Study Notes for AI (CS 440/520)
Lecture 14: Smoothing, Most Likely Explanation and Intro to Kalman Filter

Corresponding Book Chapters: 15.2-15.3-15.4
Note: These notes provide only a short summary and some highlights of the material covered in the corresponding lecture based on notes collected from students. Make sure you check the corresponding chapters. Please report any mistakes in or any other issues with the notes to the instructor.

Temporal probabilistic estimation is the problem of estimating the state of an agent, when the state changes over time.

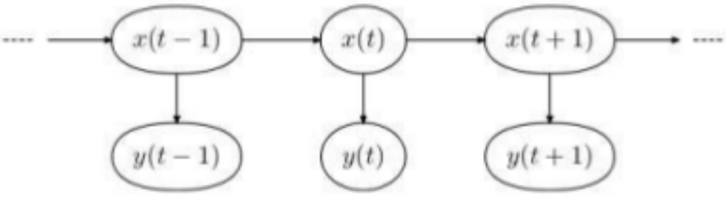

Figure 1: Dynamic bayesian network for temporal state estimation. In this figure the variable y corresponds to the evidence variables.

Figure 1 shows a simplified Dynamic Bayesian Network used to solve temporal probabilistic estimation problems. This Bayesian Network shows the relationship of states and evidences, denoted as x and y respectively. In order to use this type of model, we make use of two assumptions:

1. The Bayesian Network must be a Stationary Process. That means that the way the process changes must remain constant over time. Mathematically: $P(X_t|Parents(X_t))$ is the same $\forall$ t. For the example shown in Figure 2 the conditional probability of observing an umbrella, $P(U_t|Parents(U_t))$, is the same for all t.

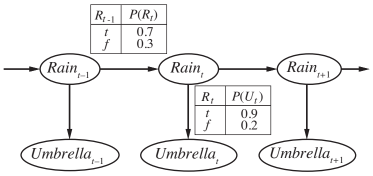

Figure 2: Dynamic Bayesian Network of the Umbrella World

2. The Bayesian Network follows a First Order Markov Assumption. The Markov Assumption states that the current state X_t depends only on a finite history of past states. This was named after Andrei Markov who first studied processes that satisfied this condition. The process illustrated in Figure 2 is a first-order Markov process since the current state X_t depends only on the previous state X_{t-1}.

Given the structure of the Bayesian Network, we must have available the following probabilities to solve problems:

- Prior Probability Distribution: $P(X_0)$.
- Transition Model: $P(X_t|X_{t-1})$.
- Observation Model: $P(E_t|X_t)$

1 Solving Temporal State Estimation Problems

With the above definitions, we can begin to solve temporal state estimation problems. We may want to find out many things about the world that our agent is in. In particular, we can estimate from observations the current, a previous or a future state of the agent. Or we might be interested in computing the path the agent has followed. These questions correspond to the four types of temporal state estimation problems: Filtering, Prediction, Smoothing, and Most Likely Explanation.

1.1 The Filtering Problem : $P(X_t|E_{1:t})$

Defining the Filtering Problem: The filtering problem poses the following question: "Given all of the evidence up to this time, what is the current state of the world?" This problem is essentially the process of extracting relevant information from sensors to guess what the current state is.

Representation of the Filtering Problem: By manipulating the above probability, we can acquire the following expression in order to evaluate the Filtering Problem:

$$P(X_t|E_{1:t}) = \alpha \cdot P(E_t|X_t) \cdot \sum_{X_{t-1}} P(X_t|X_{t-1}) \cdot P(X_{t-1}|E_{1:t-1}) \tag{1}$$

Note that the components for solving the Filtering Problem are the Observation Model, the Transition Model, and the Agent's Previous Belief.

1.2 The Prediction Problem : $P(X_{t+k}|E_{1:t})$

Defining the Prediction Problem: The Prediction Problem poses the following question: "What will the state be at some time into the future, given all the evidence up to this point?" This problem attempts to predict the future with all of the evidence we have collected so far.

Representation of the Prediction Problem: The mathematical representation of the question is: $P(X_{t+k}|E_{1:t})$ for some positive time step k. Similarly to the Filtering Problem, we can manipulate the probability in order to receive the following expression:

$$P(X_{t+k}|E_{1:t}) = \sum_{X_{t+k-1}} P(X_{t+k}|X_{t+k-1}) \cdot P(X_{t+k-1}|E_{1:t}) \tag{2}$$

A note on the prediction problem: It is important to note that the above equation will eventually converge to a stationary probability for a large value of k, which equals the prior probability $P(X_0)$.

1.3 The Smoothing Problem : $P(X_k|E_{1:t}), \quad 1 \leq k < t$

Defining the Smoothing Problem: The Smoothing Problem poses the following question: "What was the state of the world at some point in the past given all the evidence up to this point?" Essentially, we have to reconstruct past states given not only the evidence up to the time the state occurred, but also given evidences that came after that state.

Representation of the Smoothing Problem: The Smoothing Problem can be represented as follows:

$$P(X_k|E_{1:t}) = P(X_k|E_{1:k}, E_{k+1:t}) = \alpha * \overbrace{P(X_k|E_{1:k})}^{\text{filtering}} * P(E_{k+1:t}|X_k, E_{1:k}) \tag{3}$$

The above operation breaks the smoothing problem into two sub-problems. Note that the first probability is simply the Filtering Problem, which we have already discussed how to compute; however, what is the second probability? This Probability does not appear in any of the problems we have talked about so far. This new probability is called the "Backwards Message". In this context, the Filtering Problem is called the "Forward Message".

1.3.1 Computing the "Backwards Message"

The second probability in the above expression is not known to us from the problem definition, nor is it one of the other problems we need to solve. This Probability represents the Previous Backwards Message. This means that we can compute the "Backwards Message" iteratively, starting at the current state and moving backwards along the dynamic Bayesian networks toward the state in question. This is a similar operation to filtering, only in the opposite direction.

$P(E_{k+1:t}|X_k)$

$\equiv \sum_{X_{k+1}} P(E_{k+1:t}|X_k, X_{k+1}) \cdot P(X_{k+1}|X_k)$ by Conditioning on X_k.

$\equiv \sum_{X_{k+1}} P(E_{k+1:t}|X_{k+1}) \cdot P(X_{k+1}|X_k)$ by Conditional Independence.

$\equiv \sum_{X_{k+1}} P(E_{k+1}, E_{k+2:t}|X_{k+1}) \cdot P(X_{k+1}|X_k)$

$\equiv \sum_{X_{k+1}} P(E_{k+1}|X_{k+1}) \cdot P(E_{k+2:t}|X_{k+1}) \cdot P(X_{k+1}|X_k)$

Then putting this result back into Equation 3 we have our final equation for smoothing:

$$P(X_k|E_{1:k}, E_{k+1:t}) = \alpha * \overbrace{\sum_{X_{k+1}} \overbrace{P(E_{k+1}|X_{k+1})}^{\text{Observation}} * \overbrace{P(E_{k+2:t}|X_{k+1})}^{\text{Recursive Step}} * \overbrace{P(X_{k+1}|X_k}^{\text{Transition Model}}}^{\text{Backwards Message}} * \overbrace{P(X_k|E_{1:k})}^{\text{Filtering/Forward Message}} \tag{4}$$

Complexity: In solving the smoothing problem, we use both forward and backwards recursion. At each step of the recursion, in both cases, our step is a constant time process. Consequently, the overall time complexity for the sequence $E_{1:t}$ is $O(t)$. We can also compute the smoothing problem for all states in the sequence, which is more useful, and the complexity of that is $O(t^2)$, because we must perform the smoothing problem at every state in the sequence. If we are able to store information of what we have done, we can store intermediate results between smoothing operations in order to speed up the process. In this case, the space complexity of the problem becomes

$O(t)$ and the time complexity is reduced to $O(t)$. This may be prohibitively high space complexity if we want to smooth the estimation of all previous states.

1.4 Most Likely Explanation

Defining the Most Likely Explanation: The Most Likely Explanation problem poses the following question: "What is the sequence of states that most likely occurred for the current evidence to be observed?" This problem is different from the previous ones in that the query is to return an entire sequence of states.

Representation of the Most Likely Explanation: The Most Likely Explanation can be mathematically represented as $argmax_{X_1...X_t}\{P(X_1...X_t, X_{t+1}|E_{1:t+1})\}$.

Calculating the Most Likely Explanation: In order to solve a Most Likely Explanation query, we can use the Viterbi Algorithm. The Viterbi Algorithm:

$$argmax_{X_1...X_t}\{P(X_1...X_t, X_{t+1}|E_{1:t+1})\} =$$

$$\alpha \cdot P(E_{t+1}|X_{t+1}) \cdot argmax_{X_1...X_t}\{P(X_{t+1}|X_t) * argmax_{X_t}\{P(X_1...X_t|E_{1:t})\}\}$$

uses the Observation Model, the Transition Model, and the Previous estimate of the Most Likely Explanation. Notice the similarity to filtering, only that the Viterbi Algorithm is using argmax instead of a summation. The time and space requirements of this algorithm are also linear in the number of time steps t.

1.5 Example: Robot in Discrete World

Assume you have a robot in a discrete world. The prior probability distribution is provided in Figure 3. The robot has the following transition and observation models:

- Transition model: When the robot moves in a certain direction, there is a 10% chance of going backwards
- Observation model: The robot is able to sense how far away obstacles are in the Up direction. The robot is 80% of the time correct in its measurement, 10% percent chance to overestimate by 1 cell and 10% percent chance to underestimate by 1.

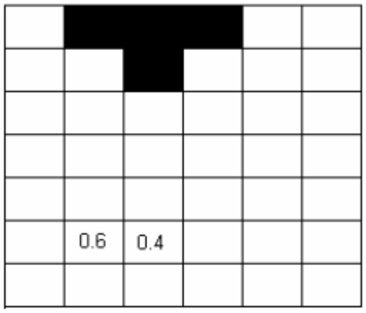

Figure 3: Prior Probability Distribution

We will apply filtering to estimate the robot's location after one move and one sensing operation. Assume that the robot moves up 1 step and then senses that the obstacle is 2 cells away in the Up direction (which means 2 empty cells between the robot's position and the obstacles). We can compute the probability of being at a cell X_1^i at time step 1, using the filtering algorithm:

$$P(X_1^i| \text{ move up 1 step, sensing input = 2 cells }) =$$
$$= \alpha P(e_1|X_1^i) * \sum_{X_0^j} P(X_1^i|X_0^j)P(X_0^j)$$

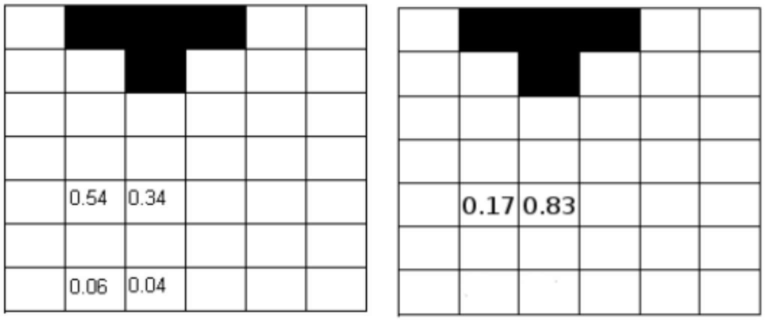

Figure 4: (left) the probability distribution after considering the move just one application of the transition model (right) the probability distribution after considering the observation: the top cell on the second column has 10% probability of being the correct cell according to the observation, while the top cell on the third column has 80% probability of being the correct cell. We multiple this probabilities with the predictive probabilities (0.54 and 0.34 respectively), then we normalize and this is how we acquire the above result.

2 Solving Problems in Continuous Space

An example of a state estimation problem in the real world, which is obviously continuous, is tracking the position of a moving agent. Other continuous problem involve the estimation of parameters such as height, mass, temperature or money. The problem that arises is that it is now impossible to specify conditional probabilities for each value of a continuous variable that we have to estimate. To solve such temporal state estimation problems in continuous spaces we can employ one of the following alternatives:

2.1 Discretize the Environment

The first and obvious solution to continuous state estimation is to discretize the environment into a grid so that we may apply the techniques we have already described. If the world can be easily discretized, then we can apply this process; however, by discretizing the world and using these solutions, we often get considerable errors in the state estimation process.

2.2 Use Convenient Probability Density Functions

Another solution is to represent the data of the world in such a way that is memory efficient and quick to reference. If we have available a probability density function with such desirable qualities, then we can keep the required number of parameters to represent the probabilities low. Thus, we will not run into the computational explosion problems that appear in the simplistic discretization mentioned above and we should be able keep track of real world agent's states with greater accuracy.

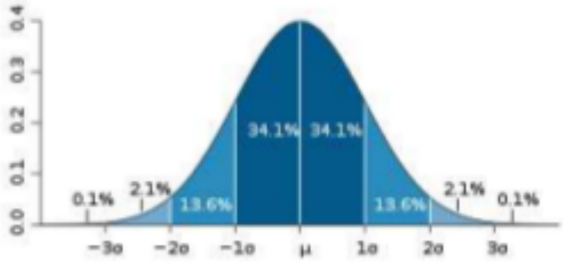

Figure 5: A Gaussian distribution. The darker color is less than one standard deviation from the mean. For the normal distribution, this accounts for about 68% of the set (dark blue), while two standard deviations from the mean (medium and dark blue) account for about 95% and three standard deviations (light, medium, and dark blue) account for about 99.7%.

Figure 5 provides an example of a Gaussian Distribution. We can use Gaussian Distributions to represent probabilities in continuous space. With the parameters mean(μ) and standard deviation(σ), we are able to keep track of a probability density function. This is desirable because we are able to keep track of a complex probability with just two parameters. As you can see from Figure 2, 99.7% of all probabilities fall within 3 standard deviations from the mean. In this context, μ is called the mean, σ is the Standard Deviation, and σ^2 is called the Variance. A Gaussian Probability Density Function is represented as:

$$\frac{1}{\sigma\sqrt{2\pi}}^{\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)} \tag{5}$$