Furthermore, a significant amount of domain knowledge is used in the design of the similarity function, because only a single example is available. This single example can be more appropriately viewed as a user requirement rather than a historical rating, because it is specified interactively. In knowledge-based systems, there is less emphasis on using historical data or ratings. Like the Rocchio method, such methods are also interactive, although the interactivity is far more sophisticated in case-based systems.

4.4.3 Bayes Classifier

The Bayes classifier is discussed in section 3.4 of Chapter 3 in collaborative filtering. However, the discussion in Chapter 3 is a non-standard use of the Bayes model in which the missing entries are predicted from the specified ones. In the context of content-based recommender systems, the problem translates to a more conventional use of the Bayes model for text classification. Therefore, we will revisit the Bayes model in the context of text classification.

In this case, we have a set $\mathcal{D}_L$ containing the training documents, and a set $\mathcal{D}_U$ containing the test documents. For ease in discussion, we will assume that the labels are binary in which users specify either a like or a dislike rating as $+1$ or -1, respectively for each of the training documents in $\mathcal{D}_L$. It is, however, relatively easy to generalize this classifier to the case where the ratings take on more than two values.

As before, assume that the rating of the ith document in $\mathcal{D}_L$ is denoted by $c_i \in \{-1, 1\}$. Therefore, this labeled set represents the user profile. There are two models that are commonly used in text data, which correspond to the Bernoulli and the multinomial models, respectively. In the following, we will discuss only the Bernoulli model. The multinomial model is discussed in detail in [22].

In the Bernoulli model, the frequencies of the words are ignored, and only the presence or absence of the word in the document is considered. Therefore, each document is treated as a binary vector of d words containing only values of 0 and 1. Consider a target document $\overline{X} \in \mathcal{D}_U$, which might correspond to the description of an item. Assume that the d binary features in $\overline{X}$ are denoted by $(x_1 \ldots x_d)$. Informally, we would like to determine $P(\text{Active user likes } \overline{X} | x_1 \ldots x_d)$. Here, each x_i is a 0-1 value, corresponding to whether or not the ith word is present in the document $\overline{X}$. Then, if the class (binary rating) of $\overline{X}$ is denoted by $c(\overline{X})$, this is equivalent to determining the value of $P(c(\overline{X}) = 1 | x_1 \ldots x_d)$. By determining both $P(c(\overline{X}) = 1 | x_1 \ldots x_d)$ and $P(c(\overline{X}) = -1 | x_1 \ldots x_d)$ and selecting the larger of the two, one can determine whether or not the active user likes $\overline{X}$. These expressions can be evaluated by using the Bayes rule and then applying a *naive assumption* as follows:

$$\begin{aligned} P(c(\overline{X}) = 1 | x_1 \ldots x_d) &= \frac{P(c(\overline{X}) = 1) \cdot P(x_1 \ldots x_d | c(\overline{X}) = 1)}{P(x_1 \ldots x_d)} \\ &\propto P(c(\overline{X}) = 1) \cdot P(x_1 \ldots x_d | c(\overline{X}) = 1) \\ &= P(c(\overline{X}) = 1) \cdot \prod_{i=1}^{d} P(x_i | c(\overline{X}) = 1) \quad \text{[Naive Assumption]} \end{aligned}$$

The naive assumption states that the occurrences of words in documents are conditionally independent events (on a specific class), and therefore one can replace $P(x_1 \ldots x_d | c(\overline{X}) = 1)$ with $\prod_{i=1}^{d} P(x_i | c(\overline{X}) = 1)$. Furthermore, the constant of proportionality is used in the first relationship because the denominator is independent of the class. Therefore, the denominator does not play any role in deciding between the relative order of the classes.

The denominator, however, does play a role in terms of ranking the propensity of *different items (documents)* to be liked by the user. This is relevant to the problem of *ranking* items for a specific user, in order of $P(c(\overline{X}) = 1|x_1 \ldots x_d)$.

In cases where such a ranking of the items is needed, the constant of proportionality is no longer irrelevant. This is particularly common in recommendation applications where it is not sufficient to determine the relative probabilities of items belonging to different rating values, but to actually rank them with respect to one another. In such cases, the constant of proportionality needs to be determined. Assume that the constant of proportionality in the relationship above is denoted by K. The constant of proportionality K can be obtained by using the fact that the sum of the probabilities of all possible instantiations of $c(\overline{X})$ should always be 1. Therefore, we have:

$$K \cdot \left[P(c(\overline{X}) = 1) \cdot \prod_{i=1}^{d} P(x_i|c(\overline{X}) = 1) + P(c(\overline{X}) = 1) \cdot \prod_{i=1}^{d} P(x_i|c(\overline{X}) = 1) \right] = 1$$

Therefore, we can derive the following value for K:

$$K = \frac{1}{P(c(\overline{X}) = 1) \cdot \prod_{i=1}^{d} P(x_i|c(\overline{X}) = 1) + P(c(\overline{X}) = -1) \cdot \prod_{i=1}^{d} P(x_i|c(\overline{X}) = -1)}$$

This approach is used to determine the probability of a user liking each possible item in $\mathcal{D}_U$. The items in $\mathcal{D}_U$ are then ranked according to this probability and presented to the user. These methods are particularly well suited to binary ratings. There are other ways of using the probability to estimate the predicted value of the ratings and rank the items when dealing with ratings that are not necessarily binary. Such methods are discussed in detail in section 3.4 of Chapter 3.

4.4.3.1 Estimating Intermediate Probabilities

The Bayes method requires the computation of intermediate probabilities such as $P(x_i|c(\overline{X}) = 1)$. So far, we have not yet discussed how these probabilities can be estimated in a data-driven manner. The main utility of the aforementioned Bayes rule is that it expresses the prediction probabilities in terms of other probabilities [e.g., $P(x_i|c(\overline{X}) = 1)$] that can be estimated more easily in a data-driven way. We reproduce the Bayes condition above:

$$P(c(\overline{X}) = 1|x_1 \ldots x_d) \propto P(c(\overline{X}) = 1) \cdot \prod_{i=1}^{d} P(x_i|c(\overline{X}) = 1)$$

$$P(c(\overline{X}) = -1|x_1 \ldots x_d) \propto P(c(\overline{X}) = -1) \cdot \prod_{i=1}^{d} P(x_i|c(\overline{X}) = -1)$$

In order to compute the Bayes probabilities, we need to estimate the probabilities on the right-hand side of the equations above. These include the prior class probabilities $P(c(\overline{X}) = 1)$ and $P(c(\overline{X}) = -1)$. Furthermore, the feature-wise conditional probabilities, such as $P(x_i|c(\overline{X}) = 1)$ and $P(x_i|c(\overline{X}) = -1)$, need to be estimated. The probability $P(c(\overline{X}) = 1)$ can be estimated as the fraction of positive training examples $\mathcal{D}_L^+$ in the labeled data $\mathcal{D}_L$. In order to reduce overfitting, Laplacian smoothing is performed by adding values proportional to a small parameter $\alpha > 0$ to the numerator and denominator.

$$P(c(\overline{X}) = 1) = \frac{|\mathcal{D}_L^+| + \alpha}{|\mathcal{D}_L| + 2 \cdot \alpha} \tag{4.10}$$

Table 4.1: Illustration of the Bayes method for a content-based system

Keyword ⇒ Song-Id ⇓	Drums	Guitar	Beat	Classical	Symphony	Orchestra	Like or Dislike
1	1	1	1	0	0	0	Dislike
2	1	1	0	0	0	1	Dislike
3	0	1	1	0	0	0	Dislike
4	0	0	0	1	1	1	Like
5	0	1	0	1	0	1	Like
6	0	0	0	1	1	0	Like
Test-1	0	0	0	1	0	0	?
Test-2	1	0	1	0	0	0	?

The value of $P(c(\overline{X}) = -1)$ is estimated in an exactly similar way. Furthermore, the conditional feature probability $P(x_i|c(\overline{X}) = 1)$ is estimated as the fraction of the instances in the positive class for which the ith feature takes on the value of x_i. Let $q^+(x_i)$ represent the number of instances in the positive class that take on the value of $x_i \in \{0, 1\}$ for the ith feature. Then, we can use a Laplacian smoothing parameter $\beta > 0$ to estimate the probability as follows:

$$P(x_i|c(\overline{X}) = 1) = \frac{q^+(x_i) + \beta}{|\mathcal{D}_L^+| + 2 \cdot \beta} \tag{4.11}$$

A similar approach can be used to estimate $P(x_i|c(\overline{X}) = -1)$. Note that the Laplacian smoothing is helpful for cases where little training data are available. In the extreme case, where $\mathcal{D}_L^+$ is empty, the probability $P(x_i|c(\overline{X}) = 1)$ would be (appropriately) estimated to be 0.5 as a kind of prior belief. Without smoothing, such an estimation would be indeterminate, because both the numerator and denominator of the ratio would be 0. Laplacian smoothing, like many regularization methods, can be interpreted in terms of the greater importance of prior beliefs when the amount of training data is limited. Although we have presented the aforementioned estimation for the case of binary ratings, it is relatively easy to generalize the estimation when there are k distinct values of the ratings. A similar type of estimation is discussed in the context of collaborative filtering in section 3.4 of Chapter 3.

4.4.3.2 Example of Bayes Model

We provide an example of the use of the Bayes model for a set of 6 training examples and two test examples. In Table 4.1, the columns correspond to features representing properties of various songs. The user like or dislike is illustrated in the final column of the table. Therefore, the final column can be viewed as the rating. The first 6 rows correspond to the training examples, which correspond to the user profile. The final pair of rows correspond to two candidate music tracks that need to be ranked for the specific user at hand. In machine learning parlance, these rows are also referred to as test instances. Note that the final (dependent variable) column is specified only for the training rows because the user like or dislike (ratings) are not known for the test rows. These values need to be predicted.

By examining the features in Table 4.1, it becomes immediately evident that the first three features (columns) might often occur in many popular music genres such as rock music, whereas the final three features typically occur in classical music. The user profile, represented by Table 4.1 clearly seems to suggest a preference for classical music over rock

music. Similarly, among the test examples, only the first of the two examples seems to match the user's interests. Let us examine how the Bayes approach is able to derive this fact in a data-driven way. For ease in computation, we will assume that Laplacian smoothing is not used, although it is important to use such smoothing methods in real applications.

By using the Bayes model, we can derive the conditional probabilities for likes and dislikes based on the observed features of the test examples:

$$\begin{aligned} P(\text{Like}|\,\textit{Test-1}) &\propto 0.5 \prod_{i=1}^{6} P(\text{Like}|x_i) \\ &= (0.5) \cdot \frac{3}{4} \cdot \frac{2}{2} \cdot \frac{3}{4} \cdot \frac{3}{3} \cdot \frac{1}{4} \cdot \frac{1}{3} \\ &= \frac{3}{128} \\ P(\text{Dislike}|\,\textit{Test-1}) &\propto 0.5 \prod_{i=1}^{6} P(\text{Dislike}|x_i) \\ &= (0.5) \cdot \frac{1}{4} \cdot \frac{0}{2} \cdot \frac{1}{4} \cdot \frac{0}{3} \cdot \frac{3}{4} \cdot \frac{2}{3} \\ &= 0 \end{aligned}$$

By normalizing the two probabilities to sum to 1, we obtain the result that $P(\text{Like}|\,\textit{Test-1})$ is 1 and $P(\text{Dislike}|\,\textit{Test-1})$ is 0. In the case of *Test-2*, exactly the opposite result is obtained where $P(\text{Like}|\,\textit{Test-2})$ is 0. Therefore, *Test-1* should be recommended to the active user over *Test-2*. This is the same result that we obtained on visual inspection of this example.

When Laplacian smoothing is used, we will not obtain such binary probability values for the various classes, although one of the classes will obtain a much higher probability than the other. In such cases, all the test examples can be ranked in order of their predicted probability of a "Like" and recommended to the user. Laplacian smoothing is advisable because a single 0-value in the product-wise form of the expression on the right-hand side of the Bayes rule can result in a conditional probability value of 0.

4.4.4 Rule-based Classifiers

Rule-based classifiers can be designed in a variety of ways, including leave-one-out methods, as well as associative methods. A detailed description of the various types of rule-based classifiers is provided in [18, 22]. In the following, we will discuss only associative classifiers because they are based on the simple principles of association rules. A discussion of rule-based methods is provided in section 3.3 of Chapter 3. Refer to that section for the basic definitions of association rules and their measures, such as *support* and *confidence*. The support of a rule defines the fraction of rows satisfying both the antecedent and the consequent of a rule. The confidence of a rule is the fraction of rows satisfying the consequent, from the rows already known to satisfy the antecedent. The concept of a row "satisfying" the antecedent or consequent is described in more detail below.

Rule-based classifiers in content-based systems are similar to rule-based classifiers in collaborative filtering. In the item-item rules of collaborative filtering, both the antecedents and consequents of rules correspond to ratings of items. The main difference is that the antecedents of the rules in collaborative filtering correspond[3] to the ratings of various items,

[3]A different approach in collaborative filtering is to leverage user-user rules. For user-user rules, the antecedents and consequents may both contain the ratings of specific users. Refer to section 3.3 of Chapter 3.