SL09. Bayesian Learning

Introduction:

- We're trying to learn the best (most probable) hypothesis H given input data and domain knowledge.

Best == Most probable

- It's the probability of some hypothesis *h* given input data *D*:

$$P_r(h \mid D)$$

- We're trying to find hypothesis *h* with the highest probability P_r:

$$argmax_{h \in H}(P_r(h \mid D))$$

Bayes Rule:

- Bayes Rule for probability states that:

$$P_r(h \mid D) = \frac{P_r(D \mid h)P_r(h)}{P_r(D)}$$

  - $P_r(h \mid D)$ → The probability of a specific hypothesis given input data (Posterior probability).
  - $P_r(D \mid h)$ → The probability of data given the hypothesis. It's the (likelihood) of seeing some particular labels associated with input points, given a world where some hypothesis *h* is true.
  - $P_r(h)$ → The prior probability of a particular hypothesis. This value encapsulates our prior belief that one hypothesis is likely or unlikely compared to other hypotheses. This is basically the domain knowledge.
  - $P_r(D)$ → The likelihood of the data under all hypotheses (A normalizing term).
  - Bayes Rule is derived from the chain rule:

$$\begin{aligned} & P_r(a,b) = P_r(a,b)P_r(b) \\ & P_r(a,b) = P_r(b,a)P_r(a) \\ then \quad \rightarrow \quad & P_r(a,b)P_r(b) = P_r(b,a)P_r(a) \\ & P_r(a,b) = \frac{P_r(b,a)P_r(a)}{P_r(b)} \end{aligned}$$

Bayesian Learning:

- Bayesian Learning algorithm:

  For each $h \in H$:

  Calculate $P_r(h \mid D) = \frac{P_r(D \mid h)P_r(h)}{P_r(D)}$

  Output:

  $h = argmax_{h \in H}(P_r(h \mid D))$

- Using this approximate probability, we can calculate the Maximum a Posteriori (MAP), which is the maximum probability hypothesis given the data across all hypotheses:

$$h_{map} = argmax_{h \in H}(P_r(h \mid D))$$

$$h_{map} = argmax_{h \in H}(\frac{P_r(D \mid h)P_r(h)}{P_r(D)})$$

- Since we're interested in finding the hypothesis with the highest probability, not the exact probability value for each hypothesis, our prior on the data isn't exactly relevant. That is, we don't care about the $P_r(D)$ term in the denominator as it affects all computations equally:

$$h_{map} = argmax_{h \in H}(P_r(D \mid h)P_r(h))$$

- We can also assume that our prior belief is uniform over all the hypotheses $h \in H$ (we equally believe in every $h \in H$), then we can drop $P_r(h)$ from the equation, ending up with the Maximum Likelihood:

$$h_{ml} = argmax_{h \in H}(P_r(D \mid h))$$

- The problem with Bayes Learning is that it's not practical to perform direct computations for large hypotheses spaces, because you have to look into every single hypothesis.

Bayesian Learning in Action:

- Assume:
  - Given noise-free training data $\{\langle x_i, d_i \rangle\}$ as examples of c.
  - $c \in H$
  - Uniform prior.
- We need to calculate $P_r(h \mid D)$:

$$P_r(h \mid D) = \frac{P_r(D \mid h)P_r(h)}{P_r(D)}$$

$$P_r(h) = \frac{1}{|H|}, \qquad because\ we\ have\ a\ uniform\ prior$$

$$P_r(D \mid h) = \begin{cases} 1 & if\ d_i = h(x_i)\ \forall x_i, d_i \in D \\ 0 & otherwise \end{cases}$$

  - This equation basically means that $P_r(D \mid h) = 1$ if $h \in VS(D)$

$$P_r(D) = \sum_{h_i \in H} P_r(D \mid h_i)\, P_r(h_i) = \sum_{h_i \in VS_{H,D}} 1 \cdot \frac{1}{|H|} = \frac{|VS|}{|H|}$$

$$P_r(h \mid D) = \frac{1 \cdot \frac{1}{|H|}}{\frac{|VS|}{|H|}} = \frac{1}{|VS|}$$

  - This means that given data D, the probability of h to be a correct hypothesis is a uniform over all the hypotheses in the version space.

Bayesian Learning with Noise:

- Assume:
  - Given $\{\langle x_i, d_i \rangle\}$
  - $d_i = f(x_i) + \varepsilon_i$
  - $\varepsilon_i \sim N(0, \sigma^2)$ → IID (Independent and Identically Distributed)
- We need to calculate $P_r(h \mid D)$:

$$h_{ml} = argmax_{h \in H}(P_r(D \mid h))$$

  - To find $P_r(D \mid h)$ for IID, we find the product of the probability of each data point given the hypothesis:

$$h_{ml} = argmax_{h \in H} \prod_i P_r(d_i \mid h)$$

  - Given a Gaussian noise:

$$h_{ml} = argmax_{h \in H} \prod_i \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(\frac{-1}{2} \cdot \frac{(d_i - h(x_i))^2}{\sigma^2}\right)}$$

  - Since we're looking for the maximum:
    1. We can remove $\frac{1}{\sqrt{2\pi\sigma^2}}$ since we're looking for the maximum.
    2. We can take the natural log ln to remove the exponential. Since ln of a product is equal to the sum of the terms, we end up with the following function.

$$h_{ml} = argmax_{h \in H} \sum_i \frac{-1}{2} \cdot \frac{(d_i - h(x_i))^2}{\sigma^2}$$

  - Again, since we're calculating the maximum, we can remove the $\frac{1}{2}$ and the σ^2:

$$h_{ml} = argmax_{h \in H} - \sum_i (d_i - h(x_i))^2$$

  - Maximizing a negative value is the same as minimizing the positive sum of this value:

$$h_{ml} = argmin_{h \in H} \sum_i (d_i - h(x_i))^2$$

- This means: If you're looking for the maximum likelihood hypothesis, you should minimize the sum of squared error.
- This model will not work if the data is corrupted with any sort of noise other than Gaussian noise.

Minimum Description Length:

$$h_{map} = argmax_{h \in H}(P_r(D \mid h)P_r(h))$$
$$h_{map} = argmax_{h \in H}[\log P_r(D \mid h) + \log P_r(h)]$$
$$h_{map} = argmin_{h \in H}[-log\, P_r(D \mid h) - log\, P_r(h)]$$

- Information theory: The optimal code for some event w with probability P_r has a length of $-\log P_r$.
- This means that in order to maximize the Maximum a Posteriori hypothesis, we need to minimize two terms that can be described as length:
  - $log\, P_r(h)$ → This is the length of the hypothesis, which is the number of bits needed to represent this hypothesis.
  - $log\, P_r(D \mid h)$ → This is the length of the data given a particular hypothesis. If the hypothesis perfectly describes the data, so we don't need any points. But if the hypothesis labels some points wrong, so we need the correct labels for these points to be able to come up with a better hypothesis. So basically this term captures the error.
- This is always a trade of, a more complex hypothesis will drive down error, while a simple hypothesis will have some error.
- We need to find the best hypothesis, which is the simplest hypothesis that minimizes error. This hypothesis is called the Minimum Description.

Bayesian Classification:

- The question in classification is "What is the best label?" not the best hypothesis.
- To find the best label, we need to do a weighted vote for every single hypothesis in the hypotheses set, where the weight is the probability $P_r(h \mid D)$.
- Now we end up trying to maximize v_{map}:

$$V_{map} = argmax_{v_j \in V} \sum_{h_i \in H} P_r(v_j \mid h_i)\, P_r(h_i \mid D)$$

- The Bayes optimal classifier is computationally very costly. This is because the posterior probability $P_r(h \mid D)$ must be computed for each hypothesis $h \in H$ and combined with the prediction $P_r(v \mid h)$ before V_{map} can be computed.