Adaptive Sampling Technique using KullbackLeibler Distance

Arun Vignesh Malarkkan
1215098183

Gowtham Sekkilar
1215181396

Raghavendran Ramakrishnan
1215325696

Faisal Alatawi
1215360666

Abstract— **Over the past few years, particle filters have been applied to a variety of state estimation problems and also resulted in a great success. In addition to assigning a probability of being in each of the possible belief states, we present a statistical approach to increase the efficiency of particle filters by adapting the size of sample sets during the estimation process. The key idea of the KLD-sampling method is to bound the approximation error introduced by the sample-based representation of the particle filter by measuring the approximation error using the KullbackLeibler distance. Our adaptive approach chooses a small number of samples if the density is focused on a small part of the state space, and it chooses a large number of samples if the uncertainty of the considered state is high.**

I. INTRODUCTION

Estimating the state of a system in a dynamic environment based on noisy sensor measurements is extremely important in varied areas as different as intelligent game agents, speech recognition, target tracking, mobile robot navigation, and computer vision. Over the past few years, particle filters have been applied with great success to a variety of state estimation problems [1]. Particle filters are used to estimate the posterior probability density over the state space of such dynamic system. The key idea of this technique is to represent probability densities by sets of samples. It is due to this representation that particle filters combine efficiently with the ability of particle filters lies in the way they place computational resources.

Since the complexity of particle filters depends on the number of samples used for estimation, several attempts have been made to make more efficient use of the available samples[2]. We introduce a novel approach to adapt the number of samples with respect to time. Our technique determines the number of samples based on statistical bounds on the sample-based approximation quality

In this paper, we implemented the Adaptive Monte Carlo sampling method in Particle Filters to design an intelligent adaptive game agent in Pacman domain that use sensors to locate and eat invisible ghosts.

II. PREVIOUS WORK

A. BAYESIAN NETWORK

Bayesian Networks also known as belief networks or causal networks are graphical models for representing multivariate probability distributions. Each variable X_i is represented as a vertex in an directed acyclic graph; the probability distribution $P(X_1, X_2, .., X_N)$ is represented in factorized form as follows:

$$P(X_1, X_2, .., X_N) = \prod_{i=1}^{N} P(X_i | \prod_{X_i})$$

Where ${}_{(}X_i)$ is the set of vertices that are X_i's parents in the graph.

A Bayesian network is fully specified by the combination of:

- The graph structure, i.e., what directed arcs exist in the graph
- The probability table $P(X_i|_{(}X_i))$ for each variable X_i

Bayes filters address the problem of estimating the state of a dynamical system from sensor measurements. The key idea of Bayes filtering is to recursively estimate the posterior probability density over the state space conditioned on the data collected so far. If we assume that the data consists of an alternating sequence of time indexed observations z_t and control measurements u_t, which describes the dynamics of the system, then the posterior at time t is called $Bel(x_t)$ and is defined by

$$Bel(x_t) = p(x_t|z_t, u_{t-1}, z_{t-1}, u_{t-2}, ..., u_0, z_0)$$

B. PARTICLE FILTERING

```
Inputs: $S_t = \left\{(x_{t-1}^{(i)}, w_{t-1}^{(i)})|i = 1, ., n\right\}$
S$_t := \emptyset; \alpha := 0$
for i := 1, . . . , n do
    //Resampling: Draw state from previous belief
        Sample an index j from the discrete
        distribution given by the weights in S$_t - 1$
    //Sampling : Predict next state
        Sample x$_t^{(i)} from\ p(x_t|x_{t-1}, u_{t-1})$
        conditioned on x$_{t-1}$ $and\ u_{t-1}$
        w$_t^{(i)} := p(z_t|x_t^{(i)})$;
        $\alpha := \alpha + w_t^{(i)}$
        S$_t := S_t \cup (x_t^{(i)}, w_t^{(i)})$
for i := 1,...., n do
        w$_t^{(i)} := w_t^{(i)}/\alpha$
return S$_t$
```

Fig. 1. The Basic Particle Filtering Algorithm

A Particle is simply an assignment of a value to set of variables and an associated weight. The probability of a proposition, given some evidence, is proportional to the weighted proportion of the weights of the particles in which the proposition is true. Particle filtering is a sampling technique that starts with a population, that is a set of particles, each which assigns a value to no variables, and has a weight of 1 and can be used for an unbound number of particles.

Particle filters lead to an approximate estimate of a solution for a complex model of a problem. Particle filters are a variant of Bayes filters which represent the belief $Bel(x_t)$ by a set S_t of n weighted samples distributed according to $Bel(x_t)$.

$$S_t = \left\{(x_t^{(i)}, w_t^{(i)})|i = 1, ., n\right\} \tag{1}$$

C. KLD-SAMPLING

The key idea of adaptive particle filters can be stated as

At each iteration of the particle filter, determine the number of samples such that, with probability 1-δ , the error between the true posterior and the sample-based approximation is less than ϵ

To derive a bound on the approximation error, we assume that the true posterior is given by a discrete, piecewise constant distribution such as discrete density tree or a multidimensional histogram[3]. For such a representation we show how to determine the number of samples so that the distance between the sample-based maximum likelihood estimate(MLE) and the true posterior does not exceed a pre-specified threshold . We denote the resulting approach as the KLD-sampling algorithm since the distance between the MILE and the true distribution is measured by the Kullback-Leibler distance. The KL distance is a measure of the difference between two probability distributions p and q:

$$K(p, q) = \sum_x p(x) log(p(x)/q(x)) \tag{2}$$

The KL-distance is never negative and is zero if and only if the two distribution are identical. It is not a metric, since it is not symmetric and does not obey the triangle property. If we choose the number of samples n as

$$n = 1/2\epsilon X^2_{k-1,1-\delta} \tag{3}$$

Then we can guarantee that with probability 1-δ, the KL-distance between the MLE and the true distribution is less than ϵ[1]. A good approximation is given by the WilsonHilferty transformation[4], which yields

$$n = \frac{k-1}{2\epsilon}\left\{1 - \frac{2}{9(k-1)} + \sqrt{\frac{2}{9(k-1}} z_{1-\delta}\right\}^3 \tag{4}$$

where $z_{(1-\delta)}$ is the upper $1-\delta$ quantile of the standard normal distribution. The values of for typical values of $z_{(1-\delta)}$ are readily available in standard statistical tables. From the above equation we can see that the required number of samples is proportional to the inverse of the error bound δ, and to the first-order linear in the number k of bins with support. Here we assume that a bin of the multinomial distribution has support if its probability is above a certain threshold(i.e., if it contains at least one particle).

The above table summarizes the update step of the KLD-sampling particle filter. We update the number of supported bins k for the predictive distribution after each sample generated. The determination of k can be done incrementally by checking for each generated sample whether it falls into an empty bin or not. The implementation of this modified particle filter is straightforward. The difference to the original algorithm is that we have to keep track of the number k of supported bins and the number n of desired samples. The bins can be implemented either as a fixed, multi-dimensional grid, or more compactly as tree structures[5]

```
Inputs: $S_t - 1 = \left\{(x_{t-1}^{(i)}, w_{t-1}^{(i)})|i = 1, ., n\right\}$ representing
    belief $Bel(x_{t-1})$, control measurements u$_{t-1}$,
    observations z$_t$, $bounds \epsilon \ and \ \gamma, bin \ size \delta$,
    minimum number of samples n$_{xmin}$
S$_t$ $:= \phi; n = 0; n_x = 0; k = 0; \alpha := 0$
do
    //Resampling: Draw state from previous belief
        Sample an index j from the discrete
        distribution given by the weights in S$_{t-1}$
    //Sampling : Predict next state
        Sample x$_t^{(i)}$ $from \ p(x_t|x_{t-1}, u_{t-1})$
        conditioned on x$_t - 1 \ and u_t - 1$
        w$_t^{(n)}$ $:= p(z_t|x_t^{(n)})$;
        $\alpha := \alpha + w_t^{(n)}$
        S$_t$ $:= S_t \cup (x_t^{(n)}, w_t^{(n)})$
        if (x$_t^{(n)}$ $falls \ into \ empty \ bin \ b)$ then
            k := k +1
            b := non-empty
            $n_x := \frac{k-1}{2\epsilon}\{1 - \frac{2}{9(k-1)} + \sqrt{\frac{2}{9(k-1)}} z_{1-\delta}\}^3$
        n := n + 1 while (n ¡ n$_x$ $and n < n_{xmin})$
for i := 1,...., n do
    w$_t^{(i)}$ $:= w_t^{(i)}/\alpha$
return S$_t$
```

Fig. 2. KLD Sampling Algorithm

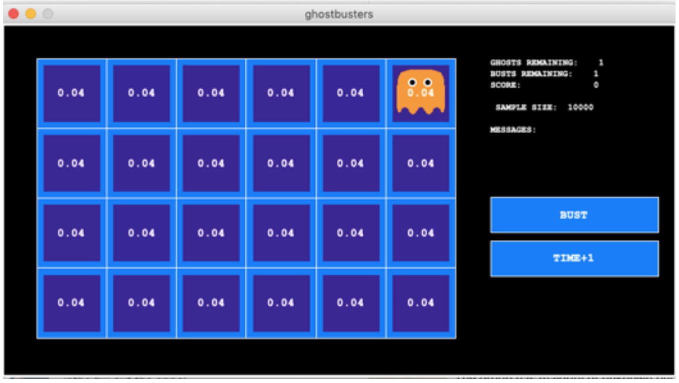
III. IMPLEMENTATIONS

We have presented the implementation of Adaptive Particle Method using KL-distance technique following the algorithm presented in Technical paper published by Dieter Fox [6]. We tested the performance by integrating the algorithm with the Pac-Man projects that were developed for the course work of CS188 at University of Berkley. [7].

The project involved a game of Ghostbuster where Pac-Man agent identifies hidden ghosts based on the inference in the form of noisy observation which is a Manhattan distance of the ghosts from the Pac-Man agent. The game ends once all the ghosts are eaten. The specific scenario which we selected to test the performance of the code involved using particle filter for representing the current belief. The particle filter used initial samples of size 5000 for approximate (noisy) inference about the location of the ghosts.

The sampling technique involved likelihood weighting, where all the evidence variables were fixed, and the samples were weighted by the evidence likelihood. This helped in taking advantage of all the samples that are observed.

The challenge with classical particle filtering technique was it didnt offer a method to dynamically vary the particle



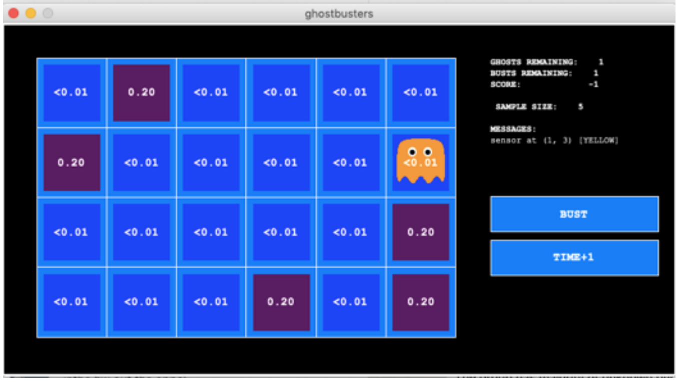
0.04	0.04	0.04	0.04	0.04	0.04
0.04	0.04	0.04	0.04	0.04	0.04
0.04	0.04	0.04	0.04	0.04	0.04
0.04	0.04	0.04	0.04	0.04	0.04

```
GHOSTS REMAINING:    1
BUSTS REMAINING:    1
SCORE:              0

 SAMPLE SIZE:  10000

MESSAGES:
```

Fig. 3. Ghost busters initial state. Here 10,000 samples are used for initial inference



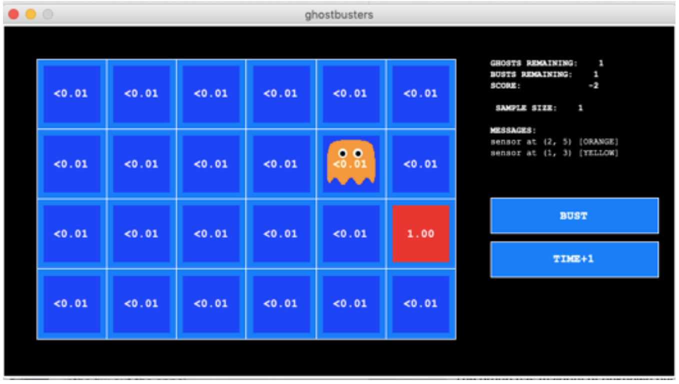
<0.01	0.20	<0.01	<0.01	<0.01	<0.01
0.20	<0.01	<0.01	<0.01	<0.01	<0.01
<0.01	<0.01	<0.01	<0.01	<0.01	0.20
<0.01	<0.01	<0.01	0.20	<0.01	0.20

```
GHOSTS REMAINING:    1
BUSTS REMAINING:    1
SCORE:             -1

 SAMPLE SIZE:    5

MESSAGES:
sensor at (1, 3) [YELLOW]
```

Fig. 4. At step 2, the number of particles dynamically reduces as the location of the Ghost is inferred with higher probability



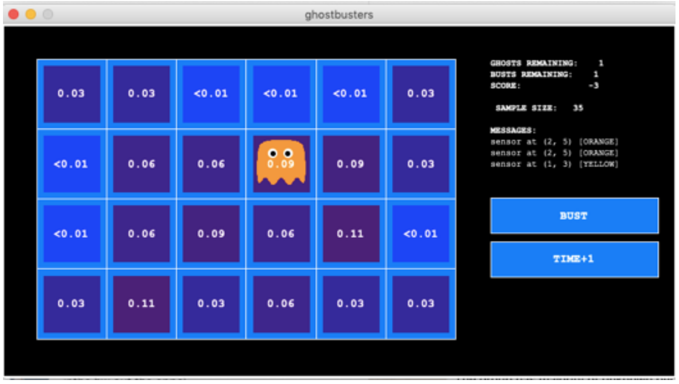
<0.01	<0.01	<0.01	<0.01	<0.01	<0.01
<0.01	<0.01	<0.01	<0.01	<0.01	<0.01
<0.01	<0.01	<0.01	<0.01	<0.01	1.00
<0.01	<0.01	<0.01	<0.01	<0.01	<0.01

```
GHOSTS REMAINING:    1
BUSTS REMAINING:    1
SCORE:             -2

 SAMPLE SIZE:    1

MESSAGES:
sensor at (2, 5) [ORANGE]
sensor at (1, 3) [YELLOW]
```

Fig. 5. The sample size is reduced to 1 as the location seems to be inferred with high certainty as next possible position



0.03	0.03	<0.01	<0.01	<0.01	0.03
<0.01	0.06	0.06	0.09	0.09	0.03
<0.01	0.06	0.09	0.06	0.11	<0.01
0.03	0.11	0.03	0.06	0.03	0.03

```
GHOSTS REMAINING:    1
BUSTS REMAINING:    1
SCORE:             -3

 SAMPLE SIZE:   35

MESSAGES:
sensor at (2, 5) [ORANGE]
sensor at (2, 5) [ORANGE]
sensor at (1, 3) [YELLOW]
```

Fig. 6. Once the position is explored, it is clear the inferred location was incorrect and number of samples peaks to 35 as the uncertainty peaks

size in consistent with the uncertainty in the environment. The traditional approach (which is also compared in this work) used a fixed number of samples which resulted in either unnecessary wastage of resources by computing the inferences of all samples or provided noisy reading due to insufficient number of samples. The Adaptive sampling method solves this issue by dynamically varying the number of particles that are required in consistence with the error margin in the prediction of the ghost location (which reflects the uncertainty in the environment). This is measured through KL distance technique. In our implementation, we used the WilsonHilferty transformation to approximate the noise in the environment and this directly influenced the number of samples that are used for inference.

The particle size was initially assigned a maximum limit (10000). Bins were setup to hold all the observed samples. They worked as counters with sample location as key Once a specific sample is randomly selected, it is compared with the samples in the bin. If it was already observed, it is ignored, and next sample is drawn. But if they are new, the counter keeping track of the number of bins is incremented and the KL distance factor is calculated. This value is then used to draw the number of samples. As a result, when the same sample is drawn multiple time (due to lesser uncertainty in the environment), it doesnt impact the sample size and it remains unchanged. But if unique samples are observed with higher weight every time, it implies uncertainty in the environment (in the ghost location), Hence it increases the sample size as the KLD is increased. This can be inferred from Figures 1 till 4 mentioned below.

IV. COMPARISON

Performance of the algorithm was evaluated by measuring the run time and resource utilization. The results of comparison between the execution of the program using traditional sampling approach and new adaptive approach are published in the below table. It showed an impressive performance boost of 74.71 Percent over traditional sampling approach.

TABLE I
PERFORMANCE COMPARISON

PARTICLE FILTER TYPE	RUN TIME	NOTES
KLD-Sampling	13958	74.71 Percent Faster
ParticleFilter	53409	

V. CONCLUSION

We have presented a statistical approach to adapting the sample set sizes of particle filters. The key idea of the KLD-sampling approach is to bound the error introduced by the sample-based belief representation.

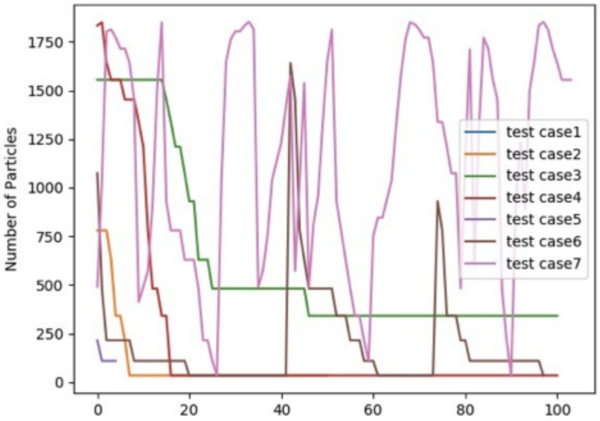


Fig. 7. Plot of particle size vs time. This plot shows the variations in particle size for different test executions of the Pac Man project

VI. Future Work

KLD-sampling opens several directions for future research. Our approach can also be extended to the case where in certain parts of the state space highly accurate estimates are needed, while in other parts a rather crude approximation is sufficient. This problem can be addressed by locally adapting the discretization to the desired approximation quality using multi-resolution tree structures[5] in combination with stratified sampling. As a result, more samples are used in important parts of the state space, while fewer samples are used in other parts.

References

[1] Dieter Fox, Adapting the Sample Size in Particle Filters through KLD-Sampling, University of Washington, 2003.

[2] Gilks, W. R., and Berzuini, C. 2001. Following a moving targetMonte Carlo inference for dynamic Bayesian models. Journal of the Royal Statistical Society, Series B 63(1):127146

[3] Koller, D., and Fratkina, R. 1998. Using learning for approximation in stochastic processes. In Proceedings of the International Conference on Machine Learning..

[4] Johnson, N., Kotz, S., and Balakrishnan, N. 1994. Continuous Univariate Distributions, Vol. 1, Wiley, New York.

[5] Moore, A. W., Schneider, J., and Deng, K. 1997. Efficient locally weighted polynomial regression predictions. In Proceedings of the International

[6] CS188-"Introduction to Artificial Intelligence"-https://inst.eecs.berkeley.edu/ cs188/fa11/projects/tracking/busters.html,University of California,Berkeley