Reinforcement Learning for Autonomous Driving - Week 1

Brandon Dominique, Hoang Huynh, Eric Av, John Nguyen

Overview

1. Basics of probability theory: Bayes Theorem, Conditional Probability, classifiers: supervised and unsupervised learning (Plenty of material via Google search).

2. Markov Chain and decision theory and automata theory (Posted on Piazza)

3. Reinforcement Learning: look for different papers: Google Scholar, arXiv, IEEEXplore

4. Meta-learning: Curated list of meta-learning papers: https://github.com/dragen1860/awesome-meta-learning

5. Ebook on meta-learning: https://www.packtpub.com/big-data-and-business-intelligence/hands-meta-learning-python (Don't need to buy, there is a 10-days trial period, which I believe should be enough, but do it at the very end when you know you can finish it. It is okay if you want to buy though)

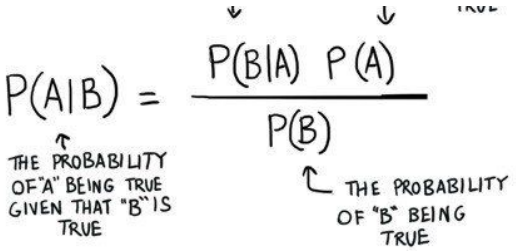
6. ROS: http://wiki.ros.org/

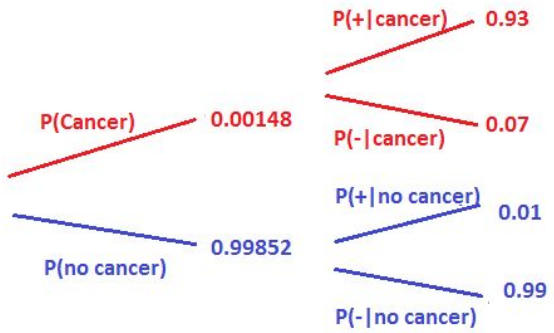
7. http://gazebosim.org/tutorials

8. http://sdformat.org/spec?ver=1.6&elem=world

Bayes Theorem

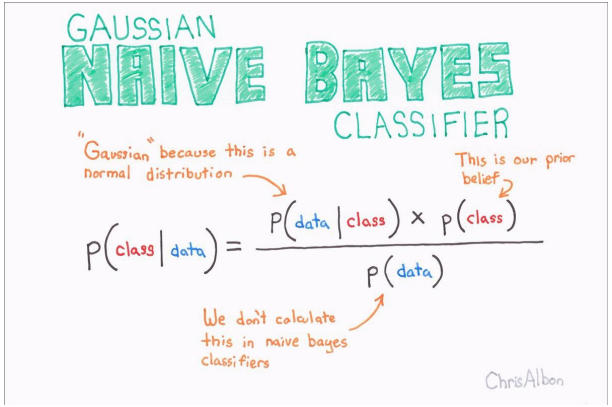
- Given two events A and B, what is the probability that event A will occur given that event B will occur as well?
- Common Example: Patients getting treated for cancer





Applications to Machine Learning

- Naive Bayes Classifier
  - Supervised
  - All features are independent of each other
- Based on the feature vector (data) for a given data point, what should it be classified as?
- Recursive Calculation



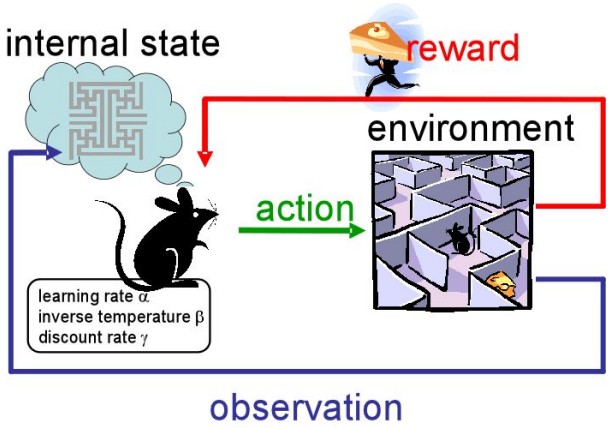
Now, the numerator of the fraction on right-hand side of the equation above is

$$P(x_1, x_2, \ldots, x_n | C_i).P(C_i) = P(x_1, x_2, \ldots, x_n, C_i)$$

$$\begin{aligned} P(x_1, x_2, \ldots, x_n, C_i) &= P(x_1 | x_2, \ldots, x_n, C_i).P(x_2, \ldots, x_n, C_i) \\ &= P(x_1 | x_2, \ldots, x_n, C_i).P(x_2 | x_3, \ldots, x_n, C_i)P(x_3, \ldots, x_n, C_i) \\ &= \ldots \\ &= P(x_1 | x_2, \ldots, x_n, C_i).P(x_2 | x_3, \ldots, x_n, C_i)...P(x_{n-1} | x_n, C_i).P(x_n | C_i).P(C_i) \end{aligned}$$

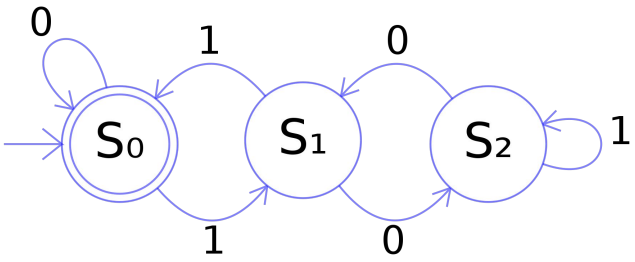
Reinforcement Learning

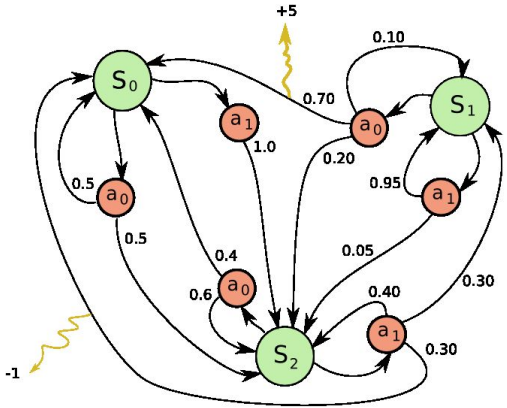
- A way for programming agents to be rewarded and punished without needing to specify how a task is to be achieved.
- Problems faced with trial and error learning interactions with the programming agents.
- The first is to search in the space of behaviors in order to find one that performs well in the environment.
- The second is to use statistical techniques and dynamic programming methods to estimate the utility of taking actions in states of the world.



Markov Chain, Decision Theory and Automata Theory

- An automata is a model of computation; it describes how an input is processed.
- A Markov Decision Process (MDP) is a stochastic finite automata; given a state and action, there are probabilities associated with what state is entered next.
- Decision Theory is the study of how to make an optimal decision with associated rewards and penalties.





Meta Learning

Source: the book "Hands-On Meta Learning with Python"

Meta learning is the process of learning to learn, where an AI optimizes one or several other AIs.

Types of Meta Learning

1/ Learning the metric space:

Learn about similarity between 2 images by computing distances of features. (Siamese networks, prototypical networks, and relation networks.)

2/ Learning the initializations:

Learn optimal initial parameter values. (MAML, Reptile, and Meta-SGD)

3/ Learning the optimizer:

We will have two networks: a base network that actually tries to learn and a meta network that optimizes the base network.

Some Problems of Meta Learning

Credit assignment problem:

-Computer needs to recognize what part of the action actually leads to the reward => need to have a ton of time to train.

Shape reward function:

-not scalable

-the computer may find the way to have that partial reward, not the way we want it to do.

What’s Next?

- Consolidate our knowledge, make sure that we’re all on the same page
- Continue Literature Review, specifically looking for papers related to Reinforcement learning in Autonomous Vehicles
  - Challenges that other people faced, Implementation Methods, etc.
  - We were told that there isn’t too much literature to review, so it’s possible that we will have to create our own implementation
- Understand how to implement a model in a simulation using tools such as ROS and Gazebo for testing our design

References

- Bayes Theorem and Cancer Screening - https://www.youtube.com/watch?v=j2tNxIaGpR4
- Naive Bayes and Machine Learning - https://www.youtube.com/watch?v=sjUDlJfdnKM
- Supervised vs Unsupervised vs Reinforcement Learning - https://www.youtube.com/watch?v=xtOg44r6dsE
- Reinforcement Learning - https://www.jair.org/index.php/jair/article/view/10166
- \section{Meta Learning}
- Meta Learning: It introduces realistic class imbalances. This varies the number of classes in each task and the size of the training set.
- \end{document}