

# Lecture 9a - Bayesian Learning

Jesse Hoey  
School of Computer Science  
University of Waterloo

March 4, 2020

Readings: Poole & Mackworth (2nd Ed.) Chapt. 10.1, 10.4

# Bayesian Learning

Basic premise:

- have a number of hypotheses or models
- don't know which one is "correct"
- Bayesians assume all are correct to a certain degree
- Have a distribution over the models
- Compute expected prediction given this average

Suppose  $X$  is input features, and  $Y$  is target feature,

$\mathbf{d} = \{x_1, y_1, x_2, y_2, \dots, x_N, y_N\}$  is evidence (data),  $x$  is a new input, and we want to know corresponding output  $y$ . We sum over all models,  $\mathbf{m} \in \mathbf{M}$

$$\begin{aligned} P(Y|x, \mathbf{d}) &= \sum_{\mathbf{m} \in \mathbf{M}} P(Y, \mathbf{m}|x, \mathbf{d}) \\ &= \sum_{\mathbf{m} \in \mathbf{M}} P(Y|\mathbf{m}, x, \mathbf{d})P(\mathbf{m}|x, \mathbf{d}) \\ &= \sum_{\mathbf{m} \in \mathbf{M}} P(Y|\mathbf{m}, x)P(\mathbf{m}|\mathbf{d}) \end{aligned}$$

# Candy Example

- Have a bag of Candy with 2 flavors (Lime, Cherry)
- Sold in bags with different ratios
  - ▶ 100% cherry
  - ▶ 75% cherry+25% lime
  - ▶ 50% cherry + 50% lime
  - ▶ 25% cherry + 75% lime
  - ▶ 100% lime
- With a random sample - what ratio is in the bag?

- **Hypotheses  $H$  (or models  $M$ )**: probabilistic theory about the world
  - ▶  $h_1$ : 100% cherry
  - ▶  $h_2$ : 75% cherry+25% lime
  - ▶  $h_3$ : 50% cherry + 50% lime
  - ▶  $h_4$ : 25% cherry + 75% lime
  - ▶  $h_5$ : 100% lime
- **Data  $D$** : evidence about the world
  - ▶  $d_1$ : 1<sup>st</sup> candy is lime
  - ▶  $d_2$ : 2<sup>nd</sup> candy is lime
  - ▶  $d_3$ : 3<sup>rd</sup> candy is lime
  - ▶ ...

We may have some prior distribution over the hypotheses:

Prior  $P(H) = [0.1, 0.2, 0.4, 0.2, 0.1]$

- Prior:  $P(H)$
- Likelihood:  $P(\mathbf{d}|H)$
- Evidence:  $\mathbf{d} = \{d_1, d_2, \dots, d_n\}$

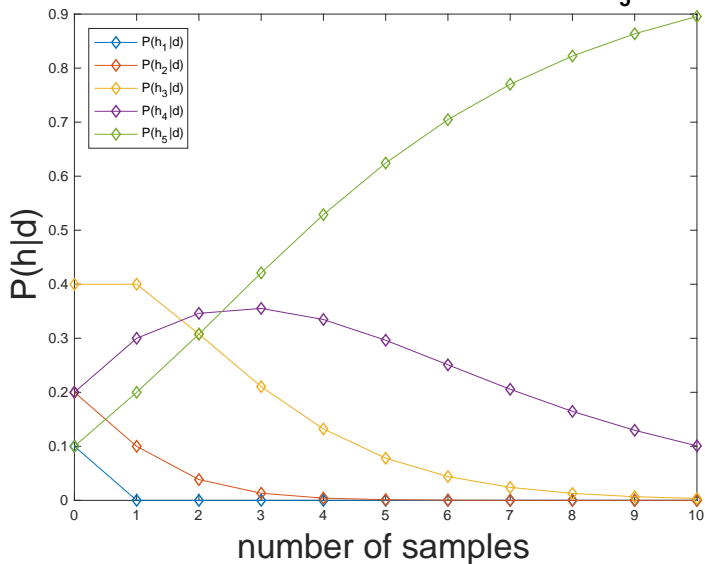
Bayesian learning: update the posterior (Bayes' theorem)

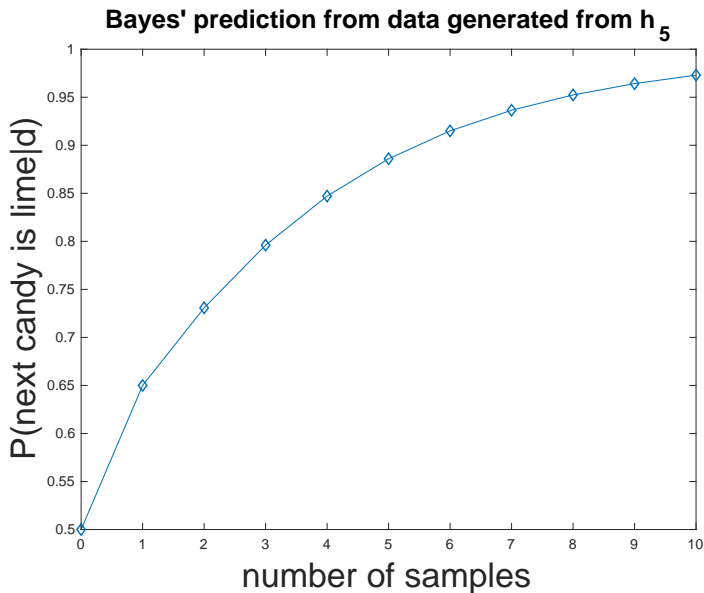
$$P(H|\mathbf{d}) \propto P(\mathbf{d}|H)P(H)$$

- want to predict  $X$ : (e.g. next candy)

$$\begin{aligned}P(X|\mathbf{d}) &= \sum_i P(X|\mathbf{d}, h_i)P(h_i|\mathbf{d}) \\ &= \sum_i P(X|h_i)P(h_i|\mathbf{d})\end{aligned}$$

- Predictions are weighted averages of the predictions of the individual hypotheses
- Hypotheses serve as "intermediaries" between raw data and prediction

Posteriors given data generated from  $h_5$ 





Bayesian learning properties:

- **Optimal**: given prior, no other prediction is correct more often than the Bayesian one
- **No overfitting**: prior/likelihood both penalise complex hypotheses

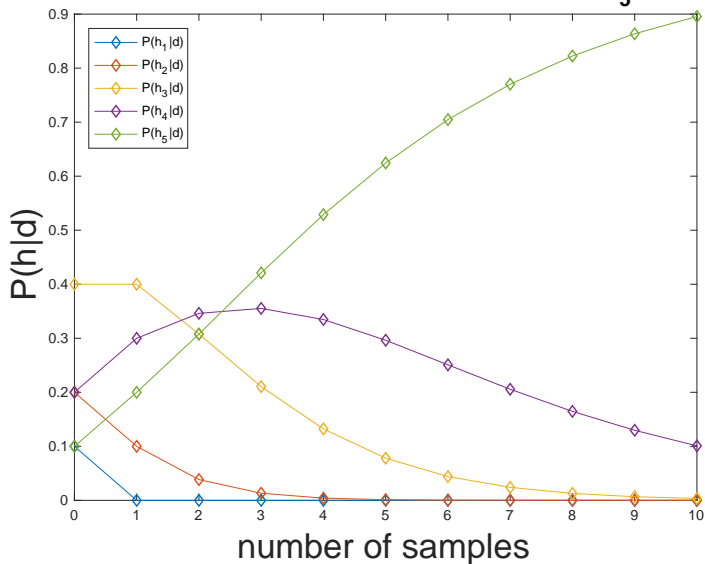
Price to pay:

- Bayesian learning may be intractable when hypothesis space is large
- sum over hypotheses space may be intractable

Solution: approximate Bayesian learning

# Maximum a posteriori

- Idea: make prediction based on **most probable hypothesis** :  
 $h_{MAP}$
- $h_{MAP} = \operatorname{argmax}_{h_i} P(h_i|d)$
- $P(X|d) \approx P(X|h_{MAP})$
- Contrast with Bayesian learning where **all hypotheses** are used

Posteriors given data generated from  $h_5$ 

# MAP properties

- MAP prediction less accurate than Bayesian one since it relies only on one hypothesis
- MAP and Bayesian predictions converge as data increases
- **no overfitting** (as in Bayesian learning)
- Finding  $h_{MAP}$  may be intractable:

$$\begin{aligned}h_{MAP} &= \operatorname{argmax}_h P(h|\mathbf{d}) \\ &= \operatorname{argmax}_h P(h)P(\mathbf{d}|h) \\ &= \operatorname{argmax}_h P(h) \prod_i P(d_i|h)\end{aligned}$$

product induces a non-linear optimisation

- can take the log to linearise

$$h_{MAP} = \operatorname{argmax}_h \left[ \log P(h) + \sum_i \log P(d_i|h) \right]$$

# Maximum Likelihood (ML)

- Idea: Simplify MAP by assuming uniform prior (i.e.  $P(h_i) = P(h_j) \forall i, j$ )

$$h_{MAP} = \operatorname{argmax}_h P(h)P(\mathbf{d}|h)$$

$$h_{ML} = \operatorname{argmax}_h P(\mathbf{d}|h)$$

- Make prediction based on  $h_{ML}$  only

$$P(X|\mathbf{d}) \approx P(X|h_{ML})$$

- ML prediction **less accurate** than Bayesian or MAP predictions since it ignores prior and relies on one hypothesis
- but ML, MAP and Bayesian converge as the amount of data increases
- more susceptible to **overfitting** : no prior
- $h_{ML}$  is often easier to find than  $h_{MAP}$

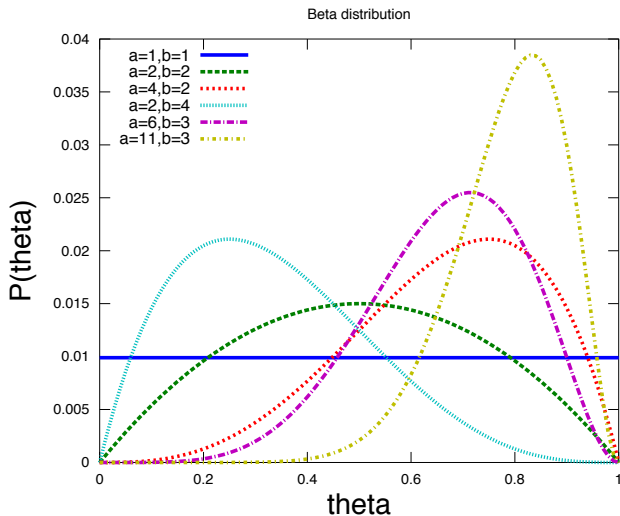
$$h_{ML} = \operatorname{argmax}_h \sum_i \log P(d_i | h)$$

# Binomial Distribution

- Generalise the hypothesis space to a continuous quantity
- $P(\text{Flavour} = \text{cherry}) = \theta$  (like a “coin weight”)
- $P(\text{Flavour} = \text{lime}) = (1 - \theta)$
- $P(k \text{ lime}, n \text{ cherry}) = \theta^n (1 - \theta)^k$  (one order)
- $P(k \text{ lime}, n \text{ cherry}) = \binom{n+k}{k} \theta^n (1 - \theta)^k$  (any order)

# Priors on Binomials

The Beta distribution  $B(\theta, a, b) = \theta^{a-1}(1 - \theta)^{b-1}$





# Bayesian classifiers

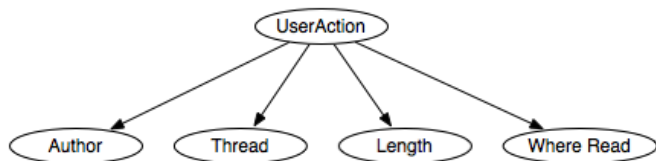
- Idea: if you knew the classification you could predict the values of features.

$$P(\text{Class}|X_1 \dots X_n) \propto P(X_1, \dots, X_n|\text{Class})P(\text{Class})$$

- Naïve Bayesian classifier:**  $X_i$  are independent of each other given the class.

Requires:  $P(\text{Class})$  and  $P(X_i|\text{Class})$  for each  $X_i$ .

$$P(\text{Class}|X_1 \dots X_n) \propto \left[ \prod_i P(X_i|\text{Class}) \right] P(\text{Class})$$



# Naïve Bayes classifier

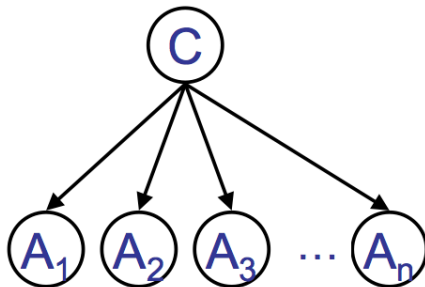
- Predict class  $C$  based on attributes  $A_i$
- Parameters:

$$\theta = P(C = \text{true})$$

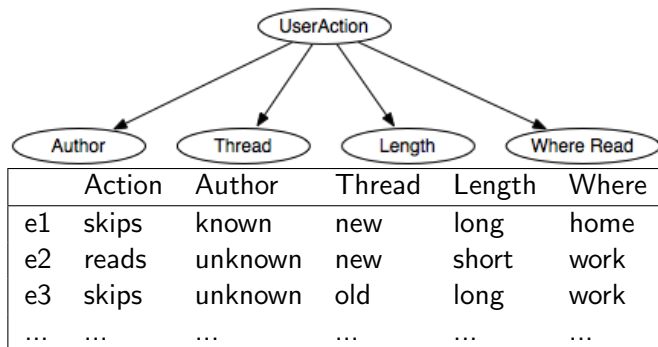
$$\theta_{i1} = P(A_i = \text{true} | C = \text{true})$$

$$\theta_{i0} = P(A_i = \text{true} | C = \text{false})$$

- Assumption:  $A_i$ s are independent given  $C$ .



# Naïve Bayes classifier



ML sets

- $\theta$  to relative frequency of reads, skips
- $\theta_{i1}$  to relative frequency of  $A_i$  given reads, skips

$$\theta_{i1} = \frac{\text{number of articles that are read and have } A_i = \text{true}}{\text{number of articles that are read}}$$

$$\theta_{i0} = \frac{\text{number of articles that are skipped and have } A_i = \text{true}}{\text{number of articles that are skipped}}$$

# Laplace correction

- If a feature never occurs in the training set, but does in the test set,
- ML may assign zero probability to a high likelihood class.
- add 1 to the numerator, and add  $d$  (arity of variable) to the denominator
- assign:

$$\theta_{i1} = \frac{(\text{number of articles that are read and have } A_i = \textit{true}) + 1}{\text{number of articles that are read} + 2}$$

$$\theta_{i0} = \frac{(\text{number of articles that are skipped and have } A_i = \textit{true}) + 1}{\text{number of articles that are skipped} + 2}$$

# Bayesian Network Parameter Learning (ML)

For fully observed data

- Parameters  $\theta_{V,pa(V)=\mathbf{v}}$
- CPTs  $\theta_{V,pa(V)=\mathbf{v}} = P(V|Pa(V) = \mathbf{v})$
- Data  $\mathbf{d}$ :

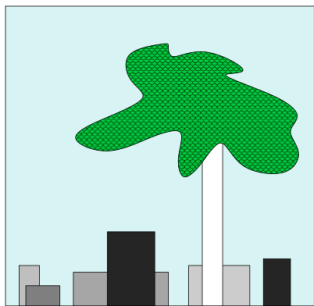
$$d_1 = \langle V_1 = v_{1,1}, V_2 = v_{2,1}, \dots, V_n = v_{n,1} \rangle$$

$$d_2 = \langle V_1 = v_{1,2}, V_2 = v_{2,2}, \dots, V_n = v_{n,2} \rangle$$

...

- Maximum likelihood: Set  $\theta_{V,pa(V)=\mathbf{v}}$  to the relative frequency of values of  $V$  given the values  $\mathbf{v}$  of the parents of  $V$

# Occam's Razor



e.g. from MacKay

[www.inference.phy.cam.ac.uk/mackay/itila/book.html](http://www.inference.phy.cam.ac.uk/mackay/itila/book.html)

# Occam's Razor

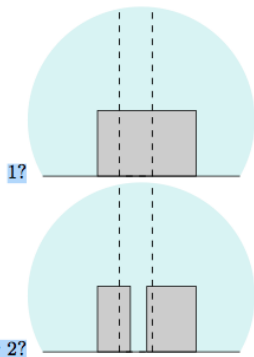
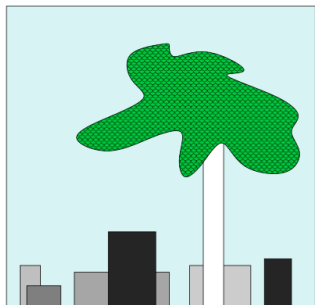


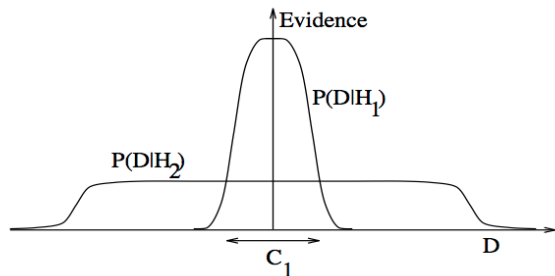
Figure 28.2. How many boxes are behind the tree?

e.g. from MacKay

[www.inference.phy.cam.ac.uk/mackay/itila/book.html](http://www.inference.phy.cam.ac.uk/mackay/itila/book.html)

# Occam's Razor

- Simplicity is encouraged in the likelihood function:
- $H_2$  is more complex than  $H_1$ ,
- so can explain more datasets  $D$ ,
- but each with lower probability





# Minimum Description Length

Bayesian learning: update the posterior (Bayes' theorem)

$$P(H|\mathbf{d}) = kP(\mathbf{d}|H)P(H)$$

So

$$-\log P(H|\mathbf{d}) = -\log P(\mathbf{d}|H) - \log P(H)$$

- first term : number of bits to encode the data given the model
- second term : number of bits to encode the model
- **MDL principle** is to choose the model that minimizes the number of bits it takes to describe both the model and the data given the model.
- MDL is equivalent to Bayesian model selection

- Supervised Learning under Uncertainty (Poole & Mackworth (2nd Ed.) chapter 7.3.2,7.5-7.6)