Lecture 9a - Bayesian Learning

Jesse Hoey
School of Computer Science
University of Waterloo

June 27, 2022

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

Bayesian Learning

Suppose X is input features, and Y is target feature, $\mathrm{d} = \{x_1, y_1, x_2, y_2, \ldots, x_N, y_N\}$ is evidence (data), x is a new input, and we want to know corresponding output y.
We sum over all models, $\mathrm{m} \in \mathrm{M}$

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

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?
- see `bayesian-learning.pdf`

Statistical Learning

- 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^{st} candy is lime
  - d_2: 2^{nd} candy is lime
  - d_3: 3^{rd} candy is lime
  - ...

We may have some prior distribution over the hypotheses:
Prior $P(H) = [0.1, 0.2, 0.4, 0.2, 0.1]$

Bayesian Learning

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

Bayesian learning: update the posterior (Bayes' theorem)

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

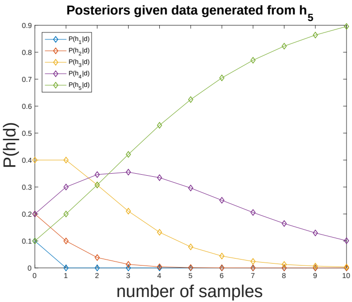
Bayesian Prediction

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

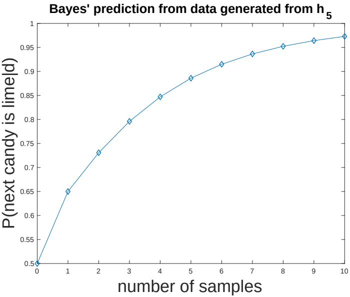
$$P(X|\mathrm{d}) = \sum_i P(X|\mathrm{d}, h_i)P(h_i|\mathrm{d})$$
$$= \sum_i P(X|h_i)P(h_i|\mathrm{d})$$

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

Posterior



Bayesian Prediction



Bayesian Learning

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:

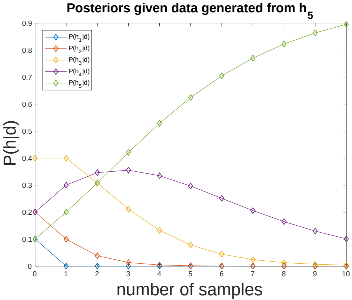
- Bayesian learning may be intractable when hypothesis space is large
- sum over hypotheses space may be intractable

Solution: approximate Bayesian learning

Maximum a posteriori

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

Posterior



MAP properties

- MAP prediction less accurate than full Bayesian 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:

$$\begin{aligned} h_{MAP} &= argmax_h P(h|\mathrm{d}) \\ &= argmax_h P(h)P(\mathrm{d}|h) \\ &= argmax_h P(h)\prod_i P(d_i|h) \end{aligned}$$

product induces a non-linear optimisation
- can take the log to linearise

$$h_{MAP} = argmax_h \left[logP(h) + \sum_i logP(d_i|h) \right]$$

Maximum Likelihood (ML)

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

$$h_{MAP} = argmax_h P(h)P(\mathrm{d}|h)$$

$$h_{ML} = argmax_h P(\mathrm{d}|h)$$

- Make prediction based on h_{ML} only

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

ML Properties

- 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}

$$h_{ML} = argmax_h \sum_i logP(d_i|h)$$

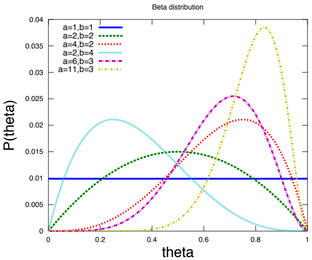
- see bayesian-learning.pdf for worked examples

Binomial Distribution

- Generalise the hypothesis space to a continuous quantity
- $P(Flavour = cherry) = \theta$ (like a "coin weight")
- $P(Flavour = lime) = (1-\theta)$
- $P(k\ lime, n\ cherry) = \theta^n(1-\theta)^k$ (one order)
- $P(k\ lime, n\ cherry) = \binom{n+k}{k}\theta^n(1-\theta)^k$ (any order)
- see bayesian-learning.pdf for worked examples

Priors on Binomials

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



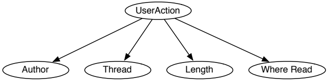
Bayesian classifiers

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

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

- Naïve Bayesian classifier: X_i are independent of each other given the class.
  Requires: $P(Class)$ and $P(X_i|Class)$ for each X_i.

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

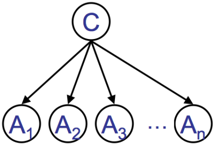


Naïve Bayes classifier

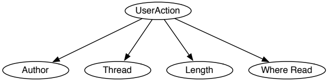
- Predict class C based on attributes A_i
- Parameters:

$$\theta = P(C = true)$$
$$\theta_{i1} = P(A_i = true|C = true)$$
$$\theta_{i0} = P(A_i = true|C = false)$$

- Assumption: A_is are independent given C.



Naïve Bayes classifier



	Action	Author	Thread	Length	Where
e1	skips	known	new	long	home
e2	reads	unknown	new	short	work
e3	skips	unknown	old	long	work
...	...	...	...	...	...

ML sets

- θ to relative frequency of reads, skips
- θ_{i1} to relative frequency of A_i given reads, skips

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

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

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:

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

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

- like a pseudocount
- see `naivebayesml.pdf`

Bayesian Network Parameter Learning (ML)

For fully observed data

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

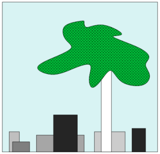
$$d_1 =< V_1 = v_{1,1}, V_2 = v_{2,1}, \ldots, V_n = v_{n,1} >$$

$$d_2 =< V_2 = v_{1,2}, V_2 = v_{2,2}, \ldots, V_n = v_{n,2} >$$

$$\ldots$$

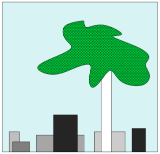
- Maximum likelihood: Set $\theta_{V,pa(V)=v}$ to the relative frequency of values of V given the the values v of the parents of V

Occam's Razor



e.g. from MacKay
`www.inference.phy.cam.ac.uk/mackay/itila/book.html`

Occam's Razor



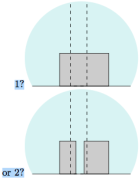
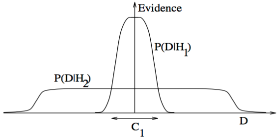


Figure 28.2. How many boxes are behind the tree?

e.g. from MacKay
`www.inference.phy.cam.ac.uk/mackay/itila/book.html`

Occam's Razor

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



Overfitting

Test set errors caused by:

- bias: the error due to the algorithm finding an imperfect model.
  - representation bias: model is too simple
  - search bias: not enough search
- variance: the error due to lack of data.
- noise: the error due to the data depending on features not modeled or because the process generating the data is inherently stochastic.
- bias-variance trade-off:
  - Complicated model, not enough data (low bias, high variance)
  - Simple model, lots of data (high bias, low variance)
- see handout `biasvariance.pdf`

Minimum Description Length

Bayesian learning: update the posterior (Bayes' theorem)

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

So

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

- 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

Next:

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