C4.5 Acknowledgments

Research on C4.5 was funded for many years by the Australian Research Council.

C4.5 is freely available for research and teaching, and source can be downloaded from http://rulequest.com/Personal/c4.5r8.tar.gz.

2 The *k*-means algorithm

2.1 The algorithm

The `k-means` algorithm is a simple iterative method to partition a given dataset into a user-specified number of clusters, k. This algorithm has been discovered by several researchers across different disciplines, most notably Lloyd (1957, 1982) [53], Forgey (1965), Friedman and Rubin (1967), and McQueen (1967). A detailed history of `k-means` alongwith descriptions of several variations are given in [43]. Gray and Neuhoff [34] provide a nice historical background for `k-means` placed in the larger context of hill-climbing algorithms.

The algorithm operates on a set of d-dimensional vectors, $D = \{\mathbf{x_i} \mid i = 1, \ldots, N\}$, where $\mathbf{x_i} \in \Re^d$ denotes the ith data point. The algorithm is initialized by picking k points in $\Re^d$ as the initial k cluster representatives or "centroids". Techniques for selecting these initial seeds include sampling at random from the dataset, setting them as the solution of clustering a small subset of the data or perturbing the global mean of the data k times. Then the algorithm iterates between two steps till convergence:

Step 1: Data Assignment. Each data point is assigned to its *closest* centroid, with ties broken arbitrarily. This results in a partitioning of the data.

Step 2: Relocation of "means". Each cluster representative is relocated to the center (mean) of all data points assigned to it. If the data points come with a probability measure (weights), then the relocation is to the expectations (weighted mean) of the data partitions.

The algorithm converges when the assignments (and hence the $\mathbf{c_j}$ values) no longer change. The algorithm execution is visually depicted in Fig. 1. Note that each iteration needs $N \times k$ comparisons, which determines the time complexity of one iteration. The number of iterations required for convergence varies and may depend on N, but as a first cut, this algorithm can be considered linear in the dataset size.

One issue to resolve is how to quantify "closest" in the assignment step. The default measure of closeness is the Euclidean distance, in which case one can readily show that the non-negative cost function,

$$\sum_{i=1}^{N} \left(\underset{j}{\operatorname{argmin}} ||\mathbf{x_i} - \mathbf{c_j}||_2^2 \right) \tag{1}$$

will decrease whenever there is a change in the assignment or the relocation steps, and hence convergence is guaranteed in a finite number of iterations. The greedy-descent nature of `k-means` on a non-convex cost also implies that the convergence is only to a local optimum, and indeed the algorithm is typically quite sensitive to the initial centroid locations. Figure 2 [1] illustrates how a poorer result is obtained for the same dataset as in Fig. 1 for a different choice of the three initial centroids. The local minima problem can be countered to some

[1] Figures 1 and 2 are taken from the slides for the book, *Introduction to Data Mining*, Tan, Kumar, Steinbach, 2006.

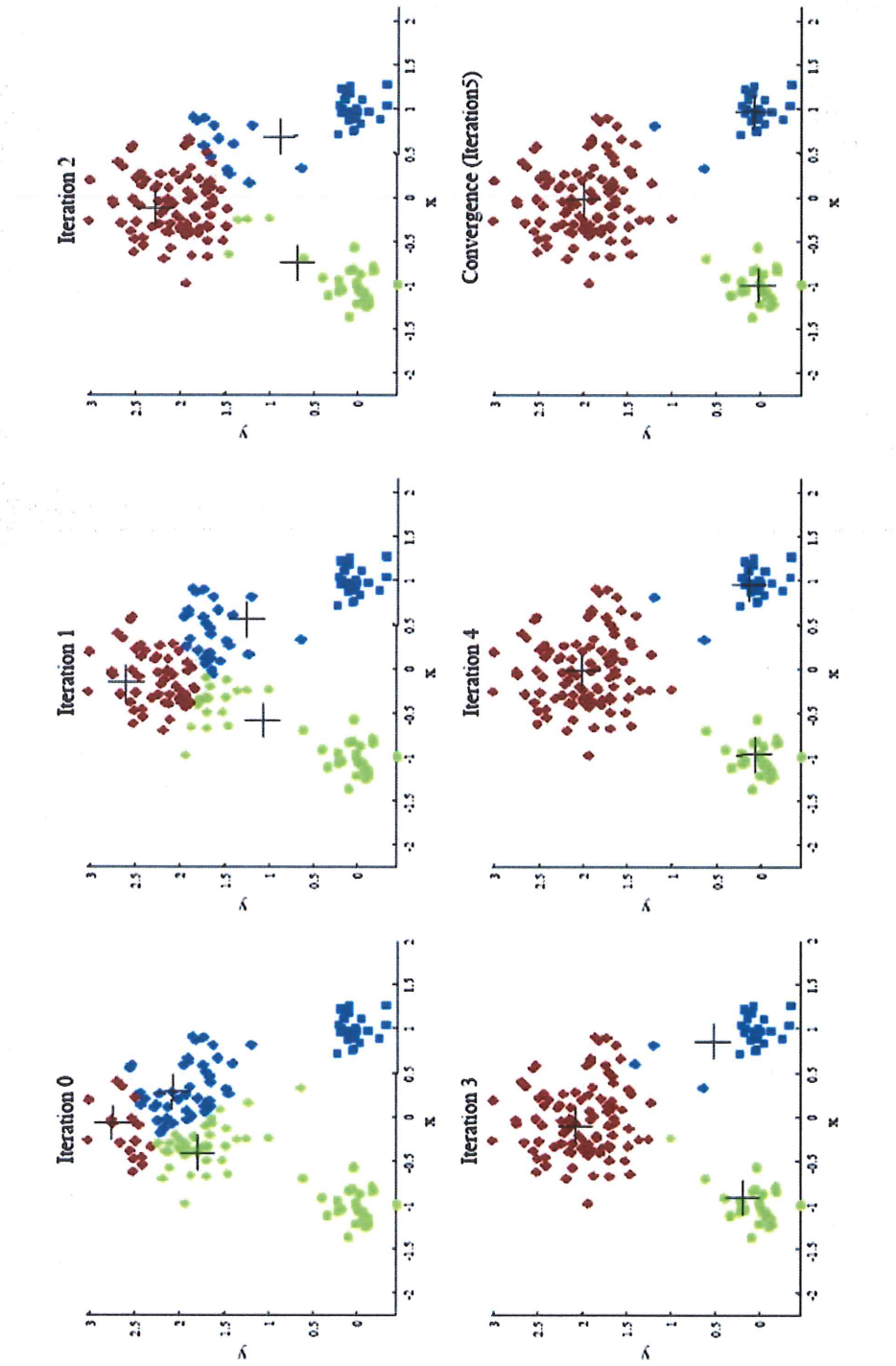

Fig. 1 Changes in cluster representative locations (indicated by '+' signs) and data assignments (indicated by color) during an execution of the `k-means` algorithm

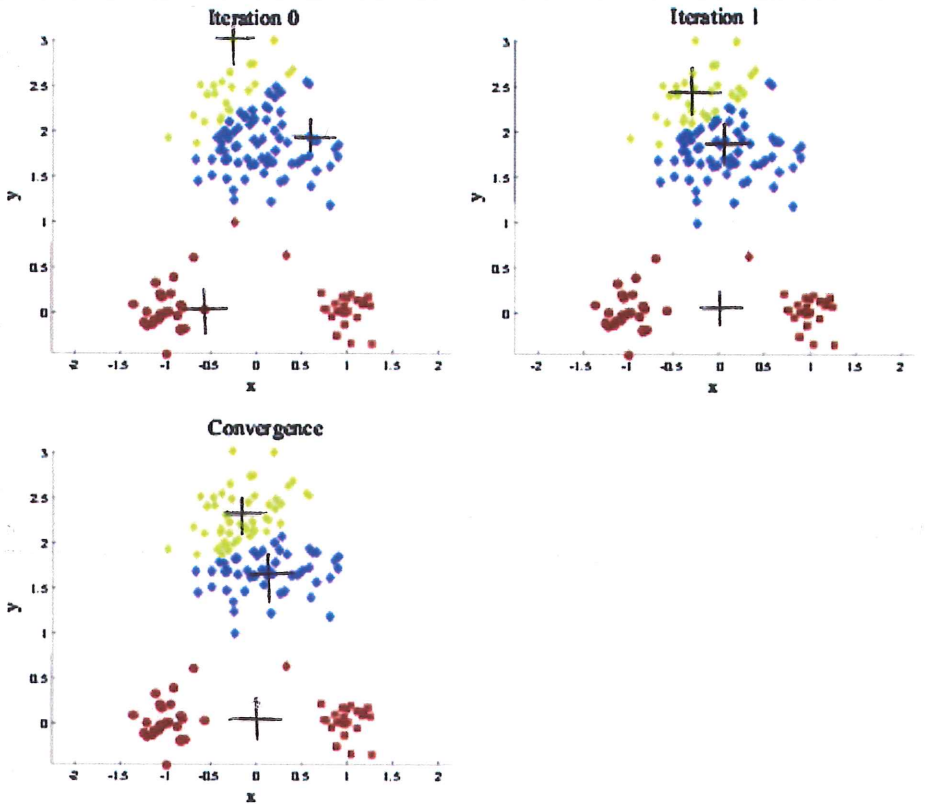

Fig. 2 Effect of an inferior initialization on the `k-means` results

extent by running the algorithm multiple times with different initial centroids, or by doing limited local search about the converged solution.

2.2 Limitations

In addition to being sensitive to initialization, the `k-means` algorithm suffers from several other problems. First, observe that `k-means` is a limiting case of fitting data by a mixture of k Gaussians with identical, isotropic covariance matrices ($\Sigma = \sigma^2\mathbf{I}$), when the soft assignments of data points to mixture components are hardened to allocate each data point solely to the most likely component. So, it will falter whenever the data is not well described by reasonably separated spherical balls, for example, if there are non-covex shaped clusters in the data. This problem may be alleviated by rescaling the data to "whiten" it before clustering, or by using a different distance measure that is more appropriate for the dataset. For example, information-theoretic clustering uses the KL-divergence to measure the distance between two data points representing two discrete probability distributions. It has been recently shown that if one measures distance by selecting any member of a very large class of divergences called Bregman divergences during the assignment step and makes no other changes, the essential properties of `k-means`, including guaranteed convergence, linear separation boundaries and scalability, are retained [3]. This result makes `k-means` effective for a much larger class of datasets so long as an appropriate divergence is used.