

is  $O(g)$ , where  $g$  is the total number of grid cells at the lowest level, which is usually much smaller than  $n$ .

Because STING uses a multiresolution approach to cluster analysis, the quality of STING clustering depends on the granularity of the lowest level of the grid structure. If the granularity is very fine, the cost of processing will increase substantially; however, if the bottom level of the grid structure is too coarse, it may reduce the quality of cluster analysis. Moreover, STING does not consider the spatial relationship between the children and their neighboring cells for construction of a parent cell. As a result, the shapes of the resulting clusters are isothetic, that is, all the cluster boundaries are either horizontal or vertical, and no diagonal boundary is detected. This may lower the quality and accuracy of the clusters despite the fast processing time of the technique.

### 10.5.2 CLIQUE: An Apriori-like Subspace Clustering Method

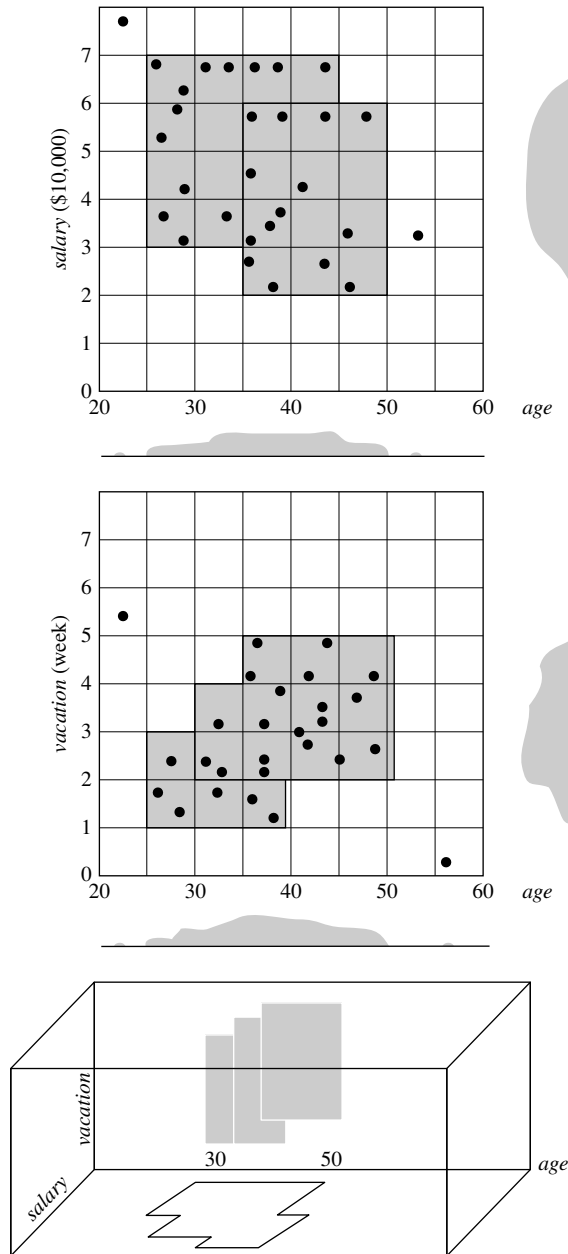
A data object often has tens of attributes, many of which may be irrelevant. The values of attributes may vary considerably. These factors can make it difficult to locate clusters that span the entire data space. It may be more meaningful to instead search for clusters within different *subspaces* of the data. For example, consider a health-informatics application where patient records contain extensive attributes describing personal information, numerous symptoms, conditions, and family history.

Finding a nontrivial group of patients for which all or even most of the attributes strongly agree is unlikely. In bird flu patients, for instance, the *age*, *gender*, and *job* attributes may vary dramatically within a wide range of values. Thus, it can be difficult to find such a cluster within the entire data space. Instead, by searching in subspaces, we may find a cluster of similar patients in a lower-dimensional space (e.g., patients who are similar to one other with respect to symptoms like high fever, cough but no runny nose, and aged between 3 and 16).

**CLIQUE** (CLustering In QUEst) is a simple grid-based method for finding density-based clusters in subspaces. CLIQUE partitions each dimension into nonoverlapping intervals, thereby partitioning the entire embedding space of the data objects into cells. It uses a density threshold to identify *dense* cells and *sparse* ones. A cell is dense if the number of objects mapped to it exceeds the density threshold.

The main strategy behind CLIQUE for identifying a candidate search space uses the monotonicity of dense cells with respect to dimensionality. This is based on the *Apriori property* used in frequent pattern and association rule mining (Chapter 6). In the context of clusters in subspaces, the monotonicity says the following. A  $k$ -dimensional cell  $c$  ( $k > 1$ ) can have at least  $l$  points only if every  $(k - 1)$ -dimensional projection of  $c$ , which is a cell in a  $(k - 1)$ -dimensional subspace, has at least  $l$  points. Consider Figure 10.20, where the embedding data space contains three dimensions: *age*, *salary*, and *vacation*. A 2-D cell, say in the subspace formed by *age* and *salary*, contains  $l$  points only if the projection of this cell in every dimension, that is, *age* and *salary*, respectively, contains at least  $l$  points.

CLIQUE performs clustering in two steps. In the first step, CLIQUE partitions the  $d$ -dimensional data space into nonoverlapping rectangular units, identifying the dense units among these. CLIQUE finds dense cells in all of the subspaces. To do so,



**Figure 10.20** Dense units found with respect to *age* for the dimensions *salary* and *vacation* are intersected to provide a candidate search space for dense units of higher dimensionality.

CLIQUE partitions every dimension into intervals, and identifies intervals containing at least  $l$  points, where  $l$  is the density threshold. CLIQUE then iteratively joins two  $k$ -dimensional dense cells,  $c_1$  and  $c_2$ , in subspaces  $(D_{i_1}, \dots, D_{i_k})$  and  $(D_{j_1}, \dots, D_{j_k})$ , respectively, if  $D_{i_1} = D_{j_1}, \dots, D_{i_{k-1}} = D_{j_{k-1}}$ , and  $c_1$  and  $c_2$  share the same intervals in those dimensions. The join operation generates a new  $(k+1)$ -dimensional candidate cell  $c$  in space  $(D_{i_1}, \dots, D_{i_{k-1}}, D_{i_k}, D_{j_k})$ . CLIQUE checks whether the number of points in  $c$  passes the density threshold. The iteration terminates when no candidates can be generated or no candidate cells are dense.

In the second step, CLIQUE uses the dense cells in each subspace to assemble clusters, which can be of arbitrary shape. The idea is to apply the Minimum Description Length (MDL) principle (Chapter 8) to use the *maximal regions* to cover connected dense cells, where a maximal region is a hyperrectangle where every cell falling into this region is dense, and the region cannot be extended further in any dimension in the subspace. Finding the best description of a cluster in general is NP-Hard. Thus, CLIQUE adopts a simple greedy approach. It starts with an arbitrary dense cell, finds a maximal region covering the cell, and then works on the remaining dense cells that have not yet been covered. The greedy method terminates when all dense cells are covered.

“How effective is CLIQUE?” CLIQUE automatically finds subspaces of the highest dimensionality such that high-density clusters exist in those subspaces. It is insensitive to the order of input objects and does not presume any canonical data distribution. It scales linearly with the size of the input and has good scalability as the number of dimensions in the data is increased. However, obtaining a meaningful clustering is dependent on proper tuning of the grid size (which is a stable structure here) and the density threshold. This can be difficult in practice because the grid size and density threshold are used across all combinations of dimensions in the data set. Thus, the accuracy of the clustering results may be degraded at the expense of the method’s simplicity. Moreover, for a given dense region, all projections of the region onto lower-dimensionality subspaces will also be dense. This can result in a large overlap among the reported dense regions. Furthermore, it is difficult to find clusters of rather different densities within different dimensional subspaces.

Several extensions to this approach follow a similar philosophy. For example, we can think of a grid as a set of fixed bins. Instead of using fixed bins for each of the dimensions, we can use an adaptive, data-driven strategy to dynamically determine the bins for each dimension based on data distribution statistics. Alternatively, instead of using a density threshold, we may use entropy (Chapter 8) as a measure of the quality of subspace clusters.

## 10.6 Evaluation of Clustering

By now you have learned what clustering is and know several popular clustering methods. You may ask, “When I try out a clustering method on a data set, how can I evaluate whether the clustering results are good?” In general, *cluster evaluation* assesses