K-Nearest-Neighbor Consistency in Data Clustering: Incorporating Local Information into Global Optimization

Chris Ding* and Xiaofeng He*

Abstract

Nearest neighbor consistency is a central concept in statistical pattern recognition, especially the kNN classification methods and its strong theoretical foundation. In this paper, we extend this concept to data clustering, requiring that for any data point in a cluster, its k-nearest neighbors and mutual nearest neighbors should also be in the same cluster. We study properties of the cluster k-nearest neighbor consistency and propose kNN and kMN consistency enforcing and improving algorithms. Extensive experiments on internet newsgroup datasets using the K-means clustering algorithm with kNN consistency enhancement show that kNN /kMN consistency can be improved significantly (about 100% for 1MN and 1NN consistencies) while the clustering accuracy is improved simultaneously. This indicates the local consistency information helps the global cluster objective function optimization.

1 Introduction

Data clustering is an extensively studied area (see [4, 7]) with a long history [9, 8]. However, due to the very large number of ways data points can be grouped into clusters, and large number of different perspectives and objectives, the procedures and thus the results are far from unique for a given dataset.

The most popular clustering method is the K-means clustering which minimizes a sum of error squared objective function. The K-means favors spherically shaped clusters. Another popular method with long history is the bottom-up hierarchical agglomerative clustering, which could produce clusters of quite different shapes depending on the choice of single-linkage, double-linkage, average linkage, etc. More recently, top-down divisive clustering [11, 3] appears to be more preferable to handle large datasets, with relatively smaller number of steps compared to the bottom-up agglomerative process. A recent trend in research is to revive the objective function approach used in the K-means . Here clustering objective functions are proposed and procedures are designed to optimize the objective function. The spectral graph partitioning approach is an example (see also [4, 11] for objective functions). Finally, all above methods provide a hard clustering, i.e., each data point is assigned to only one cluster. When clusters overlap, such a hard choice is often not clearly justified. The gaussian mixture model uses a probabilistic density distribution to model clusters, thus a data point can be assigned to several clusters with appropriate probabilities.

Cluster kNN consistency

In this paper, we propose a new concept of cluster nearest-neighbor consistency as we explain below.

> **Cluster k-Nearest-Neighbor Consistency**: For any data object in a cluster, its k-nearest neighbors should also be in the same cluster.

Our work is motivated by the k-nearest neighbor classification methods where the nearest neighbor consistency is a fundamental assumption.

kNN classification is proposed 50 years ago [5] for supervised learning. The basic assumption is that data objects in each class are distributed consistently. Therefore, suppose we are given the training dataset with known class labels for each object. A new object is inserted in the feature vector space and its kNNs are selected. The object is classified to the class which is determined by the majority voting on the class labels of its kNNs, i.e., the new object is classified to be the most consistent with its neighbors.

The kNN classification method has a strong theoretical foundation. The key theorem is:

Theorem 1 (Cover and Hart, 1967)[1]. The classification error in 1NN classification is, at most, twice of the Bayes error of optimal assignment for any data distribution in the large sample limit. □

Since the error of best possible optimal assignment is small, twice of that error is still small. Therefore, this theorem establishes the theoretical foundation of kNN classification methods[4].

*Lawrence Berkeley National Laboratory, Berkeley, CA 94720.
Supported in part by Office of Science, U.S. Department of Energy under contract number DE-AC03-76SF00098 through an LBL LDRD grant.

SAC'04, March 14-17, 2004, Nicosia, Cyprus

In this paper, we extend the notion of kNN consistency from supervised learning tasks such as classification, to unsupervised learning tasks, such as data clustering. Data clustering partitions data objects into disjoint/nonoverlapping clusters such that objects within same cluster are very similar, and objects belong different clusters are very different. It is a global optimization, i.e., it minimizes within-cluster distances and maximizes between-cluster distances. If objects within same cluster are very similar to each other, then it is likely that the nearest neighbor of any object in a cluster is also in the same cluster, i.e., the cluster kNN consistency should hold. Therefore "kNN consistency" has the same final goal as "data clustering".

Imaging the opposite. Suppose we perform a data clustering, and the resulting clusters have little kNN consistency, i.e., pick an object in a cluster and its nearest neighbor often belongs to another cluster. Such a clustering result is hardly useful and clearly contradict to the notion of "clustering".

Transitivity of cluster membership

More vigorously speaking, cluster kNN consitency defines a "transitive relationship". If x_1 is a member of cluster C_1, and x_2 is the 1NN of x_2, then by cluster 1NN consistency, x_2 should also be a member of C_1. In practical situations, this maybe not hold strictly, although we believe that is true most of time. Consider two extreme cases. For numbers a, b, c, transitivity holds strictly, i.e, $a > b$ and $b > c$ implies $a > c$. For tennis players, the fact that a beats b and b beats c often do not imply a beats c. The transitivity of cluster kNN consitency is somewhat between these two extreme cases. The technical contribution of this paper is to quantify to what extent the transitivity holds.

KNN consistency and data clustering

It appears that so far the kNN consistency has not been explored in the context of data clustering. In this paper, we explore on this issue. kNN consistency is a characteristics of cluster assignment, whether the clusters are naturally obtained, or obtained by a computational algorithm. We are interested in the relationship between kNN consistency and clustering algorithms. We ask the the following questions (Q1) Are clusters produced in standard clustering algorithms such as K-means exhbits kNN consistency? (Q2) Could the kNN consistency be enhanced in a clustering algorithm? (Q3) Would enhanced kNN consistency improve the clustering accuracy?

It should be noted that kNN consistency is a kind of local information. In the global optimization of K-means clustering, kNN is never directly considered. Therefore, the answer to question (Q1) is not necessarily positive. The comprehensive experiments in this study show that the answer is surprisingly positive (see §7). For question (Q2), we propose an algorithm to enhance the kNN consistency in a clustering algorithm (see §5) and demonstrate that the algorithm enhances kNN consistency significantly, upto 100% (see §7). Question (Q3) is the main question that motivated this study. The clustering experiments demonstrate that enhanced kNN lead to increased clustering accuracy, although this is not clear at the outset, since kNN is not a direct part of K-means clustering objective and could potentially introducing unnecessary constraints (although the final goals are similar).

K-NN consistency as unsupervised quality measure

As discussed above, "cluster kNN consistency" reflect a different aspect of the notion of "clustering" and thus provides an useful quality measure. Because kNN consistency is an "unsupervised" measure, i.e., we can compute it without the need of knowning the correct cluster solution (which is often difficult in practical situations), thus kNN consistency could be a more useful quality measure than some other quality measures such as "accuracy" which is a supervised measure requiring the knowledge of the correct clustering solution.

2 K-NN consistency in clustering

We first establish several basic properties of kNN consistency in clustering.

Definition 1. Data point kNN -consistency. Given a data point x and a cluster C_p which is a subset of the total data X. Point x is said to be kNN -consistent w.r.t. cluster C_p, if all x's kNNs are inside the cluster C_p. Furthermore, cluster C_p is said to be kNN -consistent if all its members are at least kNN -consistent w.r.t. cluster C_p.

Because x's $(k+1)$-NN subset contains its kNN subset, $(k+1)$-NN consistency implies kNN consistency.

From now on, we will skip the words "w.r.t. cluster C_p" when discussing kNN -consistency.

To capture the notion that a cluster is well-separated from other clusters, we define

Definition 2. Maximal NN-consistency. Given a data point x and a cluster C_p with size n_p. x is said to be *maximal* NN consistent every point x in C_p has its $(n_p - 1)$-NN's also in the same cluster. Furthermore, cluster C_p is said to be maximal NN-consistent if all its members are maximal NN-consistent.

Theorem 2. For x_i to be maximal NN consistent w.r.t. cluster C, no point x_j outside C_p is closer to x_i than any other point x_k inside C_p, i.e.,

$$\min_{x_j \notin C_p} d(x_i, x_j) > \max_{x_k \in C_p} d(x_i, x_k). \tag{1}$$

Furthermore, cluster C_p is maximal NN-consistent if its members satisfy

$$\min_{x_i \in C_p, x_j \notin C_p} d(x_i, x_j) > \max_{x_i \in C_p, x_k \in C_p} d(x_i, x_k), \quad (2)$$

i.e., the maximum within-cluster distance is smaller than minimum between-cluster distance. Note that all these hold when the distance $d(x_i, x_j)$ can be equivalently replaced by similarity measure $s(x_i, x_j)$ and with the directions of inequalities in Eqs.(1, 2) are reversed. $\square$

For real application datasets, clusters often overlap; A clean separation of kNNs from one cluster to another is often not possible or probable. It is often the case that for data points near the inside boundary of a cluster, some of their kNNs fall outside of the cluster. Thus the fractional (percentage) of data points inside a cluster whose entire kNN remain in the same cluster provide a measure of the kNN consistency. We therefore introduce

Definition 3. Fractional kNN -consistency. Given a cluster C_p with n_p points. If there are n_k points which are at least kNN consistent, then the fractional kNN -consistency is defined to be n_k/n_p. Clearly, if a cluster C_p is 100% kNN -consistent, then by Definition 1, C_p is kNN -consistent.

3 K-Mutual Nearest Neighbor (kMN) consistency

Note that kNN relationship is not symmetric: the fact that x is the nearest neighbor of y does not necessarily imply that y's nearest neighbor must be x.

It is convenient to define symmetric nearest neighbor relationships. This leads to so-called mutual (or reflexive) nearest neighbor[6, 2]:

Definition 4. Mutual nearest neighbor (MN). If x's nearest neighbor is y and y's nearest neighbor is x, then we say x's mutual nearest neighbor is y, and y's mutual nearest neighbor is x. In general, assume x is in the k-th NN of y and y is in the ℓ-th NN of x. Let $p = \max(k, \ell)$. We say x is the p-th mutual nearest neighbor of y and y is the p-th mutual nearest neighbor of x.

We use kMN to denote the k-th mutual nearest neighbor, and MNNs to denote mutual nearest neighbors. Note that for a given object x, some of its kMN may not exist. For example, If x's 1NN is y and y's 1NN is z, then x's 1MN does not exist. For this reason, mutual nearest neighbors are also called "isolated nearest neighbors" [10]. MNNs are a stronger and more restrictive form of neighborhoods. The adoption of MNNs defines a tighter neighborhood than the usual kNNs .

Using kMN , all the kNN consistency definitions extend to kMN . For a given point x and any p, kNN contains kMN . Also x's $(k+1)$-MN subset contains its kMN subset. Therefore, we have

Theorem 3. If x is kNN consistent w.r.t. cluster C_p, x must be kMN consistent w.r.t. cluster C_p. If cluster C_p is maximal NN consistent, cluster C_p is also maximal MN consistent. Furthermore, if x is (k+1)-MN consistent w.r.t. cluster C_p, x must be kMN consistent w.r.t. cluster C_p.

4 Applications of kNN/kMN consistency

We have introduced the kNN and kMN consistency in data clustering. Now we proceed to check them for data set. There are three cases we wish to address.

(1) If the cluster structure is already known or given, we can check whether their cluster labels are consistent or not. We call this "cluster label consistency". In many practical situations, the cluster labels will not be 100% consistent. This is because (a) the cluster labels are assigned by some procedures that do not fully take into account the desired relationship ; or (b) the desired relationship is not clearly defined to begin with. Therefore, the kNN /kMN consistency provides a useful check or validation of the cluster structure assignment.

(2) The cluster structure is not known. We wish to find maximal NN consistent or MN consistent clusters. If the dataset has natural clusters that are well separated, such as those shown in Figure 1, then most existing cluster algorithms can find them without much difficulty (after some trial or error). We will not further address this issue.

(3) The cluster structure is not known and the clusters overlap to some extent. This is the most frequently occurring case. In this case, our primary goal is to improve kNN /kMN consistency upon an existing clustering solution provided by a particular algorithm, such as the K-means . In the rest of this paper, we will study this issue in details. We propose a systematical way for improving kNN /kMN consistency. We will discuss the algorithmic details in §5 and §5. In §7, we carry out extensive experiments on Internet newsgroup dataset to demonstrate the effectiveness of our consistency improving algorithm. The experiments clearly indicate that when the kNN /kMN consistency is improved, the clustering accuracy is also improved. This is consistent with our motivation about the consistent distributions. It also confirms the basic theoretical analysis on kNN classification as explained in §1.

5 Enforcing kNN/kMN consistency

In this section, we present an effective algorithm to enforce 100% kNN consistency of the clustering results from a standard clustering algorithm such as the K-means algorithm. In next section we propose algorithm for improving the fractional kNN consistency of the clustering results.

5.1 Chain operation

Let's consider the simplest case, i.e., enforcing 1NN consistency. Suppose x is in cluster C_p and x's 1NN is x_1 is in another cluster. To improve 1NN consistency, it is attempting to simply move x_1 from another cluster to cluster C_p. Now, we identify x_1's 1NN x_2, which could be in any cluster. We then need to move x_2 to cluster C_p if x_2 is not in C_p. This situation could repeat until all 1NNs are exhausted. This results in a chain operation which is not only timing consuming, but also could be sub-optimal.

To see why it could be sub-optimal, consider a case where the chain operation involves 6 data points. It is possible that the first point x_1 is in cluster C_1, while the next 3 points $x_2, x_3, x_4$are in cluster C_2 and the last two points x_5, x_6 are in cluster C_3. In this case, we would move 5 points to cluster C_1. However, to ensure the same 1NN consistency, we only need to move x_1, x_5, x_6 to cluster C_2. In this way, not only we save computation, but the final results are more "desirable" because we distort the current cluster structure "less".

5.2 Closed neighbor set

To resolve the above chain operation problem, we propose to utilize the closed neighbor set. The idea is to first identify all points involved in the chain operation and then move them all at once in an optimal fashion.

The identification of all objects involved in a chain operation can be efficiently computed. First we prove that

Theorem 4. For kNN consistency, all objects involved in a chain operation must be a connected component in a kNN graph.

Proof. A kNN graph is formed by adding an edge between any object and its kNN . Now, one can easily see that: (a) If x, y are directly in each others kNN , then x, y must be involved in a chain operation. (b) If x, y are not directly within each others kNN , but each of them are in kNN of a third object z, then x, y must be involved in a chain operation. By repeating (b) one see that all objects in a chain operation are the nodes in kNN graph reachable from a node. Thus all nodes in a connected component of the kNN graph are involved in a single chain operation. $\square$

From Theorem 4, identification of a closed neighbor set becomes identification of the connected component in kNN graph, which can be efficiently computed in $O(N)$ time.

After the closed neighbor sets are identified, we can use a simple linear algorithm to enforce the kNN consistency.

> **Algorithm** ENFORCE.
> Pass through each of the closed neighbor set. For each closed neighbor set, gather all objects in the set to one cluster C_p. C_p is determined by majority vote as in the example in §3.1.

6 Consistency-preserving K-means

Given a clustering result, we can apply the algorithm ENFORCE to enforce the kNN consistency on the clustering result for a chosen kNN .

However, the resulting strict kNN -consistent clusters may not be the optimal results of a particular clustering algorithm, e.g, the K-means algorithm. We may further refine the clustering according to K-means clustering objective while preserving the kNN -consistency.

The popular K-means algorithm is an error minimization algorithm where the objective function is the sum of error squared, sometimes called distorsion,

$$J_{\text{Km}} = \sum_{k=1}^{K} \sum_{i \in C_k} (\mathbf{x}_i - \mathbf{c}_k)^2$$

where $\mathbf{c}_k = \sum_{i \in C_k} \mathbf{x}_i / n_k$ is the centroid of cluster C_k and $n_k = |C_k|$. Our kNN consistency-preserving K-means algorithm is the following:

Algorithm K-means-CP .

A. Initialize cluster centers $(c_1, \cdots, c_K)$.

B. Iterate (1) and (2) until converge:

(1) Assign cluster membership. Assign one closed-neighbor-set S at a time. Assign all objects of the closed-neighbor-set S to the closest cluster C_p, where the closeness is defined in average sense, i.e., $p = \arg\min_k \sum_{i \in S} (\mathbf{x}_i - \mathbf{c}_k)^2$.

(2) Update centers: $c_k = \sum_{i \in C_k} x_i / n_k$.

This algorithm has the same efficiency and convergence property as the standard K-means . The difference is that in standard K-means , each unit of cluster assignment is one object while in our case, each unit of cluster assignment is the closed-neighbor-set for enforcing strict transitivity.

Note that both consistency enforcing and preserving algorithms are target to one specific kNN consistency, say 1NN-consistency or 2MN-consistency. However, it happens often (see §7) that after one specific kNN consistency is enhanced, other kNN consistency are also enhanced to lesser extent.

We randomly select K data points as initial centroids and run K-means to convergence. We run 10 trials and select the one with the best J_{Km} objective value.

7 Experiments on Internet Newsgroups

We apply the K-means-CP clustering algorithm on Internet newsgroup articles. The 20 newsgroups dataset is from www.cs.cmu.edu/afs/cs/project/theo-11/www/naive-bayes.html. Word - document matrix is first constructed. 1000 words are selected according to the mutual information between words and documents in unsupervised manner. Standard `tf.idf` term weighting is used. Each document is normalized to 1.

We focus on two sets of 2-cluster cases and two sets of 5-cluster cases. The choice of $K = 5$ is to have some variety in the hierarchical divisive steps (we avoid $K = 4, 8$). The 4 datasets are listed below:

```
    Dataset A2:                 Dataset B2:

NG1:  alt.atheism            NG18: talk.politics.mideast
NG2:  comp.graphics          NG19: talk.politics.misc

  Dataset A5:                   Dataset B5:

NG2:  comp.graphics          NG2:  comp.graphics
NG9:  rec.motorcycles        NG3:  comp.os.ms-windows
NG10: rec.sport.baseball     NG8:  rec.autos
NG15: sci.space              NG13: sci.electronics
NG18: talk.politics.mideast  NG19: talk.politics.misc
```

In datasets A2 and A5, clusters overlap at medium level. In datasets B2 and B5, clusters overlap substantially.

Real data set often has clusters of uneven sizes. To simulate this unbalanced cluster problems, we set two different size combinations for the 5-cluster datasets A5 and B5:
(1) For balanced (even cluster size) case, we select 100 documents from each newsgroup, with a total of 500 documents per dataset. These are denoted as dataset categories A5B, B5B.
(2) For the unbalanced (uneven cluster size) case, we select 200,140,120,100,60 documents from different newsgroups. with a total of 620 documents per dataset. These are denoted as dataset categories A5U, B5U. For A2 and B2, each dataset has 200 documents, 100 from each newsgroup.

To ensure the statistics of the experiments, for each dataset category, we do 20 runs. Each run is based on randomly sampled documents from each newsgroup according to the newsgroup combination and cluster size category. In each run, we apply K-means-CP The final results are the average over the 20 random samples.

Which kNN and kMN consistencies to enhance?

To what extent the kNN and kMN can be effectively enhanced? We naturally want to improve 1NN and 1MN consistencies. But can we do better, i.e, go beyond nearest neighbors? To get an idea about how far we can go, we check the sizes of the closed neighbor set (CNS) of §5.2. An example is shown in Table 1. As k increases the size of CNS for kNN grows very rapidly, to nearly the total size of the dataset at k=3. However, CNS sizes for 1NN have mean 4.5 and max 22, therefore, they are enforceable. Also, 2MN CNS sizes have mean 2.0 and max 10, quite enforceable. 1MN CNS sizes are either 2 or 1, since an object either has a mutual NN or not. The CNS sizes for other 19 datasets for A5B are quite similar. For A5U, unbalanced cases, CNS sizes remain the same. For B5B abd B5U, CNS sizes are slightly larger. From these characteristics of the dataset, we decide to enforce 2MN and 1NN consistencies, since these CNS can be reasonably moved from one cluster to another as a whole. We

	k	1	2	3	4	5
kMN	Min_size	1	1	1	1	1
	Mean_size	1.3	2.0	3.6	8.0	14.2
	Max_size	2	10	58	347	424
	# CNS	387	255	137	62	35
kNN	Min_size	2	3	4	498	498
	Mean_size	4.5	83	249	498	498
	Max_size	22	481	494	498	498
	# CNS	111	6	2	1	1

Table 1: Sizes of CNS of §5.2 for dataset A5B.

then do a 1MN consistency-preserving refinement.

The 1NN and 2MN consistency enforcement are done as follows.
(1) Let the standard method converge.
(2) Do 1NN consistency enforcement, as in §6.
(3) Refine the cluster structure.
(4) Do 2MN consistency enforcement, as in §6. (5) Refine the cluster structure while preserving the 1MN consistency.

Results (accuracy, percentage in-consistency) immediately after convergence at step (1) and after the 1NN and 2MN enforcements and 1MN preserving refinement at step (5) are shown in Table 2.

Experiment Results

First, we compute the cluster label kNN /kMN consistencies as in §4 (1). They are listed as the first line in each dataset category. These are the average of the 20 different random sampled document sets as explained earlier. Listed are the in-consistencies. We can see that as k increases, the cluster label inconsistencies increases, as expected (the kNN and kMN in-consistencies are inclusive, e.g., for 3NN, if a 1NN or 2NN or 3NN of a point is not in the same cluster, this point is considered inconsistent). More interestingly, as the number of clusters increases, or as the cluster overlap increase (from A5 to B5 datasets), the in-consistency increases. This is consistent with our intuition that the clusters are getting more mixed or overlapped . From these numbers, one see that B5B or B5U datasets are significantly overlapped, since 25% of points has their 1NNs in an different cluster. Note that only data points near the boundary of the clusters are possibly has 1NN in-consistency. If there are 50% data points near the boundary, this means 50% of those near the boundary are inconsistent.

For the 1NN and 2MN enforcements and 1MN consistency preserving refinement, we see that all 1NN - 3NN and 1MN - 3MN consistencies are improved. 1NN and 2MN consistencies are improved by 100% in all cases. 1MN consistency is rigorously enforced. 2NN consistency is improved about 40-50%. 3NN and 3MN consistencies are also improved, about 20-40%.

Most importantly, all kNN /kMN consistencies are improved enough to be better than the corresponding clus-

	1NN		2NN		3NN		1MN		2MN		3MN		Accuracy	
	Init	Fin	Init	Fin	Init	Fin	Init	Fin	Init	Fin	Init	Fin	Init	Fin
	5.70		10.95		17.32		1.70		4.60		8.35		–	
A2	8.55	1.92	15.68	9.60	22.05	16.88	3.30	0	7.75	2.0	11.95	5.60	92.9	93.3
	12.47		23.27		31.98		5.10		11.18		16.90		–	
B2	6.70	3.12	12.65	11.35	18.50	19.65	2.05	0	5.53	1.70	9.00	6.43	67.2	75.1
	16.33		28.04		36.75		6.56		15.28		23.46		–	
A5B	16.15	7.41	28.35	22.67	37.76	34.28	6.42	0	14.54	6.85	22.95	16.74	75.2	83.1
	13.79		24.99		33.77		4.73		12.16		20.68		–	
A5U	18.39	8.27	32.04	24.48	41.59	36.77	7.26	0	16.66	7.42	26.67	17.98	76.8	76.4
	25.62		42.99		53.61		9.56		24.49		36.26		–	
B5B	23.12	9.76	39.42	29.40	50.56	45.36	8.48	0	21.59	8.90	32.81	22.72	56.3	64.3
	25.02		42.48		54.89		9.48		23.63		36.56		–	
B5U	24.97	11.27	42.98	32.52	55.73	48.59	9.87	0	23.61	11.22	36.00	25.60	55.4	56.3

Table 2: Fractional (percentage) kNN and kMN inconsistency for newsgroup dataset. For each dataset, the first line gives the label inconsistency, and the second line gives the results from K-means clustering. For each KNN or KMN, we list the initial (Init) and the final (Fin) results after incorporating consistency. The last column show the clustering accuracy. For example, for dataset B2, the label 1NN inconsistency is 12.47%; the initial and final clusters obtained by K-means has 6.7% and 3.12% 1NN inconsistency. Therefore these clusters are more 1NN consistent than the original labeling. The clustering accuracy is 67.2% initially and increases to 75.1% after 1NN consistency enhancement.

ter label consistencies. This indicates that the original cluster labels are not entirely self-consistent, and our unsupervised consistency improvements actually capture this subtle difference. Of course, the original cluster labels are not assigned by expert. When a reader on internet newsgroup submits an article, he/she submits to the newsgroup category of the existing context, even though the content of that particular article makes it more suitable to another newsgroup.

The last column of Table 2 gives the accuracy of clustering, i.e., labels assigned by the clustering algorithm comparing to its true labels. One see that improving kNN /kMN consistency also simultaneously improves the clustering accuracy slightly (comparing the left sub-column under Init to the right sub-column under Fin) for K-means for A2, B2, A5B, B5B. For A5U and B5U, the accuracy remains about same. This results are significant in that kNN /kMN consistency actually helps produce better quality clusters.

8 Summary

We propose the cluster kNN /kMN consistency as a key (unsupervised) quality measure of data clustering. We examine a number of characteristics of kNN consistency and transitivity of kNN consistency. We also propose kNN /kMN consistency enforcing and preserving algorithms. Extensive experiments on internet newsgroups indicate that kNN /kMN consistency can be significantly enhanced while the clustering accuracy is improved simultaneously.

If we enforce consistency at kMN/kNN level and allow only one connected component per cluster, this is equivalent to finding connected components in a kMN/kNN-graph. The kMN/kNN consistent K-means allows several connected components per cluster, which is thus somewhere between the connected component approach and usual K-means . Thus we have a unified framework for these algorithms.

References

[1] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE transactions in Information Theory*, 13:21–27, 1967.

[2] T.F. Cox. Reflexive nearest neighbours. *Biometrics*, 37:367–369, 1981.

[3] C. Ding and X. He. Cluster merge and split in hierarchical clustering. *Proc. IEEE Int'l Conf. Data Mining*, pages 139–146, 2002.

[4] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification, 2nd ed.* Wiley, 2000.

[5] E. Fix and J. L. Hodges. Discriminatory analysis, nonparametric discrimination: Consistency properties. *Technical Report 4, USAF School of Aviation Medicine, Randolph fiels, TX*, 1951.

[6] K.C. Gowda and G. Krishna. Agglomerative clustering using the concept of mutual nearest neighborhood. *Pattern Recognition*, 10:105–112, 1978.

[7] J. Han and M. Kamber. *Data Mining: Concepts and Techniques.* Morgan Kaufmann, 2000.

[8] J. Hartigan. *Clustering Algorithms.* John Wiley & Sons, 1975.

[9] J. MacQueen. Some methods for classification and analysis of multivariate observations. *Proc. 5th Berkeley Symposium*, pages 281–297, 1967.

[10] D.K. Pickard. Isolated nearest neighbors. *Journal of Applied Probability*, 19:444–449, 1973.

[11] Y. Zhao and G. Karypis. Criterion functions for document clustering: Experiments and analysis. *Univ. Minnesota, CS Dept. Tech Report #01-40*, 2001.