Bayesian Learning

Machine Learning, A.Y. 2022/23, Padova

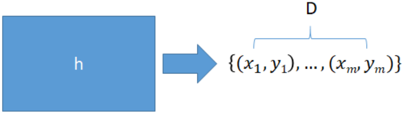
Fabio Aiolli

November 30th, 2022

Bayesian Methods

Bayesian methods provide computational techniques of learning (Naive Bayes, Bayesian Networks, etc.) but they are also useful for the interpretation/analysis of non-probabilistic algorithms:

- The observed training examples increase or decrease the probability that a hypothesis is correct
- Combination of prior knowledge on hypotheses with observed data
- Probabilistic predictions
- Classification by combining multiple hypotheses, weighted by their probability
- They define the ideal case of optimal prediction (even if computationally intractable)
- Practical difficulty: they require initial knowledge of many probabilities. If they are not initially available, they must be estimated by making appropriate assumptions about the distributions.
- Difficult in practice: computationally expensive, they require many examples for the correct estimation of the parameters.



$$P(h|D) = \frac{P(D|h)P(h)}{P(D)} = \frac{P(D|h)P(h)}{\sum_{h'} P(D|h')P(h')}$$

- $P(h)$: a priori probability of the hypothesis h
- $P(D)$: a priori probability of training data
- $P(h|D)$: probability of h given D
- $P(D|h)$: probability of D given h

Choice of the hypothesis

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

In general, we want to select the most probable hypothesis given the learning data, known as maximum a posteriori hypothesis h_{MAP}:

$$\begin{aligned} h_{MAP} &= \arg\max_{h \in H} P(h|D) \\ &= \arg\max_{h \in H} \frac{P(D|h)P(h)}{P(D)} \\ &= \arg\max_{h \in H} P(D|h)P(h) \end{aligned}$$

If we assume uniform probabilities on the hypotheses, i.e., $P(h_i) = P(h_j)$, then we can choose the so-called maximum likelihood hypothesis h_{ML}:

$$h_{ML} = \arg\max_{h \in H} P(D|h)$$

An example

Medical diagnosis: probability that a given patient has a particular form of cancer

$$
\begin{array}{rr}
P(cancer) = .008 & P(\neg cancer) = .992 \\
P(\oplus|cancer) = .98 & P(\ominus|cancer) = .02 \\
P(\oplus|\neg cancer) = .03 & P(\ominus|\neg cancer) = .97
\end{array}
$$

Suppose we observe a new patient for whom laboratory tests have given a positive result $\oplus$. What is the probability that the patient actually has cancer?

$$
\begin{aligned}
P(cancer|\oplus) \propto P(\oplus|cancer)P(cancer) &= .0078 \\
P(\neg cancer|\oplus) \propto P(\oplus|\neg cancer)P(\neg cancer) &= .0298
\end{aligned}
$$

"Brute force" learning of the hypothesis MAP

- For each hypothesis $h \in H$, compute the posterior probability

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

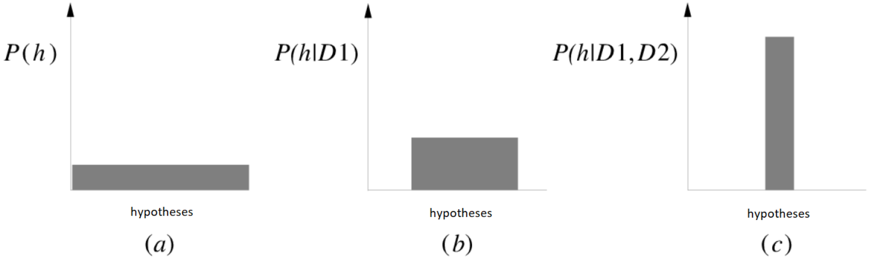
- Return the hypothesis h_{MAP} with the highest a posterior probability

$$h_{MAP} = \arg\max_{h \in H} P(h|D)$$

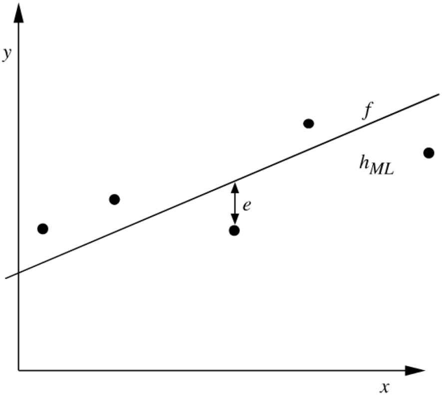
Evolution of the a posterior probability

Consistent Learner:

For simplicity we assume a uniform probability on the hypotheses, $P(h_i) = P(h_j)$, and deterministic noise-free training data ($P(D|h) = 1$ if h is consistent with D e $P(D|h) = 0$ otherwise).



Learning of a real-valued function

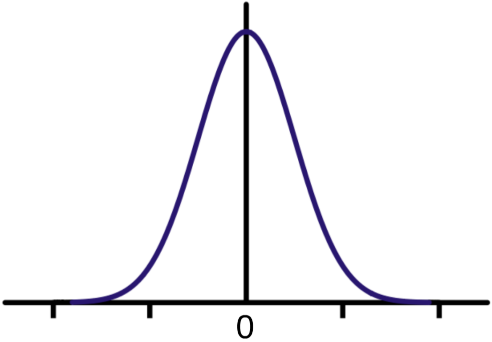


Consider any real-valued target function f, learning examples $\langle \mathbf{x}_i, d_i \rangle$, where d_i has some noise,

- $d_i = f(\mathbf{x}_i) + e_i$
- e_i is a random variable (noise) extracted independently for each $\mathbf{x}_i$ according to a Gaussian distribution with mean 0.

Then the hypothesis h_{ML} (maximum likelihood) is the one that minimizes:

$$h_{ML} = \arg\min_{h \in H} \sum_{i=0}^{m} (d_i - h(\mathbf{x}_i))^2$$



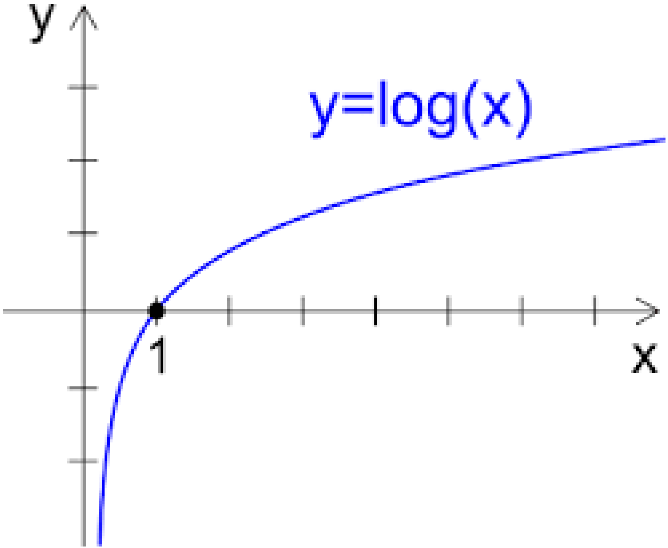
$$p(d_i|h) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(\overbrace{d_i - h(\mathbf{x}_i)}^{e_i})^2}$$

Learning of a real-valued function

$$
\begin{aligned}
h_{ML} &= \arg\max_{h \in H} p(D|h) \\
&= \arg\max_{h \in H} \prod_{i=1}^{m} p(d_i|h) \\
&= \arg\max_{h \in H} \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(d_i - h(\mathbf{x}_i))^2}
\end{aligned}
$$

which is best done by maximizing the natural logarithm.

Learning of a real-valued function



Learning of a real-valued function

$$
\begin{aligned}
h_{ML} &= \arg\max_{h \in H} \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(d_i - h(\mathbf{x}_i))^2} \\
&= \arg\max_{h \in H} \ln \left(\prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(d_i - h(\mathbf{x}_i))^2} \right) \\
&= \arg\max_{h \in H} \sum_{i=1}^{m} \ln \left(\frac{1}{\sqrt{2\pi\sigma^2}} \right) - \frac{1}{2\sigma^2}(d_i - h(\mathbf{x}_i))^2 \\
&= \arg\max_{h \in H} \sum_{i=1}^{m} -\frac{1}{2\sigma^2}(d_i - h(\mathbf{x}_i))^2 \\
&= \arg\min_{h \in H} \sum_{i=1}^{m} \frac{1}{2\sigma^2}(d_i - h(\mathbf{x}_i))^2 = \arg\min_{h \in H} \sum_{i=1}^{m} (d_i - h(\mathbf{x}_i))^2
\end{aligned}
$$

Learning a hypothesis that predicts a probability

Consider the scenario of a probabilistic function $f : X \rightarrow \{0, 1\}$.

- X might represent medical patients in terms of their symptoms and $f(x)$ might be 1 if the patient survives the desease and 0 if not;
- X might represent loan applicants in terms of their credit history and $f(x)$ might be 1 if the applicant succesfully repays the next loan and 0 if not

We want to learn a neural network (or any other real-valued approximator) $f' : X \rightarrow [0, 1]$ which predicts the probability that $f(x) = 1$ given x.

What criterion should we optimize to find an ML hypothesis for f'? To do this, we first need to define what $P(D|h)$ is,

where $D = \{\langle \mathbf{x}_1, d_1 \rangle, \ldots, \langle \mathbf{x}_n, d_n \rangle\}$ and $d_i \in \{0, 1\}$.

Learning a hypothesis that predicts a probability

$$P(D|h) = \prod_{i=1}^{m} P(\mathbf{x}_i, d_i|h) = \prod_{i=1}^{m} P(d_i|h, \mathbf{x}_i)P(\mathbf{x}_i)$$

$$P(d_i|h, \mathbf{x}_i) = \begin{cases} h(\mathbf{x}_i) & \text{if } d_i = 1 \\ 1 - h(\mathbf{x}_i) & \text{if } d_i = 0 \end{cases} = h(\mathbf{x}_i)^{d_i}(1 - h(\mathbf{x}_i))^{1-d_i}$$

$$P(D|h) = \prod_{i=1}^{m} h(\mathbf{x}_i)^{d_i}(1 - h(\mathbf{x}_i))^{1-d_i} P(\mathbf{x}_i)$$

$$\begin{aligned} h_{ML} &= \arg\max_{h \in H} \prod_{i=1}^{m} h(\mathbf{x}_i)^{d_i}(1 - h(\mathbf{x}_i))^{1-d_i} \\ &= \arg\max_{h \in H} \underbrace{\sum_{i=1}^{m} d_i \ln(h(\mathbf{x}_i)) + (1 - d_i)\ln(1 - h(\mathbf{x}_i))}_{-\text{cross entropy}} \end{aligned}$$

Most likely classification for new instances

So far we have been looking for the most likely hypothesis given the data D (that is h_{MAP})

Given a new instance $\mathbf{x}$, which is the most likely classification?

$H_{MAP}(\mathbf{x})$ classification is not necessarily the most likely classification.

Let us consider for example the following situation:

- three possible hypotheses:

$$P(h_1|D) = 0.4, \quad P(h_2|D) = 0.3, \quad P(h_3|D) = 0.3$$

- given a new instance $\mathbf{x}$,

$$h_1(\mathbf{x}) = \oplus, \quad h_2(\mathbf{x}) = \ominus, \quad h_3(\mathbf{x}) = \ominus$$

- which is the most likely classification for $\mathbf{x}$?

Bayes optimal classification

Given a class $v_j \in V$, we get:

$$P(v_j|D) = \sum_{h_i \in H} P(v_j|h_i)P(h_i|D)$$

from which it follows that the optimal (Bayes) classification of a certain instance is the class $v_j \in V$ which maximizes this probability, that is:

$$\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D)$$

Example of optimal Bayes classification

$$v_{Bayes} = \arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D)$$

Example:

$$P(h_1|D) = 0.4, \quad P(\ominus|h_1) = 0, \quad P(\oplus|h_1) = 1$$
$$P(h_2|D) = 0.3, \quad P(\ominus|h_2) = 1, \quad P(\oplus|h_2) = 0$$
$$P(h_3|D) = 0.3, \quad P(\ominus|h_3) = 1, \quad P(\oplus|h_3) = 0$$

hence:

$$\sum_{h_i \in H} P(\oplus|h_i)P(h_i|D) = 0.4 \qquad \sum_{h_i \in H} P(\ominus|h_i)P(h_i|D) = 0.6$$

and then:

$$v_{Bayes} = \arg\max_{v_j \in \{\ominus,\oplus\}} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D) = \ominus$$

Gibbs classifier

Bayes' optimal classifier can be very expensive to compute if there are many hypotheses!

Gibbs algorithm:

- Choose a hypothesis at random, with probability $P(h|D)$
- Use it to classify the new instance

Rather surprising fact: assuming that the target concepts are randomly extracted from H according to an a priori probability on H, then:

$$E[\epsilon_{Gibbs}] \leq 2E[\epsilon_{Bayes}]$$