INFO8006: Project 3 - Report

Théo Stassen - s150804
Hubar Julien - s152485

December 3, 2019

1 Bayes filter

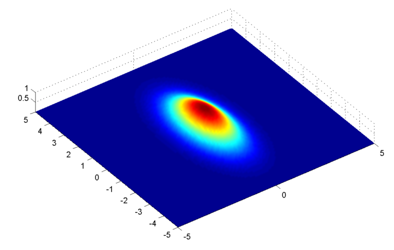
a. 

FIGURE 1: Normal distribution[1]

The sensor model of the rusty sensor is a stochastic model which takes a value (the real distance between pacman and ghost) and send at output a random value drawn in a normal distribution (with the value as mean and the given variance as variance). This output value can be seen as the perfect value plus a noise which follow a gaussian distribution.

b. The transition model can be described as :

$$P(X_{t+1,i,j}) = \sum\nolimits_{(k,l)=\text{legal neighbors of }(i,j)} P(X_{t,k,l}) * P((i,j)|(k,l),(i_p,j_p),g) \quad (1)$$

Where X_t is the set of unobservable variables, which can be seen as a matrix, with an unobservable variable for each position. $X_{t,a,b}$ is the element of the matrix for position (a,b) and time t. It value is = 1 if the ghost is at this position and 0 if not. $P(X_{t,a,b})$ is the probability of ghost to be in (a,b) at time t. $P((i,j)|(k,l))$ is the probability to go in (i,j) in time t+1 where we are in (k,l) at time t. This probability doesn't depends of the time in our case but depends of type of ghost g (the only free parameter of the transition model) and the position of pacman (i_p, j_p). This probability can be expressed as :

$$P((i,j)|(k,l),(i_p,j_p),g) = \frac{c((i,j)|(k,l),(i_p,j_p),g)}{norm((k,l),(i_p,j_p),g)} \quad (2)$$

$$norm((k,l),(i_p,j_p),g) = \sum\nolimits_{(m,n)=\text{legal neighbors of } (k,l)} c((k,l)|(m,n),(i_p,j_p),g) \tag{3}$$

Where $c((i,j)|(k,l),(i_p,j_p),g)$ is directly taken from the ghost implementation. It is equal to 1 if the step (k,l) $\rightarrow$ (i,j) decreases the ghost to pacman manhattan distance and is equal to a value which depends of the ghost type if the distance increases (1 if "confused", 2 if "afraid", 8 if "scared"). $norm((k,l),(i_p,j_p),g)$ is the sum of c values in each direction started from (k,l). It is used to have a normalized probability.

2 Implementation

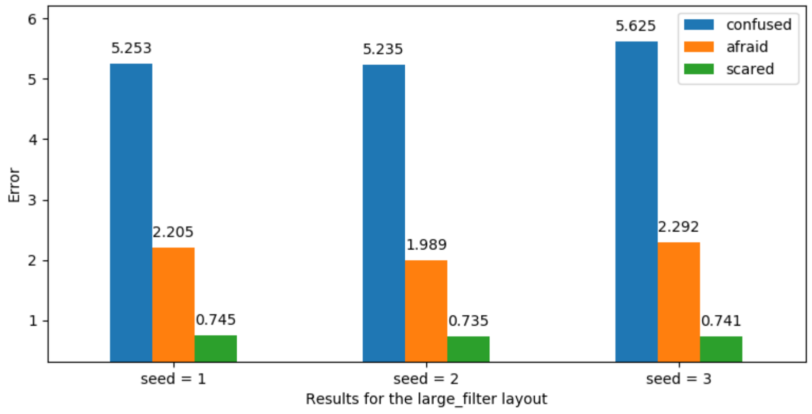
a. See `bayesfilter.py`.

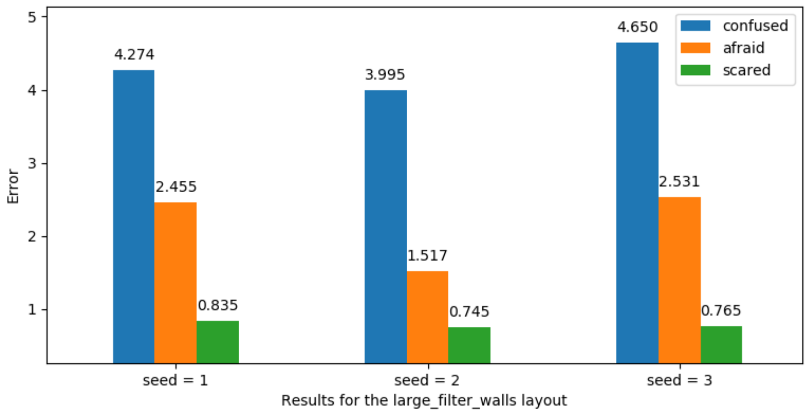
3 Experiment

a. We can consider that the distribution of probability of the ghost position is a gaussian, with the maximum probability value of the grid in the center of it. To summarise the uncertainty, we can compute the variance of this 2D distribution at a time t. The slower the variance is, the less the uncertainty is, because we had a more precise and focused information about where the ghost is.

b. To measure the quality of the belief state, we consider firstly the distance between the ground true value of the position of the ghost and the position where the probability computed is maximal. When the belief state converges, the probabilities tends to form a gaussian distribution around a convergence point, which is the maximum. This is why we take this point. If the distribution is (more or less) gaussian, estimates the quality of the belief states based only on the maximum of the gaussian distribution makes sense. We can consider only the absolute distance difference value in a time t, but it is not really precise. We choose to make the mean of these distances in the 10000 last iterations to have a very global and precise result.

c. The results are presented in the figures. We run the 3 ghost types on the two layouts three times, with three different seed. The resulting error values are the mean of 10000 last distance differences. These values vary very lightly (0.001 of difference between two execution with scared ghost).





d. As we can seen in the results, the different values of the ghost transition model parameter gives clearly different results. More the ghost is afraid by pacman (more the $c(g)$, which can be 1, 2, 8, is high), more the error is low, nowadays the seed or the layout. It is easy to understand because if $c(g)$ is high, the ghost will go, at each step, with an high probability away from the pacman. So more the ghost is afraid, more it will go directly to the far away position and will stay in that zone (where we can consider that it converges). So an scared ghost, at each time, will go and stay more stricly in the far away zone than an afraid. It is more easy to know where the ghost is, only based on the transition step, when at each step it has a really big chance to go in a precise direction that we know. In other terms, afraid and scared converges to a normal distribution of position probability, centered in one corner of the grid, but the gaussian of scared has a smallest variance. In the case of a confused ghost there are no really a place where the ghost converge. During iterations it will go in any places and the belief state will "follows" it using mainly the update step (if there was only the transition step, the belief states will vanish and converge to an equal constant probability everywhere) which indicates more or less in which direction the ghost go. But it is less precise due to the noise and so the fact that we not consider the information given by sensor as perfect (if it was the case we must do the update step by multiply by 0 the probabilities for each "bad" position and by 1 for the "good"). Concretely in the code we multiply for each position the probability by a factor linearly inversely proportional to the difference between the distance to the position we consider and the distance given by the sensor. By this method we increase the probabilities in the near distances and decrease the far ones but it is not extremely precise. That is why in mean of a lot of iteration, the error on the confused ghost is clearly bigger. We can saw also that the different layout has different results, because the configuration is not the same. In the globality of the grid, the ghost has less possibilities of moving and so there are less of chances to "miss" it, so the values for confused are better. But in the "down left corner" where the ghost will go for scared and afraid the situation is more or less equivalent, which explains that the results are in this case not better and lightly badder. The seed 2 is an exception, because in this case the ghost starts in a position which imply that it will go in completely different position depending of the layout, which is not the case in the other seeds. The results are better for afraid in the second layer because of that. The results can so vary from one particular case to another. But in general it is clear that scared gives better results than afraid, that gives better results than confused.

e. If we have a small variance the distance given by the noisy sensor is near the real distance. The belief state agent has a good evidence to use. The update step is so very efficient because we increase the probability of presence of the ghost in cases where there are really a lot of chances to have the ghost. If we increase the variance, the information has more chance to give very wrong result. An the update step will be counter productive by increasing the wrong position probabilities. Below a certain value of variance the majority of noisy distance makes sense, is not significantly wrong and the update step will be, in mean, useful. But above a certain value it is not the case. If the variance is really enormous compared to the size of the grid, there are real chances to have an absolute value of distance difference superior to the maximum distance in the grid. In this case we replace the probability computed by a very low value to avoid glitches but it means that the update is completely counterproductive. It is more interesting to not use this step and the sensor in this case.

f. Considering current position, the set of legal actions and the current belief state, we can easily creates a simple agent which has as objective to eat the ghost. At each step the goal is to decrease the ghost to pacman distance. The agent can compute the maximum of the current belief state which gives the position the most likely to have the ghost (and if it not the case considering that the distribution tends to be gaussian the real position of the ghost had big chances to be near the estimated position). The agent can choose between the legal action the one which will minimize the distance with the estimated position (by generating the successors and computing the manhattan distance for each, or by doing simple geometry calculation). It is a greedy algorithm which go straight to the ghost without taking account a possible strategy of the opponent. The efficiently of the agent will depend a lot of the variance of the noisy sensor and also on the behaviour of the ghost. Because of all the points developed above, like the fact that there is a big difference in mean between the real position and the position of the maximum of the belief state that we use there, a confused ghost will be more difficult to catch.

References

[1] https://fr.wikipedia.org/wiki/Fonction_gaussienne