

**CHENNAI – PONDICHERRY**

**Efficient kNN Classification With Different Numbers of Nearest Neighbors**

**Abstract:**

k nearest neighbor (kNN) method is a popular classification method in data mining and statistics because of its simple implementation and significant classification performance. However, it is impractical for traditional kNN methods to assign a fixed k value (even though set by experts) to all test samples. Previous solutions assign different k values to different test samples by the cross validation method but are usually time-consuming. This paper proposes a kTree method to learn different optimal k values for different test/new samples, by involving a training stage in the kNN classification. Specifically, in the training stage, kTree method first learns optimal k values for all training samples by a new sparse reconstruction model, and then constructs a decision tree (namely, kTree) using training samples and the learned optimal k values. In the test stage, the kTree fast outputs the optimal k value for each test sample, and then, the kNN classification can be conducted using the learned optimal k value and all training samples. As a result, the proposed kTree method has a similar running cost but higher classification accuracy, compared with traditional kNN methods, which assign a fixed k value to all test samples. Moreover, the proposed kTree method needs less running cost but achieves similar classification accuracy, compared with the newly kNN methods, which assign different k values to different test samples. This paper further proposes an improvement version of kTree method (namely, k\*Tree method) to speed its test stage by extra storing the information of the training samples in the leaf nodes of kTree, such as the training samples located in the leaf nodes, their kNNs, and the nearest neighbor of these kNNs. We call the resulting decision tree as k\*Tree, which enables to conduct kNN classification using a subset of the training samples in the leaf nodes rather than all training samples used in the newly