3 *Bayesian Decision Theory*

We discuss probability theory as the framework for making decisions under uncertainty. In classification, Bayes' rule is used to calculate the probabilities of the classes. We generalize to discuss how we can make rational decisions among multiple actions to minimize expected risk. We also discuss learning association rules from data.

3.1 Introduction

PROGRAMMING COMPUTERS to make inference from data is a cross between statistics and computer science, where statisticians provide the mathematical framework of making inference from data and computer scientists work on the efficient implementation of the inference methods.

Data comes from a process that is not completely known. This lack of knowledge is indicated by modeling the process as a random process. Maybe the process is actually deterministic, but because we do not have access to complete knowledge about it, we model it as random and use probability theory to analyze it. At this point, it may be a good idea to jump to the appendix and review basic probability theory before continuing with this chapter.

Tossing a coin is a random process because we cannot predict at any toss whether the outcome will be heads or tails—that is why we toss coins, or buy lottery tickets, or get insurance. We can only talk about the probability that the outcome of the next toss will be heads or tails. It may be argued that if we have access to extra knowledge such as the exact composition of the coin, its initial position, the force and its direction that is applied to the coin when tossing it, where and how it is caught, and so forth, the exact outcome of the toss can be predicted.

UNOBSERVABLE VARIABLES
OBSERVABLE VARIABLE

The extra pieces of knowledge that we do not have access to are named the *unobservable variables*. In the coin tossing example, the only *observable variable* is the outcome of the toss. Denoting the unobservables by $\mathbf{z}$ and the observable as x, in reality we have

$$x = f(\mathbf{z})$$

where $f(\cdot)$ is the deterministic function that defines the outcome from the unobservable pieces of knowledge. Because we cannot model the process this way, we define the outcome X as a random variable drawn from a probability distribution $P(X = x)$ that specifies the process.

The outcome of tossing a coin is heads or tails, and we define a random variable that takes one of two values. Let us say $X = 1$ denotes that the outcome of a toss is heads and $X = 0$ denotes tails. Such X are Bernoulli-distributed where the parameter of the distribution p_o is the probability that the outcome is heads:

$$P(X = 1) = p_o \text{ and } P(X = 0) = 1 - P(X = 1) = 1 - p_o$$

Assume that we are asked to predict the outcome of the next toss. If we know p_o, our prediction will be heads if $p_o > 0.5$ and tails otherwise. This is because if we choose the more probable case, the probability of error, which is 1 minus the probability of our choice, will be minimum. If this is a fair coin with $p_o = 0.5$, we have no better means of prediction than choosing heads all the time or tossing a fair coin ourselves!

SAMPLE

If we do not know $P(X)$ and want to estimate this from a given sample, then we are in the realm of statistics. We have a *sample*, $\mathcal{X}$, containing examples drawn from the probability distribution of the observables x^t, denoted as $p(x)$. The aim is to build an approximator to it, $\hat{p}(x)$, using the sample $\mathcal{X}$.

In the coin tossing example, the sample contains the outcomes of the past N tosses. Then using $\mathcal{X}$, we can estimate p_o, which is the parameter that uniquely specifies the distribution. Our estimate of p_o is

$$\hat{p}_o = \frac{\#\{\text{tosses with outcome heads}\}}{\#\{\text{tosses}\}}$$

Numerically using the random variables, x^t is 1 if the outcome of toss t is heads and 0 otherwise. Given the sample {heads, heads, heads, tails, heads, tails, tails, heads, heads}, we have $\mathcal{X} = \{1, 1, 1, 0, 1, 0, 0, 1, 1\}$ and the estimate is

$$\hat{p}_o = \frac{\sum_{t=1}^{N} x^t}{N} = \frac{6}{9}$$

3.2 Classification

We discussed credit scoring in section 1.2.2, where we saw that in a bank, according to their past transactions, some customers are low-risk in that they paid back their loans and the bank profited from them and other customers are high-risk in that they defaulted. Analyzing this data, we would like to learn the class "high-risk customer" so that in the future, when there is a new application for a loan, we can check whether that person obeys the class description or not and thus accept or reject the application. Using our knowledge of the application, let us say that we decide that there are two pieces of information that are observable. We observe them because we have reason to believe that they give us an idea about the credibility of a customer. Let us say, for example, we observe customer's yearly income and savings, which we represent by two random variables X_1 and X_2.

It may again be claimed that if we had access to other pieces of knowledge such as the state of economy in full detail and full knowledge about the customer, his or her intention, moral codes, and so forth, whether someone is a low-risk or high-risk customer could have been deterministically calculated. But these are nonobservables and with what we can observe, the credibility of a customer is denoted by a Bernoulli random variable C conditioned on the observables $\mathbf{X} = [X_1, X_2]^T$ where $C = 1$ indicates a high-risk customer and $C = 0$ indicates a low-risk customer. Thus if we know $P(C|X_1, X_2)$, when a new application arrives with $X_1 = x_1$ and $X_2 = x_2$, we can

$$\text{choose} \begin{cases} C = 1 & \text{if } P(C = 1|x_1, x_2) > 0.5 \\ C = 0 & \text{otherwise} \end{cases}$$

or equivalently

$$\text{choose} \begin{cases} C = 1 & \text{if } P(C = 1|x_1, x_2) > P(C = 0|x_1, x_2) \\ C = 0 & \text{otherwise} \end{cases} \tag{3.1}$$

The probability of error is $1 - \max(P(C = 1|x_1, x_2), P(C = 0|x_1, x_2))$. This example is similar to the coin tossing example except that here, the Bernoulli random variable C is conditioned on two other observable variables. Let us denote by $\mathbf{x}$ the vector of observed variables, $\mathbf{x} = [x_1, x_2]^T$.

BAYES' RULE

The problem then is to be able to calculate $P(C|\mathbf{x})$. Using *Bayes' rule*, it can be written as

$$P(C|\mathbf{x}) = \frac{P(C)p(\mathbf{x}|C)}{p(\mathbf{x})} \tag{3.2}$$

PRIOR PROBABILITY

$P(C = 1)$ is called the *prior probability* that C takes the value 1, which in our example corresponds to the probability that a customer is high-risk, regardless of the $\boldsymbol{x}$ value. It is called the prior probability because it is the knowledge we have as to the value of C *before* looking at the observables $\boldsymbol{x}$, satisfying

$$P(C = 0) + P(C = 1) = 1$$

CLASS LIKELIHOOD

$p(\boldsymbol{x}|C)$ is called the *class likelihood* and is the conditional probability that an event belonging to C has the associated observation value $\boldsymbol{x}$. In our case, $p(x_1, x_2|C = 1)$ is the probability that a high-risk customer has his or her $X_1 = x_1$ and $X_2 = x_2$. It is what the data tells us regarding the class.

EVIDENCE

$p(\boldsymbol{x})$, the *evidence*, is the marginal probability that an observation $\boldsymbol{x}$ is seen, regardless of whether it is a positive or negative example.

$$p(\boldsymbol{x}) = \sum_{C} p(\boldsymbol{x}, C) = p(\boldsymbol{x}|C = 1)P(C = 1) + p(\boldsymbol{x}|C = 0)P(C = 0) \tag{3.3}$$

POSTERIOR PROBABILITY

Combining the prior and what the data tells us using Bayes' rule, we calculate the *posterior probability* of the concept, $P(C|\boldsymbol{x})$, *after* having seen the observation, $\boldsymbol{x}$.

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}$$

Because of normalization by the evidence, the posteriors sum up to 1:

$$P(C = 0|\boldsymbol{x}) + P(C = 1|\boldsymbol{x}) = 1$$

Once we have the posteriors, we decide by using equation 3.1. For now, we assume that we know the prior and likelihoods; in later chapters, we discuss how to estimate $P(C)$ and $p(\boldsymbol{x}|C)$ from a given training sample.

In the general case, we have K mutually exclusive and exhaustive classes; $C_i, i = 1, \ldots, K$; for example, in optical digit recognition, the input is a bitmap image and there are ten classes. We have the prior probabilities satisfying

$$P(C_i) \geq 0 \text{ and } \sum_{i=1}^{K} P(C_i) = 1 \tag{3.4}$$

$p(\boldsymbol{x}|C_i)$ is the probability of seeing $\boldsymbol{x}$ as the input when it is known to belong to class C_i. The posterior probability of class C_i can be calculated as

$$P(C_i|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|C_i)P(C_i)}{p(\boldsymbol{x})} = \frac{p(\boldsymbol{x}|C_i)P(C_i)}{\sum_{k=1}^{K} p(\boldsymbol{x}|C_k)P(C_k)} \tag{3.5}$$

BAYES' CLASSIFIER

and for minimum error, the *Bayes' classifier* chooses the class with the highest posterior probability; that is, we

$$\text{choose } C_i \text{ if } P(C_i|\boldsymbol{x}) = \max_k P(C_k|\boldsymbol{x}) \tag{3.6}$$

3.3 Losses and Risks

It may be the case that decisions are not equally good or costly. A financial institution when making a decision for a loan applicant should take into account the potential gain and loss as well. An accepted low-risk applicant increases profit, while a rejected high-risk applicant decreases loss. The loss for a high-risk applicant erroneously accepted may be different from the potential gain for an erroneously rejected low-risk applicant. The situation is much more critical and far from symmetry in other domains like medical diagnosis or earthquake prediction.

LOSS FUNCTION
EXPECTED RISK

Let us define action α_i as the decision to assign the input to class C_i and λ_{ik} as the *loss* incurred for taking action α_i when the input actually belongs to C_k. Then the *expected risk* for taking action α_i is

$$R(\alpha_i|\boldsymbol{x}) = \sum_{k=1}^{K} \lambda_{ik} P(C_k|\boldsymbol{x}) \tag{3.7}$$

and we choose the action with minimum risk:

$$\text{choose } \alpha_i \text{ if } R(\alpha_i|\boldsymbol{x}) = \min_k R(\alpha_k|\boldsymbol{x}) \tag{3.8}$$

0/1 LOSS

Let us define K actions $\alpha_i, i = 1, \ldots, K$, where α_i is the action of assigning $\boldsymbol{x}$ to C_i. In the special case of the *0/1 loss* case where

$$\lambda_{ik} = \begin{cases} 0 & \text{if } i = k \\ 1 & \text{if } i \neq k \end{cases} \tag{3.9}$$

all correct decisions have no loss and all errors are equally costly. The risk of taking action α_i is

$$\begin{aligned} R(\alpha_i|\boldsymbol{x}) &= \sum_{k=1}^{K} \lambda_{ik} P(C_k|\boldsymbol{x}) \\ &= \sum_{k \neq i} P(C_k|\boldsymbol{x}) \\ &= 1 - P(C_i|\boldsymbol{x}) \end{aligned}$$

because $\sum_k P(C_k|\boldsymbol{x}) = 1$. Thus to minimize risk, we choose the most probable case. In later chapters, for simplicity, we will always assume this case and choose the class with the highest posterior, but note that this is indeed a special case and rarely do applications have a symmetric, 0/1 loss. In the general case, it is a simple postprocessing to go from posteriors to risks and to take the action to minimize the risk.

In some applications, wrong decisions—namely, misclassifications—may have very high cost, and it is generally required that a more complex—for example, manual—decision is made if the automatic system has low certainty of its decision. For example, if we are using an optical digit recognizer to read postal codes on envelopes, wrongly recognizing the code causes the envelope to be sent to a wrong destination.

REJECT In such a case, we define an additional action of *reject* or *doubt*, α_{K+1}, with $\alpha_i, i = 1, \ldots, K$, being the usual actions of deciding on classes $C_i, i = 1, \ldots, K$ (Duda, Hart, and Stork 2001).

A possible loss function is

$$\lambda_{ik} = \begin{cases} 0 & \text{if } i = k \\ \lambda & \text{if } i = K+1 \\ 1 & \text{otherwise} \end{cases} \tag{3.10}$$

where $0 < \lambda < 1$ is the loss incurred for choosing the $(K+1)$st action of reject. Then the risk of reject is

$$R(\alpha_{K+1}|\boldsymbol{x}) = \sum_{k=1}^{K} \lambda P(C_k|\boldsymbol{x}) = \lambda \tag{3.11}$$

and the risk of choosing class C_i is

$$R(\alpha_i|\boldsymbol{x}) = \sum_{k \neq i} P(C_k|\boldsymbol{x}) = 1 - P(C_i|\boldsymbol{x}) \tag{3.12}$$

The optimal decision rule is to

$$\begin{array}{lll} \text{choose } C_i & \text{if } R(\alpha_i|\boldsymbol{x}) < R(\alpha_k|\boldsymbol{x}) \text{ for all } k \neq i \text{ and} \\ & R(\alpha_i|\boldsymbol{x}) < R(\alpha_{K+1}|\boldsymbol{x}) \\ \text{reject} & \text{if } R(\alpha_{K+1}|\boldsymbol{x}) < R(\alpha_i|\boldsymbol{x}), i = 1, \ldots, K \end{array} \tag{3.13}$$

Given the loss function of equation 3.10, this simplifies to

$$\begin{array}{lll} \text{choose } C_i & \text{if } P(C_i|\boldsymbol{x}) > P(C_k|\boldsymbol{x}) \text{ for all } k \neq i \text{ and} \\ & P(C_i|\boldsymbol{x}) > 1 - \lambda \\ \text{reject} & \text{otherwise} \end{array} \tag{3.14}$$

This whole approach is meaningful if $0 < \lambda < 1$: If $\lambda = 0$, we always reject; a reject is as good as a correct classification. If $\lambda \geq 1$, we never reject; a reject is as costly as, or costlier than, an error.

3.4 Discriminant Functions

DISCRIMINANT FUNCTIONS

Classification can also be seen as implementing a set of *discriminant functions*, $g_i(\boldsymbol{x}), i = 1, \ldots, K$, such that we

$$\text{choose } C_i \text{ if } g_i(\boldsymbol{x}) = \max_k g_k(\boldsymbol{x}) \tag{3.15}$$

We can represent the Bayes' classifier in this way by setting

$$g_i(\boldsymbol{x}) = -R(\alpha_i|\boldsymbol{x})$$

and the maximum discriminant function corresponds to minimum conditional risk. When we use the 0/1 loss function, we have

$$g_i(\boldsymbol{x}) = P(C_i|\boldsymbol{x})$$

or ignoring the common normalizing term, $p(\boldsymbol{x})$, we can write

$$g_i(\boldsymbol{x}) = p(\boldsymbol{x}|C_i)P(C_i)$$

DECISION REGIONS

This divides the feature space into K *decision regions* $\mathcal{R}_1, \ldots, \mathcal{R}_K$, where $\mathcal{R}_i = \{\boldsymbol{x}|g_i(\boldsymbol{x}) = \max_k g_k(\boldsymbol{x})\}$. The regions are separated by *decision boundaries*, surfaces in feature space where ties occur among the largest discriminant functions (see figure 3.1).

When there are two classes, we can define a single discriminant

$$g(\boldsymbol{x}) = g_1(\boldsymbol{x}) - g_2(\boldsymbol{x})$$

and we

$$\text{choose} \begin{cases} C_1 & \text{if } g(\boldsymbol{x}) > 0 \\ C_2 & \text{otherwise} \end{cases}$$

An example is a two-class learning problem where the positive examples can be taken as C_1 and the negative examples as C_2. When $K = 2$, the classification system is a *dichotomizer* and for $K \geq 3$, it is a *polychotomizer*.

DICHOTOMIZER
POLYCHOTOMIZER

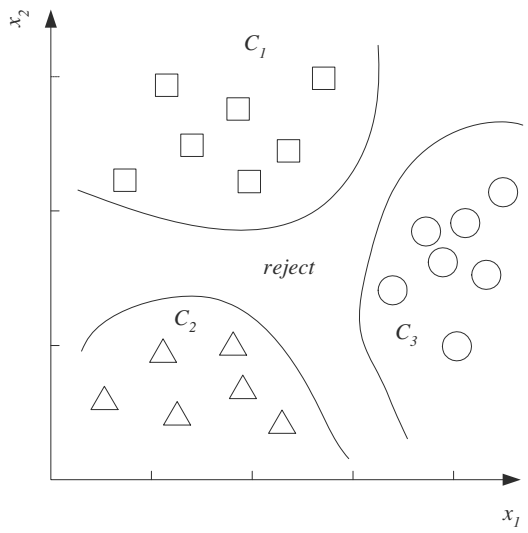

Figure 3.1 Example of decision regions and decision boundaries.

3.5 Utility Theory

UTILITY THEORY

In equation 3.7, we defined the expected risk and chose the action that minimizes expected risk. We now generalize this to *utility theory*, which is concerned with making rational decisions when we are uncertain about the state. Let us say that given evidence $\mathbf{x}$, the probability of state S_k is calculated as $P(S_k|\mathbf{x})$. We define a *utility function*, U_{ik}, which measures how good it is to take action α_i when the state is S_k. The *expected utility* is

UTILITY FUNCTION

EXPECTED UTILITY

$$EU(\alpha_i|\mathbf{x}) = \sum_k U_{ik} P(S_k|\mathbf{x}) \tag{3.16}$$

A rational decision maker chooses the action that maximizes the expected utility

$$\text{Choose } \alpha_i \text{ if } EU(\alpha_i|\mathbf{x}) = \max_j EU(\alpha_j|\mathbf{x}) \tag{3.17}$$

In the context of classification, decisions correspond to choosing one of the classes, and maximizing the expected utility is equivalent to minimizing expected risk. U_{ik} are generally measured in monetary terms, and this gives us a way to define the loss matrix λ_{ik} as well. For example, in

defining a reject option (equation 3.10), if we know how much money we will gain as a result of a correct decision, how much money we will lose on a wrong decision, and how costly it is to defer the decision to a human expert, depending on the particular application we have, we can fill in the correct values U_{ik} in a currency unit, instead of 0, λ, and 1, and make our decision so as to maximize expected earnings.

Note that maximizing expected utility is just one possibility; one may define other types of rational behavior, for example, minimizing the worst possible loss.

In the case of reject, we are choosing between the automatic decision made by the computer program and human decision that is costlier but assumed to have a higher probability of being correct. Similarly one can imagine a cascade of multiple automatic decision makers, which as we proceed are costlier but have a higher chance of being correct; we are going to discuss such cascades in chapter 17 where we talk about combining multiple learners.

3.6 Association Rules

ASSOCIATION RULE

An *association rule* is an implication of the form $X \rightarrow Y$ where X is the *antecedent* and Y is the *consequent* of the rule. One example of association rules is in *basket analysis* where we want to find the dependency between two items X and Y. The typical application is in retail where X and Y are items sold, as we discussed in section 1.2.1.

BASKET ANALYSIS

In learning association rules, there are three measures that are frequently calculated:

SUPPORT

- *Support* of the association rule $X \rightarrow Y$:

$$\text{Support}(X, Y) \equiv P(X, Y) = \frac{\#\{\text{customers who bought } X \text{ and } Y\}}{\#\{\text{customers}\}} \tag{3.18}$$

CONFIDENCE

- *Confidence* of the association rule $X \rightarrow Y$:

$$\begin{aligned} \text{Confidence}(X \rightarrow Y) \equiv P(Y|X) &= \frac{P(X, Y)}{P(X)} \\ &= \frac{\#\{\text{customers who bought } X \text{ and } Y\}}{\#\{\text{customers who bought } X\}} \end{aligned} \tag{3.19}$$

LIFT
INTEREST

- *Lift*, also known as *interest* of the association rule $X \rightarrow Y$:

$$\text{Lift}(X \rightarrow Y) = \frac{P(X,Y)}{P(X)P(Y)} = \frac{P(Y|X)}{P(Y)} \tag{3.20}$$

There are other measures as well (Omiecinski 2003), but these three, especially the first two, are the most widely known and used. *Confidence* is the conditional probability, $P(Y|X)$, which is what we normally calculate. To be able to say that the rule holds with enough confidence, this value should be close to 1 and significantly larger than $P(Y)$, the overall probability of people buying Y. We are also interested in maximizing the *support* of the rule, because even if there is a dependency with a strong confidence value, if the number of such customers is small, the rule is worthless. Support shows the statistical significance of the rule, whereas confidence shows the strength of the rule. The minimum support and confidence values are set by the company, and all rules with higher support and confidence are searched for in the database.

If X and Y are independent, then we expect lift to be close to 1; if the ratio differs—if $P(Y|X)$ and $P(Y)$ are different—we expect there to be a dependency between the two items: If the lift is more than 1, we can say that X makes Y more likely, and if the lift is less than 1, having X makes Y less likely.

These formulas can easily be generalized to more than two items. For example, $\{X, Y, Z\}$ is a three-item set, and we may look for a rule, such as $X, Z \rightarrow Y$, that is, $P(Y|X, Z)$. We are interested in finding all such rules having high enough support and confidence and because a sales database is generally very large, we want to find them by doing a small number of passes over the database. There is an efficient algorithm, called *Apriori* (Agrawal et al. 1996) that does this, which has two steps: (1) finding frequent itemsets, that is, those which have enough support, and (2) converting them to rules with enough confidence, by splitting the items into two, as items in the antecedent and items in the consequent:

APRIORI ALGORITHM

1. To find frequent itemsets quickly (without complete enumeration of all subsets of items), the Apriori algorithm uses the fact that for $\{X, Y, Z\}$ to be frequent (have enough support), all its subsets $\{X, Y\}$, $\{X, Z\}$, and $\{Y, Z\}$ should be frequent as well—adding another item can never increase support. That is, we only need to check for three-item sets all of whose two-item subsets are frequent; or, in other words, if a two-item set is known not to be frequent, all its supersets can be pruned and need not be checked.

We start by finding the frequent one-item sets and at each step, inductively, from frequent k-item sets, we generate candidate $k+1$-item sets and then do a pass over the data to check if they have enough support. The Apriori algorithm stores the frequent itemsets in a hash table for easy access. Note that the number of candidate itemsets will decrease very rapidly as k increases. If the largest itemset has n items, we need a total of $n+1$ passes over the data.

2. Once we find the frequent k-item sets, we need to convert them to rules by splitting the k items into two as antecedent and consequent. Just like we do for generating the itemsets, we start by putting a single consequent and $k-1$ items in the antecedent. Then, for all possible single consequents, we check if the rule has enough confidence and remove it if it does not.

   Note that for the same itemset, there may be multiple rules with different subsets as antecedent and consequent. Then, inductively, we check whether we can move another item from the antecedent to the consequent. Rules with more items in the consequent are more specific and more useful. Here, as in itemset generation, we use the fact that to be able to have rules with two items in the consequent with enough confidence, each of the two rules with single consequent by itself should have enough confidence; that is, we go from one consequent rules to two consequent rules and need not check for all possible two-term consequents (exercise 7).

HIDDEN VARIABLES

It should be kept in mind that a rule $X \rightarrow Y$ need not imply causality but just an association. In a problem, there may also be *hidden variables* whose values are never known through evidence. The advantage of using hidden variables is that the dependency structure can be more easily defined. For example, in basket analysis when we want to find the dependencies among items sold, let us say we know that there is a dependency among "baby food," "diapers," and "milk" in that a customer buying one of these is very much likely to buy the other two. Instead of representing dependencies among these three, we may designate a hidden node, "baby at home," as the hidden cause of the consumption of these three items. Graphical models that we will discuss in chapter 16 allow us to represent such hidden variables. When there are hidden nodes, their values are estimated given the values of observed nodes and filled in.

3.7 Notes

Making decisions under uncertainty has a long history, and over time humanity has looked at all sorts of strange places for evidence to remove the uncertainty: stars, crystal balls, and coffee cups. Reasoning from meaningful evidence using probability theory is only a few hundred years old; see Newman 1988 for the history of probability and statistics and some very early articles by Laplace, Bernoulli, and others who have founded the theory.

Russell and Norvig (1995) give an excellent discussion of utility theory and the value of information, also discussing the assignment of utilities in monetary terms. Shafer and Pearl 1990 is an early collection of articles on reasoning under uncertainty.

Association rules are successfully used in many data mining applications, and we see such rules on many Web sites that recommend books, movies, music, and so on. The algorithm is very simple and its efficient implementation on very large databases is critical (Zhang and Zhang 2002; Li 2006). Later, we will see in chapter 16 about graphical models how to generalize from association rules to concepts that need not be binary and where associations can be of different types, also allowing hidden variables.

3.8 Exercises

LIKELIHOOD RATIO

1. In a two-class problem, the *likelihood ratio* is
$$\frac{p(\boldsymbol{x}|C_1)}{p(\boldsymbol{x}|C_2)}$$
Write the discriminant function in terms of the likelihood ratio.

LOG ODDS

2. In a two-class problem, the *log odds* is defined as
$$\log \frac{P(C_1|\boldsymbol{x})}{P(C_2|\boldsymbol{x})}$$
Write the discriminant function in terms of the log odds.
3. In a two-class, two-action problem, if the loss function is $\lambda_{11} = \lambda_{22} = 0$, $\lambda_{12} = 10$, and $\lambda_{21} = 1$, write the optimal decision rule.
4. Propose a three-level cascade where when one level rejects, the next one is used as in equation 3.10. How can we fix the λ on different levels?
5. Somebody tosses a fair coin and if the result is heads, you get nothing; otherwise you get \$5. How much would you pay to play this game? What if the win is \$500 instead of \$5?

6. Generalize the confidence and support formulas for basket analysis to calculate k-dependencies, namely, $P(Y|X_1, \ldots, X_k)$.
7. Show that as we move an item from the consequent to the antecedent, confidence can never increase: confidence$(ABC \rightarrow D) \geq$ confidence$(AB \rightarrow CD)$.
8. Associated with each item sold in basket analysis, if we also have a number indicating how much the customer enjoyed the product, for example, in a scale of 0 to 10, how can you use this extra information to calculate which item to propose to a customer?
9. Show example transaction data where for the rule $X \rightarrow Y$:
   (a) Both support and confidence are high.
   (b) Support is high and confidence is low.
   (c) Support is low and confidence is high.
   (d) Both support and confidence are low.

3.9 References

Agrawal, R., H. Mannila, R. Srikant, H. Toivonen, and A. Verkamo. 1996. "Fast Discovery of Association Rules." In *Advances in Knowledge Discovery and Data Mining*, ed. U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, 307–328. Cambridge, MA: MIT Press.

Duda, R. O., P. E. Hart, and D. G. Stork. 2001. *Pattern Classification*, 2nd ed. New York: Wiley.

Li, J. 2006. "On Optimal Rule Discovery." *IEEE Transactions on Knowledge and Data Discovery* 18: 460–471.

Newman, J. R., ed. 1988. *The World of Mathematics*. Redmond, WA: Tempus.

Omiecinski, E. R. 2003. "Alternative Interest Measures for Mining Associations in Databases." *IEEE Transactions on Knowledge and Data Discovery* 15: 57–69.

Russell, S., and P. Norvig. 1995. *Artificial Intelligence: A Modern Approach*. New York: Prentice Hall.

Shafer, G., and J. Pearl, eds. 1990. *Readings in Uncertain Reasoning*. San Mateo, CA: Morgan Kaufmann.

Zhang, C., and S. Zhang. 2002. *Association Rule Mining: Models and Algorithms*. New York: Springer.