Introduction to artificial intelligence

Academic year 2018-2019

Pacman Part 3 : Reasoning over time

Bolland Julien - Gilson Maxence

(s161622 - s162425)

1 Convergence of the belief state

To analyse the convergence of our algorithm on the *observer.lay* map, we chose to use the Shannon's entropy method. To do so, we ran one hundred times the game for each $w \in [1, 3, 5]$ and each $p \in [0.01, 0.25, 0.5, 0.75, 1]$ [1]. In total, we ran $100 \times 3 \times 5 = 1500$ games [2] in order to have enough data to get reliable conclusions about our convergence. Note that for each game, we only use one ghost, but the conclusions we make below are the same for each present ghost in the game, if we chose to use more of them.

Thereby, we produced the graphs as shown in Figures 1 to 3 in annex, at the end of this report.

First of all, we can see entropy as a measure of disorder. Indeed the higher is the entropy, the more disordered is the data. Oppositely, the lower is the entropy, the less disordered is the data. In our case, that means that our Bayesian filter is more precise, *i.e.* the sonar works best, if the entropy has a low value.

Then, when we compare the Figures 1 to 3, we can conclude that the pacman's sonar is more precise when we have a lower w and a higher p. In other words, we have a better prediction of the ghost's location when the sonar's discrete distribution $(2w + 1)^2$, which is around the unknown position x_t of the ghost, is low and when the ghost is forced to go to the East direction as long as East is a legal move.

Indeed, having a low discrete distribution implies taking into account a smaller probability of being far from the true ghost's position. This leads to a better approximation of the true ghost's position. Thus, the sonar is better. Moreover, when we know that the ghost will choose the East as the next direction as long as it is a legal move (which happens when $p = 1$), we can deduct his future movement. This leads to a better approximation of the true ghost's position.

2 Discussion

First of all, let's discuss about the graphs we obtained in Figures 1 to 3. We chose to use 50 time steps for the three graphs. We could have taken more time steps, but after some tries, we noticed that convergence is the same for more time steps. Besides, 50 time steps allowed us to spare more time playing each of the 1500 games we ran. The conclusions we made above are then correct for these time steps. Then we can see the entropy plummeting after one or two time steps. This behaviour is due to the original distribution of the beliefState matrix. Indeed, at the very first iteration of the game, the beliefState matrix is uniformly distributed, because the ghost can appear in any game's square, as far as it is a legal position. Then, the first iteration is made, the ghost's positions are approximated and we can then have a better prediction of these positions.

1. Note that we didn't use $p = 0$, because the program raised an "out of bound" error. Therefore, we approached 0 by 0.1.

2. Where 3 and 5 are respectively the number of w and p we considered.

Moreover, as we ran 100 times the same game configuration, *i.e.* we use the same w and p, we had to plot the average range in which the entropy appeared for each game. In other words, every points that are making the graphs are the mean of the 100 entropies we obtained in each game and at each time steps. Of course, for a same game configuration, the position of a ghost can be different and then the distribution can vary in many different ways. This implies that the entropy, at a certain time step, can take several values. Taking the mean of these values allows us to draw conclusions as a general behaviour, but we don't have to forget that a behaviour of a game distribution can be totally different from one game to another.

As we ran a lot of games, we noticed a strange behaviour of our Bayesian filter. After a while, the ghosts were going through the maze again and again, we saw that the filter took a mosaic shape, as shown in Figures 4 to 6. Actually, this behaviour happens when a ghost is trapped in a corner or particularly when the sonar gives the exact position of the ghost (in a corner or not). The ghost has therefore a maximum of 4 possibilities of movement, knowing that the actual position has a probability of 0 of being the next position. The probability to go to this or this particular direction is computed and then, at the position at $t+1$, it also has a maximum of 4 possible moves, etc.. All this leads, after a few time steps, to a mosaic shaped filter. It is easier to see this behaviour when $w = 3$, but we can observe it also for others w.

As a final discussion, in order to loose less computing time during the loops in our codes, we decide to browse the game matrix $\forall x \in [w, N-w]$ and $\forall y \in [w, M-w]$, because of the 'playable' game area, *i.e.* the area where the ghosts can move is contained between these boundaries.

3 Possible improvements

As possible improvement, we could run a lot more games to have a proper vision of our belief state's convergence. However this implies a much longer computing time and we chose a compromise between the running time of our tests and the obtained graphs, the latter being fully acceptable as it is.

Far over, we didn't take into account the case when an evidence is located on a wall. In our code, we considered an evidence located on a wall as a legal position and we computed the transition model with the legal positions surrounding this evidence's position. As a ghost can't be on a wall, this case shouldn't produce any transition model.

As a final improvement, we could have improved the value of the entropy at the beginning of the game. The beliefState matrix is a $N \times M$ matrix, which contains elements with the same value at the beginning of the game. This is due to a uniform distribution in the maze layout. As the 'playable' area is smaller than the $N \times M$ maze layout, the elements $e_{ij} : \forall i \in [w, N-w], \forall j \in [w, M-w]$ could be the same because of a uniform distribution on the playable area, and zero anywhere else. This would reduce the entropy value for the first time step.

Annex :

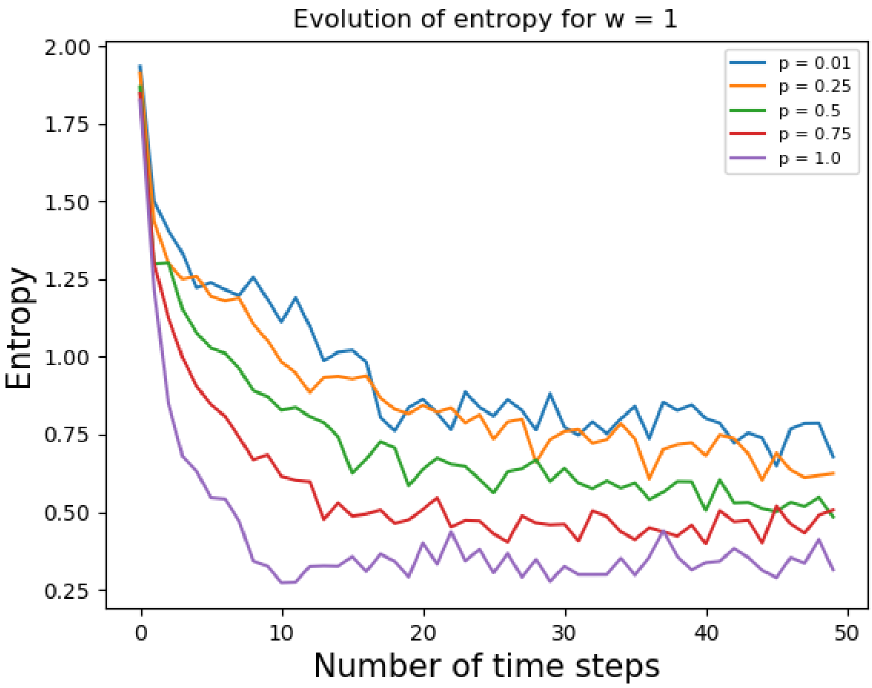

FIGURE 1 – Evolution of the entropy for $w = 1$ and $p \in [0.01, 0.25, 0.5, 0.75, 1]$

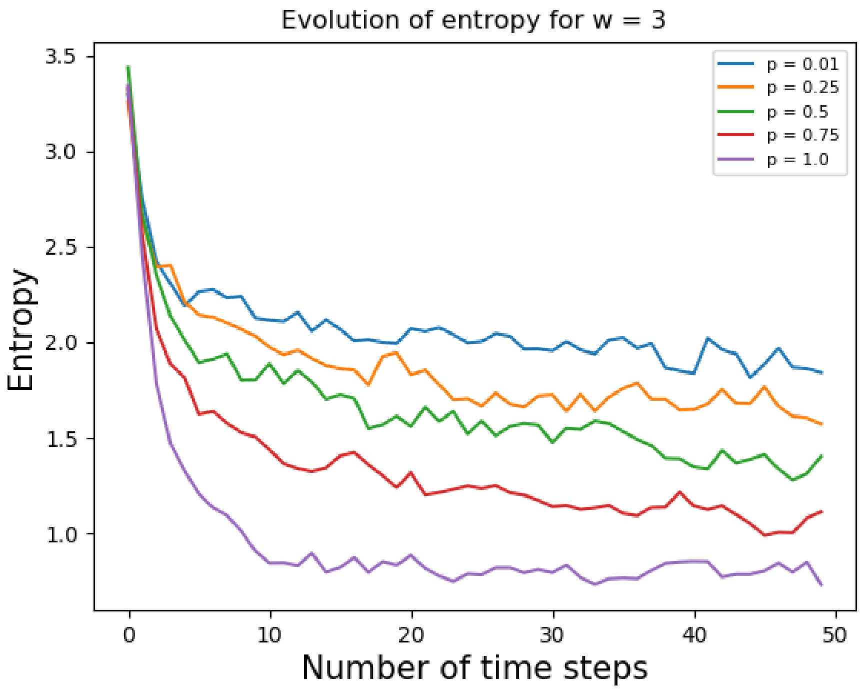

FIGURE 2 – Evolution of the entropy for $w = 3$ and $p \in [0.01, 0.25, 0.5, 0.75, 1]$

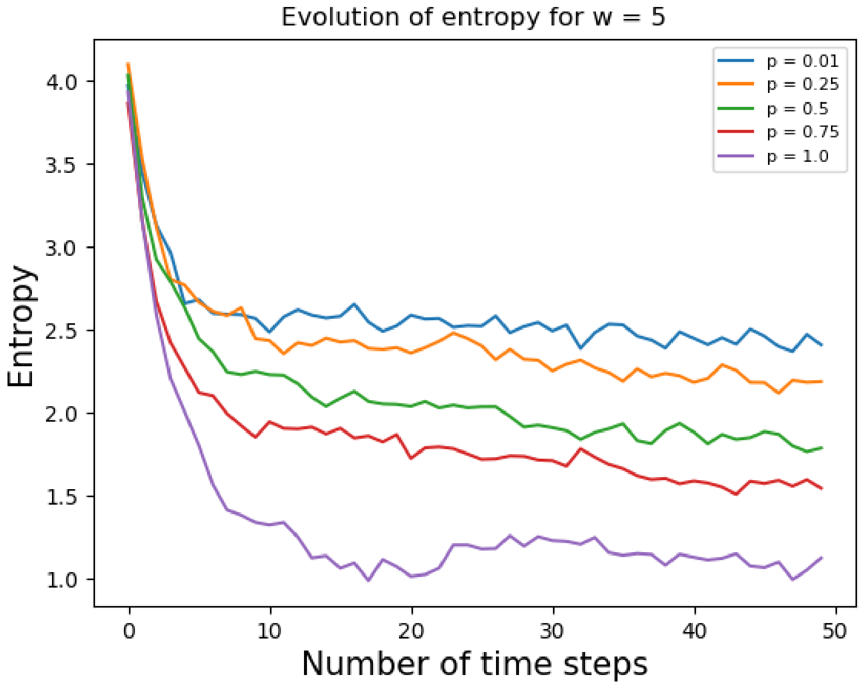

FIGURE 3 – Evolution of the entropy for $w = 5$ and $p \in [0.01, 0.25, 0.5, 0.75, 1]$

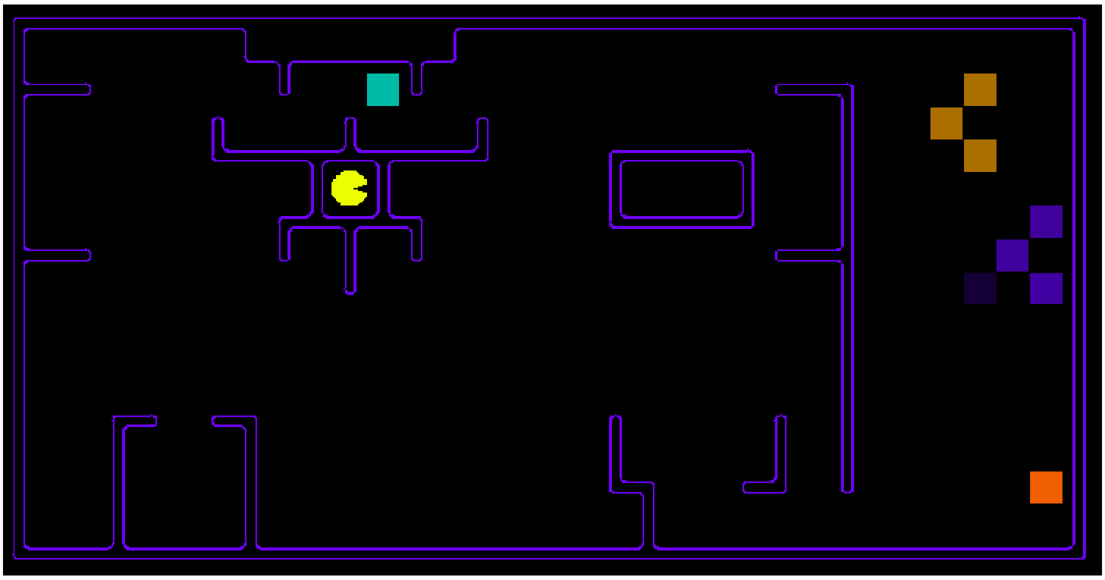

FIGURE 4 – Apparition of a mosaic shaped filter for a game with $w = 1$, $p = 0.5$, 4 ghosts and less than 100 time steps

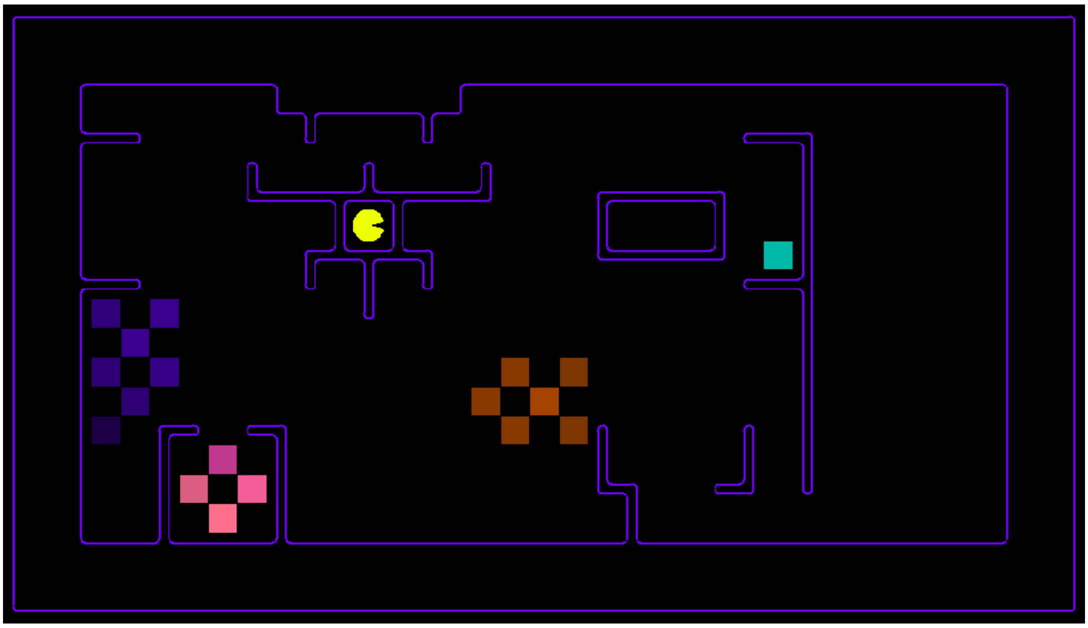

FIGURE 5 – Apparition of a mosaic shaped filter for a game with $w = 3$, $p = 0.5$, 4 ghosts and more than 200 time steps

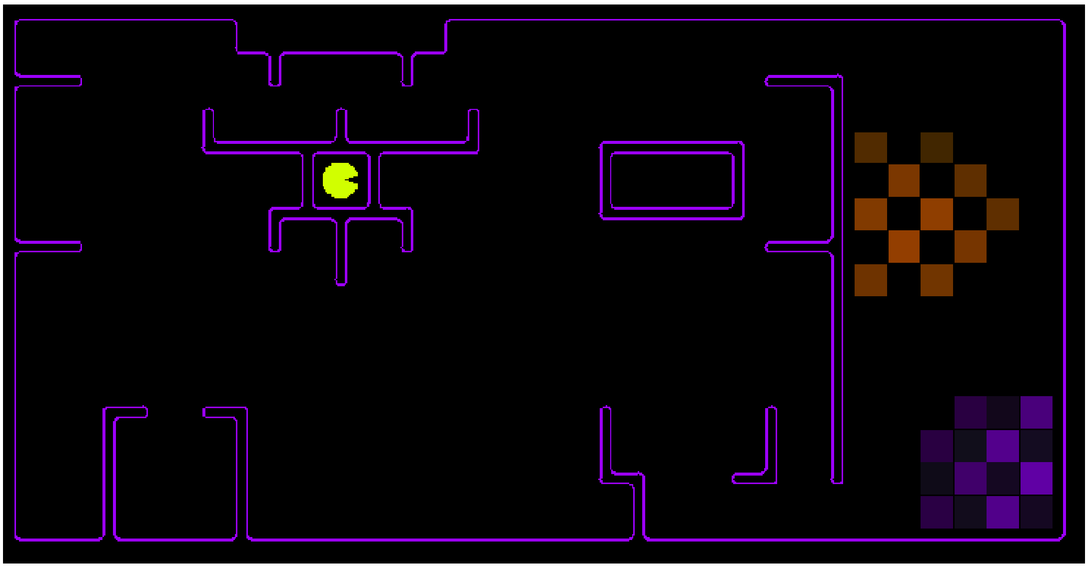

FIGURE 6 – Apparition of a mosaic shaped filter for a game with $w = 3$, $p = 0.5$, 2 ghosts and more than 200 time steps