So far we just covered what is the most probable hypothesis given the training data. However we want to just find the **most probable classifications given the training data**.

Most probable classification $\neq$ classification generated by h_{MAP}

The most probable classification is obtained by combining the predictions of all hypothesis, weighted by their posterior probabilities.

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

Where $P(v_j|D)$ is the probability that the correct classification is v_j.

Bayes Optimal Classifier

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

A theoretical classifier that effectively *just* combines the results of "all models", weighted by their posterior probability; the probability that the model describes the true underlying explanation of the dataset.

Naive Bayes Classifier

Applies to learning tasks where each instance x is described by a conjunction of attribute values $< a_1, a_2, ..., a_n >$ and where the target function $f(x)$ can take on any value from some finite set V (the possible classification labels).

$$\begin{aligned} v_{MAP} &= \arg\max_{v_j \in V} P(v_j|a_1, a_2, ..., a_n) \\ &= \arg\max_{v_j \in V} \frac{P(a_1, a_2, ..., a_n|v_j)P(v_j)}{P(a_1, a_2, ..., a_n)} \\ &= \arg\max_{v_j \in V} P(a_1, a_2, ..., a_n|v_j)P(v_j) \end{aligned}$$

Now $P(v_j)$ can be estimated similar to how we handle mode, the crux here is the needed simplification for $P(a_1, a_2, ..., a_n|v_j)$... The issue is finding the frequency of all instances with exactly $a_1, a_2, a_3, ..., a_n$ attributes such that are classified as v_i... (similar approach to finding $P(v_j)$) even in the case of binary

attributes, all possible instances would be 2^n times the number of classification labels. So as the attribute list climbs, the dataset size needed grows significantly.

Hence we simplify and introduce an assumption; *Attribute values are conditionally independent given the target value.*

Quickly reviewing types of independence with the following **recursive** definitions;

- Absolute independence of X,Y; $P(X,Y) = P(X|Y)P(Y) = P(X)P(Y)$
- Conditional independence of X,Y given Z; $P(X,Y|Z) = P(X|Y,Z)P(Y|Z) = P(X|Z)P(Y|Z)$

Thus we simplify too;

$$P(a_1, a_2, ..., a_n|v_j) = \prod_i P(a_i|v_j)$$

Which is important since we drastically reduce the number of terms to evaluate, and hence gives the dataset the chance to provide a good estimate. No explicit search through hypothesis space H, rather just counting frequencies!

So if the assumption is correct ofcourse then Naive Bayes classifications are MAP (maximal a posterior probability) classifications.

So the training phase of these classifiers is purely calculating each of these atomic probabilities right, ready for rapid multiplication later.

Smoothing

If any of these atomic probabilities evaluates to zero, we have a very serious destruction of information. Since $x \cdot 0 = 0$ ofcourse... thus there are techniques, such as **Laplace smoothing** that effectively modifies each atomic probability in the following way;

$$P(A_k = a_i|v_j) = \frac{n_{ij} + 1}{nj + m}$$

- n_{ij} is the number of training examples with class labels v_j and attribute label a_j
- n_j is the number of training examples with class label v_j
- m is the number of unique attributes values of A_k

There's also the issue of missing attribute values for new instances... all we do there is just omit those in the calculations. Makes complete sense as we are not gaining information from the fact that these variables are missing.

Extends beyond just a *few missing variables...* if all are missing, this approach reduces to the mode classifier.