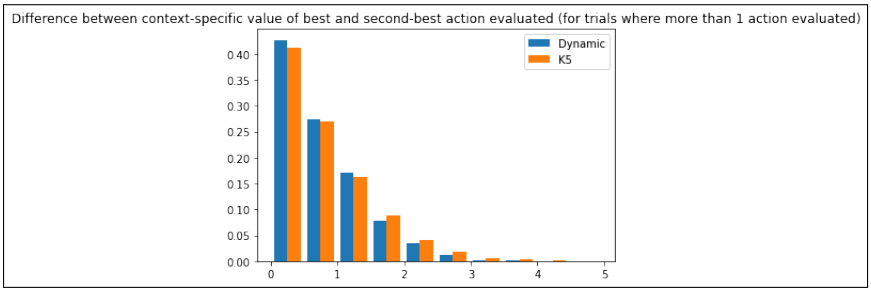

Figure 18: Distribution over the value of the last action considered

If anything, it looks like the dynamic agent has on average a lower difference between best and second best action! This effect decreases as the K for the fixed-K agent increases.

And now where we only look at trials in which the dynamic agent evaluated more than 1 action:

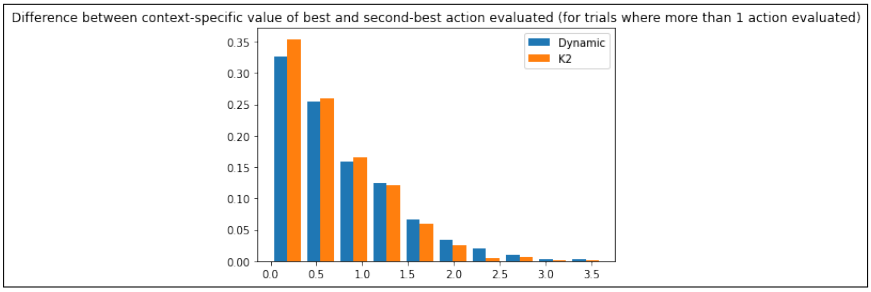

Figure 19: Distribution over the value of the last action considered

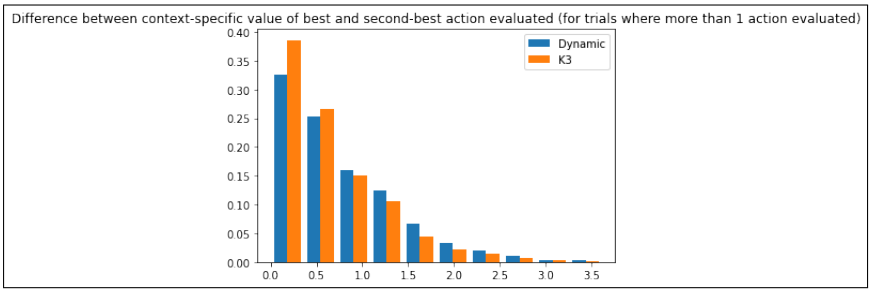

Figure 20: Distribution over the value of the last action considered

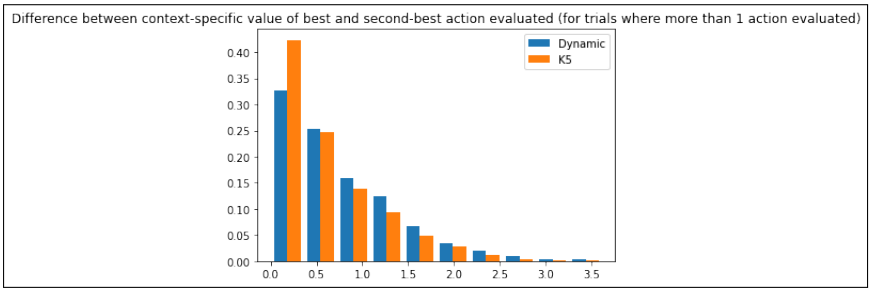

Figure 21: Distribution over the value of the last action considered

Here I see the opposite effect, where the dynamic agent has a long tail of actions where it has a much larger difference between the best and second best action, and that this effect grows as K for the fixed-K agent grows. This trend makes sense to me because inherently the more actions an agent evaluates from the same Gaussian distribution, the more likely it is that those values are closer together.

3.3 Manipulation

Note: Didn't get to the stuff below

Manipulation is good, better design - Many vs few options good; switch between them - are people able to adapt? - As sigma changes, that changes optimal K - You can't know ex ante whether high or low sigma context, but there is variability of that kind - Dynamic model able to handle that better (?) - Easy to do in the months paradigm - Associate random values with each of the month - Explore two training regimes: 1. Train people that may is most valuable month, November next most valuable - Train so that February is most valuable, etc... - Dependent measure is number of items that come to mind - Variability should be higher for dynamic

Dip a toe in the water of experimental predictions - Sure thing: use randomly assigned month-values (correlated with true value or anti-correlated) and detect relationship between correlation and number of months that came to mind - How could fail - Could be underpowered - Effects Adam observed: reward on probability of being in consideration set - On the order of 4,5,6% to have thought of something than not - Simulation - Generate dataset where just sample 3-4 months randomly, ask P(set contains may/November - by far the most common answers given) - May/November vs months at the bottom of checkmark - What is their probability of being in the consideration set - Alternative to strong dynamic model - Heuristic: keep sampling until get something of "V" or better (letter close enough to goal) - Monte Carlo simulation, in each round sample months according to Adam's chart where May/November either in best/worse position - How many did I sample before hitting on either may or November - How many observation would it take for me to reliably detect that on average the set sizes were smaller when in the better vs worse position - What would sample size/power need to be like? - Maybe not that useful - suppose: null is that they think of all months with equal probability - dynamic: strong - Composition of consideration sets of size 1 - Overrepresented by good solutions to the word problem - !-¿¿ Adam is about to run a huge experiment using this paradigm - Ask him to get people to report order to us? - If it can be done in a way so that it is totally separable

Take some references in Adam's paper - Anyone collected data on order - Many predictions boil down to the order, so that data would be helpful

References