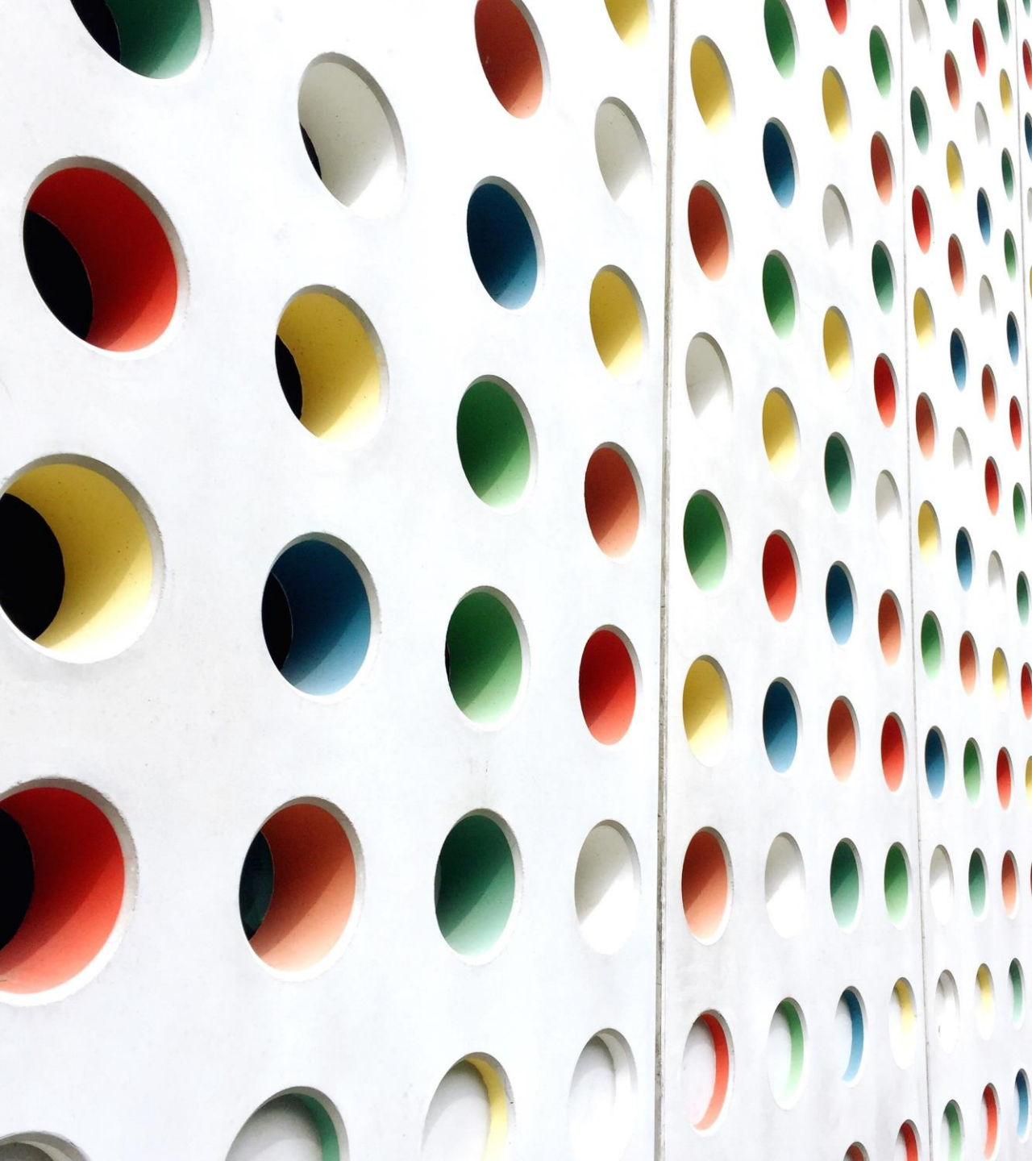


3 Generative models for discrete data

Discrete data

Bayesian concept learning

The beta-binomial model

Naive Bayes classifiers

Introduction

Classify a feature vector $\mathbf{x}$ by applying Bayes rule to a generative classifier of the form

$$p(y = c|\mathbf{x}, \boldsymbol{\theta}) \propto p(\mathbf{x}|y = c, \boldsymbol{\theta})p(y = c|\boldsymbol{\theta})$$

The key to using such models is specifying a suitable form for the class-conditional density $p(\mathbf{x}|y = c, \boldsymbol{\theta})$ which defines what kind of data we expect to see in each class

In this chapter, we focus on the case where the observed data are discrete symbols.

We also discuss how to infer the unknown parameters $\boldsymbol{\theta}$ of such models.

Generative vs Discriminative Models

Generative model - models the actual distribution of each class

1. Assume some functional form for $\boldsymbol{p(y)}$, $\boldsymbol{p(x|y)}$
2. Estimate parameters of $\boldsymbol{p(y)}, \boldsymbol{p(x|y)}$ directly from training data
3. Use Bayes rule to calculate $p(y|x)$

Discriminative model - models the decision boundary between the classes

1. Assume some functional form for $p(y|x)$
2. Estimate parameters of $p(y|x)$ directly from training data

Bayesian concept learning

Consider how a child learns to understand the meaning of a word, such as "**dog**". Presumably the child's parents point out positive examples of this concept, saying such things as, "look at the cute dog!", or "mind the doggy", etc. However, it is very unlikely that they provide negative examples, by saying "look at that non-dog".

We can think of learning the meaning of a word as equivalent to **concept learning**, which in turn is equivalent to binary classification. To see this, define $f(\mathbf{x}) = 1$ if $\mathbf{x}$ is an example of the concept C, and $f(\mathbf{x}) = 0$ otherwise. Then the goal is to learn the indicator function f, which just defines which elements are in the set C.

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

Bayesian concept learning - the number game

Josh Tenenbaum's PhD thesis

The game proceeds as follows. I choose some **simple arithmetical concept** C, such as "prime number" or "a number between 1 and 10". I then give you a series of **randomly chosen positive examples** $D = \{x_1, \ldots, x_N\}$ drawn from C, and ask you whether some new test case $\tilde{x}$ belongs to C (classify $\tilde{x}$).

Bayesian concept learning - the number game

Suppose, for simplicity, that all numbers are integers between 1 and 100.

Now suppose I tell you "16" is a positive example of the concept. What other numbers do you think are positive?

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

We can represent this as a probability distribution, $p(\tilde{x}|D)$, which is the probability that $\tilde{x} \in C$ given the data D for any $\tilde{x} \in \{1, \ldots, 100\}$. This is called the **posterior predictive distribution**.

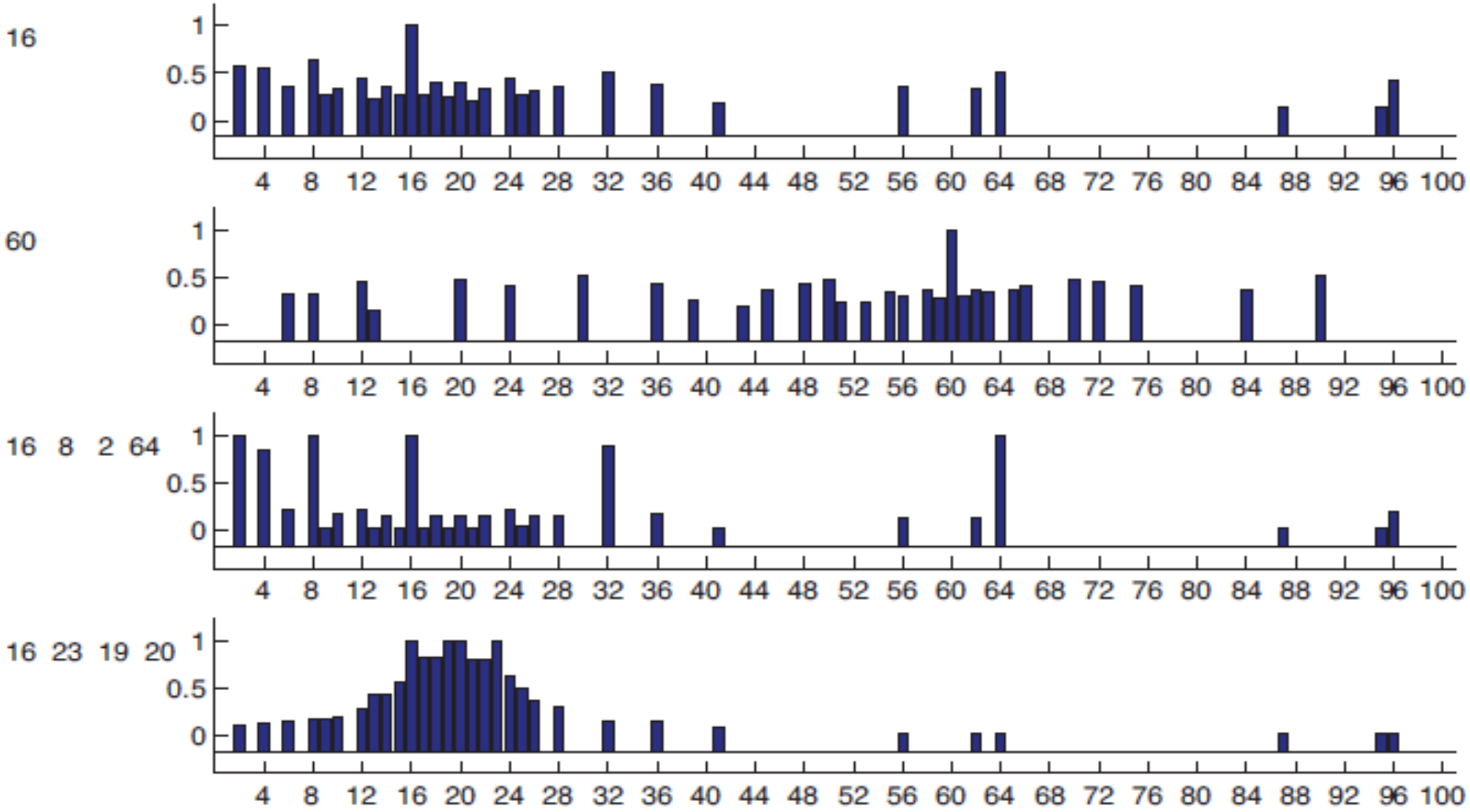
Bayesian concept learning - the number game

Now suppose I tell you that 8, 2 and 64 are also positive examples. Now you may guess that the hidden concept is "powers of two". This is an example of induction.

Given this hypothesis, the predictive distribution is quite specific, and puts most of its mass on powers of 2.

If instead I tell you the data is $D = \{16, 23, 19, 20\}$, you will get a different kind of generalization gradient.

Examples


16
60
16 8 2 64
16 23 19 20
1
0.5
0
4 8 12 16 20 24 28 32 36 40 44 48 52 56 60 64 68 72 76 80 84 88 92 96 100

Bayesian concept learning - the number game

How can we explain this behavior and emulate it in a machine?

The classic approach to induction is to suppose we have a **hypothesis space** of concepts, H, such as: odd numbers, even numbers, all numbers between 1 and 100, powers of two, all numbers ending in j (for $0 \leq j \leq 9$), etc.

The subset of H that is consistent with the data D is called the **version space**. As we see more examples, the version space shrinks, and we become increasingly certain about the concept.

However, the version space is not the whole story. After seeing $D = \{16\}$, there are many consistent rules; how do you combine them to predict if $\tilde{x} \in C$? Also, after seeing $D = \{16, 8, 2, 64\}$, why did you choose the rule "powers of two" and not, say, "all even numbers", or "powers of two except for 32", both of which are equally consistent with the evidence?

Likelihood

We must explain why we chose $h_{two} \triangleq$ "powers of two", and not, say, $h_{even} \triangleq$ "even numbers" after seeing $D = \{16, 8, 2, 64\}$, given that both hypotheses are consistent with the evidence. The key intuition is that we want to avoid **suspicious coincidences**. If the true concept was even numbers, how come we only saw numbers that happened to be powers of two?

To formalize this, let us assume that examples are sampled uniformly at random from the extension of a concept. Tenenbaum calls this the **strong sampling assumption**. Given this assumption, the probability of independently sampling N items (with replacement) from h is given by

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

Likelihood

This crucial equation embodies what Tenenbaum calls the **size principle**, which means the model favors the simplest (smallest) hypothesis consistent with the data. This is more commonly known as **Occam's razor**.

To see how it works, let $D = \{16\}$. Then $p(D|h_{two}) = 1/6$, since there are only 6 powers of two less than 100, but $p(D|h_{even}) = 1/50$, since there are 50 even numbers. So the likelihood that $h = h_{two}$ is higher than if $h = h_{even}$.

After 4 examples, the likelihood of h_{two} is $(1/6)^4 = 7.7 * 10^{-4}$, whereas the likelihood of h_{even} is $(1/50)^4 = 1.6 * 10^{-7}$. This is a likelihood ratio of almost 5000:1 in favor of h_{two} .

This quantifies our earlier intuition that $D = \{16, 8, 2, 64\}$ would be a very suspicious coincidence if generated by h_{even}.

Prior

Suppose $D = \{16, 8, 2, 64\}$. Given this data, the concept *h'*= "powers of two except 32" is more likely than *h'*= "powers of two", since *h* 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.

Of course, every individual's prior may be different. This subjective aspect of Bayesian reasoning is a source of much controversy, since it means, that different individuals will reach different answers. In fact, they presumably not only have different priors, but also different hypothesis spaces.

Prior

Although the subjectivity of the prior is controversial, it is actually quite useful. If you are told the numbers are from some arithmetic rule, then given 1200, 1500, 900 and 1400, you may think 400 is likely but 1183 is unlikely. But if you are told that the numbers are examples of healthy cholesterol levels, you would probably think 400 is unlikely and 1183 is likely.

We see that the prior is the mechanism by which background knowledge can be brought to bear on a problem. Without this, rapid learning (from small samples sizes) is impossible.

So, what prior should we use? In the example, 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. We also make the concepts even and odd more likely apriori and include two "unnatural" concepts, namely "powers of 2, plus 37" and "powers of 2, except 32", but give them low prior weight.

Posterior

The posterior is simply the likelihood times the prior, normalized. In this context we have

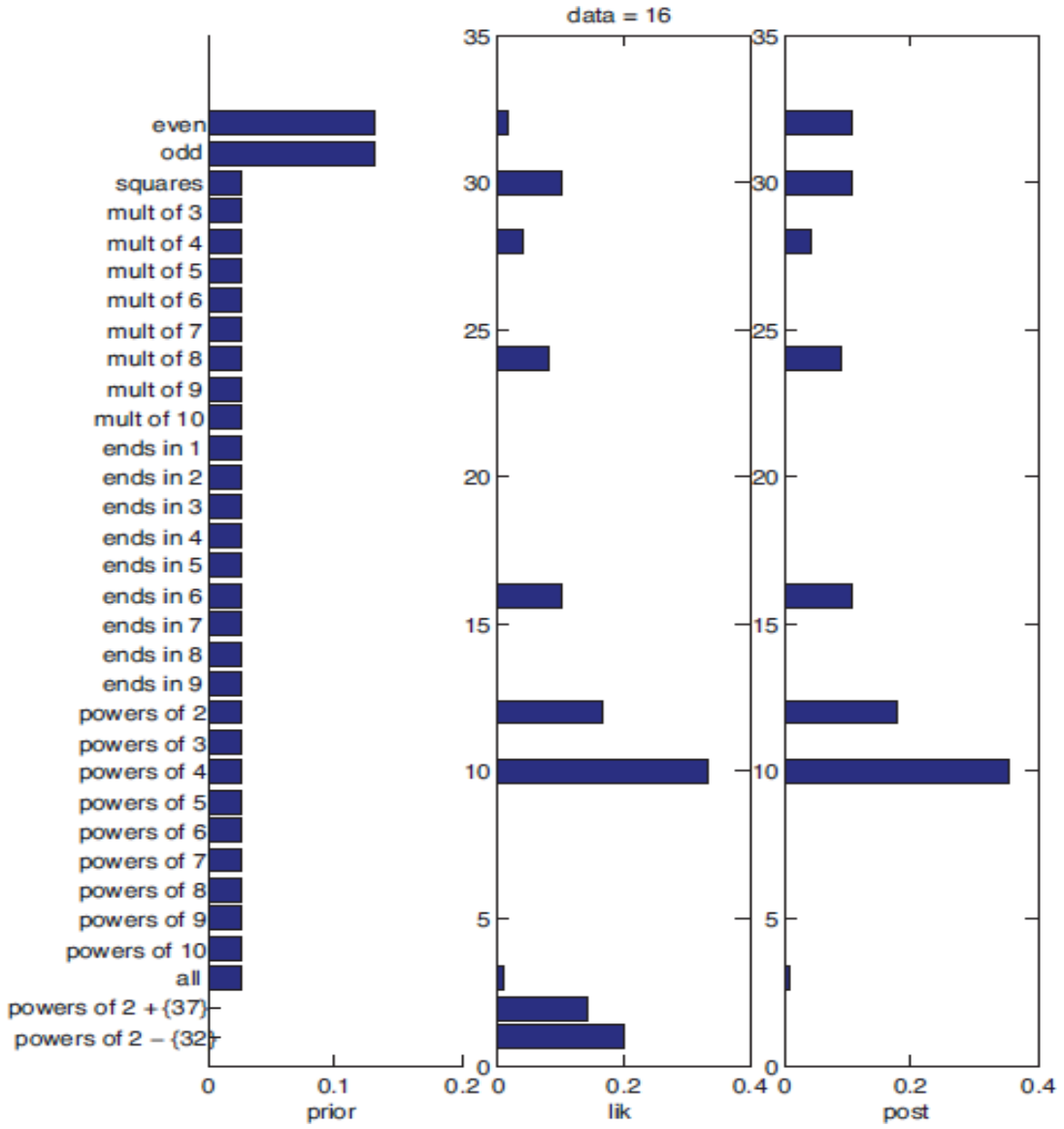
$$p(h|\mathcal{D}) = \frac{p(\mathcal{D}|h)p(h)}{\sum_{h' \in \mathcal{H}} p(\mathcal{D}, h')} = \frac{p(h)\mathbb{I}(\mathcal{D} \in h)/|h|^N}{\sum_{h' \in \mathcal{H}} p(h')\mathbb{I}(\mathcal{D} \in h')/|h'|^N}$$

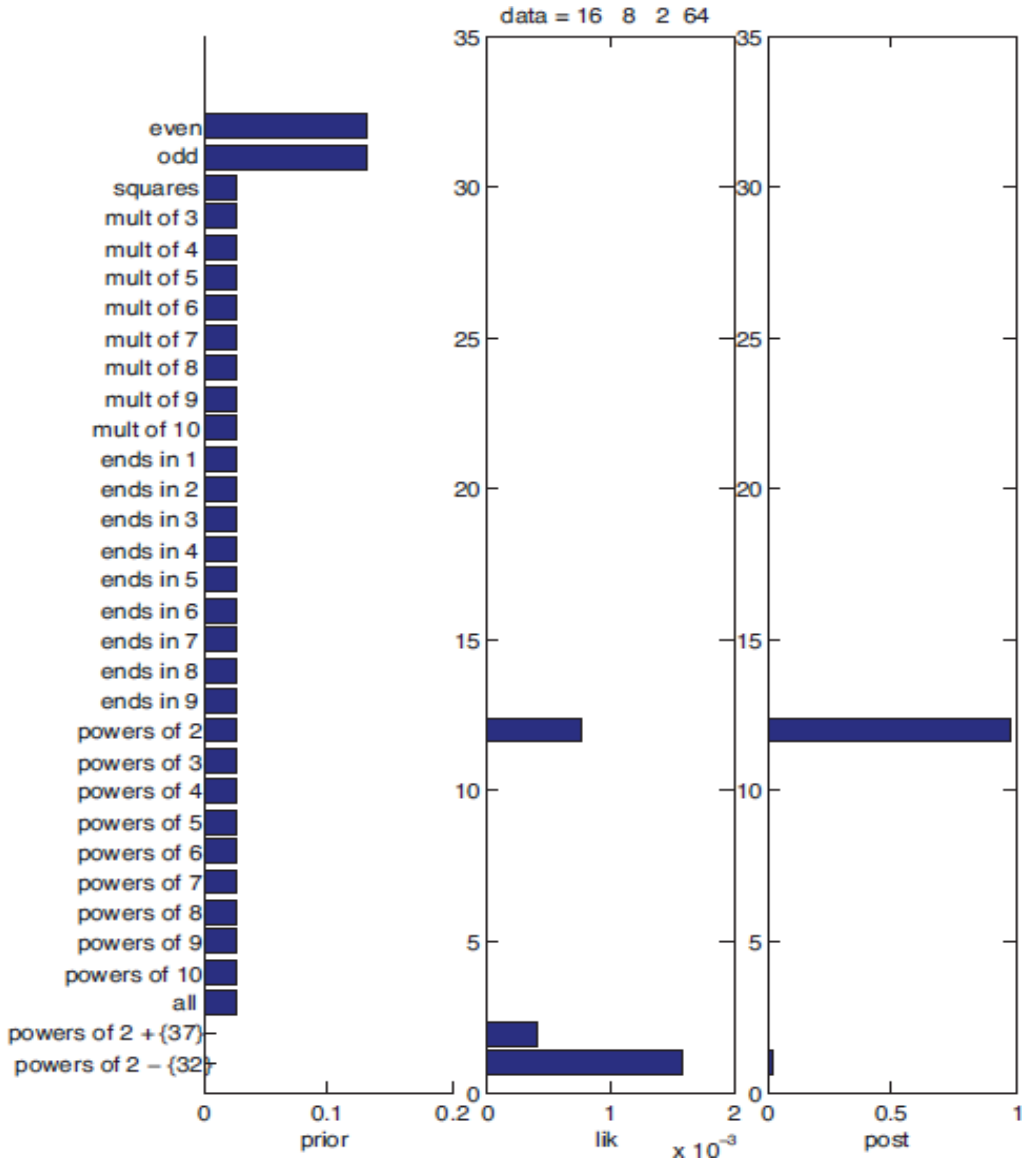
In general, when we have enough data, the posterior $p(h|D)$ becomes peaked on a single concept, namely the MAP estimate

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

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

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


data = 16
even
odd
squares
mult of 3
mult of 4
mult of 5
mult of 6
mult of 7
mult of 8
mult of 9
mult of 10
ends in 1
ends in 2
ends in 3
ends in 4
ends in 5
ends in 6
ends in 7
ends in 8
ends in 9
powers of 2
powers of 3
powers of 4
powers of 5
powers of 6
powers of 7
powers of 8
powers of 9
powers of 10
all
powers of 2 + {37}
powers of 2 − {32}
0
5
10
15
20
25
30
35
0
0.1
0.2
prior
0
0.2
0.4
lik
0
0.2
0.4
post


data = 16 8 2 64
even
odd
squares
mult of 3
mult of 4
mult of 5
mult of 6
mult of 7
mult of 8
mult of 9
mult of 10
ends in 1
ends in 2
ends in 3
ends in 4
ends in 5
ends in 6
ends in 7
ends in 8
ends in 9
powers of 2
powers of 3
powers of 4
powers of 5
powers of 6
powers of 7
powers of 8
powers of 9
powers of 10
all
powers of 2 + {37}
powers of 2 − {32}
0
5
10
15
20
25
30
35
0
0.1
0.2
prior
0
1
2
x 10^-3
lik
0
0.5
1
post

Posterior

Note that the MAP estimate can be written as

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

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**

$$\hat{h}^{mle} \triangleq \underset{h}{\operatorname{argmax}}\, p(\mathcal{D}|h) = \underset{h}{\operatorname{argmax}} \log p(\mathcal{D}|h)$$

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.

Posterior

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

Thus, we say that Bayesian inference (and ML estimation) are consistent estimators.

We also say that the hypothesis space is identifiable in the limit, meaning we can recover the truth in the limit of infinite data.

If our hypothesis class is not rich enough to represent the "truth" (which will usually be the case), we will converge on the hypothesis that is as close as possible to the truth.

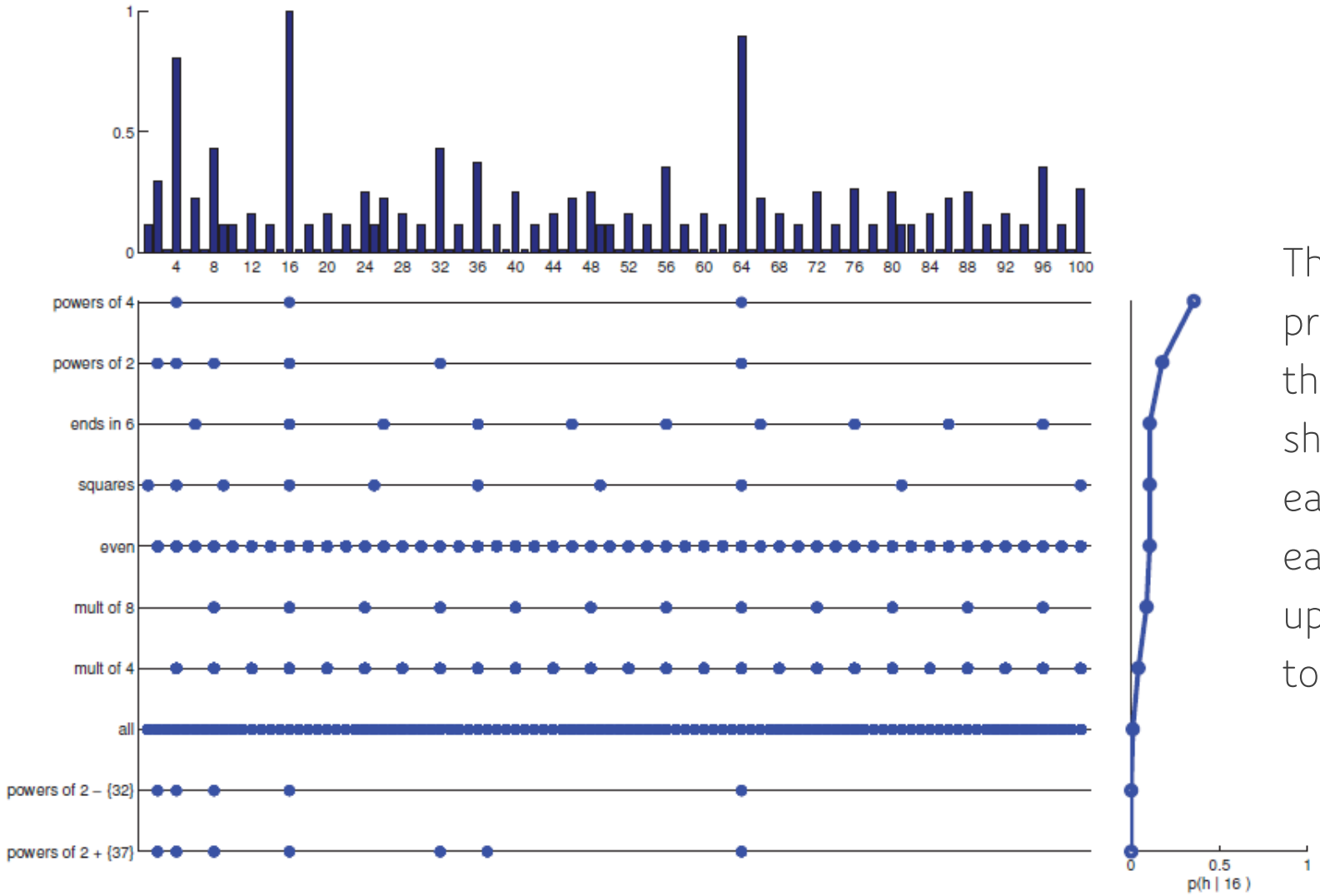
Posterior predictive distribution

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 (this is the basis of the scientific method).

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

$$p(\tilde{x} \in C|\mathcal{D}) = \sum_h p(y = 1|\tilde{x}, h)p(h|\mathcal{D})$$

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



The dots at the bottom show the predictions from each hypothesis; the vertical curve on the right shows the weight associated with each hypothesis. If we multiply each row by its weight and add up, we get the distribution at the top.

Posterior predictive distribution

When we have a small and/or ambiguous dataset, the posterior $p(h|D)$ is vague, which induces a broad predictive distribution. However, once we have "figured things out", the posterior becomes a delta function centered at the MAP estimate. In this case, the predictive distribution becomes

$$p(\tilde{x} \in C|\mathcal{D}) = \sum_h p(\tilde{x}|h)\delta_{\hat{h}}(h) = p(\tilde{x}|\hat{h})$$

This is called a **plug-in approximation** to the predictive density and is very widely used, due to its simplicity. However, in general, this under-represents our uncertainty, and our predictions will not be as "smooth" as when using BMA.

Posterior predictive distribution

For example, suppose we observe $D = \{16\}$. If we use the simple prior above, the minimal consistent hypothesis is "all powers of 4", so only 4, 16 and 64 get a non-zero probability of being predicted. This is of course an example of overfitting.

Given $D = \{16, 8, 2, 64\}$, the MAP hypothesis is "all powers of two". Thus the plug-in predictive distribution gets broader (or stays the same) as we see more data: it starts narrow, but is forced to broaden as it sees more data.

In contrast, in the Bayesian approach, we start broad and then narrow down as we learn more, which makes more intuitive sense. In particular, given $D = \{16\}$, there are many hypotheses with non-negligible posterior support, so the predictive distribution is broad. However, when we see $D = \{16, 8, 2, 64\}$, the posterior concentrates its mass on one hypothesis, so the predictive distribution becomes narrower.

So the predictions made by a plug-in approach and a Bayesian approach are quite different in the small sample regime, although they converge to the same answer as we see more data.

A more complex prior

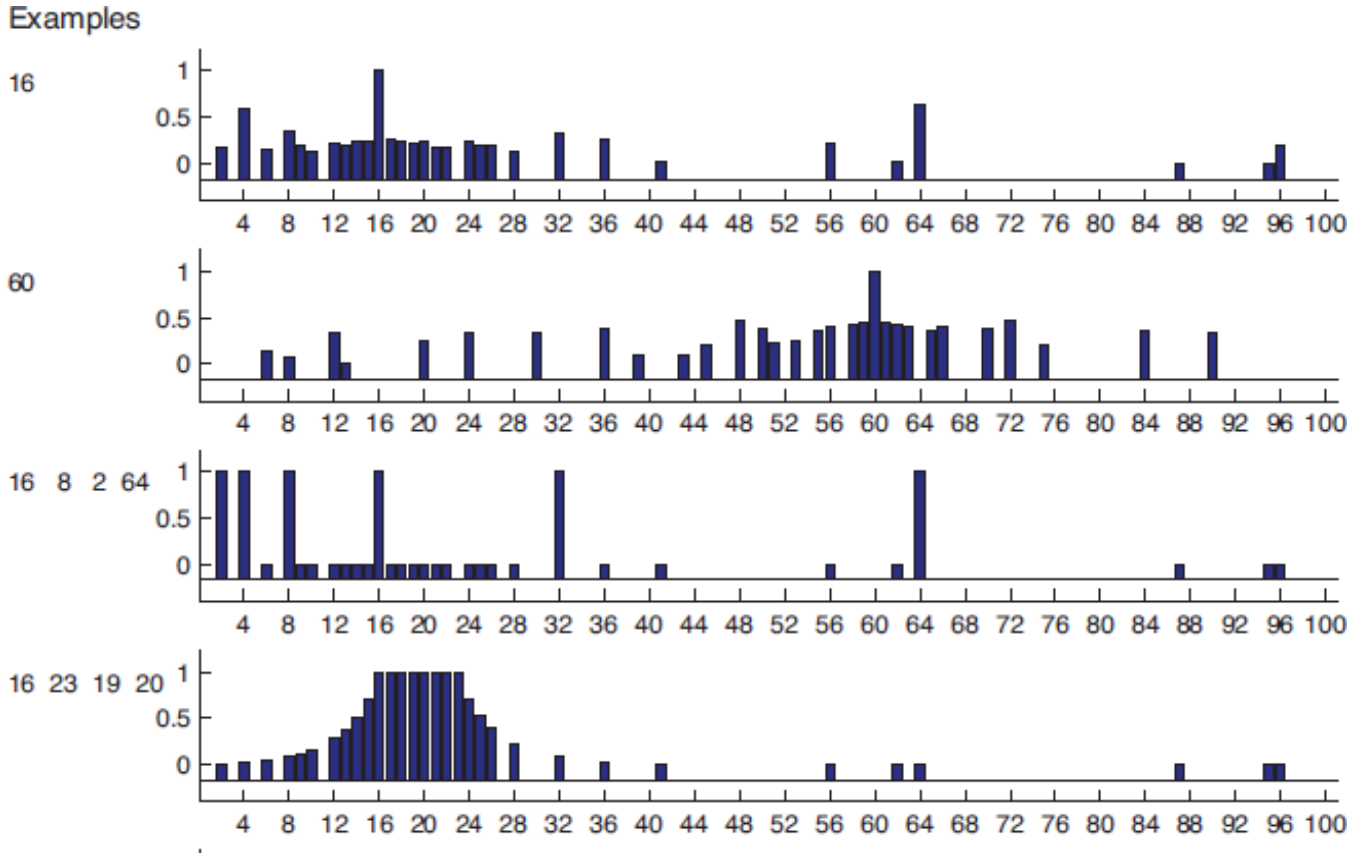
To model human behavior, Tenenbaum used a slightly more sophisticated prior which was derived by analysing some experimental data of how people measure similarity between numbers.

The result is a set of arithmetical concepts similar to those mentioned above, plus all intervals between n and m for

$$1 \leq n, m \leq 100.$$

Thus, the prior is a mixture of two priors, one over arithmetical rules, and one over intervals:

$$p(h) = \pi_0 p_{rules}(h) + (1 - \pi_0) p_{interval}(h)$$



Beta-binomial model

The beta-binomial model

The number game involved inferring a distribution over a discrete variable drawn from a finite hypothesis space, $h \in H$, given a series of discrete observations.

This made the computations particularly simple: we just needed to sum, multiply and divide.

However, in many applications, the unknown parameters are continuous, so the hypothesis space is (some subset) of R^K, where K is the number of parameters.

This complicates the mathematics, since we have to replace sums with integrals. However, the basic ideas are the same.

The beta-binomial model

We will illustrate this by considering the problem of inferring the probability that a coin shows up heads, given a series of observed coin tosses.

Although this might seem trivial, it turns out that this model forms the basis of many of the methods we will consider later in this book, including naive Bayes classifiers, Markov models, etc.

It is historically important, since it was the example which was analyzed in Bayes' original paper of 1763.

We will follow our now-familiar recipe of specifying the likelihood and prior and deriving the posterior and posterior predictive.

The beta-binomial model - Likelihood

Suppose $X_i \sim Ber(\theta)$, where $X_i = 1$ represents "heads", $X_i = 0$ represents "tails", and $\theta \in [0,1]$ is the rate parameter (probability of heads).

The likelihood has the form

$$p(\mathcal{D}|\theta) = \theta^{N_1}(1-\theta)^{N_0}$$

where we have N_1 heads and N_0 tails.

These two counts are called the sufficient statistics of the data, since this is all we need to know about D to infer θ.

More formally, we say $s(D)$ is a sufficient statistic for data D if

$$p(\theta|D) = p(\theta|s(D))$$

Consequently, if we have two datasets with the same sufficient statistics, we will infer the same value for θ.

The beta-binomial model - Likelihood

Now suppose the data consists of the count of the number of heads N_1 observed in a fixed number $N = N_1 + N_0$ of trials. In this case, we have $N_1 \sim \mathrm{Bin}(N, \theta)$, where Bin represents the binomial distribution, which has the following pmf:

$$\mathrm{Bin}(k|n,\theta) \triangleq \binom{n}{k} \theta^k (1-\theta)^{n-k}$$

$\binom{n}{k}$ is a constant independent of θ

So any inferences we make about θ will be the same whether we observe the counts, $D = (N_1, N)$, or a sequence of trials, $D = \{x_1, \ldots, x_N\}$.

The beta-binomial model - Prior

We need a prior which has support over the interval [0, 1].

To make the math easier, it would be convenient if the prior had the same form as the likelihood, i.e., if the prior looked like

$$p(\theta) \propto \theta^{\gamma_1}(1-\theta)^{\gamma_2}$$

for some prior parameters γ_1 and γ_2. If this were the case, then we could easily evaluate the posterior by simply adding up the exponents:

$$p(\theta) \propto p(\mathcal{D}|\theta)p(\theta) = \theta^{N_1}(1-\theta)^{N_0}\theta^{\gamma_1}(1-\theta)^{\gamma_2} = \theta^{N_1+\gamma_1}(1-\theta)^{N_0+\gamma_2}$$

When the prior and the posterior have the same form, we say that the prior is a **conjugate prior** for the corresponding likelihood. Conjugate priors are widely used because they simplify computation, and are easy to interpret, as we see below.

The beta-binomial model - Prior

In the case of the Binomial, the conjugate prior is the beta distribution,

$$\text{Beta}(\theta|a,b) \propto \theta^{a-1}(1-\theta)^{b-1}$$

The parameters of the prior are called **hyper-parameters**. We can set them in order to encode our prior beliefs.

For example, to encode our beliefs that θ has mean 0.7 and standard deviation 0.2, we set $a = 2.975$ and $b = 1.275$.

If we know "nothing" about θ, except that it lies in the interval [0, 1], we can use a uniform prior, which is a kind of uninformative prior. The uniform distribution can be represented by a beta distribution with $a = b = 1$.

The beta-binomial model - Posterior

If we multiply the likelihood by the beta prior we get the following posterior:

$$p(\theta|D) \propto \text{Bin}(N_1|\theta, N_0 + N_1)\text{Beta}(\theta|a, b) = \text{Beta}(\theta|N_1 + a, N_0 + b)$$

In particular, the posterior is obtained by adding the prior hyper-parameters to the empirical counts. For this reason, the hyper-parameters are known as **pseudo counts**. The strength of the prior, also known as the **effective sample size** of the prior, is the sum of the pseudo counts, $a + b$; this plays a role analogous to the data set size, $N_1 + N_0 = N$.

The beta-binomial model - Posterior

Note that updating the posterior sequentially is equivalent to updating in a single batch. To see this, suppose we have two data sets D_a and D_b with sufficient statistics N_1^a N_0^a and N_1^b N_0^b . Let $N_1 = N_1^a + N_1^b$ and $N_0 = N_0^a + N_0^b$ be the sufficient statistics of the combined datasets.

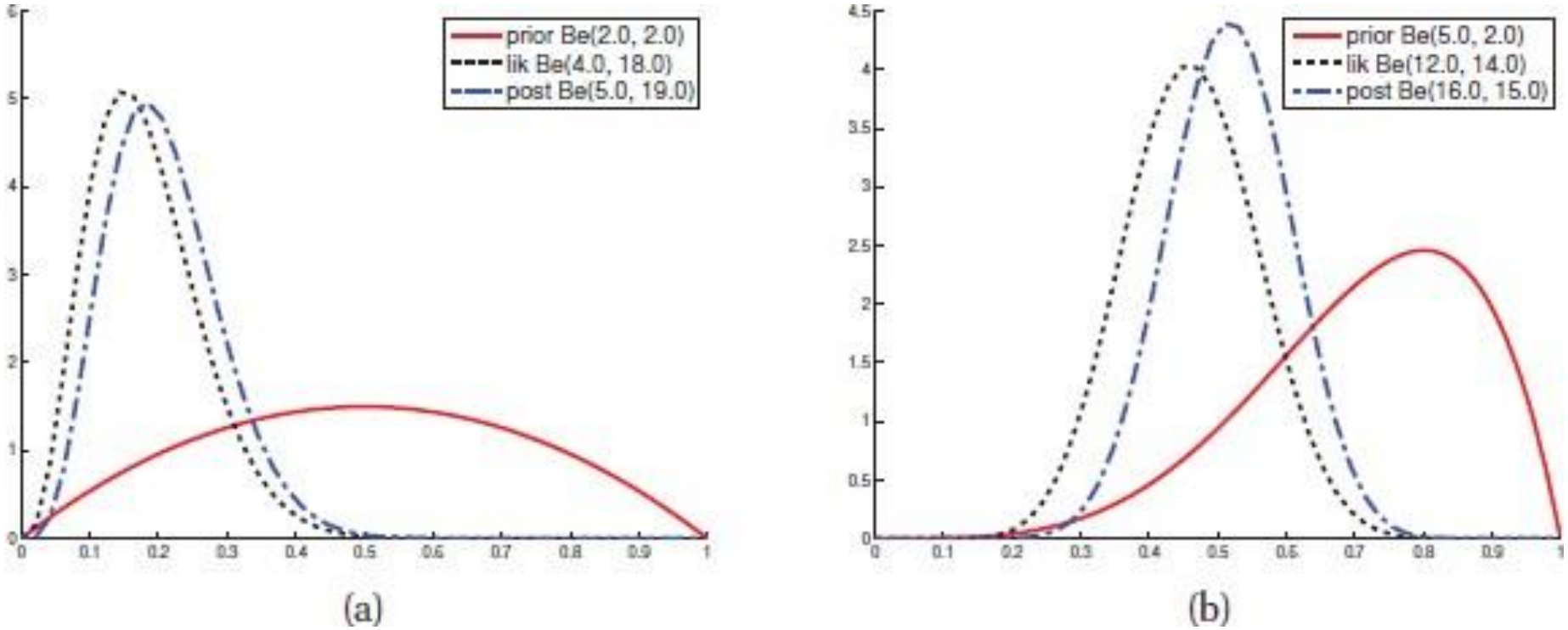
In batch mode, we have

$$p(\theta|\mathcal{D}_a, \mathcal{D}_b) \propto \text{Bin}(N_1|\theta, N_1 + N_0)\text{Beta}(\theta|a, b) \propto \text{Beta}(\theta|N_1 + a, N_0 + b)$$

In sequential mode, we have

$$\begin{aligned} p(\theta|\mathcal{D}_a, \mathcal{D}_b) &\propto p(\mathcal{D}_b|\theta)p(\theta|\mathcal{D}_a) \\ &\propto \text{Bin}(N_1^b|\theta, N_1^b + N_0^b)\text{Beta}(\theta|N_1^a + a, N_0^a + b) \\ &\propto \text{Beta}(\theta|\ N_1^a + N_1^b + a, N_0^a + N_0^b + b) \end{aligned}$$

This makes Bayesian inference particularly well-suited to **online learning**.



(a) an example where we update a weak Beta(2,2) prior with a peaked likelihood function, corresponding to a large sample size; we see that the posterior is essentially identical to the likelihood: since the data has overwhelmed the prior.

(b) an example where we update a strong Beta(5,2) prior with a peaked likelihood function; now we see that the posterior is a "compromise" between the prior and likelihood.

Posterior mean and mode

The MAP estimate is given by (the mode of the corresponding beta distribution)

$$\hat{\theta}_{MAP} = \frac{a + N_1 - 1}{a + b + N - 2}$$

If we use a uniform prior, then the MAP estimate reduces to the MLE, which is just the empirical fraction of heads:

$$\hat{\theta}_{MLE} = \frac{N_1}{N}$$

By contrast, the posterior mean is given by:

$$\overline{\theta} = \frac{a + N_1}{a + b + N}$$

Posterior mean and mode

We will now show that the posterior mean is convex combination of the prior mean and the MLE, which captures the notion that the posterior is a compromise between what we previously believed and what the data is telling us

Let $\alpha_0 = a + b$ be the equivalent sample size of the prior, which controls its strength, and let the prior mean be $m_1 = a/\alpha_0$. Then the posterior mean is given by

$$\mathbb{E}[\theta|\mathcal{D}] = \frac{\alpha_0 m_1 + N_1}{N + \alpha_0} = \frac{\alpha_0}{N + \alpha_0} m_1 + \frac{N}{N + \alpha_0} \frac{N_1}{N} = \lambda m_1 + (1 - \lambda)\hat{\theta}_{MLE}$$

Where $\lambda = \frac{\alpha_0}{N + \alpha_0}$ is the ratio of the prior to posterior equivalent sample size.

So the weaker the prior, the smaller is $\boldsymbol{\lambda}$, and hence the closer the posterior mean is to the MLE

Posterior variance

The mean and mode are point estimates, but it is useful to know how much we can trust them. The variance of the posterior is one way to measure this. The variance of the Beta posterior is given by

$$\mathrm{var}\left[\theta|\mathcal{D}\right] = \frac{(a+N_1)(b+N_0)}{(a+N_1+b+N_0)^2(a+N_1+b+N_0+1)}$$

We can simplify this formidable expression in the case that $N \gg a, b$ to get

$$\mathrm{var}\left[\theta|\mathcal{D}\right] \approx \frac{N_1 N_0}{NNN} = \frac{\hat{\theta}(1-\hat{\theta})}{N}$$

Hence the "error bar" in our estimate, is given by

$$\sigma = \sqrt{\mathrm{var}\left[\theta|\mathcal{D}\right]} \approx \sqrt{\frac{\hat{\theta}(1-\hat{\theta})}{N}}$$

We see that the uncertainty goes down at a rate of $1/\sqrt{N}$. Note, however, that the uncertainty (variance) is maximized when $\hat{\theta} = 0.5$, and is minimized when $\hat{\theta}$ is close to 0 or 1.

The beta-binomial model - Posterior predictive distribution

So far, we have been focusing on inference of the unknown parameter(s). Let us now turn our attention to prediction of future observable data. Consider predicting the probability of heads in a single future trial under a *Beta*(*a*, *b*) posterior. We have

$$\begin{aligned} p(\tilde{x} = 1|\mathcal{D}) &= \int_0^1 p(x = 1|\theta) p(\theta|\mathcal{D}) d\theta \\ &= \int_0^1 \theta \, \mathrm{Beta}(\theta|a, b) d\theta = \mathbb{E}\left[\theta|\mathcal{D}\right] = \frac{a}{a+b} \end{aligned}$$

Thus we see that the mean of the posterior predictive distribution is equivalent to plugging in the posterior mean parameters.

Overfitting and the black swan paradox

Suppose instead that we plug-in the MLE, i.e., we use

$$p(\tilde{x}|\mathcal{D}) \approx \mathrm{Ber}(\tilde{x}|\hat{\theta}_{MLE})$$

Unfortunately, this approximation can perform quite poorly when the sample size is small. For example, suppose we have seen $N = 3$ tails in a row. The MLE $\hat{\theta} = 0/3 = 0$, since this makes the observed data as probable as possible. However, using this estimate, we predict that heads are impossible.

This is called the **zero count problem** or the sparse data problem, and frequently occurs when estimating counts from small amounts of data. One might think that in the era of "big data", such concerns are irrelevant, but note that once we partition the data based on certain criteria – such as the number of times a specific person has engaged in a specific activity – the sample sizes can become much smaller (personalized recommendation of web pages). Thus, Bayesian methods are still useful, even in the big data regime.

Overfitting and the black swan paradox

The zero-count problem is analogous to a problem in philosophy called the **black swan paradox**. This is based on the ancient Western conception that all swans were white. In that context, a black swan was a metaphor for something that could not exist.

Black swans were discovered in Australia by European explorers in the 17th Century.

This paradox was used to illustrate the problem of induction, which is the problem of how to draw general conclusions about the future from specific observations from the past.

Overfitting and the black swan paradox

Let us now derive a simple Bayesian solution to the problem.

We will use a uniform prior, so $a = b = 1$. In this case, plugging in the posterior mean:

$$p(\tilde{x} = 1|\mathcal{D}) = \frac{N_1 + 1}{N_1 + N_0 + 2}$$

This justifies the common practice of adding 1 to the empirical counts, normalizing and then plugging them in, a technique known as **add-one smoothing**.

The beta-binomial model – Example 1

Suppose we have the data of 30 coin tosses. The sufficient statistics of the data are:

- Number of times the coin landed heads - $N_1 = 18$
- Number of times the coin landed tails - $N_0 = 12$

Infer the probability that a coin shows up heads in the next toss

- $p(x = \text{“heads”}) = ?$

The beta-binomial model – Example 1

First, we need to find the distribution (posterior) of θ *based on:*

- A prior belief of fairness of the coin (Prior)
- The data we have seen (Likelihood)

Once we have calculated θ, then the task of predicting whether the next coin toss will be heads or tails is simple

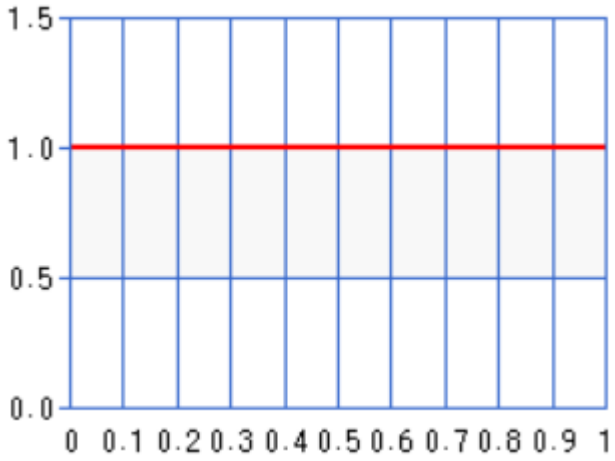
- We either plug in a point estimate of θ in $\mathbf{X}_i \sim \mathbf{Ber}(\theta)$ or integrate over the posterior of θ, $p(\theta|D)$

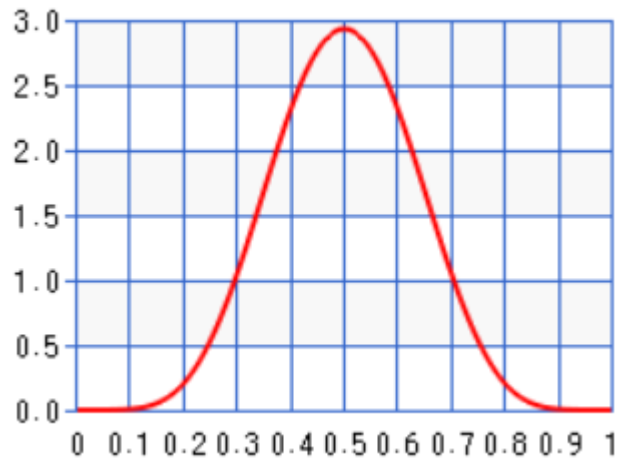
The beta-binomial model – Example 1 Prior

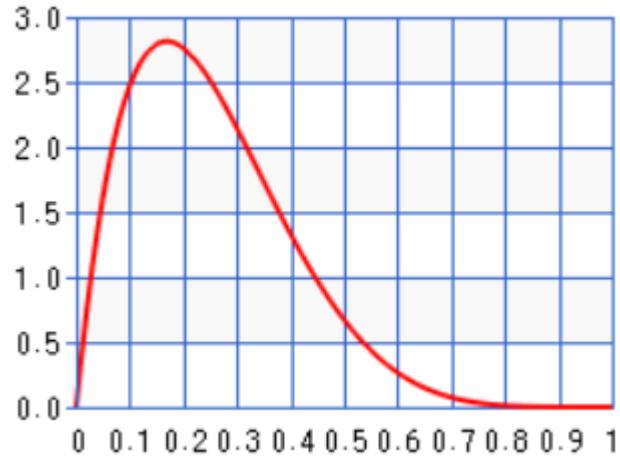
As we saw above, the prior used for the model is the Beta distribution

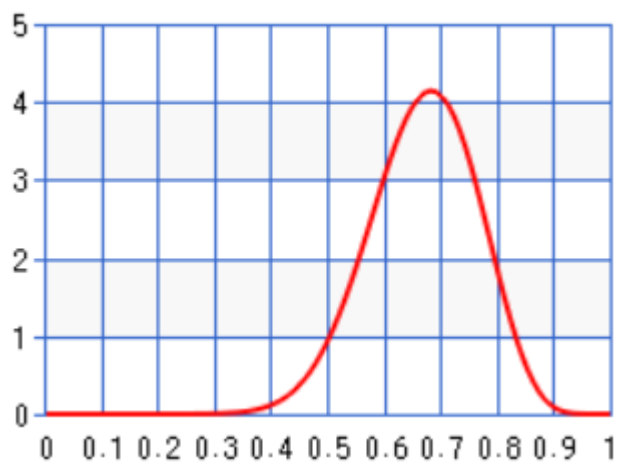
We will look at four cases

1. Uninformative (uniform) prior: $\text{Beta}(\theta \mid a,b)$, where $a=b=1$
2. Strong prior that the coin is fair: $\text{Beta}(\theta \mid a,b)$, where $a=7$, $b=7$
3. Weak prior that the coin is unfair (lands "tails" more often): $\text{Beta}(\theta \mid a,b)$, where $a=2$, $b=6$
4. Strong prior that the coin is unfair (lands "heads" more often): $\text{Beta}(\theta \mid a,b)$, where $a=16$, $b=8$







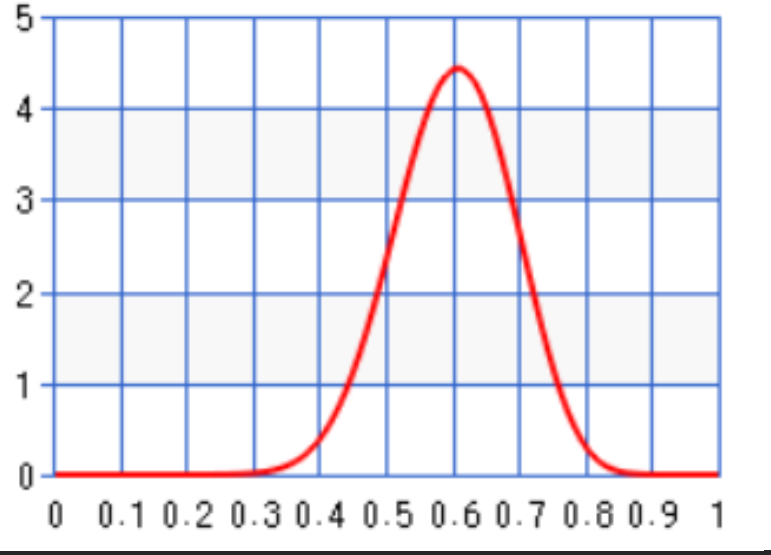


The beta-binomial model – Example 1 Likelihood

The likelihood function $p(D|\theta)$ *is* independent of the prior so it will be the same for all the cases

We can see that the maximum of this function is 0.66 corresponding to N_1/N

- This is technically the maximum likelihood estimate (MLE)

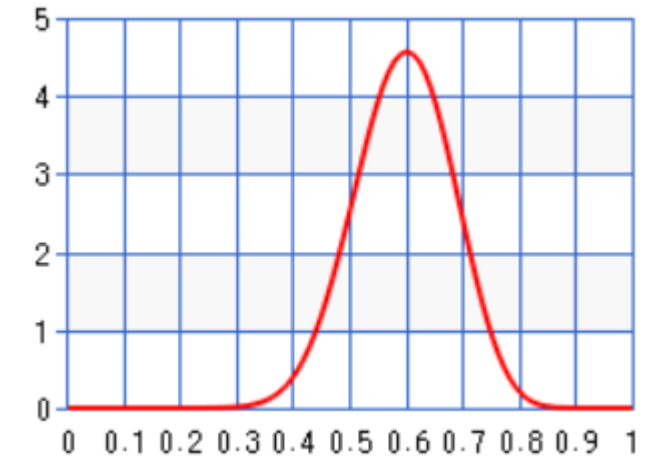


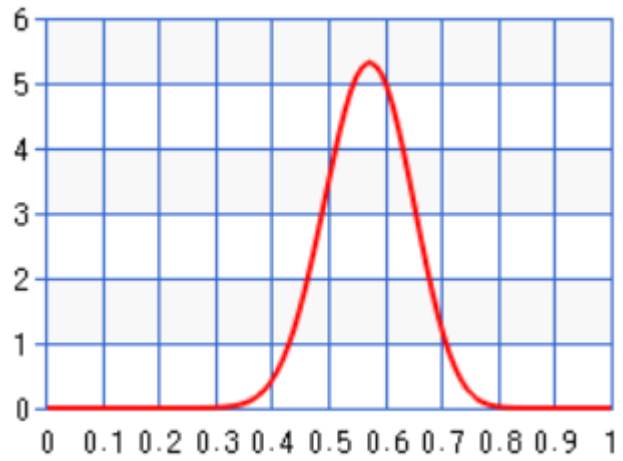
The beta-binomial model – Example 1 Posterior

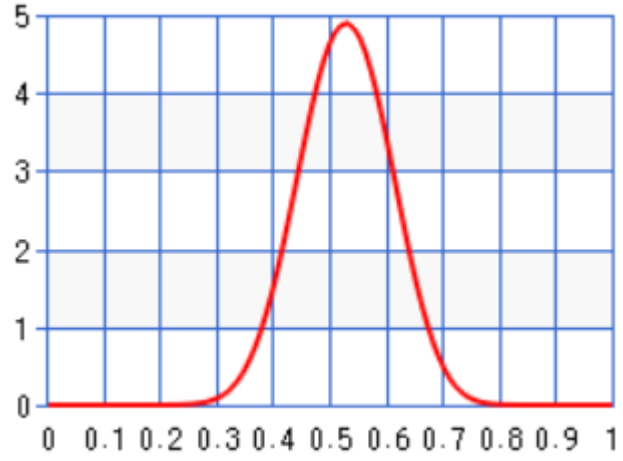
The posterior is calculated as a product of the prior and the likelihood

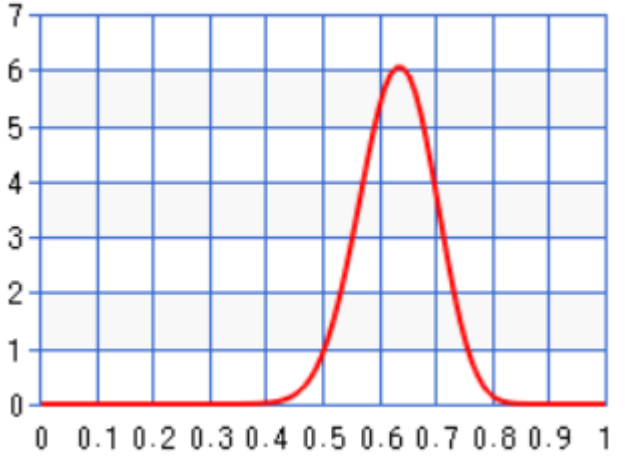
We again analyze the four cases

1. Uninformative prior
2. Strong prior that the coin is fair
3. Weak prior that the coin is unfair (lands "tails" more often)
4. Strong prior that the coin is unfair (lands "heads" more often)









The beta-binomial model – Example 1 Posterior mean and mode

The MLE is the same irrelevant of prior - $\frac{18}{30} = 0.6$

$$\hat{\theta}_{MLE} = \frac{N_1}{N}$$

The MAP estimate for the four cases

1. Uninformative prior - $\frac{1+18-1}{1+1+30-2} = \frac{18}{30} = 0.6$
2. Moderate prior that the coin is fair $\frac{7+18-1}{7+7+30-2} = \frac{24}{42} = 0.57$
3. Weak prior that the coin is unfair (lands "tails" more often) $\frac{2+18-1}{2+6+30-2} = \frac{19}{36} = 0.52$
4. Strong prior that the coin is unfair (lands "heads" more often) $\frac{16+18-1}{16+8+30-2} = \frac{33}{52} = 0.63$

$$\hat{\theta}_{MAP} = \frac{a + N_1 - 1}{a + b + N - 2}$$

The posterior mean for the four cases

1. Uninformative prior $\frac{1+18}{1+1+30} = \frac{19}{32} = 0.59$
2. Moderate prior that the coin is fair $\frac{7+18}{7+7+30} = \frac{25}{44} = 0.57$
3. Weak prior that the coin is unfair (lands "tails" more often) $\frac{2+18}{2+6+30} = \frac{20}{38} = 0.52$
4. Strong prior that the coin is unfair (lands "heads" more often) $\frac{16+18}{16+8+30} = \frac{34}{54} = 0.63$

$$\bar{\theta} = \frac{a + N_1}{a + b + N}$$

The beta-binomial model – Example 1 Posterior variance

The variance for the four cases

$$var[\theta|D] = \frac{(a + N_1)(b + N_0)}{(a + N_1 + b + N_0)^2(a + N_1 + b + N_0 + 1)}$$

1. Uninformative prior – $\mathbf{var}[\theta|D]$ = 0.007
2. Moderate prior that the coin is fair – $\mathbf{var}[\theta|D]$ = 0.005
3. Weak prior that the coin is unfair (lands "tails" more often) – $\mathbf{var}[\theta|D]$ = 0.006
4. Strong prior that the coin is unfair (lands "heads" more often) - $\mathbf{var}[\theta|D]$ = 0.004

The beta-binomial model - Posterior predictive distribution

For the uninformative prior (case 1)

- *If we use the posterior distribution the probability that a new coin will land heads is p(x="heads) = 0.59*
- *If we plug in the MAP estimate the probability that a new coin will land heads is p(x="heads) = 0.6*
- *If we plug in the ML estimate the probability that a new coin will land heads is p(x="heads) = 0.6*

For the strong prior (case 4)

- *If we use the posterior distribution the probability that a new coin will land heads is p(x="heads) = 0.63*
- *If we plug in the MAP estimate the probability that a new coin will land heads is p(x="heads) = 0.63*
- *If we plug in the ML estimate the probability that a new coin will land heads is p(x="heads) = 0.6*

Naïve Bayes

$$p(x_1, \dots, x_n|C_k) = p(x_1|C_k)p(x_2, \dots, x_n|C_k, x_1)$$
$$= p(x_1|C_k)p(x_2|C_k, x_1)p(x_3, \dots, x_n|C_k, x_1, x_2)$$
$$= p(x_1|C_k)p(x_2|C_k, x_1) \dots p(x_n|C_k, x_1, x_2, \dots, x_{n-1})$$

Naive Bayes classifiers

Classify a vector of features by specifying the class conditional distribution, $p(x_j| y = c)$.

The simplest approach is to assume the features are conditionally independent given the class label

$$p(x|y = c, \theta) = \prod_{j=1}^{D} p(x_j|y = c, \theta_{jc})$$

The resulting model is called a **naive Bayes classifier**

Naive Bayes classifiers

The form of the class-conditional density depends on the type of each feature. We give some possibilities below:

- In the case of real-valued features, we can use the Gaussian distribution:

$$p(\mathbf{x}|y=c,\boldsymbol{\theta}) = \prod_{j=1}^{D} \mathcal{N}(x_j|\mu_{jc},\sigma_{jc}^2)$$

- In the case of binary features, $x_j \in \{0,1\}$, we can use the Bernoulli distribution:

$$p(\mathbf{x}|y=c,\boldsymbol{\theta}) = \prod_{j=1}^{D} \mathrm{Ber}(x_j|\mu_{jc})$$

- In the case of categorical features, $x_j \in \{1,\ldots,K\}$, we can use the multinoulli distribution:

$$p(\mathbf{x}|y=c,\boldsymbol{\theta}) = \prod_{j=1}^{D} \mathrm{Cat}(x_j|\boldsymbol{\mu}_{jc}),$$

Model fitting

The probability for a single data case is given by

$$p(\mathbf{x}_i, y_i|\boldsymbol{\theta}) = p(y_i|\boldsymbol{\pi}) \prod_j p(x_{ij}|\boldsymbol{\theta}_j) = \prod_c \pi_c^{\mathbb{I}(y_i=c)} \prod_j \prod_c p(x_{ij}|\boldsymbol{\theta}_{jc})^{\mathbb{I}(y_i=c)}$$

The probability of each class $p(y_i)$ is given by

$$\hat{\pi}_c = \frac{N_c}{N}$$

The maximum likelihood depends on the type of distribution for each feature. If the features are binary $p(x_{jc})$ is given by

$$\hat{\theta}_{jc} = \frac{N_{jc}}{N_c}$$

Bayesian naive Bayes

The trouble with maximum likelihood is that it can overfit.

A simple solution to overfitting is to be Bayesian

Often, we just take $a=1$ and $b=1$, corresponding to **add-one smoothing** or **Laplace smoothing**.

In other words, to compute the posterior, we just update the prior counts with the empirical counts from the likelihood.

Using the model for prediction

At test time, the goal is to compute

$$p(y = c|x, D) \propto p(y = c|D) \prod_{j=1}^{D} p(x_j|y = c, D)$$

The correct Bayesian procedure is to integrate out the unknown parameters, but we often just use a plug-in approximation (like MLE) for each parameter

Feature selection using mutual information

As NBC is fitting a joint distribution over potentially many features, it can suffer from overfitting.

In addition, the run-time cost is $O(D)$, which may be too high for some applications.

One common approach to tackling both of these problems is to perform **feature selection**

Evaluate the relevance of each feature separately, and then take the top K, where K is chosen based on some tradeoff between accuracy and complexity.

This approach is known as variable **ranking**, **filtering**, or **screening**.

One way to measure relevance is to use MI between feature $\mathbf{X}_j$ and the class label $\mathbf{Y}$

$$I(X,Y) = \sum_{x_j} \sum_{y} p(x_j, y) \log \frac{p(x_j, y)}{p(x_j)p(y)}$$

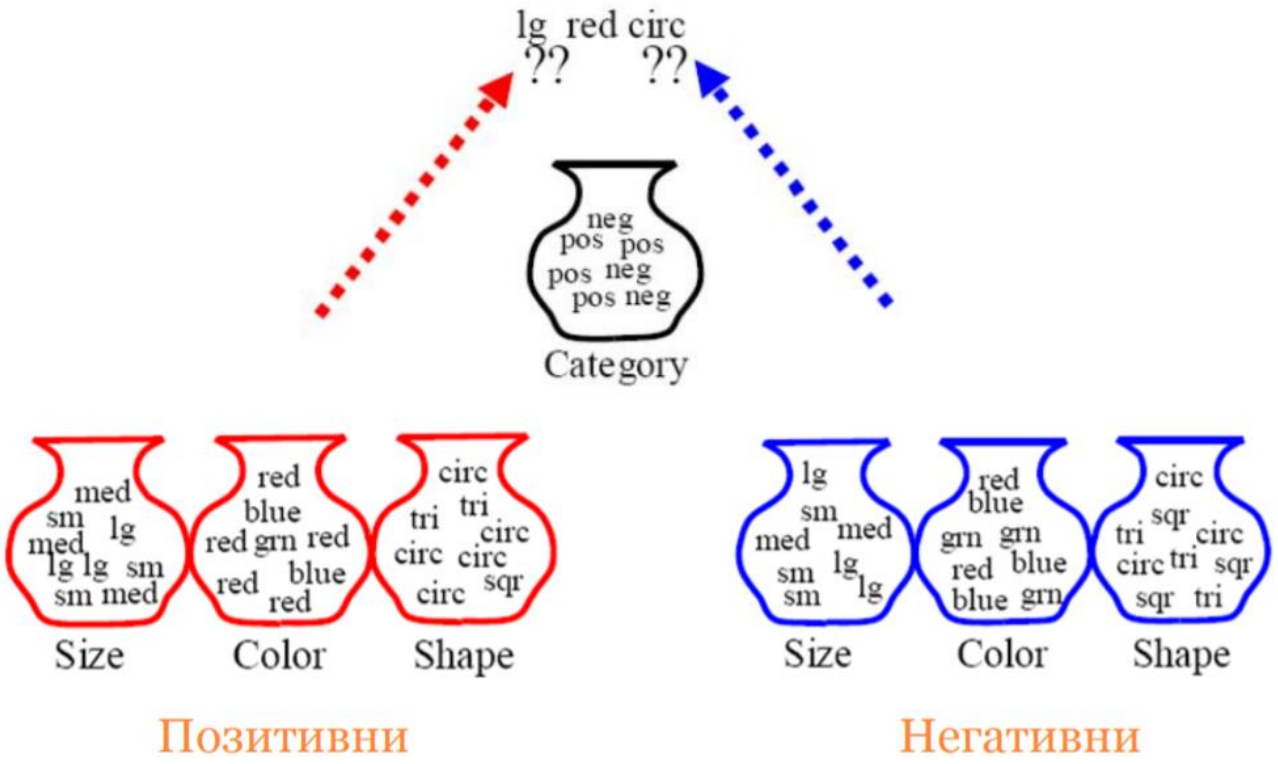
Naïve Bayes: Example 1

Suppose we have a dataset with features:

- Size (small, medium, large)
- Color(red, blue, green)
- Shape (circle, square, triangle)
- **Class** (positive, negative)

The goal is to now classify a new instance:

- x=[medium, red, circle]
- y=?



Naïve Bayes: Example 1

First, we need to calculate the parameters based on the data

- the prior probabilities of each class $p(y_c)$
- the likelihood of the data $p(x_{jc}|y_c)$

All the probabilities needed for the parameters are calculated in this example and shown in the table

Features are discrete

Probability	positive	negative
P(Y)	0.5	0.5
P(small | Y)	0.4	0.4
P(medium | Y)	0.1	0.2
P(large | Y)	0.5	0.4
P(red | Y)	0.9	0.3
P(blue | Y)	0.05	0.3
P(green | Y)	0.05	0.4
P(square | Y)	0.05	0.4
P(triangle | Y)	0.05	0.3
P(circle | Y)	0.9	0.3

Naïve Bayes: Example 1

Probability	positive	negative
P(Y)	0.5	0.5
P(medium | Y)	0.1	0.2
P(red | Y)	0.9	0.3
P(circle | Y)	0.9	0.3

Goal: classify the new test instance

First, we find the cells of the table which provide the info we need

- The row pertaining to $p(y_c)$ which gives us values for π
- The rows pertaining to $p(x_{jc}|y_c)$ which give us values for θ_{jc}
- Once we have these values, we just multiply them to gain the posterior probability of each class

p(pos|X)=p(pos|m,r,c)=p(m,r,c|pos)*p(pos)/p(X)=
=p(m|pos)*p(r|pos)*p(c|pos)*p(pos)/p(X)=0,1*0,9*0,9*0,5/p(X)=0,0495/p(X)

p(neg|X)=p(m,r,c|neg)*p(neg)/P(X)=
=p(m|neg)*p(r|neg)*p(c|neg)*p(neg)/P(X)=0,2*0,3*0,3*0,5/P(X)=0,009/P(X)

p(pos|X)+p(neg|X)=1

0,0495/p(X)+0,009/p(X)=1

=> p(X)=0,0585

Naïve Bayes: Example 2

Now we have a new, similar dataset with the same features as before, based on the following table

The goal is the same as before, classify:

- x=[medium, red, circle]
- y=?

Ex	Size	Color	Shape	Category
1	small	red	circle	positive
2	large	red	circle	positive
3	small	red	triangle	negitive
4	large	blue	circle	negitive

Naïve Bayes: Example 2

We calculate the probabilities corresponding to the data and calculate the posterior probabilities for each class just as before

$$P(\text{positive} \mid X) = 0.5 * 0.0 * 1.0 * 1.0 / P(X) = 0$$

$$P(\text{negative} \mid X) = 0.5 * 0.0 * 0.5 * 0.5 / P(X) = 0$$

Problem: in the table we have no data pertaining to size=*medium*

The resulting probability of $p(medium|Y) = 0$, leading to the final posterior probabilities of each class being 0

Probability	positive	negative
P(*Y*)	0.5	0.5
P(small | *Y*)	0.5	0.5
P(medium | *Y*)	0.0	0.0
P(large | *Y*)	0.5	0.5
P(red | *Y*)	1.0	0.5
P(blue | *Y*)	0.0	0.5
P(green | *Y*)	0.0	0.0
P(square | *Y*)	0.0	0.0
P(triangle | *Y*)	0.0	0.5
P(circle | *Y*)	1.0	0.5

Naïve Bayes: Example 2

To solve this problem we use a Bayesian approach

- Add priors to each parameter that don't allow them to be 0
- In the simplest case, we use add-one smoothing or Laplacian smoothing

The new probabilities are

$$p(small|positive) = \frac{1+1}{2+3} = 0.4$$

$$p(medium|positive) = \frac{0+1}{2+3} = 0.2$$

$$p(large|positive) = \frac{1+1}{2+3} = 0.4$$

Now we can re-calculate the posterior probability for each class

Example 3: E-mail classification

Using Naive Bayes we assume that words are independent of each other. An assumption that is not correct but does not hurt too much in the task.

To simplify with the assumption, we assume that the position of the words does not matter. Calculate $P(w_j | y_k)$ i.e. the probability that the word w_j appears in a message for a given class y_k.

Create a dictionary: make a list of all the words that appear in the training set (very high or very low frequency words can be discarded)

Example 3

SPAM

OFFER IS SECRET

CLICK SECRET LINK

SECRET SPORTS LINK

HAM

PLAY SPORTS TODAY
WENT PLAY SPORTS
SECRET SPORTS EVENT
SPORT IS TODAY
SPORT COSTS MONEY

- Questions:
  - Size of the dictionary? 13 words
  - P(SPAM) = 3/8

SPAM

OFFER IS SECRET

CLICK SECRET LINK

SECRET SPORTS LINK

HAM

PLAY SPORTS TODAY
WENT PLAY SPORTS
SECRET SPORTS EVENT
SPORT IS TODAY
SPORT COSTS MONEY

- Questions:
  - P("SECRET" | SPAM) = 1/3
  - P("SECRET" | HAM) = 1/15

SPAM	HAM
OFFER IS SECRET	PLAY SPORTS TODAY
CLICK SECRET LINK	WENT PLAY SPORTS
SECRET SPORTS LINK	SECRET SPORTS EVENT
	SPORT IS TODAY
	SPORT COSTS MONEY

- Questions:
  - MESSAGE M = "SPORTS"
  - P(SPAM | M) = 1/6 Applying Bayes' Rule

p(spam|sports)=p(sports|spam)*p(spam)/p(sports)=
=(1/9 * 3/8) / (6/24) = 1/6

p(ham|sports)=p(sports|ham)*p(ham)/p(sports)=
=(5/15 * 5/8) / (6/24) = 5/6

SPAM	HAM
OFFER IS SECRET	PLAY SPORTS TODAY
CLICK SECRET LINK	WENT PLAY SPORTS
SECRET SPORTS LINK	SECRET SPORTS EVENT
	SPORT IS TODAY
	SPORT COSTS MONEY

- Questions:
  - MESSAGE M = "SECRET IS SECRET"
  - P(SPAM | M) = 25/26 Applying Bayes' Rule

p(spam|"sec is sec")=p("sec is sec"|spam)*p(spam)/p(X)=
=p(sec|spam)*p(is|spam)*p(sec|spam)*p(spam)/p(X)=
= 1/3 * 1/9 * 1/3 * 3/8 / p(X) = 25/26

p(ham|"sec is sec")=p("sec is sec"|ham)*p(ham)/p(X)=
=p(sec|ham)*p(is|ham)*p(sec|ham)*p(ham)/p(X)=
= 1/15 *1/15 * 1/15 *5/8 / p(X) = 1/26

p(X)=3/(3*9*3*8)+5/(15*15*15*8)

SPAM	HAM
OFFER IS SECRET	PLAY SPORTS TODAY
CLICK SECRET LINK	WENT PLAY SPORTS
SECRET SPORTS LINK	SECRET SPORTS EVENT
	SPORT IS TODAY
	SPORT COSTS MONEY

- Questions:
  - MESSAGE M = “TODAY IS SECRET”
  - P(SPAM | M) = 0 Applying Bayes’ Rule

Naïve Bayes: Example 4

Finally, we look at an example where the features are continuous random variables

- Therefore, we model them using the Gaussian distribution instead of a Bernoulli or Categorical distribution

Using the data shown in the table we need to calculate the parameters of the Gaussian for each feature for each class independently

- calculate π for the prior probabilities of each class $p(y_c)$
- calculate μ and σ^2 for the likelihood of the data $p(x_{jc}|y_c)$

sex	height (feet)	weight (lbs)	foot size(inches)
male	6	180	12
male	5.92 (5'11")	190	11
male	5.58 (5'7")	170	12
male	5.92 (5'11")	165	10
female	5	100	6
female	5.5 (5'6")	150	8
female	5.42 (5'5")	130	7
female	5.75 (5'9")	150	9

	male			female		
	height	weight	footsize	height	weight	footsize
mean	5.86	176.25	11.25	5.42	132.50	7.50
variance	0.04	122.92	0.92	0.10	558.33	1.67

Example 4

The goal again is to classify (find the posterior probabilities of each class for) a new instance

- x=[6, 130, 8]

We again use the same formula as before

$$posterior(male) = \frac{P(male)p(height|male)p(weight|male)p(footsize|male)}{evidence}$$

$$posterior(female) = \frac{P(female)p(height|female)p(weight|female)p(footsize|female)}{evidence}$$

The evidence is the same for both classes so we don't need to estimate it to choose the most probable class. However, if we need the probability of each class we can calculate it as

$$evidence = P(male)p(height|male)p(weight|male)p(footsize|male) + P(female)p(height|female)p(weight|female)p(footsize|female)$$

Example 4

We calculate the probability distribution of each feature for each class

$$P(male) = 0.5$$

$$p(height|male) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(\frac{-(6-\mu)^2}{2\sigma^2}\right) = 1.57$$

$$p(weight|male) = 5.98 \cdot 10^{-6}$$

$$p(footsize|male) = 1.31 \cdot 10^{-3}$$

$$P(female) = 0.5$$

$$p(height|female) = 0.223$$

$$p(weight|female) = 0.017$$

$$p(footsize|female) = 0.287$$

0.000544014649533

Finally, we calculate the posterior distribution as their product

- *posterior* (male) = 6.19 * 10^{-9} / evidence
- *posterior* (female) = 5.38 * 10^{-4} / evidence
- Therefore, the new data point is assigned class = female

Naïve Bayes classifier

The model is called "naive" since we do not expect the features to be independent, even conditional on the class label.

However, even if the naive Bayes assumption is not true, it often results in classifiers that work well.

One reason for this is that the model is quite simple (it only has $O(CD)$ parameters, for C classes and D features), and hence it is relatively immune to overfitting.

Handles noise pretty well because of the independence assumption

- Doesn't fully adapt to the data