Bayesian Concept Learning

Intelligent Systems and Control

2019

Sepehr Maleki

University of Lincoln
School of Engineering

Concept Learning

Think of how a child learns the word “dog”.

- The child’s parents point out positive examples of this concept (e.g., “look at the cute dog!”, or “mind the doggy”, etc.).
- It is very unlikely that they provide negative examples, by saying “look at that non-dog”.
- Negative examples may be obtained during an active learning process (e.g., he child says “look at the dog” and the parent says “that’s a cat, dear, not a dog”).

We can think of learning the meaning of a word as equivalent to *concept learning*.

(S. Maleki 2019)

A Binary Classification Problem

Concept learning can be seen as a binary classification problem:

$$f(x) = \begin{cases} 1 & x \in \mathcal{C} \\ 0 & \text{Otherwise} \end{cases}.$$

The goal is to learn the indicator function f, which just defines which elements are in the set $\mathcal{C}$.

NB. Standard binary classification techniques require positive and negative examples. By contrast, we will devise a way to learn from positive examples alone.

The Number Game

Suppose I have a set of numbers. I show you a positive example of the set $\mathcal{C}$ (e.g., 16). Then I ask if you think other numbers (e.g., $17, 6, 32, 99$) also belong in this set?

It's hard to tell with only one example, so your predictions will be quite vague. Presumably numbers that are similar in some sense to 16 are more likely. **But similar in what way?**

- 17 is similar, because it is "close by".
- 6 is similar because it has a digit in common.
- 32 is similar because it is also even and a power of 2.
- 99 does not seem similar.

Thus some numbers are more likely than others.

(S. Maleki 2019)

The Number Game

The number game can be represented as a probability distribution, $p(\tilde{x}|\mathcal{D})$, which is the probability that $\tilde{x} \in \mathcal{C}$ given the data $\mathcal{D} \subset \mathcal{C}$. This is called the *posterior predictive distribution*.

As we continue the game, I tell you that $8, 2$ and 64 are also positive examples (i.e., $\mathcal{D} = \{16, 8, 2, 64\}$). Now you may guess that the hidden concept is "powers of two". This is an example of *induction*.

If instead I tell you the data is $\mathcal{D} = \{16, 23, 19, 20\}$, you will have a different induction.

(S. Maleki 2019)

Machine Concept Learning

- Suppose we have a hypothesis space of concepts, $\mathcal{H}$ (e.g., powers of two, odd numbers, etc.).
- The subset of $\mathcal{H}$ which is consistent with the observed data $\mathcal{D}$ is called the *version space*.
- As we see more examples, the version space shrinks and we become increasingly certain about the concept.

How can we explain the concept learning and emulate it in a machine?

(S. Maleki 2019)

Machine Concept Learning

Version space is not the whole story. After seeing $\mathcal{D} = \{16\}$, there are many consistent rules; how do you combine them to predict if $\tilde{x} \in \mathcal{C}$?

- after seeing $\mathcal{D} = \{16, 8, 2, 64\}$, why did you choose the rule "powers of two" and not, say, "all even numbers"?

There is a Bayesian explanation for this!

Let's define two hypotheses:

$$\begin{cases} h_{two} & \text{power of two} \\ h_{even} & \text{Even numbers} \end{cases}.$$

The key intuition is that we want to avoid suspicious coincidences.

(S. Maleki 2019)

Machine Concept Learning

For simplicity, assume all numbers are integers between 1 and 100. Then the *extensions* of the two concepts are given by:

$$\begin{cases} h_{two} : \{2, 4, 8, 16, 32, 64\} \\ h_{even} : \{2, 4, 6, 8, 10, \ldots, 100\} \end{cases} .$$

Assume that examples (positive) are sampled uniformly at random from the extension of a concept. The probability of independently sampling N items (with replacement) from h is given by:

$$p(\mathcal{D}|h) = \left[\frac{1}{size(h)}\right]^N = \left[\frac{1}{|h|}\right]^N .$$

This means that the model favours the simplest (smallest) hypothesis consistent with the data (alson known as Occam's razor).

(S. Maleki 2019)

Likelihood

In the number game, consider $\mathcal{D} = \{16\}$. Then:

$$\begin{cases} p(\mathcal{D}|h_{two}) = \frac{1}{6} \\ p(\mathcal{D}|h_{even}) = \frac{1}{50} \end{cases}$$

So the likelihood that $h = h_{two}$ is higher than if $h = h_{even}$. After seeing new examples $\mathcal{D} = \{16, 8, 2, 64\}$:

$$\begin{cases} p(\mathcal{D}|h_{two}) = (\frac{1}{6})^4 \\ p(\mathcal{D}|h_{even}) = (\frac{1}{50})^4 \end{cases}$$

This is a *likelihood ratio* of almost $5000 : 1$ and quantifies our earlier intuition that $\mathcal{D} = \{16, 8, 2, 64\}$ would be a very suspicious coincidence if generated by h_{even}.

(S. Maleki 2019)

Prior

Consider a different hypothesis for the data $\mathcal{D} = \{16, 8, 2, 64\}$:

$$h' : \text{ “powers of two, except 32”}$$

h' is even more likely than h_{two} since it does not need to explain the coincidence that 32 is missing from the set of examples.

However, the hypothesis $h' =$ “powers of two except 32” seems “conceptually unnatural”.

We can capture such intuition by assigning low *prior* probability to unnatural concepts.

Prior

Prior is the mechanism by which background knowledge can be brought to bear on a problem.

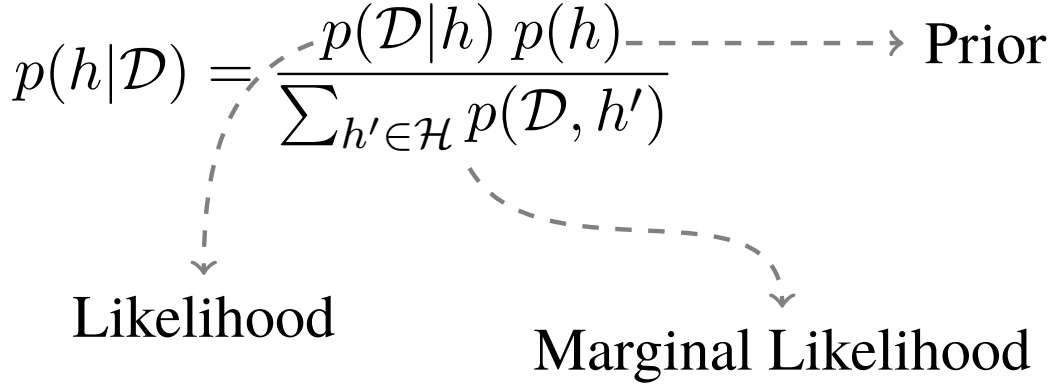
So, what prior should we use?

- Let us use a simple prior which puts uniform probability on 30 simple arithmetical concepts, such as “even numbers”, “odd numbers”, “prime numbers”, “numbers ending in 9”, etc.
- To make things more interesting, we make the concepts even and odd more likely a-priori.
- We also include two “unnatural” concepts, namely “powers of 2, plus 37” and “powers of 2, except 32”, but give them low prior weight.

(S. Maleki 2019)

Posterior

The posterior is simply the likelihood times the prior, normalised:



Often just:

$$p(h|\mathcal{D}) \propto p(\mathcal{D}|h)\, p(h)\ .$$

Marginal likelihood (normalisation factor) is used to make sure that the posterior probability is normalised and sums up to 1.

Numbers Game

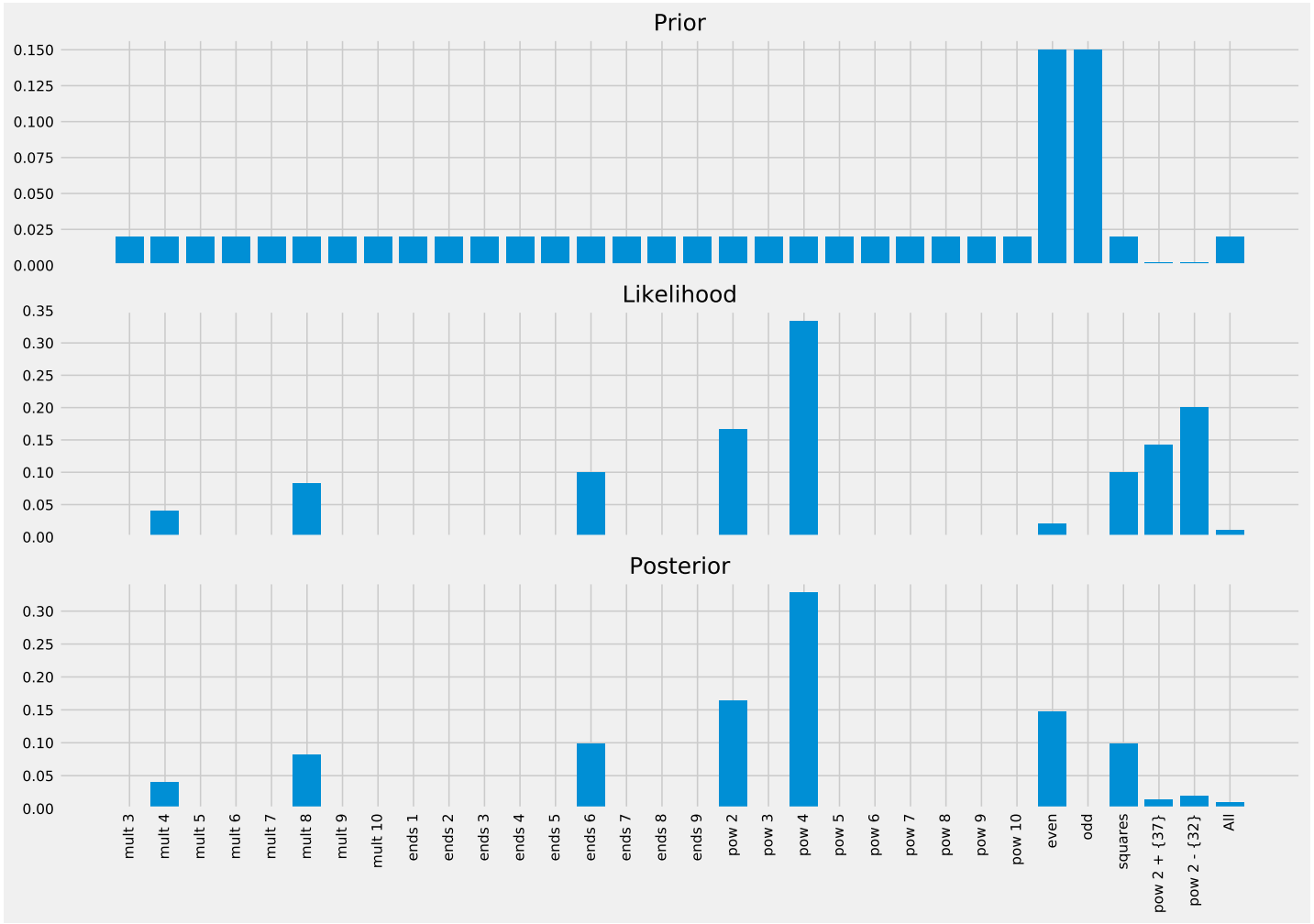


Figure: Prior, likelihood and posterior for $\mathcal{D} = \{16\}$.

(S. Maleki 2019)

Numbers Game

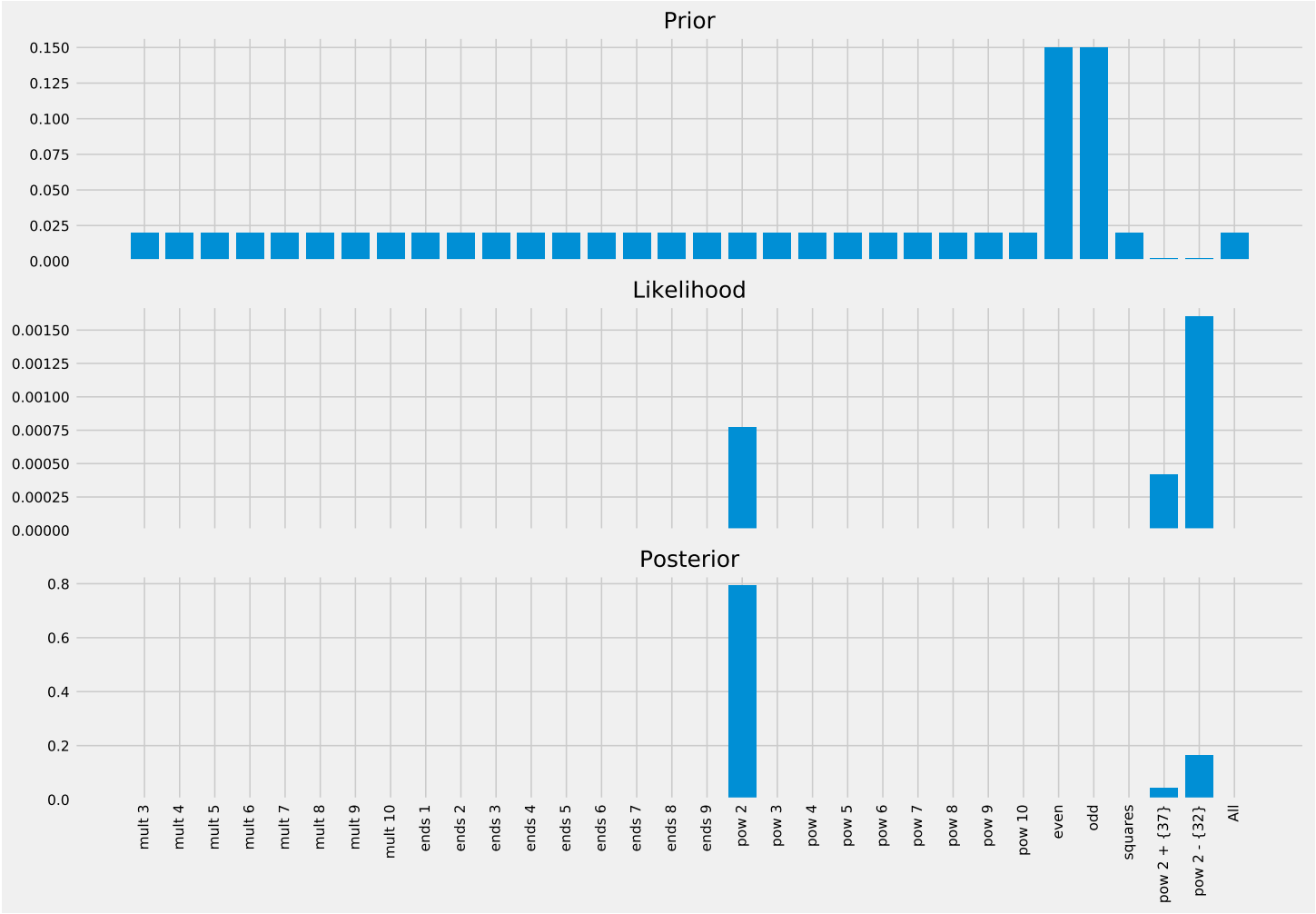


Figure: Prior, likelihood and posterior for $\mathcal{D} = \{16, 8, 2, 64\}$.

(S. Maleki 2019)

MAP Estimate

In general, when we have enough data, the posterior $p(h|\mathcal{D})$ becomes peaked on a single concept, namely the MAP estimate, i.e.,

$$p(h|\mathcal{D}) \rightarrow \delta_{\hat{h}^{MAP}}(h) \ ,$$

where $\hat{h}^{MAP} = \text{argmax}_h \, p(h|\mathcal{D})$ is the posterior mode, and δ is the Dirac measure defined by:

$$\delta_x(A) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases} \ .$$

MAP estimate can also be written as:

$$\hat{h}^{MAP} = \underset{h}{\text{argmax}} \, p(\mathcal{D}|h)p(h) = \underset{h}{\text{argmax}} [\log p(\mathcal{D}|h) + \log p(h)]$$

(S. Maleki 2019)

Maximum Likelihood Estimate (MLE)

Since the likelihood term depends exponentially on N, and the prior stays constant, as we get more and more data, the MAP estimate converges towards the maximum likelihood estimate or MLE:

$$\hat{h}^{mle} \triangleq \operatorname*{argmax}_h p(\mathcal{D}|h) = \operatorname*{argmax}_h \log p(\mathcal{D}|h) \ .$$

In other words, if we have enough data, we see that the data overwhelms the prior. In this case, the MAP estimate converges towards the MLE.

If the true hypothesis is in the hypothesis space, then the MAP/ ML estimate will converge upon this hypothesis.

Posterior Predictive Distribution

The posterior is our internal belief state about the world.

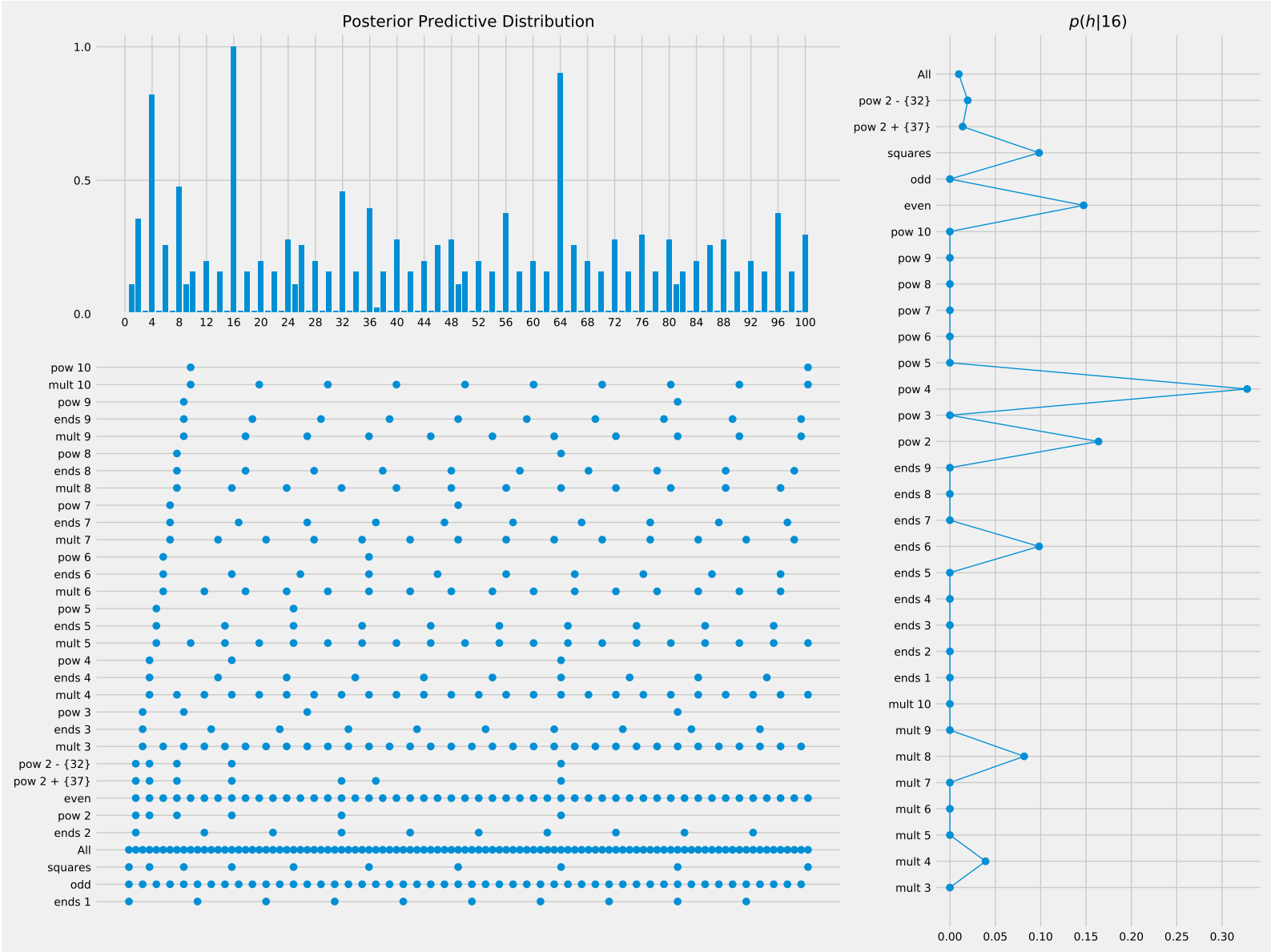
The way to test if our beliefs are justified is to use them to predict objectively observable quantities.

Specifically, the posterior predictive distribution in this context is given by:

$$p(\tilde{x} \in \mathcal{C}|\mathcal{D}) = \sum_{h \in \mathcal{H}} p(\tilde{x} \in \mathcal{C}|h)p(h|\mathcal{D}) \ .$$

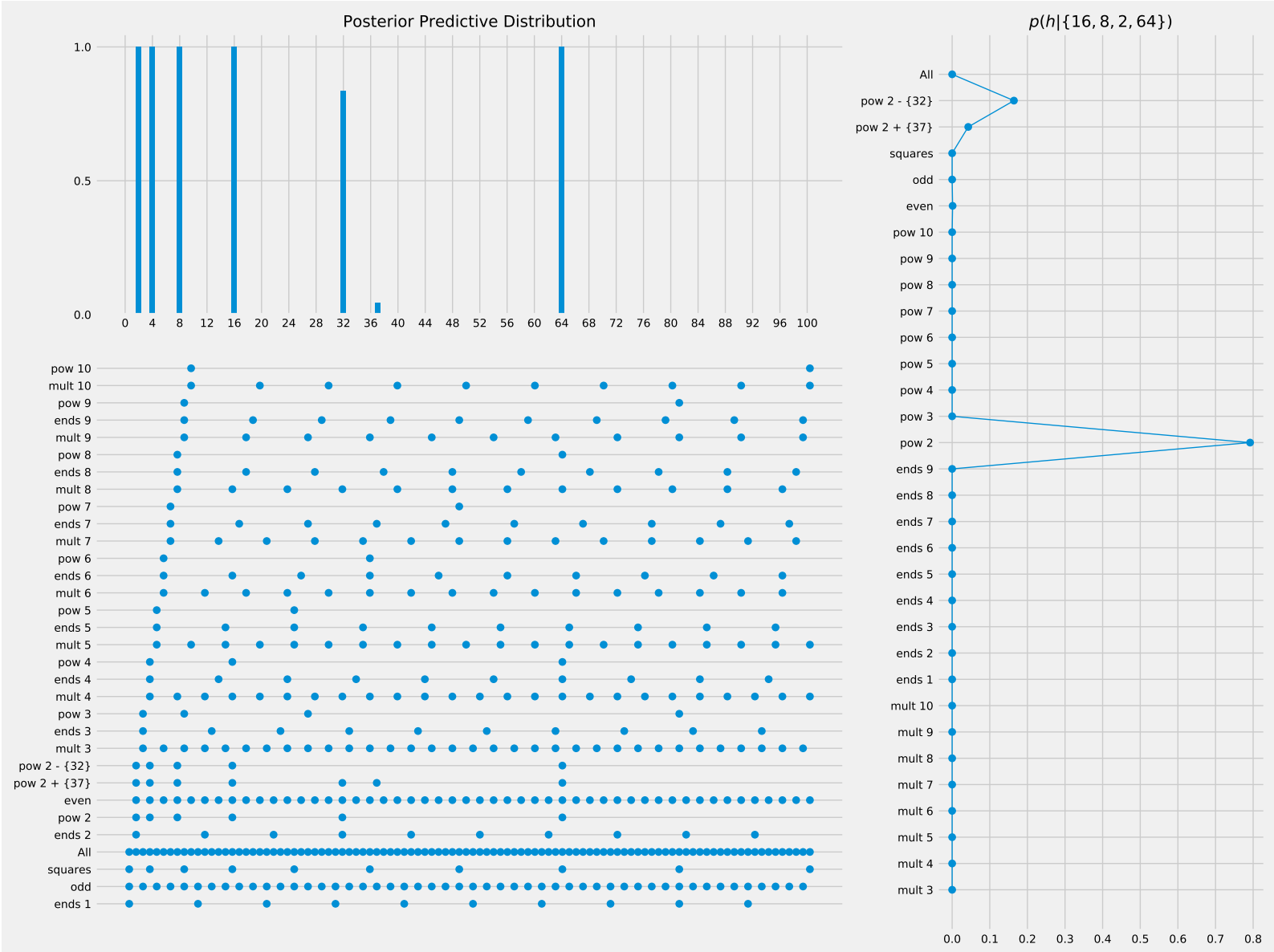
This is just a weighted average of the predictions of each individual hypothesis and is called *Bayes model averaging*.

Posterior Predictive Distribution - $\mathcal{D} = \{16\}$



(S. Maleki 2019)

Posterior Predictive Distribution - $\mathcal{D} = \{16, 8, 2, 64\}$



(S. Maleki 2019)

Naive Bayes Classifiers

Consider a data-set $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{c_1, c_2, ..., c_k\}$. A new feature vector $\mathbf{x}_{new}$ belongs to the class c_i if and only if:

$$p(c_i|\mathbf{x}_{new}) > p(c_j|\mathbf{x}_{new}) \ ,$$

for $1 \leq j \leq m, \ i \in \{1, 2, ..., k\}, i \neq j$.

Thus we find the class that maximises $p(c_i|\mathbf{x}_{new})$. The class c_i for which $p(c_i|\mathbf{x}_{new})$ is maximised is called the maximum posteriori hypothesis.

Naive Bayes Classifiers

By Bayes' theorem:

$$p(c_i|\mathbf{x}_{new}) = \frac{p(\mathbf{x}_{new}|c_i)p(c_i)}{p(\mathbf{x}_{new})} \ .$$

As $p(\mathbf{x}_{new})$ is the same for all classes, only $p(\mathbf{x}_{new}|c_i)p(c_i)$ needs to be maximised.

If the prior probabilities for classes, $p(c_i)$, are unknown, then it is commonly assumed they are equally likely and therefore we would only maximise $p(\mathbf{x}_{new}|c_i)$.

Naive Bayes Classifiers

Given data sets with many features, it would be computationally expensive to compute $p(\mathbf{x}_{new}|c_i)$.

To reduce the computation, we make the "naive" assumption of class conditional independence (hence the model's name).

Mathematically, this means:

$$p(\mathbf{x}_{new}|c_i) \approx \prod_{j=1}^{d} p(x_j|c_i) \ .$$

Where the probabilities $p(x_1|c_i), p(x_2|c_i), ..., p(x_d|c_i)$ can easily be estimated from the training set.

(S. Maleki 2019)

Example

RID	age	income	student	credit	C_i: buy
1	youth	high	no	fair	C_2: no
2	youth	high	no	excellent	C_2: no
3	middle-aged	high	no	fair	C_1: yes
4	senior	medium	no	fair	C_1: yes
5	senior	low	yes	fair	C_1: yes
6	senior	low	yes	excellent	C_2: no
7	middle-aged	low	yes	excellent	C_1: yes
8	youth	medium	no	fair	C_2: no
9	youth	low	yes	fair	C_1: yes
10	senior	medium	yes	fair	C_1: yes
11	youth	medium	yes	excellent	C_1: yes
12	middle-aged	medium	no	excellent	C_1: yes
13	middle-aged	high	yes	fair	C_1: yes
14	senior	medium	no	excellent	C_2: no

$$\mathbf{x}^\star = (age = youth, income = medium, student = yes, credit = fair)$$

(S. Maleki 2019)

Example

The data samples are described by features: *age, income, student, and credit*.

The class label, *buy*, tells whether the person buys a computer, has two distinct values, yes (class c_1) and no (class c_2).

We need to maximise $p(\mathbf{x}^\star|c_i)p(c_i)$ for $i = 1, 2$. $p(c_i)$ can be estimated from the training samples:

$$p(buy = yes) = \frac{9}{14}, \quad p(buy = no) = \frac{5}{14} \ .$$

Example

To compute $p(\mathbf{x}^\star|c_i)$, for $i = 1, 2$, we compute the following conditional probabilities:

$$p(age = youth|buy = yes) = \frac{2}{9}, \quad p(age = youth|buy = no) = \frac{3}{5}$$

$$p(income = medium|buy = yes) = \frac{4}{9}, \; p(income = medium|buy = no) = \frac{2}{5}$$

$$p(student = yes|buy = yes) = \frac{6}{9}, \quad p(student = yes|buy = no) = \frac{1}{5}$$

$$p(credit = fair|buy = yes) = \frac{6}{9}, \quad p(credit = fair|buy = no) = \frac{2}{5}$$

Example

Therefore:

$$\begin{aligned} p(\mathbf{x}^\star|buy = yes) = & p(age = youth|buy = yes) \\ & p(income = medium|buy = yes) \\ & p(student = yes|buy = yes) \\ & p(credit = fair|buy = yes) \\ & = \frac{2}{9}\ \frac{4}{9}\ \frac{6}{9}\ \frac{6}{9} = 0.044\ . \end{aligned}$$

Similarly,

$$p(\mathbf{x}^\star|buy = no) = \frac{3}{5}\ \frac{2}{5}\ \frac{1}{5}\ \frac{2}{5} = 0.019\ .$$

Example

To find the class that maximises $p(\mathbf{x}^\star|c_i)p(c_i)$, we compute:

$$p(\mathbf{x}^\star|buy = yes)p(buy = yes) = 0.028$$
$$p(\mathbf{x}^\star|buy = no)p(buy = no) = 0.007\ .$$

Thus the naive Bayesian classifier predicts $buy = yes$ for sample $p(\mathbf{x}^\star|c_i)$.