CS480 – Introduction to Artificial Intelligence

Topic: Learning – 1
Bayesian Network Parameter Estimation

Mustafa Bilgic

http://www.cs.iit.edu/~mbilgic

https://twitter.com/bilgicm

Learning

- What's learning?
- Intro to Chapter 18: *"In which we describe agents that can improve their behavior through diligent study of their own experiences."*
- We do not make any philosophical statements about whether the agent is *truly* learning
- *"An agent is learning if it improves its performance on future tasks after making observations about the world."*

Why Learn and not program directly?

- We cannot anticipate all possible situations that the agent might find itself in
- Time/location/context changes knowledge and rules
- We might not know the solution crisp enough to program it
- We might not have time to encode all the knowledge

What to learn?

- Which action to take in a state (state → action)
- Outcomes of our actions (action → state)
- Mapping percepts to world states (percept → state)
- Utility of the states (state → utility)
- and more…

Feedback

- Unsupervised learning
  - No feedback; the agent discovers patterns in the data
  - E.g., clustering, dimensionality reduction, outlier detection
- Supervised learning
  - Feedback: input-output pairs
  - E.g., classification, regression, ranking
- Reinforcement learning
  - Feedback: rewards

Episodic vs Sequential

- Supervised and unsupervised learning are often episodic
  - E.g., speech recognition, medical diagnosis, credit score prediction, …
- Reinforcement learning is often sequential
  - E.g., game playing

Machine Learning

- ML is used to supplement several applications of AI
- Even though all the rage is now about deep learning, DL is a subfield of ML, and ML is a subfield of AI
- Example
  - Agents can combine the powers of search and ML to play games
  - Robots can use ML to make sense of their percepts and model the world, but they need to use search and planning to achieve goals

We'll Cover

1. Bayesian network parameter estimation
2. Supervised learning
    1. Naïve Bayes
    2. Logistic regression
    3. Overviews of decision trees and neural networks
3. Reinforcement learning

1. Bayesian network parameter estimation

- Given:
  - A set of random variables, V_i
    - E.g., age, gender, cholesterol level, etc.
  - A Bayesian network structure over these variables
    - E.g., a doctor can point out the most important correlations and causations
  - Data
    - E.g., existing patient records, where some or all V_i are known
- Goal:
  - Estimate the parameters needed for the Bayesian network, i.e., P(V_i | parentsOf(V_i))

Known Bayesian network structure

- In this class, we assume the structure is given
- How reasonable is this assumption?
  - In some domains, the expert might provide a reasonable structure to start with
- There are many methods that learn the structure of the Bayesian network from data
  - Those topics are covered in the CS583 – Probabilistic Graphical Models course in detail

Parameter estimation for BNs

- Assume the network structure is given over variables V_i
- Let d_j be a fully observed instance
  - $d_j = <V_1=\text{t}, V_2=\text{f}, \ldots, V_n=\text{t}>$
- The data $\mathcal{D}$ consists of fully observed instances
  - $\mathcal{D} = \{d_1, d_2, \ldots, d_m\}$
- Estimate the network parameters $P(V_i \mid \text{parents}(V_i))$
- Two approaches
  1. Maximum likelihood estimation
  2. Bayesian estimation

Simplest case – one variable

- Imagine we have a thumbtack
- Flip it, and it comes as heads or tails

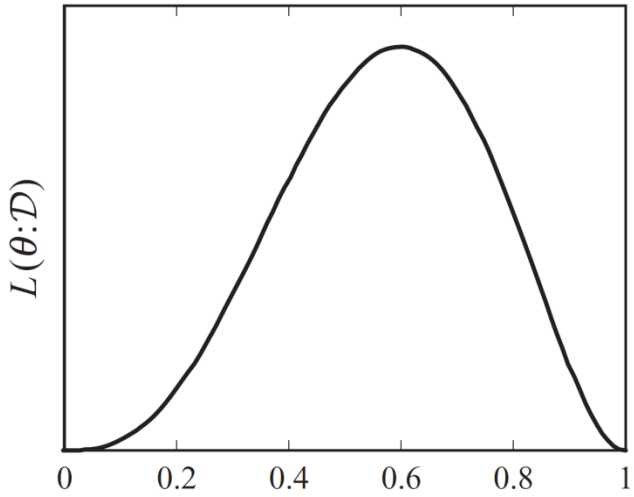
- P(Heads) = θ, P(Tails) = 1- θ
- Assume we flip it 100 times and it comes head 30 times
- What is θ?

THUMBTACK TOSSES

- Assume we have a set of thumbtack tosses
  - $\mathcal{D} = \{d_1, d_2, \ldots, d_{100}\}$
- Assume we have 30 heads and 70 tails
- P(Heads) = θ, P(Tails) = 1- θ
- θ can be any number between 0 and 1
- We have an infinite number of choices
  - θ=0, …, θ=0.3, …, θ=0.5, …, θ=1
- We want to formulate an objective function f(θ: $\mathcal{D}$), where, given 30 heads and 70 tails, f(θ: $\mathcal{D}$) achieves its maximum when θ=0.3
  - Any ideas?

Likelihood

- What is the probability, or *likelihood*, of seeing the sequence H, T, T, H, H?
  - $\theta*(1-\theta)*(1-\theta)*\theta*\theta = \theta^3(1-\theta)^2$



When is $L(\theta:\mathcal{D})$ maximum?

LIKELIHOOD/LOG-LIKELIHOOD

- Number of heads = k, number of tails = m-k
- Likelihood: $L(\theta:\mathcal{D}) = \theta^k(1-\theta)^{m-k}$
- Log-likelihood: $l(\theta:\mathcal{D}) = k\log\theta+(m-k)\log(1-\theta)$
- Note that $L(\theta:\mathcal{D})$ achieves its maximum for θ that maximizes $l(\theta:\mathcal{D})$
- Find θ that maximizes the log-likelihood
- Take derivate of $l(\theta:\mathcal{D})$ w.r.t. θ and set it to zero

Maximum likelihood for a multinomial

- Domain of X is {A, B, C}
- We see A a times, B b times, and C c times.
- P(X=A) is p, P(X=B) is q, and P(C) = $1 - p - q$
- What are p and q?
- Proof?

Let's see a few examples

- Simple structure
  - $X \rightarrow Y$
- General structure
  - The key is that the parameters for each variable can be optimized independently
  - Examples

Bayesian estimation

- Assume we flip a coin 10 times and we get 4 Heads, 6 Tails
  - What is P(C=H)?
- What if we repeat the flips 10M times and we get 4M Heads and 6M Tails?
- Bayesian estimation will let us encode our *prior knowledge*

To cut it short, (I mean really short)

- We'll encode our prior knowledge as a set of "imaginary" counts
- For example, we will assume that we have already seen α heads and β tails
- Assume we flip a coin 10 times and we get 4 Heads, 6 Tails
  - P(C=heads) = $(4 + \alpha) / (10 + \alpha + \beta)$
  - $\alpha = 0, \beta = 0$; P(C=h) = 4/10 = 0.4
  - $\alpha = 1, \beta = 1$; P(C=h) = 5/12 = 0.417
  - $\alpha = 10, \beta = 10$; P(C=h) = 14/30 = 0.467
  - $\alpha = 100, \beta = 100$; P(C=h) = 104/210 = 0.495
- Assume we flip a coin 1000 times and we get 400 Heads, 600 Tails
  - P(C=heads) = $(400 + \alpha) / (1000 + \alpha + \beta)$
  - $\alpha = 0, \beta = 0$; P(C=h) = 400/1000 = 0.4
  - $\alpha = 1, \beta = 1$; P(C=h) = 401/1002 = 0.4002
  - $\alpha = 10, \beta = 10$; P(C=h) = 410/1020 = 0.402
  - $\alpha = 100, \beta = 100$; P(C=h) = 500/1200 = 0.417

Imaginary counts

- Note that imaginary counts can be applied to any categorical variable, not necessarily just binary variables
- Also helps with dealing zero probabilities
- When all imaginary counts are 1, this is called Laplace smoothing
  - E.g, $\alpha = 1$, $\beta = 1$
- Let's see some examples

Next

- Supervised Learning