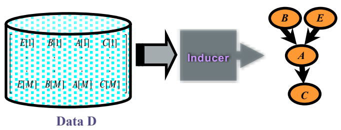
Bayesian Learning

(Ch. 20.1–20.2)



1

Bayesian Learning

- Bayesian probability: the view of probability as a **measure** of belief, as opposed to being a **frequency**
  - Does not mean that past statistics are ignored
  - Statistics of what has happened in the past **are** the knowledge that is conditioned on and used to update belief.
- **Models** are mathematical formulations of observed events
- **Parameters** are factors in the models
  - Specifically, those affecting observations

Mackworth & Poole Ch. 6
www.analyticsvidhya.com/blog/2016/06/bayesian-statistics-beginners-simple-english

2

Naïve Bayes

First, make the simplest possible independence assumption:

- Each attribute is independent of the values of the other attributes, **given the class variable** (the label)
  - In restaurants: Cuisine is independent of Patrons, **given a decision to stay**
- Embodied in a belief network where:
  - Features are nodes
  - Target variable (the classification) has no parents
  - The classification is the only parent of each input feature
- This requires:
  - Probability distributions $P(C)$ for target variable C (the classes, e.g., + or -)
  - $P(F_i \mid C)$ for each input feature F_i

3

Formulation Terms

- C: a **class**
  - What we're trying to classify into – e.g., positive (spam) or negative (not spam), cat or dog, yellow or not, …
- Example, data point, training datum, etc: a single example from which to learn, e.g., 
- F: a feature vector
  - F_1 .. F_n hold the values of each feature for some specific data point (so F_1 might be R, F_2 = G, F_3 = G, …)

4

Bayesian Formulation

For each example (training datum), predict C (the class) by conditioning on observed input features and by querying the classification

- The probability of class C given $F_1, \ldots, F_n$:
  $p(C \mid F_1, \ldots, F_n) = p(C)\, p(F_1, \ldots, F_n \mid C) / P(F_1, \ldots, F_n)$
- Denominator: normalizing constant to make probabilities sum to 1, which we call **α**

  $p(C \mid F_1, \ldots, F_n) = \alpha\, p(C)\, p(F_1, \ldots, F_n \mid C)$

  - Denominator does not depend on class
  - Therefore, not needed to determine the most likely class

5

Bayesian Formulation

- The probability of class C given $F_1, \ldots, F_n$ is:
  $p(C \mid F_1, \ldots, F_n) = p(C)\, p(F_1, \ldots, F_n \mid C) / P(F_1, \ldots, F_n)$
  $= \alpha\, p(C)\, p(F_1, \ldots, F_n \mid C)$
- Assumption: each feature is conditionally independent of the other features given C. Then:
  $p(C \mid F_1, \ldots, F_n) = \alpha\, p(C)\, \Pi_i\, p(F_i \mid C)$
- We can estimate each of these conditional probabilities from the observed **counts** in the training data:
  $p(F_i \mid C) = N(F_i, C) / N(C)$

6

Bayesian Formulation

- Given a data point with inputs $F_1=v_1,\ldots,F_k=v_k$:
- Use Bayes' rule to compute **posterior probability distribution** of the example's classification, C:
  - $$P(C \mid F_1=v_1,\ldots,F_k=v_k) = \frac{(P(F_1=v_1,\ldots,F_k=v_k \mid C) \times P(C))}{(P(F_1=v_1, \ldots, F_k=v_k))}$$
    $$= \frac{(P(F_1=v_1 \mid C)\times \cdots \times P(F_k=v_k \mid C)\times P(C))}{(\sum_C P(F_1=v_1 \mid C)\times \cdots \times P(F_k=v_k \mid C) \times P(C))}$$

7

Bayesian Formulation

- Given a data point with inputs $F_1=v_1,\ldots,F_k=v_k$:
- Use Bayes' rule to compute **posterior probability distribution** of the example's classification, C:

So for each possible class, you can calculate the probability of a new datum belonging to that class. The highest probability is the classification output.

8

Naive Bayes: Example

- p(Wait | Cuisine, Patrons, Rainy?)

  $= \alpha\, p(\text{Wait})\, p(\text{Cuisine} \mid \text{Wait})\, p(\text{Patrons} \mid \text{Wait})\, p(\text{Rainy?} \mid \text{Wait})$

naive Bayes assumption: is it reasonable?

9

9

Naive Bayes: Analysis

- Easy to implement
- Outperforms many more complex algorithms
  - Should almost always be used for baseline comparisons
- Works well when the independence assumption is appropriate
  - Often appropriate for **natural kinds**: classes that exist because they are useful in distinguishing the objects that humans care about

But…

- Can't capture interdependencies between variables (obviously)
- For that, we need Bayes nets!

10

10

- Binary Features: long, sweet, yellow (or not)
- What we know:
  - 50% are bananas
  - 30% are oranges
  - 20% are other fruits

Fruit	Long	Sweet	Yellow	Total
Banana	400	350	450	500
Orange	0	150	300	300
Other	100	150	50	200
Total	500	650	800	1000

- And:
  - 500 bananas: Long=400 (0.8), Sweet=350 (0.7), Yellow=450 (0.9)
  - 300 oranges: Long=0, Sweet= 150 (0.5),Yellow=300 (1.0)
  - 200 other: Long=100 (0.5), Sweet=150 (0.75), Yellow=50 (0.25)
- We are given a new fruit that is *Long*, *Sweet*, and *Yellow*.
- Set this up as a Bayes' reasoning problem.

11

- Binary Features: long, sweet, yellow (or not)
- What we know:
  - 50% are bananas
  - 30% are oranges
  - 20% are other fruits

Fruit	Long	Sweet	Yellow	Total
Banana	400	350	450	500
Orange	0	150	300	300
Other	100	150	50	200
Total	500	650	800	1000

- And:
  - 500 bananas: Long=400 (0.8), Sweet=350 (0.7), Yellow=450 (0.9)
  - 300 oranges: Long=0, Sweet= 150 (0.5),Yellow=300 (1.0)
  - 200 other: Long=100 (0.5), Sweet=150 (0.75), Yellow=50 (0.25)
- We are given a new fruit that is *Long*, *Sweet*, and *Yellow*.
- Set this up as a Bayes' reasoning problem. What are the odds of this new thing being a banana? An orange? An other?

Example from: https://towardsdatascience.com/all-about-naive-bayes-8e13cef044cf

12

Learning in Bayesian Networks

13

13

Naive Bayes: Analysis

- Easy to implement
- Outperforms many more complex algorithms
  - Should almost always be used for baseline comparisons
- Works well when the independence assumption is appropriate
  - Often appropriate for **natural kinds**: classes that exist because they are useful in distinguishing the objects that humans care about

But…

- Can't capture interdependencies between variables (obviously)
- For that, we need Bayes nets!

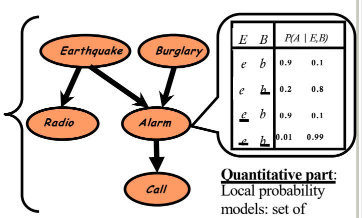
14

14

Quick Review: Bayes Nets

Qualitative part:

- Statistical independence statements (causality!)
- Directed acyclic graphs (DAG)
  - Nodes - **random variables of interest** (exhaustive, mutually exclusive states)
  - Edges - direct (causal-ish) influence



Quantitative part: Local probability models: set of conditional probability distributions.

15

Bayesian Learning: Bayes' Rule

- "Model" = learned belief about how the universe works
  - E.g., a fully trained classifier
- New idea: Instead of choosing the single most likely model or finding the set of all models consistent with training data, **compute the posterior probability of every model given the training examples**
- **Bayesian learning:** Compute posterior probability distribution of the class of a new example, conditioned on its input features and **all training examples**

21

21

Bayesian Learning: Bayes' Rule

- Given some **model space** (set of hypotheses h_i) and **evidence** (data D):
  - $P(h_i|D) = \alpha\, P(D|h_i)\, P(h_i)$
- We assume observations are independent of each other, given a model (hypothesis), so:
  - $P(h_i|D) = \alpha \prod_j P(d_j|h_i)\, P(h_i)$
- To predict the value of some unknown quantity C (e.g., the class label for a future observation):
  - $P(C|D) = \sum_i P(C|D, h_i)\, P(h_i|D) = \sum_i P(C|h_i)\, P(h_i|D)$

These are equal by our independence assumption

22

22

Example

- New example has inputs and target features (class variables)
  - Inputs: $X=x$
  - Target features: Y
  - e: **set** of training examples
- Goal: compute **$P(Y|X=x, e)$**
  - The **probability distribution** of target variables given the inputs and the examples
- A **model** is assumed to have generated the examples; M is set of models
- Then: $P(Y|x, e) = \sum_{m\in M} P(Y, m \,|x, e)$ $= \sum_{m\in M} P(Y \,|\, m, x, e) \times P(m|x, e)$ $= \sum_{m\in M} P(Y \,|\, m, x) \times P(m|e)$
- Bayes' rule: $P(m|e) = (P(e|m)\times P(m))/(P(e))$
- **Weight of each model** depends on how well it predicts the data, **plus** its prior probability

Details: http://artint.info/html/ArtInt_196.html

23

Bayesian Learning, 3 Ways

- **BMA (Bayesian Model Averaging)**
  - Don't just choose one hypothesis; instead, make predictions based on the weighted average of all hypotheses (or some set of best hypotheses)
- **MAP (Maximum *A Posteriori*) hypothesis**
  - Choose hypothesis with highest *a posteriori** probability, given data
  - **Maximize p(h_i | D)**
  - Generally easier than Bayesian learning
  - Closer to Bayesian prediction as more data arrives
- **MLE (Maximum Likelihood Estimate)**
  - Assume all hypotheses are equally likely *a priori***; best hypothesis maximizes the **likelihood** (i.e., probability of **data** given **hypothesis**)
  - **Maximize p(D | h_i)**

* *afterwards*

** *beforehand*

24

Bayesian Learning

- **BMA (Bayesian Model Averaging) –** average predictions of hypotheses
- **MAP (Maximum *A Posteriori*) hypothesis –** Maximize p(h_i | D)
- **MLE (Maximum Likelihood Estimate) –** Maximize p(D | h_i)
- **MDL (Minimum Description Length) principle:** Use some encoding to model the **complexity** of the hypothesis, and the fit of the data to the hypothesis, then **minimize** the overall description of h_i + D

25

Example: Coin Toss

- **Models** mathematically formulate observed events
- **Parameters** are factors in the models affecting outcomes
- **Toin Coss Example**
  - **Fairness of coin** is the parameter, θ;
  - **Outcome** of the events is data, D
    - E.g. 100 flips, heads = 72, tails = 28
  - Given (D), what is the **probability** this coin is fair (θ=0.5)?
  - Bayes' rule: P(θ|D) = (P(D|θ) × P(θ))/P(D)

www.analyticsvidhya.com/blog/2016/06/bayesian-statistics-beginners-simple-english

26

Example: Coin Toss

- Bayes : P(θ|D) = (P(D|θ) × P(θ))/P(D)
- **P(θ) is the prior:** the strength of our belief in the fairness of coin before the toss
  - Can have any degree of fairness between 0 and 1
- **P(D|θ) is the likelihood of observing this result** given distribution for θ
  - Probability of observing that number of heads in a particular number of flips, given a fair coin
- **P(D) is evidence**: the probability of observed data
  - Determined by summing (or integrating) across all possible values of θ, weighted by how strongly we believe in those particular values of θ
- **P(θ|D) is the posterior**: belief of our parameters after observing the evidence

www.analyticsvidhya.com/blog/2016/06/bayesian-statistics-beginners-simple-english

27

Example: Coin Toss

- Bayes : P(θ|D) = (P(D|θ) × P(θ))/P(D)
- P(θ) ... in befo...
  - C...
- P(D... dist...
  - Pr... of fli...
- P(D) is evidence: the probability of observed data
  - Determined by summing (or integrating) across all possible values of θ, weighted by how strongly we believe in those particular values of θ
- **P(θ|D) is the posterior**: belief of our parameters after observing the evidence

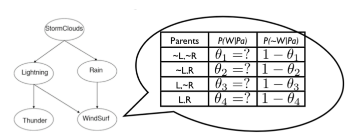
The point: If we had multiple **hypotheses** about the fairness of the coin, then this tells us the **probability** of seeing a certain sequence of flips for **each possible fairness** (hypothesis).

www.analyticsvidhya.com/blog/2016/06/bayesian-statistics-beginners-simple-english

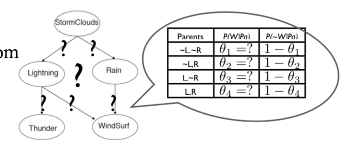
28

Learning in Bayes Nets

- Parameter Learning/Estimation: infer Θ from data, given G



- Structure Learning: inferring G and Θ from data



-

29

Project Break

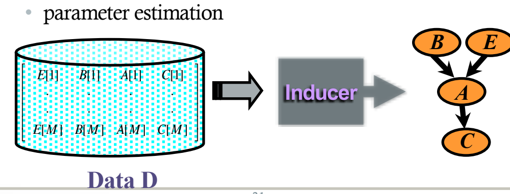
- Approach?
- Functions, inputs and outputs?
- NOT pseudocode
- **Questions?**

30

30

Learning Bayesian Networks

- Given training set $\boldsymbol{D} = \{\boldsymbol{x}[1], \ldots, \boldsymbol{x}[M]\}$
- Find B that best matches $\boldsymbol{D}$
  - model selection
  - parameter estimation



31

31

Parameter Estimation

- Assume known structure
- Goal: estimate BN param
  - entries in local probability models, P(X | Paren
- A good parameterization $\boldsymbol{\theta}$ is **likely** to ge[...]e observed data:

$$L(\theta : \boldsymbol{D}) = P(\boldsymbol{D} \mid \theta) = \prod_m P(x[m] \mid \theta)$$

- Maximum Likelihood Estimation (MLE) Principle: Choose $\boldsymbol{\theta}^*$ to maximize L

i.i.d. samples
independent and identically distributed (i.i.d.) if each random variable has the same probability distribution as the others and all are mutually independent

32

32

Sufficient Statistics

- Sufficient statistic: a function s(D) of data that summarizes relevant information computing the likelihood

$$s(\mathbf{D}) = s(\mathbf{D}') \Rightarrow L(\theta \mid \mathbf{D}) = L(\theta \mid \mathbf{D}')$$

- Sufficient statistics tell us all there is to know about data.

33

33

Parameter Estimation II

- Likelihood **decomposes** per the structure of the network
  → we get a separate estimation task for each parameter
- The MLE (maximum likelihood estimate) solution:
  - For each value x of a node X
  - And each instantiation $\boldsymbol{u}$ of $Parents(X)$
  - Just need to collect the counts for every combination of parents and children observed in the data

$$\theta^*_{x|u} = \frac{N(x,u)}{N(u)}$$

sufficient statistics

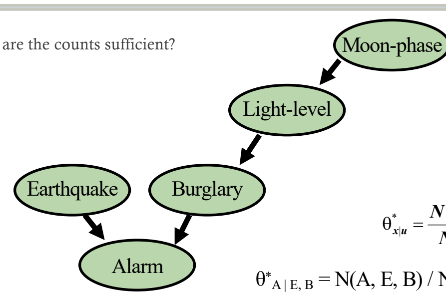
  - MLE: equivalent to assuming uniform prior over parameter values

34

34

Sufficient Statistics: Example

Why are the counts sufficient?



$$\theta^*_{x|u} = \frac{N(x,u)}{N(u)}$$

$\theta^*_{A \mid E, B} = N(A, E, B) / N(E, B)$

35

35

Examples

- Thumbtack tossing:
  - $(m_h, m_t) = (3,7)$. MLE: $\theta = 0.3$.
  - Reasonable. Data suggest that the thumbtack is biased toward tail.
- Coin tossing:
  - Case 1: $(m_h, m_t) = (3,7)$. MLE: $\theta = 0.3$. Not reasonable.
  - Our experience (prior) suggests strongly that coins are fair, hence $\theta=1/2$.
  - The size of the data set is too small to convince us this particular coin is biased.
  - The fact that we get (3, 7) instead of (5, 5) is probably due to randomness.
- Case 2: $(m_h, m_t) = (30{,}000, 70{,}000)$. MLE: $\theta = 0.3$. Reasonable.
  - Data suggest that the coin is after all biased, overshadowing our prior.
  - MLE does not differentiate these cases – does not take prior information into account.

37

37

Model Selection

Goal: Select the best network structure, given the data

Input:
- Training data
- Scoring function

Output:
- A network that maximizes the score

- This is NP-hard!

38

38

Structure Selection: Scoring

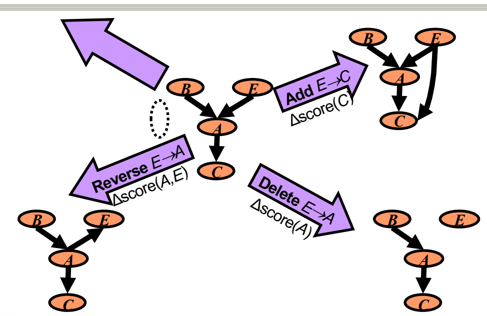
- Bayesian: prior over parameters and structure
- Find balance between model complexity and fit to data

Marginal likelihood ***Prior***

- Score (G:D) = log P(G|D) α log [P(D|G) P(G)]
- Marginal likelihood just comes from our parameter estimates
- Prior on structure can be any measure we want; typically a function of the network complexity

39

Heuristic Search



40

Variations on a Theme

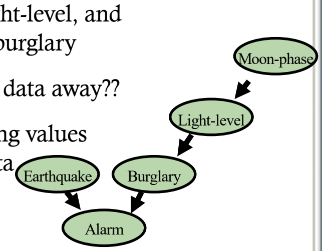
- **Known structure, fully observable**: only need to do parameter estimation
- **Unknown structure, fully observable:** do heuristic search through structure space, then parameter estimation
- **Known structure, missing values:** use expectation maximization (EM) to estimate parameters
- **Known structure, hidden variables:** apply adaptive probabilistic network (APN) techniques
- **Unknown structure, hidden variables:** too hard to solve!

42

42

Handling Missing Data

- Suppose that in some cases, we observe earthquake, alarm, light-level, and moon-phase, but not burglary
- Should we throw that data away??
- **Idea**: Guess the missing values based on the other data



43

43

EM (Expectation Maximization)

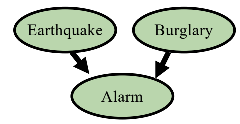
- **Guess** probabilities for nodes with **missing values** (e.g., based on other observations)
- **Compute the probability distribution** over the missing values, given our guess
- **Update the probabilities** based on the guessed values
- **Repeat** until convergence

44

44

EM Example

- Suppose we have observed Earthquake and Alarm but not Burglary for an observation on November 27
- We estimate the CPTs based on the *rest* of the data
- We then estimate P(Burglary) for November 27 from those CPTs
- Now we recompute the CPTs as if that estimated value had been observed
- Repeat until convergence!



45

45