**Cumulative culture model**

**Learning algorithms**

A cultural system consists of n traits, each characterised by s complexity levels. For each complexity level, agents learn which trait provides the best payoff. The payoffs for all traits at each complexity level are drawn from an exponential distribution with rate 1 , then squared, doubled, and rounded, so that each payoff is an integer. While the raw payoff for a trait at level s does not predict the payoff of the same trait at level s+1, we implemented one-step dependencies between traits to restrict how much of the space of cultural traits agents can explore and model the effects of path dependence characteristic of cumulative culture. Thus the payoffs at level s are adjusted using a gaussian window that restricts the payoffs at level s depending on what trait the agent picked at level s-1.

1. Penalty dependencies

The payoffs for traits 1 to *n* at level s are penalised according to a gaussian window such that the payoffs decrease when the difference between the trait chosen at level *s-1* and trait chosen at trait *s* increases. For instance, if trait *i* was chosen at level *s-1* and trait *j* was chosen at level *s*, then the payoff this agent will get at level *s* is:

.

Then, the total payoff of an agent who knows s levels is

We implemented agents with different abilities to assess the depth of dependencies across complexity levels. The simplest agent is a one-step learner. This algorithm ignores dependencies entirely and, for each complexity level, picks the trait with the highest payoff. A more advanced algorithm is a two-step learner, which assesses payoffs two levels at a time, thus taking into account dependencies, and picks the trait that maximise the sum of the two payoffs. The two step learner can be a “chunk” learner, assessing two levels at a time, learning the two traits that maximise the payoff for the two levels, then jumping to the next two levels, and so on, or it could be a “window” learner, which assesses two levels at a time, picking the two traits that maximise the payoff, but only learning the first, then moving on to assessing level 2 and 3, learning the trait at level 2 that maximises the payoff for the two levels combined, and so on. Similarly, a three-step learner assesses the sum of payoffs of 3 consecutive levels and either learns all three levels and advances three levels, in the case of a “chunk” learner, or picks the first and advances one level, in the case of a “window” learner. We implemented a four-step learner and a five-step learner as well, stopping at 5 because above this value the task becomes computationally intractable, although in principle a “perfect” learner would analyse all s levels at once.

1. Bonus dependencies

Instead of penalising the payoffs according to which trait was chosen at the previous level, we can add a bonus proportional to the payoff of the same trait at the previous level, which achieves an equivalent purpose, incentivising a narrow exploration of the cultural space. Similarly to the implementation of the penalty dependence, the payoff for level *s* if trait I was chosen at level *s-1* and trait *j* was chosen at level *s* is:

Just as before, we implemented a one, two, three, four, and five-step “window” learner.

**Task difficulty**

In order to assess how the difficulty of the task changes across the parameter space, we explored different values for the number of traits in the cultural space, the number of complexity levels, as well as the standard deviation of the Gaussian window. Our measure of difficulty in this setting is the payoff associated with the repertoire of each one the one, two, three, four, or five-step learners. The expectation is that a lower standard deviation will be associated with a lower payoff, as it restricts the space of cultural traits that can be explored and, implicitly, decreases the chances of finding the trait with the highest payoffs. Similarly, an increase in the number of traits should be associated with a relative decrease in payoffs, particularly for low values of the standard deviation. While a larger number of complexity levels will be naturally associated with a higher payoff simply because there are more payoffs to count, we have no expectations with regards to how it will affect the relative payoffs.

**Results**

**Algorithm comparison**

As expected, a higher standard deviation allows a wider exploration of the cultural space and leads to a higher payoff. This is true for all three algorithms presented below, as indicated by the final repertoire, as well as the total payoff associated with the repertoire.

It seems that the “window” learning algorithm performs better than the “chunk” learning algorithm under a penalty dependency structure, at least for lower levels of the standard deviation, when the task is harder. The window learning algorithm seems to achieve perfect performance.

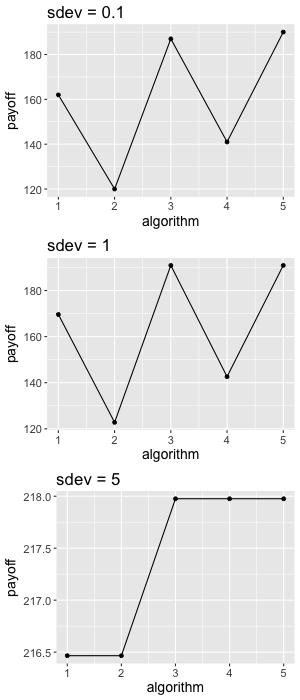
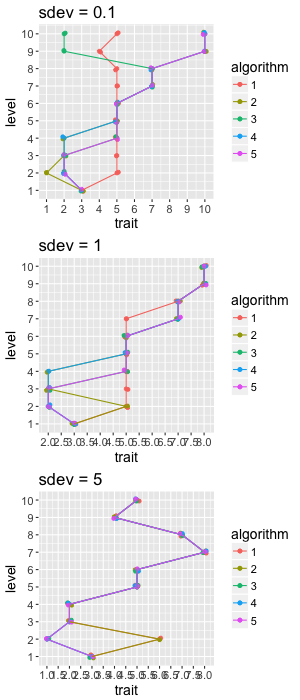


Fig. 1 – the path through the cultural space for all 5 **chunk learning algorithms** with a **penalty dependency** structure at different values of the standard deviation (left); payoffs associated with the five algorithms (right)

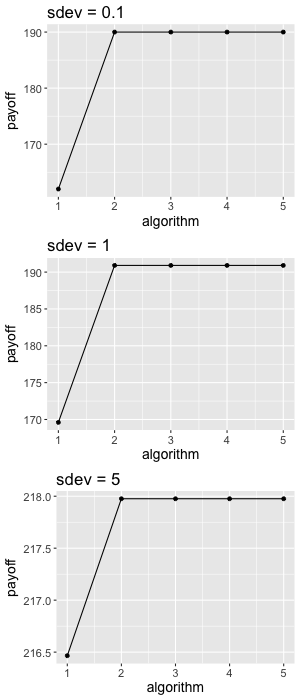
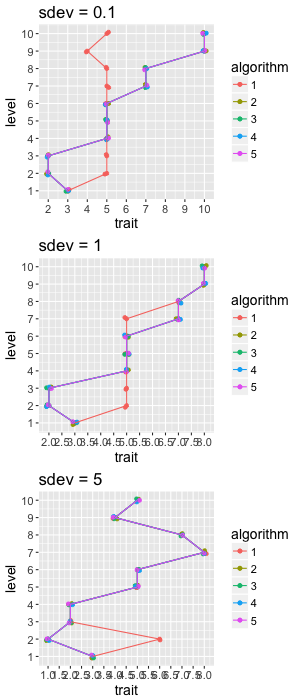


Fig. 2 – the path through the cultural space for all 5 **window learning algorithms** with a **penalty dependency** structure at different values of the standard deviation (left); payoffs associated with the five algorithms (right) – for the **same reward matrix as above**

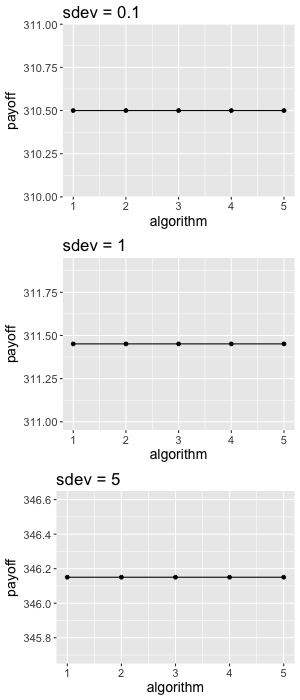
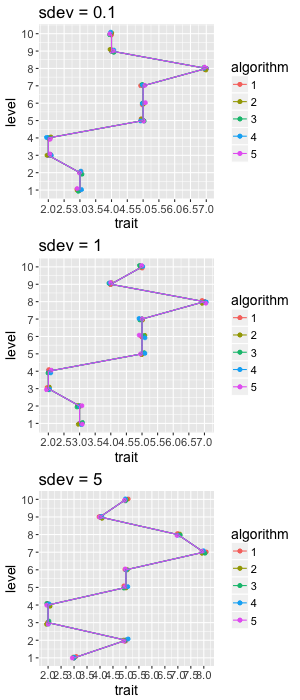


Fig. 3 – the path through the cultural space for all 5 **window learning algorithms** with a **bonus dependency** structure at different values of the standard deviation (left); payoffs associated with the five algorithms (right) – for the **same reward matrix as above**

**Standard deviation**

To assess the effect of the standard deviation more thoroughly, we ran 100 repeats of the three algorithms for varying values of the number of traits, number of levels, and the standard deviation, presented below.

For the chunk learning algorithms, each progressive increase in the number of steps the agent can see ahead and use to guide learning is associated with an increase in the payoff, at least for harder tasks at lower values of the standard deviation. The window learning with a penalty dependency structure, however, experiences a steep increase between the one-shot and two-shot learner, but very little improvement for higher values of foresight. Learners in a bonus dependency setting show very little, if any, improvement with the number of steps they can see ahead, potentially because of a ceiling effect.

**N = 5 traits and S = 5 levels**

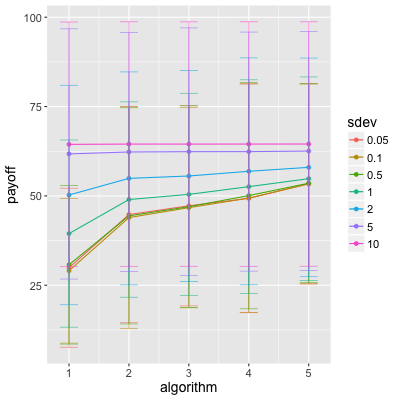
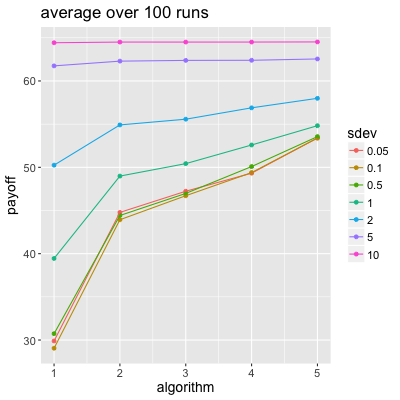


Fig. 4 – average payoff for all 5 **chunk learning algorithms** with a **penalty dependency** structure for a range of values of standard deviation for **5 traits and 5 complexity levels**

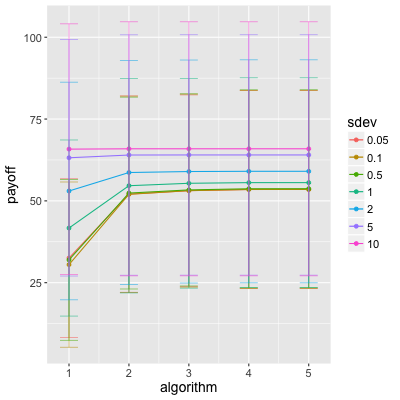
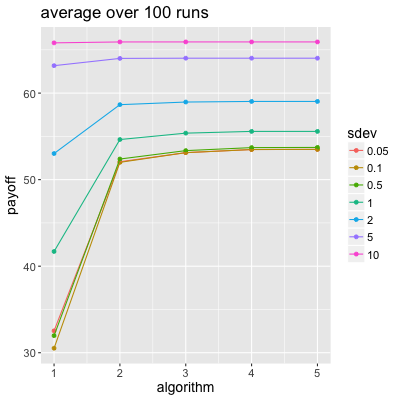


Fig. 5 – average payoff for all 5 **window learning algorithms** with a **penalty dependency** structure for a range of values of standard deviation for **5 traits and 5 complexity levels**

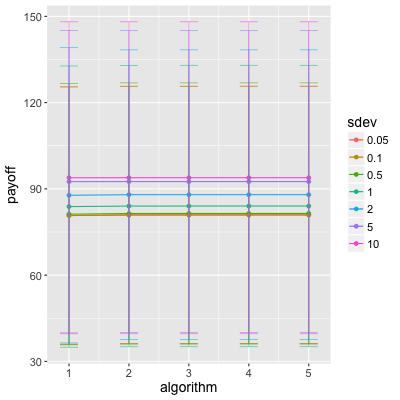
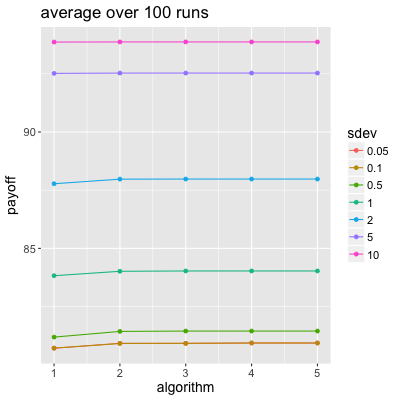


Fig. 6 – average payoff for all 5 **window learning algorithms** with a **bonus dependency** structure for a range of values of standard deviation for **5 traits and 5 complexity levels**

**N = 10 and S = 10**

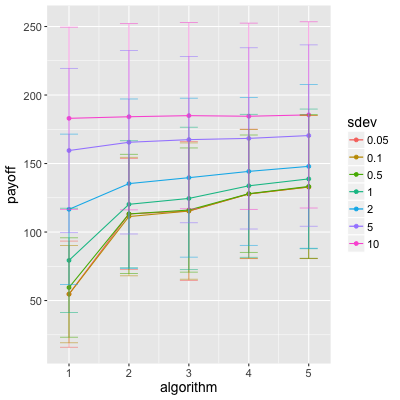
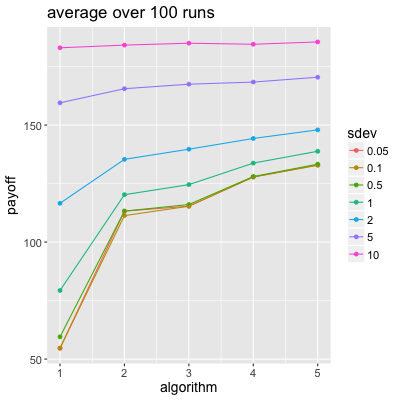


Fig. 7 – average payoff for all 5 **chunk learning algorithms** with a **penalty dependency** structure for a range of values of standard deviation for **10 traits and 10 complexity levels**

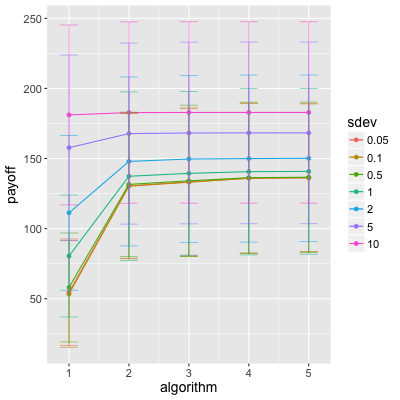
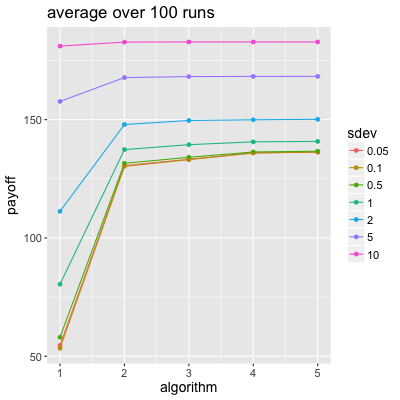


Fig. 8 – average payoff for all 5 **window learning algorithms** with a **penalty dependency** structure for a range of values of standard deviation for **10 traits and 10 complexity levels**

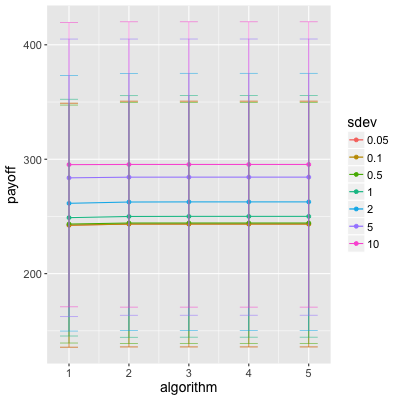
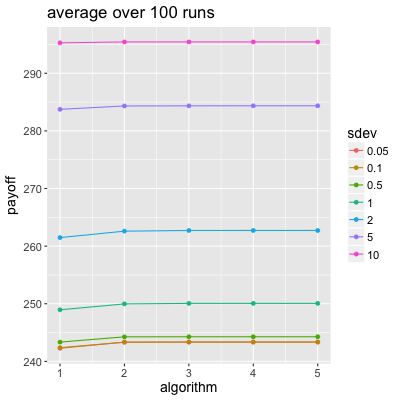


Fig. 9 – average payoff for all 5 **window learning algorithms** with a **bonus dependency** structure for a range of values of standard deviation for **10 traits and 10 complexity levels**