BAYESIAN LEARNING

Outline

- Statistical Learning
- Learning with Complete Data
- Learning with Hidden Variables: The EM Algorithm

Based on the textbook by Stuart Russell & Peter Norvig:

Artificial Intelligence, A Modern Approach (2nd Edition)

Sections 20.1–20.3

1. STATISTICAL LEARNING

- The **data**, i.e. instantiations of some or all random variables describing the domain, serve as evidence.
- **Hypotheses** are probabilistic theories of how the domain works.
- The aim is to make a *prediction* concerning an unknown quantity X given some data and hypotheses.
- In **Bayesian learning**, the probability of each hypothesis is calculated, given the data, and predictions are made on that basis.
- Predictions are made by using *all* the hypotheses, weighted by their probabilities, rather than by using a single "best" hypothesis.

Example. Our favorite *Surprise* candy comes in two flavors, cherry and lime, but they are wrapped in an indistinguishable way.

The candy is sold in large (indistinguishable) bags containing various mixtures of the two flavors:

1. 100% cherry
2. 75% cherry and 25% lime
3. 50% cherry and 50% lime
4. 25% cherry and 75% lime
5. 100% lime

Given a new bag of candy, the random variable H (for *hypothesis*) denotes the type of the bag, with possible values h_1 through h_5.

☞ The agent needs to infer a probabilistic model of the world.

Bayesian Learning

- Let $\mathbf{D}$ represent all the data with observed value $\mathbf{d}$.
- The probability of each hypothesis h_i is obtained by Bayes' rule:

$$P(h_i \mid \mathbf{d}) = \alpha P(\mathbf{d} \mid h_i) P(h_i).$$

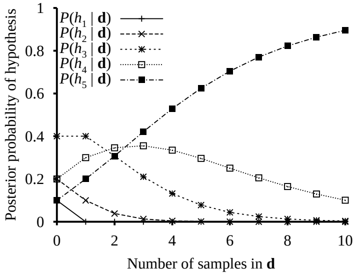
- Assuming that each h_i specifies a complete distribution for an unknown quantity X, Bayesian learning is characterized by

$$\mathbf{P}(X \mid \mathbf{d}) = \sum_i \mathbf{P}(X \mid \mathbf{d}, h_i) P(h_i \mid \mathbf{d}) = \sum_i \mathbf{P}(X \mid h_i) P(h_i \mid \mathbf{d}).$$

- The key quantities are the **hypothesis** prior $P(h_i)$ and the **likelihood** of the data under each hypothesis $P(\mathbf{d} \mid h_i)$.
- If the observations are *independently and identically distributed* (**i.i.d.** for short), then $P(\mathbf{d} \mid h_i) = \prod_j P(d_j \mid h_i)$.

Example. For the candy example, the prior distribution over $h_1, \ldots, h_5$ is given by $\langle 0.1, 0.2, 0.4, 0.2, 0.1 \rangle$, as advertised by the manufacturer.

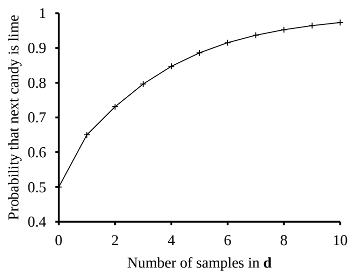
- If the bag is really an all-lime bag (h_5) and the first 10 candies are consequently all lime, then $P(\mathbf{d} \mid h_3) = 0.5^{10}$.
- The posterior probabilities of the five hypotheses change as the sequence of 10 lime candies is observed:



Example. The probability that the next candy is lime becomes

$$P(d_{N+1} = lime) = \sum_{i=1}^{5} P(d_{N+1} = lime \mid h_i) P(h_i \mid d_1 = lime, \ldots, d_N = lime).$$

When $N = 0$, we obtain $P(d_1 = lime) = \sum_{i=1}^{5} P(d_1 = lime \mid h_i) P(h_i)$ $= 0.0 \times 0.1 + 0.25 \times 0.2 + 0.5 \times 0.4 + 0.75 \times 0.2 + 1.0 \times 0.1 = 0.5$.



☞ The true hypothesis eventually dominates Bayesian prediction.

MAP and ML Hypotheses

- Unfortunately, the hypothesis space is usually very large or infinite which makes the Bayesian approach intractable.
- A common approximation is to use **maximum a posteriori** (MAP) **hypothesis** h_{MAP} — a hypothesis h_i that maximizes $P(h_i \mid \mathbf{d})$:

$$\mathbf{P}(X \mid \mathbf{d}) \approx \mathbf{P}(X \mid h_{\mathrm{MAP}}).$$

- To determine h_{MAP}, it is sufficient to maximize $P(\mathbf{d} \mid h_i)P(h_i)$, or alternatively, to minimize $-\log_2 P(\mathbf{d} \mid h_i) - \log_2 P(h_i)$.
- In some cases (recall the subjective nature of priors), the prior probabilities $P(h_i)$ can be assumed to be **uniformly** distributed.
- Then maximizing $P(\mathbf{d} \mid h_i)$ produces a **maximum-likelihood** (ML) **hypothesis** h_{ML} — a special case of h_{MAP}.

2. LEARNING WITH COMPLETE DATA

- A **parameter learning task** is about finding the numerical parameters for a probability model having a fixed structure.
- Data are **complete** when each data point contains values for every variable in the probability model being learned.
- Complete data greatly simplifies parameter learning.
- We will consider parameter learning in two simple settings:
  1. Maximum-likelihood parameter learning
  2. Naive Bayes models
- See the course book for further examples such as continuous models and strategies to learn Bayes network structure.

Maximum Likelihood Parameter Learning

- Suppose arbitrary cherry–lime proportions in the candy example.
- The **parameter** θ is the proportion of cherry candies.
- Out of N candies, the likelihood of c cherries and $l = N - c$ limes is

$$P(\mathbf{d} \mid h_\theta) = \prod_{j=1}^{N} P(d_j \mid h_\theta) = \theta^c (1-\theta)^l.$$

- This can be maximized by maximizing

$$L(\mathbf{d} \mid h_\theta) = \log P(\mathbf{d} \mid h_\theta) = c \log \theta + l \log(1-\theta).$$

- By setting $\frac{dL(\mathbf{d}|h_\theta)}{d\theta} = 0$, one obtains a ML hypothesis $\theta = \frac{c}{c+l} = \frac{c}{N}$.
- As a shortcoming, the ML hypothesis assigns zero probability to events (e.g., no cherry candies) that have not yet been observed.

Naive Bayes Models

- The **naive Bayes** model consists of a class/root node C and a number of attribute variables $X_1, \ldots, X_n$ as leaves.
- In the Boolean case, there are only $2n+1$ parameters in the model: $P(C = true) = \theta$ and for each $1 \leq i \leq n$,

$$P(X_i = true | C = true) = \theta_{(i,1)} \text{ and } P(X_i = true | C = false) = \theta_{(i,2)}.$$

- Attributes are assumed conditionally independent given the class.
- For observed attribute values $x_1, \ldots, x_n$ and a class C,

$$\mathbf{P}(C \mid x_1, \ldots, x_n) = \alpha \mathbf{P}(C) \prod_{i=1}^{n} \mathbf{P}(x_i \mid C).$$

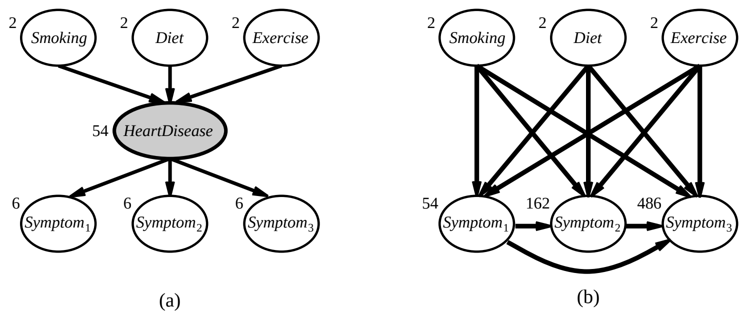
- No search is required to find the ML naive Bayes hypothesis.
- Boosting yields a very effective general-purpose learning algorithm.

3. LEARNING WITH HIDDEN VARIABLES

- Many real-world problems have **hidden** or **latent** variables which are not observable in the data available for learning.
- Latent variables can dramatically reduce the number of parameters required to specify a Bayes network.
- This, in turn, can significantly decrease the amount of data needed to learn the parameters.
- The **expectation-maximization** (EM) algorithm enables learning in the presence of hidden variables in a very general way.

Example

- In a Bayes network describing heart diseases, each variable has three possible values: *mone*, *moderate*, and *severe*.
- The removal of the only hidden variable *HeartDisease* increases the number of parameters from 78 to 708.



Learning Bayesian Networks with Hidden Variables

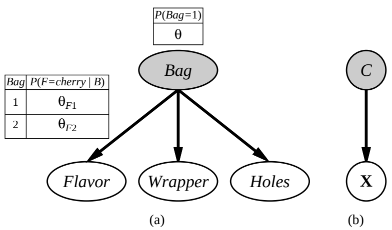
- We will consider **mixture distributions** where the data are generated from k independent component distributions.
- The probability of particular attribute values $\mathbf{x}$ is given by

$$P(\mathbf{x}) = \sum_{i=1}^{k} P(\mathbf{x}|C=i)P(C=i)$$

where variable C, with values $1, \ldots, k$, denotes the component.

Example

- In a generalized candy domain, candies are described by three features: *Flavor*, *Wrapper*, and *Hole*.
- The distribution of candies in each bag is described by a *naive Bayes model*: the features are independent given the bag.



- Given two bags, the parameters for the nodes of the network are θ, θ_{F1}/θ_{F2}, θ_{W1}/θ_{W2}, and θ_{H1}/θ_{H2}. See Figure (a) above.

Example (continued)

- A test data were generated using actual parameters $\theta = 0.5$, $\theta_{F1} = \theta_{W1} = \theta_{H1} = 0.8$, and $\theta_{F2} = \theta_{W2} = \theta_{H2} = 0.3$:

	$W = red$		$W = green$	
	$H=1$	$H=0$	$H=1$	$H=0$
$F = cherry$	273	93	104	90
$F = lime$	79	100	94	167

- For numerical simplicity, the parameter values are initialized as

$$\theta^{(0)} = 0.6,\ \theta^{(0)}_{F1} = \theta^{(0)}_{W1} = \theta^{(0)}_{H1} = 0.6, \text{ and } \theta^{(0)}_{F2} = \theta^{(0)}_{W2} = \theta^{(0)}_{H2} = 0.4:$$

for one iteration of the EM algorithm.

Example (continued)

- The parameter θ for the bag variable B is revised as follows:

$$\begin{aligned}\theta^{(1)} &= \hat{N}(B=1)/N = \tfrac{1}{N}\textstyle\sum_{j=1}^{N} P(B=1 \mid f_j, w_j, h_j) \\ &= \frac{1}{N}\sum_{j=1}^{N} \frac{P(f_j \mid B=1)P(w_j \mid B=1)P(h_j \mid B=1)P(B=1)}{\sum_i P(f_j \mid B=i)P(w_j \mid B=i)P(h_j \mid B=i)P(B=i)} \\ &\approx 0.6124.\end{aligned}$$

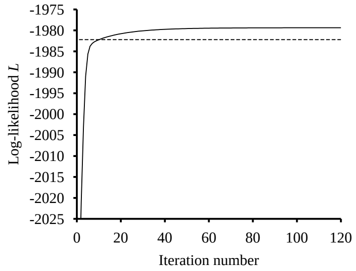
- Other parameters, such as θ_{F1}, are revised by *expected* counts

$$\sum_{j:F_j=cherry} P(B=1 \mid F_j = cherry, wrapper_j, holes_j)$$

which can be calculated using standard Bayes network algorithms.

Example (finished)

- After completing the process, the new parameter values are:
  $\theta^{(1)} = 0.6124, \quad \theta^{(1)}_{F1} = 0.6684, \quad \theta^{(1)}_{W1} = 0.6483, \quad \theta^{(1)}_{H1} = 0.6558,$
  $\theta^{(1)}_{F2} = 0.3887, \quad \theta^{(1)}_{W2} = 0.3817, \quad \theta^{(1)}_{H2} = 0.3827.$
- The log likelihood of the data increases very rapidly:



- The new model soon fits better than the original ($L \approx -1982$).

Learning Hidden Markov Models (HMMs)

- The goal is to learn the transition probabilities of HMMs given a (set of) observation sequence as data.
- As shown earlier, any HMM can be represented as a dynamic BN with a single discrete state variable.
- In HMMs, the transition probability $\theta_{ijt} = P(X_{t+1} = j \mid X_t = i)$ is fixed, i.e., $\theta_{ijt} = \theta_{ij}$, for all points of time t.
- To estimate the probability of a transition from state i to state j,
$$\theta_{ij} = \frac{\sum_t \hat{N}(X_{t+1} = j, X_t = i)}{\sum_t \hat{N}(X_t = i)}.$$
- Expected counts are computed by any HMM inference algorithm.

General Form of the EM Algorithm

- The treatment of hidden variables is based on computing their expected values for each example.
- Then parameters can be recomputed using the expected values as if they were observed values.
- In general, the EM algorithm can be characterized by
$$\theta^{(i+1)} = \arg\max_{\theta} \sum_{\mathbf{z}} P(\mathbf{Z} = \mathbf{z} \mid \mathbf{x}, \theta^{(i)}) L(\mathbf{x}, \mathbf{Z} = \mathbf{z} \mid \theta)$$
  where $\mathbf{x}$ and $\mathbf{Z}$, respectively, denote observed values and hidden variables *in all examples*, and θ denotes all parameters.
- The type of parameters varies from case to case.

SUMMARY

- **Bayesian learning** methods formulate learning as a form of probabilistic inference: observations are used to update a prior distribution over hypotheses.
- This approach implements Ockham's razor principle but quickly becomes intractable for complex hypothesis spaces.
- **Maximum a posteriori** (MAP) and **maximum likelihood** (ML) learning are more tractable approximations of Bayesian learning.
- **Naive Bayes** learning scales particularly well.
- When some variables are hidden, local maximum likelihood solutions can be found using the EM algorithm.