Bayesian Learning

Bayesian Learning

Features of Bayesian learning methods:

- Each observed training example can incrementally decrease or increase the estimated probability that a hypothesis is correct.
  - This provides a more flexible approach to learning than algorithms that completely eliminate a hypothesis if it is found to be inconsistent with any single example.
- Prior knowledge can be combined with observed data to determine the final probability of a hypothesis. In Bayesian learning, prior knowledge is provided by asserting
  - a prior probability for each candidate hypothesis, and
  - a probability distribution over observed data for each possible hypothesis.
- Bayesian methods can accommodate hypotheses that make probabilistic predictions
- New instances can be classified by combining the predictions of multiple hypotheses, weighted by their probabilities.
- Even in cases where Bayesian methods prove computationally intractable, they can provide a standard of optimal decision making against which other practical methods can be measured.

Difficulties with Bayesian Methods

- Require initial knowledge of many probabilities
  - When these probabilities are not known in advance they are often estimated based on background knowledge, previously available data, and assumptions about the form of the underlying distributions.
- Significant computational cost is required to determine the Bayes optimal hypothesis in the general case (linear in the number of candidate hypotheses).
  - In certain specialized situations, this computational cost can be significantly reduced.

Bayes Theorem

- In machine learning, we try to determine the ***best hypothesis*** from some hypothesis space H, given the observed training data D.
- In Bayesian learning, the ***best hypothesis*** means the ***most probable*** hypothesis, given the data D plus any initial knowledge about the prior probabilities of the various hypotheses in H.
- Bayes theorem provides a way to calculate the probability of a hypothesis based on its prior probability, the probabilities of observing various data given the hypothesis, and the observed data itself.

Bayes Theorem

P(h) is ***prior probability of hypothesis h***

- P(h) to denote the initial probability that hypothesis h holds, before observing training data.
- P(h) may reflect any background knowledge we have about the chance that h is correct. If we have no such prior knowledge, then each candidate hypothesis might simply get the same prior probability.

P(D) is ***prior probability of training data D***

- The probability of D given no knowledge about which hypothesis holds

P(h|D) is ***posterior probability of h given D***

- P(h|D) is called the ***posterior probability*** of ***h,*** because it reflects our confidence that ***h*** holds after we have seen the training data ***D.***
- The posterior probability P(h|D) reflects the influence of the training data ***D,*** in contrast to the prior probability P(h), which is independent of D.

P(D|h) is ***posterior probability of D given h***

- The probability of observing data ***D*** given some world in which hypothesis ***h*** holds.
- Generally, we write ***P(xly)*** to denote the probability of **event x** given **event y.**

Bayes Theorem

- In ML problems, we are interested in the probability P(h|D) that h holds given the observed training data D.
- Bayes theorem provides a way to calculate the posterior probability P(h|D), from the prior probability P(h), together with P(D) and P(D|h).

Bayes Theorem: $$P(h \mid D) = \frac{P(D \mid h)\, P(h)}{P(D)}$$

- P(h|D) increases with P(h) and P(D|h) according to Bayes theorem.
- P(h|D) decreases as P(D) increases, because the more probable it is that D will be observed independent of h, the less evidence D provides in support of h.

Bayes Theorem

Sample Space for events A and B

A holds	T	T	F	F	T	F	T
B holds	T	F	T	F	T	F	F

$P(A) = 4/7$ $P(B) = 3/7$ $P(B|A) = 2/4$ $P(A|B) = 2/3$

Is Bayes Theorem correct?

$P(B|A) = P(A|B)P(B) / P(A) = (2/3 * 3/7) / 4/7 = 2/4$ ➔ CORRECT

$P(A|B) = P(B|A)P(A) / P(B) = (2/4 * 4/7) / 3/7 = 2/3$ ➔ CORRECT

Maximum A Posteriori (MAP) Hypothesis, hMAP

- The learner considers some set of candidate hypotheses H and it is interested in finding the ***most probable hypothesis*** $h \in H$ given the observed data D
- Any such maximally probable hypothesis is called a ***maximum a posteriori (MAP) hypothesis*** h_{MAP}.
- We can determine the MAP hypotheses by using Bayes theorem to calculate the posterior probability of each candidate hypothesis.

$$h_{MAP} \equiv \underset{h \in H}{\operatorname{argmax}} P(h|D)$$

$$= \underset{h \in H}{\operatorname{argmax}} \frac{P(D|h)P(h)}{P(D)}$$

$$= \underset{h \in H}{\operatorname{argmax}} P(D|h)P(h)$$

Example - Does patient have cancer or not?

- The test returns a correct positive result in only 98% of the cases in which the disease is actually present, and a correct negative result in only 97% of the cases in which the disease is not present.
- Furthermore, .008 of the entire population have cancer.

P(cancer) = .008 P(notcancer) = .992
P(+|cancer) = .98 P(-|cancer) = .02
P(+|notcancer) = .03 P(-|notcancer) = .97

- A patient takes a lab test and the result comes back positive.

P(+|cancer) P(cancer) = .98 * .008 = .0078
P(+|notcancer) P(notcancer) = .03 * .992 = .0298 ➔ h_{MAP} *is notcancer*

- Since P(cancer|+) + P(notcancer|+) must be 1

P(cancer|+) = .0078 / (.0078+.0298) = .21
P(notcancer|+) = .0298 / (.0078+.0298) = .79

Expectation is a linear operator defined as follows:

$$E(X) = \sum_{x_i \in E} x_i P(X = x_i)$$

Variance of a linear operator is defined as follows:

$$Var(X) = E((X - E(X|\theta))^2)$$

Let us introduce the indicator variable $X = I_A(w)$

$$X = I_A(w) = \begin{cases} 1, & \text{if } w \in A. \\ 0, & \text{otherwise.} \end{cases}$$

What is the expected value of A?

A Bernoulli r.v. x takes values in $\{0, 1\}$

$$p(x|\theta) = \theta^x (1-\theta)^{(1-x)}$$

What is the expectation of a Bernoulli Distribution?
What is the variance of a Bernoulli Distribution?
What is the distribution of N independent coin tosses?

Maximum Likelihood Estimation

- Step 1: Given n data, $x_{1:n} = \{x_1, x_2, \ldots, x_n\}$ write down the expression for the joint distribution of the data:

$$p(x_{1:n}|\theta) = \prod_{i=1}^{n} P(x_i|\theta)$$

- Step 2:Compute the log-likelihood
- Step 3:Differentiate and equate to zero to find the estimate of θ

What is the distribution of N independent coin tosses?Compute the MLE for Bernoulli θ

(Hint: Take m as number of 1s, and n as number of coin flips)

Problem Setup

A Bernoulli random variable represents the outcome of a single trial, with:

- $X = 1$ (success) with probability θ,
- $X = 0$ (failure) with probability $1 - \theta$.

The probability mass function is:

$$P(X|\theta) = \theta^X (1 - \theta)^{1-X}.$$

Given a dataset of n independent observations $\{x_1, x_2, \ldots, x_n\}$, where each $x_i \in \{0, 1\}$, the goal is to estimate θ using the MLE.

Likelihood Function

The likelihood function is the joint probability of observing all n samples:

$$L(\theta) = P(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n|\theta).$$

For independent observations:

$$L(\theta) = \prod_{i=1}^{n} P(x_i|\theta).$$

Substituting the Bernoulli PMF:

$$L(\theta) = \prod_{i=1}^{n} \theta^{x_i}(1-\theta)^{1-x_i}.$$

Log-Likelihood Function

Taking the logarithm of the likelihood for simplification:

$$\ell(\theta) = \ln L(\theta) = \sum_{i=1}^{n} \left[x_i \ln \theta + (1 - x_i) \ln(1 - \theta) \right].$$

Simplify:

$$\ell(\theta) = \left(\sum_{i=1}^{n} x_i \right) \ln \theta + \left(\sum_{i=1}^{n} (1 - x_i) \right) \ln(1 - \theta).$$

Let:

- $S = \sum_{i=1}^{n} x_i$ (the total number of successes).

The log-likelihood becomes:

$$\ell(\theta) = S \ln \theta + (n - S) \ln(1 - \theta).$$

Maximizing the Log-Likelihood

To find the MLE, differentiate $\ell(\theta)$ with respect to θ and set it to zero:

$$\frac{\partial \ell(\theta)}{\partial \theta} = \frac{S}{\theta} - \frac{n-S}{1-\theta}.$$

Set $\frac{\partial \ell(\theta)}{\partial \theta} = 0$:

$$\frac{S}{\theta} = \frac{n-S}{1-\theta}.$$

Simplify:

$$S(1-\theta) = (n-S)\theta.$$

Expand:

$$S - S\theta = n\theta - S\theta.$$

Rearrange:

$$S = n\theta.$$

Solve for θ:

$$\theta = \frac{S}{n}.$$

MLE for θ

The MLE for θ is:

$$\hat{\theta} = \frac{\text{Number of successes}}{\text{Total number of trials}} = \frac{\sum_{i-1}^{n} x_i}{n}.$$

Bayesian Learning
(Bayesian Learning Procedure)

- Step 1: Given n data, $x_{1:n} = \{x_1, x_2, \ldots, x_n\}$, write down the expression for the likelihood:

$$p(x_{1:n}|\theta) = \theta^m(1-\theta)^{(n-m)} \text{ (for a coin model)}$$

- Step 2: Specify a prior: $p(\theta)$
- Step 3:Compute the posterior

$$p(\theta|x_{1:n}) = \frac{p(x_{1:n}|\theta)p(\theta)}{p(x_{1:n})}$$

where $p(x_{1:n}) = \int p(x_{1:n}|\theta)p(\theta)d\theta$ is called Marginal Likelihood

$$p(\theta|x_{1:n}) \propto p(x_{1:n}|\theta)p(\theta)$$

Bayesian Learning
(Bayesian Learning Procedure)

- Step 1: Given n data, $x_{1:n} = \{x_1, x_2, \ldots, x_n\}$, write down the expression for the likelihood:

$$p(x_{1:n}|\theta) = \theta^m(1-\theta)^{(n-m)}$$

- Step 2: Specify a prior: $p(\theta)$
  we know θ is continuous and it is in the range of $0 \leq \theta \leq 1$
  What is $P(\theta)$?
  We usually take $P(\theta) = Beta(\alpha, \beta)$

$$Beta(\alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)+\Gamma(\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1};$$

α and β are called hyper parameters

where $\Gamma(Z) = \int_0^{\inf} exp^x x^{Z-1} dx$

$log(Z) = \int_1^Z \frac{1}{X} dx$ and $E(\theta) = \frac{\alpha}{\alpha+\beta}$

Bayesian Learning
(Bayesian Learning Procedure)

- Step 3: Compute the posterior:

$$p(\theta|x_{1:n}) \propto p(x_{1:n}|\theta)p(\theta)$$
$$p(\theta|x_{1:n}) \propto \theta^m(1-\theta)^{n-m}\theta^{\alpha-1}(1-\theta)^{\beta-1}$$
$$p(\theta|x_{1:n}) \propto \theta^{m+\alpha-1}(1-\theta)^{n-m+\beta-1}$$
$$p(\theta|x_{1:n}) \propto \theta^{\alpha'}(1-\theta)^{\beta'}$$

where $\alpha' = m + \alpha - 1$ and $\beta' = n - m + \beta - 1$

$$\therefore p(\theta|x_{1:n}) = \frac{\Gamma(\alpha'+\beta')}{\Gamma(\alpha')+\Gamma(\beta')}\theta^{\alpha'}(1-\theta)^{\beta'}$$

Bayesian Learning

Suppose we observe the data $x_{1:6} = \{1, 1, 1, 1, 1, 1\}$, where each x_i comes from the same Bernoulli distribution (i.e iid). What is a good guess of θ ?

- Compute the posterior and use it mean as the estimate
  - Using a prior Beta(2,2)
  - Using a prior Beta(1, 0.01)

Bayesian Learning
with Gaussian Prior for a coin model

Intractable Integration is a bottleneck in Bayesian Learning

- Let assume the likelihood is Bernoulli

$$p(x_{1:n}|\theta) = \prod_{i=1}^{n} P(x_i|\theta)$$
$$= \prod_{i=1}^{n} \theta^{x_i}(1-\theta)^{1-x_i}$$

- Let also assume the θ has a Gaussian Prior (typically we take the $\mu \approx 0$

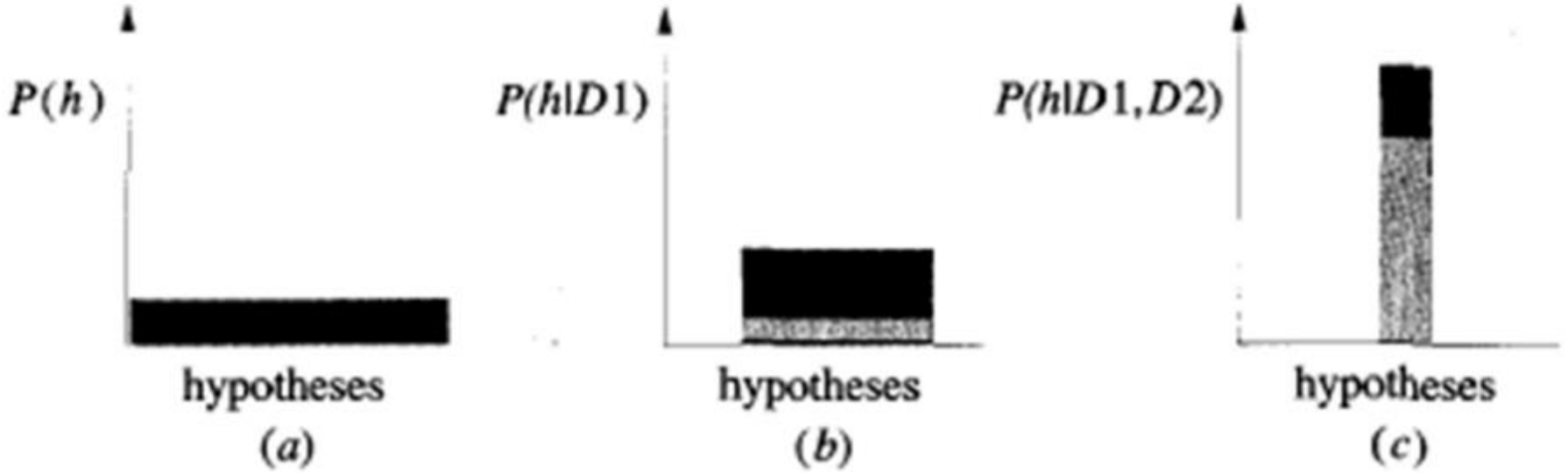
$$p(\theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(\theta-\mu)'(\theta-\mu)\right)$$

- Then Posterior distribution is

$$p(\theta|x_{1:n}) = \frac{p(x_{1:n}|\theta)p(\theta)}{p(x_{1:n})}$$
$$z = p(x_{1:n}) = \int p(x_{1:n}|\theta)p(\theta)d\theta$$

- Z is known as Normalization Constant and will not easily be computed using the Numerical Techniques. Instead of exactly computing this value, Monte Carlo will approximate it.

Evolution of posterior probabilities P(h|D) with increasing training data.



(a) Uniform priors assign equal probability to each hypothesis.
As training data increases first to Dl (b), then to Dl ∧ D2 (c),

the posterior probability of inconsistent hypotheses becomes zero, while posterior probabilities increase for hypotheses remaining in the version space.

Prior can be based on:

- **Domain knowledge**: If prior information is available, use it to assign probabilities.
- **Non-informative priors**: If no prior knowledge exists, assign equal probabilities to all hypotheses (uniform prior).

Maximum Likelihood and Least-Squared Error Hypotheses

- Many learning approaches such as neural network learning, linear regression, and polynomial curve fitting try to learn a continuous-valued target function.

- **Under certain assumptions any learning algorithm that minimizes the squared error between the output hypothesis predictions and the training data will output a *MAXIMUM LIKELIHOOD HYPOTHESIS*.**

- The significance of this result is that it provides a Bayesian justification (under certain assumptions) for many neural network and other curve fitting methods that attempt to minimize the sum of squared errors over the training data.

Bayes Optimal Classifier

- Normally we consider:
  - What is the most probable ***hypothesis*** given the training data?
- We can also consider:
  - what is the most probable ***classification*** of the new instance given the training data?

- Consider a hypothesis space containing three hypotheses, hl, h2, and h3.
  - Suppose that the posterior probabilities of these hypotheses given the training data are .4, .3, and .3 respectively.
  - Thus, hl is the MAP hypothesis.
  - Suppose a new instance x is encountered, which is classified positive by *hl,* but negative by *h2* and h3.
  - Taking all hypotheses into account, the probability that x is positive is .4 (the probability associated with *h1),* and the probability that it is negative is therefore .6.
  - The most probable classification (negative) in this case is different from the classification generated by the MAP hypothesis.

Bayes Optimal Classifier

- The most probable classification of the new instance is obtained by combining the predictions of all hypotheses, weighted by their posterior probabilities.
- If the possible classification of the new example can take on any value v_j from some set V, then the probability $P(v_j \mid D)$ that the correct classification for the new instance is v_j :

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

- **Bayes optimal classification:**

$$\underset{v_j \in V}{\operatorname{argmax}} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D)$$

Bayes Optimal Classifier - Ex

$$P(h_1|D) = .4, \; P(\ominus|h_1) = 0, \; P(\oplus|h_1) = 1$$
$$P(h_2|D) = .3, \; P(\ominus|h_2) = 1, \; P(\oplus|h_2) = 0$$
$$P(h_3|D) = .3, \; P(\ominus|h_3) = 1, \; P(\oplus|h_3) = 0$$

Probabilities:

$$\sum_{h_i \in H} P(\oplus|h_i) P(h_i|D) = .4$$

$$\sum_{h_i \in H} P(\ominus|h_i) P(h_i|D) = .6$$

Result:

$$\underset{v_j \in \{\oplus, \ominus\}}{\operatorname{argmax}} \sum_{h_i \in H} P(v_j|h_i) P(h_i|D) = \ominus$$

Bayes Optimal Classifier

- Although the Bayes optimal classifier obtains the best performance that can be achieved from the given training data, it can be quite costly to apply.
  - The expense is due to the fact that it computes the posterior probability for every hypothesis in H and then combines the predictions of each hypothesis to classify each new instance.

- An alternative, less optimal method is the Gibbs algorithm:

1. Choose a hypothesis ***h*** from H at random, according to the posterior probability distribution over H.
2. Use ***h*** to predict the classification of the next instance ***x.***

Naive Bayes Classifier

- One highly practical Bayesian learning method is Naive Bayes Learner (***Naive Bayes Classifier).***
- The naive Bayes classifier applies to learning tasks where each instance ***x*** is described by a conjunction of attribute values and where the target function f (x) can take on any value from some finite set V.
- A set of training examples is provided, and a new instance is presented, described by the tuple of attribute values $(a_1, a_2 \ldots a_n)$.
- The learner is asked to predict the target value (classification), for this new instance.

Naive Assumption

In Naive Bayes, the assumption of feature independence simplifies the calculation of $P(X|C)$. If there are multiple features $X = (X_1, X_2, ..., X_n)$, the model assumes that these features are conditionally independent given the class C. This leads to the following simplification:

$$P(X|C) = P(X_1|C)P(X_2|C)...P(X_n|C)$$

Thus, the classifier calculates the likelihood of the features given the class as the product of the individual likelihoods of each feature.

Classification Rule

To classify a new instance, the Naive Bayes classifier computes the posterior probability for each class C using Bayes' Theorem. The class with the highest posterior probability is chosen as the predicted class. Mathematically, we compute:

$$\hat{C} = \arg\max_C P(C) \prod_{i-1}^{n} P(X_i|C)$$

Naive Bayes Classifier - Ex

Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Naive Bayes Classifier -Ex

- New instance to classify:

  (Outlook=sunny, Temperature=cool, Humidity=high, Wind=strong)
- Our task is to predict the target value (yes or no) of the target concept *PlayTennis* for this new instance.

$$v_{NB} = \underset{v_j \in \{yes,no\}}{\text{argmax}} \; P(v_j) \prod_i P(a_i|v_j)$$

$$= \underset{v_j \in \{yes,no\}}{\text{argmax}} \; P(v_j) \; \mathbf{P(Outlook=sunny|v_j)\; P(Temperature=cool|v_j)\; P(Humidity=high|v_j)\; P(Wind=strong|v_j)}$$

Naive Bayes Classifier -Ex

- P(P1ayTennis = yes) = 9/14 = .64
- P(P1ayTennis = no) = 5/14 = .36

$$P(yes)\ P(sunny|yes)\ P(cool|yes)\ P(high|yes)\ P(strong|yes) = .0053$$

$$P(no)\ P(sunny|no)\ P(cool|no)\ P(high|no)\ P(strong|no) = .0206$$

➔ Thus, the naive Bayes classifier assigns the target value ***PlayTennis*** = ***no*** to this new instance, based on the probability estimates learned from the training data.

- Furthermore, by normalizing the above quantities to sum to one we can calculate the conditional probability that the target value is ***no,*** given the observed attribute values.

$$.0206 / (.0206 + .0053) = .795$$

Practical tips – Laplace smoothing

- **Laplace Smoothing** (also known as **additive smoothing**) is a technique used in probability and statistics to handle the problem of zero probabilities in categorical data.
- In **Naive Bayes classification**, Laplace smoothing ensures that no probability is ever exactly zero, which can otherwise cause issues during classification.

Why is Laplace Smoothing Needed?

In Naive Bayes, the probability of a class C given the feature X is computed as:

$$P(C|X) \propto P(C) \cdot \prod_{i-1}^{n} P(X_i|C)$$

If $P(X_i|C) = 0$ for any feature X_i, the entire product becomes zero, regardless of the other features. This happens when a feature X_i is absent from the training data for the given class C.

For example, if you're classifying documents and encounter a word in the test document that was never seen in the training data for a specific class, the likelihood for that class becomes zero, leading to incorrect or undefined results.

How Laplace Smoothing Works

Laplace smoothing adds a small positive constant (typically 1) to the count of each feature, preventing any probability from being zero.

Formula for Smoothed Probability

Without smoothing:

$$P(X_i|C) = \frac{\text{Count of } X_i \text{ in class } C}{\text{Total count of all features in class } C}$$

With Laplace smoothing (additive smoothing):

$$P(X_i|C) = \frac{\text{Count of } X_i \text{ in class } C + \alpha}{\text{Total count of all features in class } C + \alpha \cdot \text{Vocabulary size}}$$

Where:

- α is the smoothing parameter (commonly $\alpha = 1$).
- **Vocabulary size** is the total number of unique features (e.g., words) in the dataset.

Let's classify a document as either **Spam** or **Not Spam** using the Naive Bayes classifier.

Step 1: Prepare the Training Data

We have the following labeled dataset (documents and their corresponding labels):

Document	Label
"Buy cheap meds"	Spam
"Cheap car insurance"	Spam
"Meeting tomorrow at 10 AM"	Not Spam
"Project deadline extended"	Not Spam

We also assume the vocabulary (unique words in all documents) is:

$$\text{Vocabulary: } \{\text{buy, cheap, meds, car, insurance, meeting, tomorrow, project, deadline, extended}\}$$

Step 2: Calculate Priors

The prior probability of a class is the fraction of documents in that class:

$$P(\text{Spam}) = \frac{\text{Number of Spam Documents}}{\text{Total Documents}} = \frac{2}{4} = 0.5$$

$$P(\text{Not Spam}) = \frac{\text{Number of Not Spam Documents}}{\text{Total Documents}} = \frac{2}{4} = 0.5$$

Step 3: Calculate Likelihoods

The likelihood $P(\text{word}|\text{class})$ is calculated using word frequencies in each class. To avoid zero probabilities, we'll use **Laplace Smoothing**:

$$P(\text{word}|\text{class}) = \frac{\text{Count of word in class} + 1}{\text{Total words in class} + \text{Vocabulary size}}$$

Step 3: Calculate Likelihoods

The likelihood $P(\text{word}|\text{class})$ is calculated using word frequencies in each class. To avoid zero probabilities, we'll use **Laplace Smoothing**:

$$P(\text{word}|\text{class}) = \frac{\text{Count of word in class} + 1}{\text{Total words in class} + \text{Vocabulary size}}$$

For Spam:

- Total words in Spam: 6 ("buy", "cheap", "meds", "cheap", "car", "insurance")
- Vocabulary size: 10
- Example calculations:
  - $P(\text{buy}|\text{Spam}) = \frac{1+1}{6+10} = \frac{2}{16} = 0.125$
  - $P(\text{cheap}|\text{Spam}) = \frac{2+1}{6+10} = \frac{3}{16} = 0.1875$
  - Similarly, compute for other words.

For Not Spam:

- Total words in Not Spam: 8 ("meeting", "tomorrow", "at", "10", "AM", "project", "deadline", "extended")
- Vocabulary size: 10
- Example calculations:
  - $P(\text{meeting}|\text{Not Spam}) = \frac{1+1}{8+10} = \frac{2}{18} = 0.111$
  - $P(\text{cheap}|\text{Not Spam}) = \frac{0+1}{8+10} = \frac{1}{18} = 0.055$

Step 4: Classify a New Document

Suppose we want to classify the document: "**cheap meds project**".

Calculate Posterior for Spam:

$$P(\text{Spam}|\text{cheap, meds, project}) \propto P(\text{Spam}) \cdot P(\text{cheap}|\text{Spam}) \cdot P(\text{meds}|\text{Spam}) \cdot P(\text{project}|\text{Spam})$$

Substitute values:

$$P(\text{Spam}) = 0.5,\ P(\text{cheap}|\text{Spam}) = 0.1875,\ P(\text{meds}|\text{Spam}) = 0.125,\ P(\text{project}|\text{Spam}) = 0.0625$$

$$P(\text{Spam}|\text{cheap, meds, project}) \propto 0.5 \cdot 0.1875 \cdot 0.125 \cdot 0.0625 = 0.000732$$

Calculate Posterior for Not Spam:

$$P(\text{Not Spam}|\text{cheap, meds, project}) \propto P(\text{Not Spam}) \cdot P(\text{cheap}|\text{Not Spam}) \cdot P(\text{meds}|\text{Not Spam}) \cdot P(\text{project}|\text{Not Spam})$$

Substitute values:

$$P(\text{Not Spam}) = 0.5,\ P(\text{cheap}|\text{Not Spam}) = 0.055,\ P(\text{meds}|\text{Not Spam}) = 0.055,\ P(\text{project}|\text{Not Spam}) = 0.111$$

$$P(\text{Not Spam}|\text{cheap, meds, project}) \propto 0.5 \cdot 0.055 \cdot 0.055 \cdot 0.111 = 0.000168$$

Comparison: Since $P(\text{Spam}|\text{cheap, meds, project}) > P(\text{Not Spam}|\text{cheap, meds, project})$,
the document is classified as **Spam**.

Conclusion

Using the Naive Bayes classifier, we predicted the document "**cheap meds project**" as **Spam**.