validation criteria may be available to evaluate a clustering. Thus, a number of *internal* criteria may be defined to validate the quality of a clustering. The major problem with internal criteria is that they may be biased toward one algorithm or the other, depending on how they are defined. In some cases, external validation criteria may be available when a test data set is synthetically generated, and therefore the true (ground-truth) clusters are known. Alternatively, for real data sets, the class labels, if available, may be used as proxies for the cluster identifiers. In such cases, the evaluation is more effective. Such criteria are referred to as *external validation criteria*.

6.9.1 Internal Validation Criteria

Internal validation criteria are used when no external criteria are available to evaluate the quality of a clustering. In most cases, the criteria used to validate the quality of the algorithm are borrowed directly from the objective function, which is optimized by a particular clustering model. For example, virtually any of the objective functions in the k-representatives, EM algorithms, and agglomerative methods could be used for validation purposes. The problem with the use of these criteria is obvious in comparing algorithms with disparate methodologies. A validation criterion will always favor a clustering algorithm that uses a similar kind of objective function for its optimization. Nevertheless, in the absence of external validation criteria, this is the best that one can hope to achieve. Such criteria can also be effective in comparing two algorithms using the same broad approach. The commonly used internal evaluation criteria are as follows:

1. *Sum of square distances to centroids:* In this case, the centroids of the different clusters are determined, and the sum of squared (SSQ) distances are reported as the corresponding objective function. Smaller values of this measure are indicative of better cluster quality. This measure is obviously more optimized to distance-based algorithms, such as k-means, as opposed to a density-based method, such as *DBSCAN*. Another problem with SSQ is that the absolute distances provide no meaningful information to the user about the quality of the underlying clusters.

2. *Intracluster to intercluster distance ratio:* This measure is more detailed than the SSQ measure. The idea is to sample r pairs of data points from the underlying data. Of these, let P be the set of pairs that belong to the same cluster found by the algorithm. The remaining pairs are denoted by set Q. The average intercluster distance and intracluster distance are defined as follows:

$$Intra = \sum_{(\overline{X_i},\overline{X_j})\in P} dist(\overline{X_i},\overline{X_j})/|P| \tag{6.43}$$

$$Inter = \sum_{(\overline{X_i},\overline{X_j})\in Q} dist(\overline{X_i},\overline{X_j})/|Q|. \tag{6.44}$$

   Then the ratio of the average intracluster distance to the intercluster distance is given by $Intra/Inter$. Small values of this measure indicate better clustering behavior.

3. *Silhouette coefficient:* Let $Davg_i^{in}$ be the average distance of $\overline{X_i}$ to data points *within* the cluster of $\overline{X_i}$. The average distance of data point $\overline{X_i}$ to the points in each cluster (other than its own) is also computed. Let $Dmin_i^{out}$ represent the minimum of these

(average) distances, over the other clusters. Then, the silhouette coefficient S_i *specific to the ith object*, is as follows:

$$S_i = \frac{Dmin_i^{out} - Davg_i^{in}}{\max\{Dmin_i^{out}, Davg_i^{in}\}}. \tag{6.45}$$

The overall silhouette coefficient is the average of the data point-specific coefficients. The silhouette coefficient will be drawn from the range $(-1, 1)$. Large positive values indicate highly separated clustering, and negative values are indicative of some level of "mixing" of data points from different clusters. This is because $Dmin_i^{out}$ will be less than $Davg_i^{in}$ only in cases where data point $\overline{X_i}$ is closer to at least one other cluster than its own cluster. One advantage of this coefficient is that the absolute values provide a good intuitive feel of the quality of the clustering.

4. *Probabilistic measure:* In this case, the goal is to use a mixture model to estimate the quality of a particular clustering. The centroid of each mixture component is assumed to be the centroid of each discovered cluster, and the other parameters of each component (such as the covariance matrix) are computed from the discovered clustering using a method similar to the M-step of EM algorithms. The overall log-likelihood of the measure is reported. Such a measure is useful when it is known from domain-specific knowledge that the clusters *ought* to have a specific shape, as is suggested by the distribution of each component in the mixture.

The major problem with internal measures is that they are heavily biased toward particular clustering algorithms. For example, a distance-based measure, such as the silhouette coefficient, will not work well for clusters of arbitrary shape. Consider the case of the clustering in Fig. 6.11. In this case, some of the *point-specific* coefficients might have a negative value for the correct clustering. Even the overall silhouette coefficient for the correct clustering might not be as high as an incorrect k-means clustering, which mixes points from different clusters. This is because the clusters in Fig. 6.11 are of arbitrary shape that do not conform to the quality metrics of distance-based measures. On the other hand, if a density-based criterion were designed, it would also be biased toward density-based algorithms. The major problem in relative comparison of different methodologies with internal criteria is that all criteria attempt to define a "prototype" model for goodness. The quality measure very often only tells us *how well the prototype validation model matches the model used for discovering clusters*, rather than anything intrinsic about the underlying clustering. This can be viewed as a form of *overfitting*, which significantly affects such evaluations. At the very least, this phenomenon creates uncertainty about the reliability of the evaluation, which defeats the purpose of evaluation in the first place. This problem is fundamental to the unsupervised nature of data clustering, and there are no completely satisfactory solutions to this issue.

Internal validation measures do have utility in some practical scenarios. For example, they can be used to compare clusterings by a similar class of algorithms, or different runs of the same algorithm. Finally, these measures are also sensitive to the number of clusters found by the algorithm. For example, two different clusterings cannot be compared on a particular criterion when the number of clusters determined by different algorithms is different. A fine-grained clustering will typically be associated with superior values of many internal qualitative measures. Therefore, these measures should be used with great caution, because of their tendency to favor specific algorithms, or different settings of the same algorithm. Keep in mind that clustering is an *unsupervised* problem, which, by definition, implies that there is no well-defined notion of a "correct" model of clustering in the absence of external criteria.

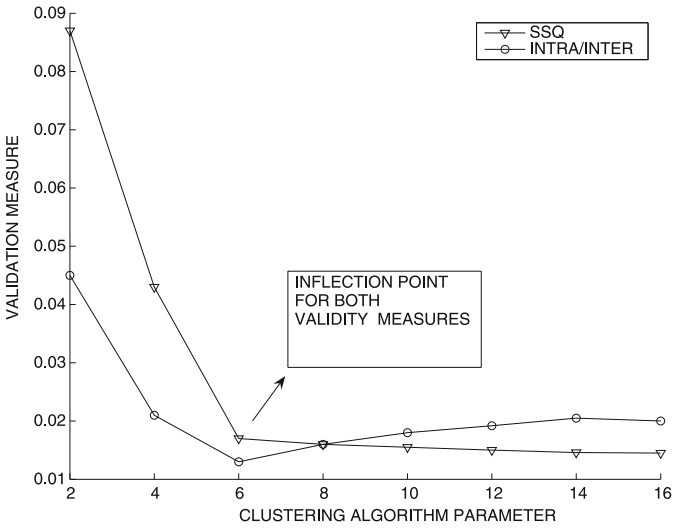

Figure 6.24: Inflection points in validity measures for parameter tuning

6.9.1.1 Parameter Tuning with Internal Measures

All clustering algorithms use a number of parameters as input, such as the number of clusters or the density. Although internal measures are inherently flawed, a limited amount of parameter tuning can be performed with these measures. The idea here is that the variation in the validity measure may show an inflection point (or "elbow") at the correct choice of parameter. Of course, because these measures are flawed to begin with, such techniques should be used with great caution. Furthermore, the shape of the inflection point may vary significantly with the nature of the parameter being tuned, and the validation measure being used. Consider the case of k-means clustering where the parameter being tuned is the number of clusters k. In such a case, the SSQ measure will always reduce with the number of clusters, though it will reduce at a sharply lower rate after the inflection point. On the other hand, for a measure such as the ratio of the intra-cluster to inter-cluster distance, the measure will reduce until the inflection point and then may increase slightly. An example of these two kinds of inflections are illustrated in Fig. 6.24. The X-axis indicates the parameter being tuned (number of clusters), and the Y-axis illustrates the (relative) values of the validation measures. In many cases, if the validation model does not reflect either the natural shape of the clusters in the data, or the algorithmic model used to create the clusters very well, such inflection points may either be misleading, or not even be observed. However, plots such as those illustrated in Fig. 6.24 can be used in conjunction with visual inspection of the scatter plot of the data and the algorithm partitioning to determine the correct number of clusters in many cases. Such tuning techniques with internal measures should be used as an informal rule of thumb, rather than as a strict criterion.

6.9.2 External Validation Criteria

Such criteria are used when ground truth is available about the true clusters in the underlying data. In general, this is not possible in most real data sets. However, when synthetic data is generated from known benchmarks, it is possible to associate cluster identifiers with the generated records. In the context of real data sets, these goals can be *approximately* achieved with the use of class labels when they are available. The major risk with the use of class labels is that these labels are based on application-specific properties of that data set and may not reflect the natural clusters in the underlying data. Nevertheless, such criteria

Cluster Indices	1	2	3	4
1	97	0	2	1
2	5	191	1	3
3	4	3	87	6
4	0	0	5	195

Figure 6.25: Confusion matrix for a clustering of good quality

Cluster Indices	1	2	3	4
1	33	30	17	20
2	51	101	24	24
3	24	23	31	22
4	46	40	44	70

Figure 6.26: Confusion matrix for a clustering of poor quality

are still preferable to internal methods because they can usually avoid *consistent* bias in evaluations, when used over multiple data sets. In the following discussion, the term "class labels" will be used interchangeably to refer to either cluster identifiers in a synthetic data set or class labels in a real data set.

One of the problems is that the number of natural clusters in the data may not reflect the number of class labels (or cluster identifiers). The number of class labels is denoted by k_t, which represents the true or ground-truth number of clusters. The number of clusters determined by the algorithm is denoted by k_d. In some settings, the number of true clusters k_t is equal to the number of algorithm-determined clusters k_d, though this is often not the case. In cases where $k_d = k_t$, it is particularly helpful to create a *confusion matrix*, which relates the mapping of the true clusters to those determined by the algorithm. Each row i corresponds to the class label (ground-truth cluster) i, and each column j corresponds to the points in *algorithm-determined* cluster j. Therefore, the (i, j)th entry of this matrix is equal to the number of data points in the true cluster i, which are mapped to the algorithm-determined cluster j. The sum of the values across a particular row i will always be the same across different clustering algorithms because it reflects the size of ground-truth cluster i in the data set.

When the clustering is of high quality, it is usually possible to permute the rows and columns of this confusion matrix, so that only the diagonal entries are large. On the other hand, when the clustering is of poor quality, the entries across the matrix will be more evenly distributed. Two examples of confusion matrices are illustrated in Figs. 6.25 and 6.26, respectively. The first clustering is obviously of much better quality than the second.

The confusion matrix provides an intuitive method to visually assess the clustering. However, for larger confusion matrices, this may not be a practical solution. Furthermore, while confusion matrices can also be created for cases where $k_d \neq k_t$, it is much harder to assess the quality of a particular clustering by visual inspection. Therefore, it is important to design hard measures to evaluate the overall quality of the confusion matrix. Two commonly used measures are the *cluster purity*, and *class-based Gini index*. Let m_{ij} represent the number of data points from class (ground-truth cluster) i that are mapped to (algorithm-determined) cluster j. Here, i is drawn from the range $[1, k_t]$, and j is drawn from the range $[1, k_d]$. Also assume that the number of data points in true cluster i are denoted by N_i, and the number of data points in algorithm-determined cluster j are denoted by M_j. Therefore, the number of data points in different clusters can be related as follows:

$$N_i = \sum_{j=1}^{k_d} m_{ij} \qquad \forall i = 1 \ldots k_t \tag{6.46}$$

$$M_j = \sum_{i=1}^{k_t} m_{ij} \qquad \forall j = 1 \ldots k_d \tag{6.47}$$

A high-quality algorithm-determined cluster j should contain data points that are largely dominated by a single class. Therefore, for a given algorithm-determined cluster j, the number of data points P_j in its *dominant class* is equal to the maximum of the values of m_{ij} over different values of ground truth cluster i:

$$P_j = \max_i m_{ij}. \tag{6.48}$$

A high-quality clustering will result in values of $P_j \leq M_j$, which are very close to M_j. Then, the overall purity is given by the following:

$$\text{Purity} = \frac{\sum_{j=1}^{k_d} P_j}{\sum_{j=1}^{k_d} M_j}. \tag{6.49}$$

High values of the purity are desirable. The cluster purity can be computed in two different ways. The method discussed above computes the purity of each algorithm-determined cluster (with respect to ground-truth clusters), and then computes the aggregate purity on this basis. The second way can compute the purity of each ground-truth cluster with respect to the algorithm-determined clusters. The two methods will not lead to the same results, especially when the values of k_d and k_t are significantly different. The mean of the two values may also be used as a single measure in such cases. The first of these measures, according to Eq. 6.49, is the easiest to intuitively interpret, and it is therefore the most popular.

One of the major problems with the purity-based measure is that it only accounts for the dominant label in the cluster and ignores the distribution of the remaining points. For example, a cluster that contains data points predominantly drawn from two classes, is better than one in which the data points belong to many different classes, even if the cluster purity is the same. To account for the variation across the different classes, the Gini index may be used. This measure is closely related to the notion of entropy, and it measures the level of *inequality* (or confusion) in the distribution of the entries in a row (or column) of the confusion matrix. As in the case of the purity measure, it can be computed with a row-wise method or a column-wise method, and it will evaluate to different values. Here the column-wise method is described. The Gini index G_j for column (algorithm-determined cluster) j is defined as follows:

$$G_j = 1 - \sum_{i=1}^{k_t} \left(\frac{m_{ij}}{M_j}\right)^2. \tag{6.50}$$

The value of G_j will be close to 0 when the entries in a column of a confusion matrix are skewed, as in the case of Fig. 6.25. When the entries are evenly distributed, the value will be close to $1 - 1/k_t$, which is also the upper bound on this value. The average Gini coefficient is the weighted average of these different column-wise values where the weight of G_j is M_j:

$$G_{average} = \frac{\sum_{j=1}^{k_d} G_j \cdot M_j}{\sum_{j=1}^{k_d} M_j}. \tag{6.51}$$

Low values of the Gini index are desirable. The notion of the Gini index is closely related to the notion of entropy E_j (of algorithm-determined cluster j), which measures the same intuitive characteristics of the data:

$$E_j = -\sum_{i=1}^{k_t} \left(\frac{m_{ij}}{M_j}\right) \cdot \log\left(\frac{m_{ij}}{M_j}\right). \tag{6.52}$$

Lower values of the entropy are indicative of a higher quality clustering. The overall entropy is computed in a similar way to the Gini index, with the use of cluster specific entropies.

$$E_{average} = \frac{\sum_{j=1}^{k_d} E_j \cdot M_j}{\sum_{j=1}^{k_d} M_j}. \tag{6.53}$$

Finally, a pairwise precision and pairwise recall measure can be used to evaluate the quality of a clustering. To compute this measure, all pairs of data points within the same algorithm-determined cluster are generated. The fraction of pairs which belong to the same ground-truth clusters is the precision. To determine the recall, pairs of points within the same ground-truth clusters are sampled, and the fraction that appear in the same algorithm-determined cluster are computed. A unified measure is the *Fowlkes-Mallows* measure, which reports the geometric mean of the precision and recall.

6.9.3 General Comments

Although cluster validation is a widely studied problem in the clustering literature, most methods for cluster validation are rather imperfect. Internal measures are imperfect because they are typically biased toward one algorithm or the other. External measures are imperfect because they work with class labels that may not reflect the true clusters in the data. Even when synthetic data is generated, the method of generation will implicitly favor one algorithm or the other. These challenges arise because clustering is an *unsupervised* problem, and it is notoriously difficult to validate the quality of such algorithms. Often, the only true measure of clustering quality is its ability to meet the goals of a specific application.

6.10 Summary

A wide variety of algorithms have been designed for the problem of data clustering, such as representative-based methods, hierarchical methods, probabilistic methods, density-based methods, graph-based methods, and matrix factorization-based methods. All methods typically require the algorithm to specify some parameters, such as the number of clusters, the density, or the rank of the matrix factorization. Representative-based methods, and probabilistic methods restrict the shape of the clusters but adjust better to varying cluster density. On the other hand, agglomerative and density-based methods adjust better to the shape of the clusters but do not adjust to varying density of the clusters. Graph-based methods provide the best adjustment to varying shape and density but are typically more expensive to implement. The problem of cluster validation is a notoriously difficult one for unsupervised problems, such as clustering. Although external and internal validation criteria are available for the clustering, they are often biased toward different algorithms, or may not accurately reflect the internal clusters in the underlying data. Such measures should be used with caution.

6.11 Bibliographic Notes

The problem of clustering has been widely studied in the data mining and machine learning literature. The classical books [74, 284, 303] discuss most of the traditional clustering methods. These books present many of the classical algorithms, such as the partitioning and hierarchical algorithms, in great detail. Another book [219] discusses more recent methods