SNS COLLEGE OF TECHNOLOGY

Coimbatore-35

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A++' Grade

Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF INFORMATION TECHNOLOGY

19CSE303 – ARTIFICIAL INTELLIGENCE

III YEAR IV SEM

UNIT V – **LEARNING**

TOPIC : Statistical Learning

Statistical Learning

- Data – instantiations of some or all of the random variables describing the domain; they are *evidence*
- Hypotheses – probabilistic theories of how the domain works
- The Surprise candy example: two flavors in very large bags of 5 kinds, indistinguishable from outside
  - h1: 100% cherry – P(c|h1) = 1, P(l|h1) = 0
  - h2: 75% cherry + 25% lime
  - h3: 50% cherry + 50% lime
  - h4: 25% cherry + 75% lime
  - h5: 100% lime

- Problem formulation
  - Given a new bag, random variable H denotes the bag type (h1 – h5); D_i is a random variable (cherry or lime); after seeing D_1, D_2, ..., D_N, *predict* the flavor (value) of D_{N-1}.
- Bayesian learning
  - Calculates the probability of each hypothesis, given the data and makes predictions on that basis
    - $P(h_i|\mathbf{d}) = \alpha P(\mathbf{d}|h_i)P(h_i)$, where **d** are observed values of **D**
  - To make a prediction about an unknown quantity X
    - $P(X|\mathbf{d})=\Sigma_i P(X|\mathbf{d},h_i)P(h_i|\mathbf{d})=\Sigma_i P(X|h_i)P(h_i|\mathbf{d})=\Sigma_i P(X|h_i)P(\mathbf{d}|h_i)P(h_i)/P(\mathbf{d})$

    assuming that h_i determines a probability distribution over X

    Predictions use a likelihood-weighted average over hypotheses
    - Hypothesis *prior* $P(h_i)$ and *likelihood* of the data under each h_i, $P(\mathbf{d}|h_i)$;
      - One distribution for $P(h_i)$ is <0.1, 0.2, 0.4, 0.2, 0.1>.
    - h_i are *intermediaries* between raw data and predictions
    - No need to pick one best-guess hypothesis

- $P(\mathbf{d}|h_i) = \Pi_j P(d_j|h_i)$ assuming observations are i.i.d. (independent, and identically distributed)
- How each $P(h_i|\mathbf{d})$ changes when a sequence of 10 lime candies is observed (Fig. 20.1)
  - True hypothesis eventually dominates the Bayesian prediction – the feature of Bayesian learning
  - Bayesian prediction is optimal; its hypothesis space is usually very large or infinite
  - Let's see how these curves (Fig.20.1 a) are obtained
    - P(h2|l1) = ; P(h2|l1,l2) = ; P(h2|l1,l2,l3) = ; …
  - How to obtain Fig.20.1 b – $P(X|\mathbf{d})$ for X = lime

Common approximations

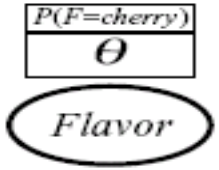
- MAP (maximum a posteriori): $P(X|\mathbf{d}) \sim P(X|h_{MAP})$
  - P(lime|l1,l2,l3) = 0.8 (from Fig 20.1b), but after seeing l1,l2,and l3, P(h5|l1,l2,l3) is the max, P(lime|h5)=1.
  - h_{MAP} is h_i that maximizes $P(h_i|\mathbf{d}) \cong P(\mathbf{d}|h_i)P(h_i)$ (Fig 20.1a)
    - Finding MAP hypotheses is much easier than Bayes Learning
    - B and MAP learning use the prior to penalize complexity
    - MAP learning chooses h_i that compresses data most, or minimum description length (MDL)
- ML can be obtained from MAP if $P(h_i)$ is uniform
  - h_{ML} is h_i that maximizes $P(\mathbf{d}|h_i)$ (Why?)
  - It is reasonable when (1) no preferable hypothesis a priori, (2) data set is large
- In short, $P(X|\mathbf{d}) \cong \Sigma_i P(X|h_i)P(\mathbf{d}|h_i)P(h_i)$ – (Bayes learning)
  - $\cong P(X|h_{MAP})P(\mathbf{d}|h_{MAP})P(h_{MAP})$ – (MAP)
  - $\cong P(X|h_{ML})P(\mathbf{d}|h_{ML})$ – (ML)

Learning with Complete Data

- Parameter learning - to find the numerical parameters for a probability model whose structure is fixed
- Data are complete when each data point contains values for every variable in the model
- *Maximum-likelihood* parameter learning: discrete model
  - Two examples: one and three parameters
    - To be seen in next 3 slides (From the book authors' slides)
  - With complete data, ML parameter learning problem for a Bayesian network decomposes into separate learning problems, one for each parameter
  - A significant problem with ML learning – 0 may be assigned to some events that have not been observed
    - Various tricks are used to avoid this problem. One is to assign count = 1 instead of 0 to each event

ML parameter learning in Bayes nets

Bag from a new manufacturer; fraction θ of cherry candies?
Any θ is possible: continuum of hypotheses h_θ
θ is a parameter for this simple (binomial) family of models



Suppose we unwrap N candies, c cherries and $\ell = N - c$ limes
These are i.i.d. (independent, identically distributed) observations, so

$$P(\mathbf{d}|h_\theta) = \prod_{j=1}^{N} P(d_j|h_\theta) = \theta^c \cdot (1-\theta)^\ell$$

Maximize this w.r.t. θ—which is easier for the log-likelihood:

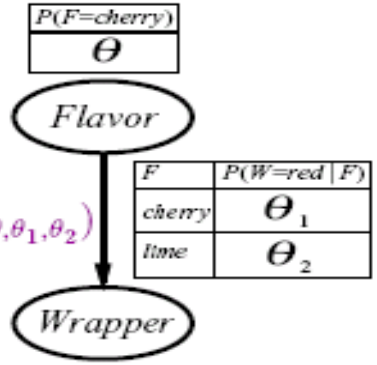
$$L(\mathbf{d}|h_\theta) = \log P(\mathbf{d}|h_\theta) = \sum_{j=1}^{N} \log P(d_j|h_\theta) = c\log\theta + \ell\log(1-\theta)$$

$$\frac{dL(\mathbf{d}|h_\theta)}{d\theta} = \frac{c}{\theta} - \frac{\ell}{1-\theta} = 0 \quad \Rightarrow \quad \theta = \frac{c}{c+\ell} = \frac{c}{N}$$

Seems sensible, but causes problems with 0 counts!

Multiple parameters

Red/green wrapper depends probabilistically on flavor:



Likelihood for, e.g., cherry candy in green wrapper:

$$\begin{aligned} &P(F = cherry, W = green|h_{\theta,\theta_1,\theta_2}) \\ &= P(F = cherry|h_{\theta,\theta_1,\theta_2})P(W = green|F = cherry, h_{\theta,\theta_1,\theta_2}) \\ &= \theta \cdot (1 - \theta_1) \end{aligned}$$

N candies, r_c red-wrapped cherry candies, etc.:

$$P(\mathbf{d}|h_{\theta,\theta_1,\theta_2}) = \theta^c(1-\theta)^\ell \cdot \theta_1^{r_c}(1-\theta_1)^{g_c} \cdot \theta_2^{r_\ell}(1-\theta_2)^{g_\ell}$$

$$\begin{aligned} L &= [c \log\theta + \ell \log(1-\theta)] \\ &+ [r_c \log\theta_1 + g_c \log(1-\theta_1)] \\ &+ [r_\ell \log\theta_2 + g_\ell \log(1-\theta_2)] \end{aligned}$$

Multiple parameters contd.

Derivatives of L contain only the relevant parameter:

$$\frac{\partial L}{\partial \theta} = \frac{c}{\theta} - \frac{\ell}{1-\theta} = 0 \qquad \Rightarrow \quad \theta = \frac{c}{c+\ell}$$

$$\frac{\partial L}{\partial \theta_1} = \frac{r_c}{\theta_1} - \frac{g_c}{1-\theta_1} = 0 \qquad \Rightarrow \quad \theta_1 = \frac{r_c}{r_c+g_c}$$

$$\frac{\partial L}{\partial \theta_2} = \frac{r_\ell}{\theta_2} - \frac{g_\ell}{1-\theta_2} = 0 \qquad \Rightarrow \quad \theta_2 = \frac{r_\ell}{r_\ell+g_\ell}$$

With complete data, **parameters can be learned separately**

After optimization

- So, we are convinced by the previous optimization procedure
- What's the learning from data?
  - Estimate the probabilities from data
  - The details can be seen in the example of NBC

Naïve Bayes models

- The most common Bayesian network model used in machine learning
  - Figure 20.3 Comparison between DT and NBC
- It assumes that the attributes are conditionally independent of each other, given class
  - Fig 20.2(b) is a NBC with the parameters for i^{th} instance as:
    $\Theta=P(C=T)$, $\Theta_{i1}=P(X_{i1}=T|C=T)$, $\Theta_{i2}=P(X_{i2}=T|C=F)$
    $\Theta=P(Cherry)$, $\Theta_{i1}=P(Red|Cherry)$, $\Theta_{i2}=P(Red|\neg Cherry)$
- A deterministic prediction can be obtained by choosing the most likely class
  - $\mathbf{P}(C|x_1,x_2,\ldots,x_n) = \alpha\mathbf{P}(C)\,\Pi_i\,\mathbf{P}(x_i|C)$
- NBC has no difficulty with nosy data
  - For n Boolean attributes, there are just 2n+1 parameters, no search is required for finding h_{ML}

Learning with Hidden Variables

- Many real-world problems have hidden variables which are not observable in the data available for learning.
- Question: If a variable (disease) is not observed, why not construct a model without it?
- Answer: Hidden variables can dramatically reduce the number of parameters required to specify a Bayesian network. This results in the reduction of needed amount of data for learning.
  - Fig. 20.7 from 708 parameters to 78.

EM: Learning mixtures of Gaussians

- The unsupervised clustering problem (Fig 20.8 with 3 components): $P(\mathbf{x}) = \Sigma^{k}_{i=1} P(C=i)P(\mathbf{x}|C=i)$
  - If we knew which component generated each x_j, we can get μ,σ
  - If we knew the parameters of each component, we know which c_i should x_j belong to. However, we do not know either, …
- EM – expectation and maximization
  - Pretend we know the parameters of the model and then to infer the probability that each x_j belongs to each component; iterate until convergence.
- For the mixture of Gaussians, initialize the mixture model parameters arbitrarily; and iterate the following
  - E-step: Compute $p_{ij} = P(C=i|\mathbf{x}_j) = \alpha P(\mathbf{x}_j|C=i)P(C=i)$
    $P(C=i) = p_i = \Sigma_j p_{ij}$
  - M-step: Compute the new $\mu_i = \Sigma_j p_{ij}\mathbf{x}_j/p_i$, $\Sigma_i p_{ij}\mathbf{x}_j\mathbf{x}_j^T/p_i$, $w_i = p_i$ (component weight)

- E-step computes the expected value p_{ij} of the *hidden indicator variables* Z_{ij}, where Z_{ij} is 1 if x_j was generated by *i*-th component, 0 otherwise
- M-step finds the new values of the parameters that maximize the log likelihood of the data, given the expected values of Z_{ij}
- Fig 20.8(c) is learned from Fig. 20.8(a)
  - Fig 20.9(a) plots the log likelihood of the data as EM progresses
    - EM increases the log likelihood of the data at each iteration.
    - EM can reach a local maximum in likelihood.

EM: Learning Bayesian networks with hidden variables

- Fig 20.10: two bags of mixed candies described by 3 features: Flavor, Wrapper, and Hole
  - The candy distribution is described by a naïve Bayes: the features are independent, given the bag
  - The parameters are: Θ – the probability that a candy from bag 1; Θ_{F1} and Θ_{F2} – the probabilites that the flavor is cherry given that a candy from bag 1 and bag2; Θ_{W1} and Θ_{W2} for red wrapper from bag1 and bag 2; and Θ_{H1} and Θ_{H2} that the candy has a hole from bag1 and bag2.
- EM details are …

Instance-based Learning

- Parametric vs. nonparametric learning
  - Learning focuses on fitting the *parameters* of a restricted family of probability models to an unrestricted data set
  - Parametric learning methods are often simple and effective, but can oversimplify what's really happening
  - Nonparametric learning allows the hypothesis complexity to grow with the data
  - IBL is nonparametric as it constructs hypotheses directly from the training data.

Nearest-neighbor models

- The key idea: Neighbors are similar
  - Density estimation example: estimate x's probability density by the density of its neighbors
  - Connecting with table lookup, NBC, decision trees, …
- How define neighborhood N
  - If too small, no any data points
  - If too big, density is the same everywhere
  - A solution is to define N to contain k points, where k is large enough to ensure a meaningful estimate
    - For a fixed k, the size of N varies (Fig 21.12)
    - The effect of size of k (Figure 21.13)
    - For most low-dimensional data, k is usually between 5-10

K-NN for a given query x

- Which data point is nearest to x?
  - We need a distance metric, D(x1, x2)
  - Euclidean distance D_E is a popular one
  - When each dimension measures something different, it is inappropriate to use D_E (Why?)
  - Important to standardize the scale for each dimension
    - Mahalanobis distance is one solution
  - Discrete features should be dealt with differently
    - Hamming distance
- Use k-NN to predict
- High dimensionality poses another problem
  - The nearest neighbors are usually a long way away!
    - A d-dimensional hypercube of side $b \leq 1$, its volume is b^d, given N data points, for each neighborhood to have k points, how big is b?

Summary

- Bayesian learning formulates learning as a form of probabilistic inference, using the observations to update a prior distribution over hypotheses.
- Maximum a posteriori (MAP) selects a single most likely hypothesis given the data.
- Maximum likelihood simply selects the hypothesis that maximizes the likelihood of the data (= MAP with a uniform prior).
- EM can find local maximum likelihood solutions for hidden variables.
- Instance-based models use the collection of data to represent a distribution.
  - Nearest-neighbor method