The purpose of mining over categorical and metric attributes, as described in Chapter 7, relies on exhaustive enumeration. There is another way of drawing useful information from quantitative association rules leading to optimization problems such as the following one.

**Problem**

Find one or several ranges of values (i.e., instantiations) for one or several attributes that maximize support (or confidence) under the constraint that confidence (or support) satisfies a given threshold condition.

The new type of problems has a more limited purpose since it focuses attention on only one rule. Moreover, it deals with optimization. It was introduced in a limited form by Fukuda et al. in [FMMT96a, b] and it was next extended by Rastogi et al. [RS98, 99, BRS99], who proposed several new algorithms. Optimizing association rules is an alternative to quantitative rule mining when the purpose is not to discover all rules, but to understand closely the behavior of a given rule. For example, suppose a company is selling products via the Internet. The customers are requested to fill in forms on the purchases, so that the company can maintain a purchase database. Now suppose that, for some reason, the company is interested in understanding the association:

age, sex, profession $\Rightarrow$ product.

Many questions may be formulated that can be translated into optimization problems. For instance:

1. What is the range of age values that maximizes the support (or confidence) of the rule, under the constraint that confidence (or support) is at least “a given threshold” and that:
   - sex = female,
   - profession = employee
   - product = “a given product.”

Note that the support of the rule for the range corresponds to the number of purchases made by the customers with ages in this range.
2. What are the q (at most) ranges of age values that maximize the support (or confidence) of the rule, under the constraint that confidence (or support) is at least: “a given threshold” and that:

- sex = female,
- profession = employee
- product = “a given product.”

Only one attribute appeared as uninstantiated in the problems previously stated. Similar problems involving several attributes can be formulated as well. For instance:

1'. What is the pair (ranges of age values, profession category) that maximizes the support (or confidence) of the rule, under the constraint that confidence (or support) is at least: “a given threshold” and that:

- sex = female,
- product = “a given product.”

Problem (2') can be derived in the same way as we derived problem (1').

In this chapter, we discuss three rule optimization problems: maximize confidence under the constraint that support exceeds a given threshold (MC\S problem); maximize support under the constraint that confidence exceeds a given threshold (MS\C problem); and maximize the gain measure (MG problem; see Sections 8.1.1, 8.2.1, and 8.4.1 for definitions of the gain measure). Each optimization problem gives rise to four problem instances, depending on how many attributes are left uninstantiated (problem dimensionality: e.g., one in problem 1, or two in problem 1') and on how many instantiations of the uninstantiated attributes are expected in the solution (solution multiplicity: e.g., one in problem 1, or q at most in problem 2 and 2'). Such a classification is motivated by the observed variations in computational complexity. A problem with d as dimensionality and q as solution multiplicity is denoted a “d-q-type problem.” In this chapter, we consider the following problems.

a. 1-1-type problems (e.g., problem 1). All problems of that type (MC\S, MS\C, and MG) are solved by algorithms performing with linear time complexity. Solving these problems is the topic of Section 8.1.

b. d-1-type problems (e.g., problem 1'). All problems of that type (MC\S, MS\C, and MG) are solved by algorithms performing with polynomial time complexity. The degree of polynomials depends on d (i.e., the number of uninstantiated attributes). Solving these problems is the topic of Section 8.2.

c. 1-q-type problems (e.g., problem 2). A weak form of MS\C and MG can be solved by algorithms performing with (low degree) polynomial time complexity. The strong form of MS\C is shown to be NP-hard in [RS98]. MC\S is also shown to be NP-hard even when the problem dimension is 1. Solving these problems is the topic of Section 8.3.