Approach to Evaluate Clustering using Classification Labelled Data

by

Tuong Luu

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2010

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Cluster analysis has been identified as a core task in data mining for which many different algorithms have been proposed. The diversity, on one hand, provides us a wide collection of tools. On the other hand, the profusion of options easily causes confusion. Given a particular task, users do not know which algorithm is good since it is not clear how clustering algorithms should be evaluated. As a consequence, users often select clustering algorithm in a very adhoc manner.

A major challenge in evaluating clustering algorithms is the scarcity of real data with a "correct" ground truth clustering. This is in stark contrast to the situation for classification tasks, where there are abundantly many data sets labeled with their correct classifications. As a result, clustering research often relies on labeled data to evaluate and compare the results of clustering algorithms.

We present a new perspective on how to use labeled data for evaluating clustering algorithms, and develop an approach for comparing clustering algorithms on the basis of classification labeled data. We then use this approach to support a novel technique for choosing among clustering algorithms when no labels are available.

We use these tools to demonstrate that the utility of an algorithm depends on the specific clustering task. Investigating a set of common clustering algorithms, we demonstrate that there are cases where each one of them outputs better clusterings. In contrast to the current trend of looking for a superior clustering algorithm, our findings demonstrate the need for a variety of different clustering algorithms.

Acknowledgments

First, I would like to thank my supervisor, Shai Ben-David, for all his support throughout my graduate studies and for all the insightful and constructive discussions that greatly helped me with the completion of this thesis.

I would also like to thank my sister Khanh Luu, who have accompanied me throughout my Master program at University of Waterloo.

Additionally, I would like to thank Rita Ackerman for her great team work and efforts in our collaboration. Also, thanks to all my current and previous office mates, who created the great atmosphere of both work and fun: Ting Liu, Tyler Lu, Ke Deng, and David Pal.

I would especially like to thank Jakub Gawryjolek, who was patient enough to deal with me being extremely busy over the last couple of months. I also greatly appreciate your suggestions in implementing the algorithms. You have been the greatest motivation and inspiration to me.

Dedication

I would like to dedicate this thesis to my parents. They have instilled in me the confidence and the determination to achieve set goals. Without their great support, understanding and encouragement to reach further, over all the years of my education, completing this thesis would not be possible.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Clustering is a central data analysis tool with a wide range of applications. Everyday, new data are clustered, new clustering criteria are introduced and new algorithms are proposed. Not surprisingly, there are many clustering algorithms. Faced with a concrete clustering task, users need to choose an appropriate clustering algorithm. Currently, such decisions are often made in a very ad hoc manner. Of course, users are aware of the costs involved in employing different clustering algorithms (software purchasing costs, running times, memory requirements, needs for data preprocessing etc.) but these considerations do not address the outcomes that these algorithms may produce.

One of the inherent difficulties in evaluating the qualitative performance of clustering algorithms is the scarcity of real life data with a "correct", ground truth, clustering. This is in stark contrast to the situation for classification tasks, where there are abundantly many data sets labeled with their correct classifications.

One solution to this issue, that has been reported in the clustering literature ([24], [5]) is to view the classes defined by the data labels as the correct clusters that the clustering algorithm should detect. We argue that this approach has serious flaws. Clustering is aimed to group domain elements in a way that corresponds to their similarities. While it does make sense to assume that similar objects tend to have similar labels, the converse may not be true. As a simple example, consider the task of clustering cars for the purpose of evaluating their market value, say, classifying them into 5 price categories. The claim that cars that are similar (in terms of some relevant features such as make, brand, age, involvement in past accidents etc.) may have similar market value, makes much more sense than the converse claim that all cars in the same price bracket are similar (in terms of the above mentioned features).

We propose a new technique for evaluating clustering algorithms that is based on this observation. Our approach accounts for the possibility that valid clusterings of labeled data have more clusters than there are labels in the data. The key component of our approach is a new *impurity measure* that is based on label homogeneity within clusters, but does not penalize for partitioning similarly labeled data. While points with distinct labels should be assigned to different clusters, there may be natural groups of points that share the same label, which may be mutually non-similar. In particular, we view the number of labels only as a lower bound on the desired number of clusters.

With this tool for revaluating the quality of clusterings, we investigate a number of basic questions in clustering theory by applying common clustering algorithms to classification labeled real data sets, and using our tool to evaluate the resulting clusterings. Our message is two-fold. We show that different algorithms produce very different results and that different tasks require different clustering tools. Our second message is the development and demonstration of a clustering model-selection tool that is based on the stability of clustering results over random perturbations of the input data set.

With the growing number of applications that require clustering, new algorithms are often being proposed. It is common to see papers that develop a novel clustering tool and demonstrate the advantages of that tool by showing that it outperforms others on some specific data sets. This trend has created a substantial volume of clustering algorithms that is virtually impossible to navigate through. As a result, many clustering users continue to rely on the most well-known, easily accessible, clustering algorithms. To aid clustering users, studies have been conducted with the aim of identifying which well-known algorithm is best, either in general or for a specific domain.

We argue that the usefulness of a clustering algorithm depends on the specific clustering task. Investigating a set of common clustering algorithms, we demonstrate that there are cases where each one of them outputs better clusterings. That is, for every algorithm, there are data where it finds better clusterings than all other algorithms considered.

Our findings illustrate the advantage of having a variety of clustering algorithms. In addition to showing that the performance of an algorithm is task dependent, we also show that different algorithms often output substantially different clusterings. Therefore, having choice when it comes to what algorithm to use increases the chances of finding a good clustering. These results also demonstrate the need for a new perspective; instead of searching for overall superior clustering algorithms, researchers should aim to provide users with tools for selecting a clustering algorithm

according to their specific tasks.

In order to evaluate clusterings, we define a new notion: impurity measure. Our impurity measure makes use of cluster labels. However, in most clustering scenarios no such labels are available. We develop an approach for selecting a clustering algorithm that does not require access to labels. We support this approach by examining data with classification label information using the impurity measure.

Our new approach is based on the sensitivity of an algorithm to data perturbation. Meaningful clusters are usually well-separated. When an algorithm outputs a good clustering on some data, we expect that if we slightly perturb the data, the algorithm will output a similar clustering.

We empirically demonstrate that sensitivity to data perturbation is positively correlated with low impurity. Algorithms that produce clusterings that are robust to small perturbations of the underlying distances score well under the impurity measure. Since testing robustness to perturbation requires no auxiliary information about the clustering, and is computationally efficient, this makes for a realistic criterion for selecting a clustering algorithm.

The thesis is organized as follows. The next chapter discusses related previous work. Chapter 3 lists our basic definitions, notation as well as brief summary of each clustering algorithm we consider. Next, we present our impurity measure for clustering evaluation in Chapter 4. We then demonstrate in Chapter 5 that different clustering algorithms may output substantially different clusterings on the same data using a clustering-similarity measure. Next, we compare some popular clustering algorithms using the impurity measure in Chapter 6. We then propose the sensitivity-based approach for selecting a clustering algorithm in the absence of data labels, supporting it using the impurity measure in Chapter 7. Finally, in Chapter 8 we present our conclusions.

Chapter 2

Related Work

There are many diverse studies comparing clustering algorithms. Virtually every paper that proposes a new clustering algorithm compares it with some previous algorithms. Since clustering is an important data exploration tool for a wide range applications, there are many application-specific comparative studies.

[20] studied three clustering algorithms which are widely used for Wireless Ad Hoc Networks, namely the Lowest-ID, the Highest Degree and the Extended Robust Re-clustering algorithm. Their work aims to investigate which are the factors that significantly affect the re-clustering performance such as cluster head modification rate, number of the generated clusters, reliability metrics, cluster head availability probability or end to end message delivery ratio.

[26] and [17] compared different document clustering techniques. The first compared partitional to agglomerative clustering algorithms, and proposed a new algorithm while the second studied K-means and agglomerative algorithms. The performance measure used in both papers are F-measure and entropy measure, which are both specific measure for clustering documents. [26] concluded partitional algorithms generally perform better than agglomerative algorithms and [17] stated that K-means was as good or better than agglomerative approaches.

As in bio-informatics, [2] conducted a comparison of four algorithms: Hierarchical clustering, K-means, PAM and SOM while [23] compared two implementations of K-means: Lloyd's method and Progressive Greedy. [2] concluded that K-means generated clusters with slightly better structural quality while hierarchical algorithms are bad. [23] focused on the running times and distance efficiency instead of clustering quality and stated Lloyd's method outperforms the other implementation in both measures.

[12] compared clustering methods for marketing research such as Single Linkage, Average Linkage, Complete Linkage, Ward's Method, K-means, Iterative partitioning methods and Hill-climbing methods.

The above papers are different from ours since we use data from a variety of different applications. This will create a more general comparison among clustering algorithms since we do not restrict our test data to any specific domain. Instead, we test the algorithms across different data sets and notice their performance. In addition to task-specific studies, there are a few empirical comparisons of clustering algorithms for general purposes.

[14] studied five algorithms: Complete Linkage, AutoClass, Self-Organizing Map, Growing Hierarchical Self-organizing Map and Generative Topographic Mapping. The author(s) focused on the robustness of the algorithms in terms of parameter selection, the ease of handling and the information gained from the result representation instead of clustering quality, which is our focus. In addition, [14] only tested the algorithms against one toy data set and one high dimensional data set.

[18] and [19] proposed a modified version of Single Linkage and demonstrated that it outperforms the traditional linkage algorithms. In both papers, clustering quality was determined by comparing the truth clustering and the clustering produced by the algorithms using Rand-index.

[5] tested four algorithms: K-means, Fuzzy c-means, Mountain and Subtractive against one heart-disease data set while we provides the comparison across many different real data. Root mean squared error and accuracy were used to determine clustering quality. As the result, [5] concluded that K-means is fairly better than the rest.

[24] presented a survey of clustering algorithms along with a small experiment with 3 data sets. Similarly, [4] conducted a survey of clustering algorithms and presented an empirical study of the algorithms on 4 data sets. In both studies, the authors used the number of labels as the correct number of clusters and measured the quality of clusterings by counting the percentage of errors.

We, in this work, consider a set of clustering algorithms that are popular in literature and compare them on a diversity of real data. Clustering algorithms can be compared based on different criteria, such as usability ([14]), execution time ([11]), CPU time and memory space ([25]), while our emphasis here is on the quality of the output of clustering algorithms.

Since there is a scarcity of ground-truth clustering data, researchers often rely on classification data to evaluate the quality of clustering algorithms. In such cases,

the standard approach is to view the classes defined by the data labels as the correct clusters that the clustering algorithm should detect (see, for example, [5], [21], [4], and [24]). The main distinction of our approach lies in how we utilize labeled data to evaluate clustering quality. We will only use the number of labels as a lower bound for the number of clusters.

Lastly, we do not try to determine a winning algorithm among the ones considered as in [5], [26] or [17]. Instead, we demonstrate the diversity of clustering algorithms and that each algorithm is needed for some data sets. Therefore, there is no overall superior winner.

Chapter 3

Notation and Background

3.1 Notation

A *k-clustering* $C = \{C_1, C_2, \ldots, C_k\}$ of a data set $X \subseteq \mathbf{R}^m$ is a partition of X into k disjoint subsets (so, $\bigcup_i C_i = X$).

A *clustering* of X is a k-clustering of X for some $1 \leq k \leq |X|$. We use the letter k to represent the number of clusters.

For a clustering C, let $|C|$ denote the number of clusters in C and $|C_i|$ denote the number of points in a cluster C_i. Let $|X|$ denote the number of elements in data set X.

3.2 Clustering distance: Variation of Information

In order to evaluate the difference between clusterings, we need a precise measure of clustering similarity. Various of such measures have been proposed, and some of the most popular are Jaccard index ([1]), Rand index ([13]), and Variation of Information ([9]). We have selected one such common measure, Variation of Information (VI) proposed by Meila, which was shown to satisfy many desirable properties [9]. Our empirical studies seem to yield very similar results when other common measures of difference between clusterings are employed.

Intuitively, Variation of Information [9] represents how much information one clustering gives about the other. $H(C)$, the entropy associated with a k-clustering $C = \{C_1, \ldots, C_k\}$, is defined as:

$$H(C) = -\sum_{i=1}^{k} P(i) \log P(i) \text{ where } P(i) = \frac{|C_i|}{|X|}$$

The mutual information between two clusterings C and C' is defined as

$$I(C, C') = \sum_{i=1}^{k} \sum_{i'=1}^{k'} P(i, i') \log \frac{P(i, i')}{P(i)P(i')} \text{ where } P(i, i') = \frac{|C_i \cap C'_{i'}|}{|X|}$$

where k and k' are the total number of clusters in clustering C and C' respectively.

The *Variation of Information* between clusterings C and C' is then defined as:

$$VI(C, C') = H(C) + H(C') - 2I(C, C')$$

This measure was shown that it was lower-bounded by 0, when the 2 clusterings are exactly the same and upper-bounded by $\log(|X|)$, when one clustering has all points in 1 cluster and the other clustering has all points in their own cluster.

In order to make the results interpretable across all data sets, instead of presenting the actual values of VI, which varies from 0 to $\log(|X|)$, we normalize VI values by $\log(|X|)$.

3.3 Clustering Algorithms

We consider five of the most commonly used clustering algorithms in literature: Average Linkage [16], Complete Linkage [6], Ward's Method ([22]), K-means ([7]) and Nearest Neighbor ([3]). Due to their popularity, we only present short description of each algorithm.

3.3.1 Average Linkage

Belonging to Agglomerative algorithm class, Average Linkage starts with $|X|$ clusters and each of them includes exactly one data point. A series of merge operations are then followed out, which finally leads to the desired number of clusters. Given the desired number of clusters, the algorithm is as follows:

1. Initialize each data point in its own cluster
2. For each pair of clusters C_i and C_j, compute:

$$D(C_i, C_j) = \frac{\sum_{x \in C_i, y \in C_j} d(x,y)}{|C_i|.|C_j|}$$

   where d is the distance measure between any two points in the data set.

3. Repeat from step 2 until the number of clusters equals to k:
   (a) Determine the pair C_i and C_j such that $D(C_i, C_j)$ is minimal
   (b) Merge cluster C_i and C_j
   (c) Update each $D(C_i, C_j)$ accordingly.

Figure 3.1 illustrates how Average Linkage determines the distance between any two clusters.

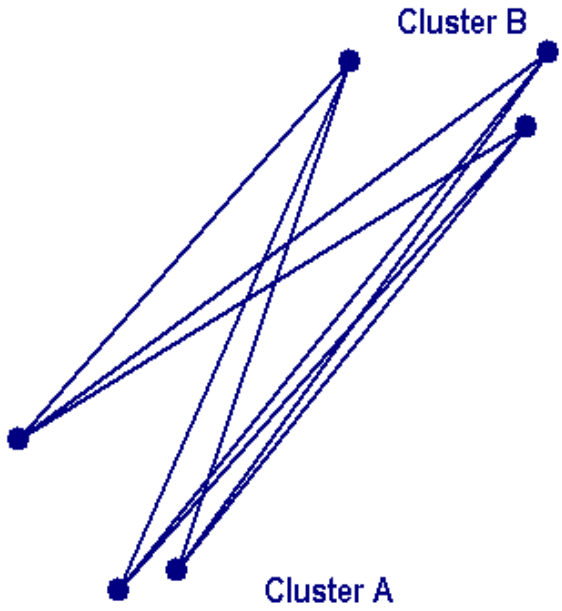

Figure 3.1: Average Linkage Algorithm [15]

3.3.2 Complete Linkage

Complete Linkage is very similar to Average Linkage in terms of how the clusters are created. The difference between them lies in how the distance between any 2 clusters are calculated. In Complete Linkage, the distance between two clusters is determined by the maximum distance between any two points in the two clusters. Given the desired number of clusters, the specific algorithm is as follows:

1. Initialize each data point in its own cluster
2. For each pair of clusters C_i and C_j, compute:

$$D(C_i, C_j) = \max\{d(x, y) : where\, x \in C_i\, and\, y \in C_j\}$$

   where d is the distance measure between points in the data set.

3. Repeat from step 2 until the number of clusters equals to k:
   (a) Determine the pair C_i and C_j such that $D(C_i, C_j)$ is minimal

(b) Merge cluster C_i and C_j

(c) Update each $D(C_i, C_j)$ accordingly.

Figure 3.2 demonstrates how Complete Linkage determines the distance between any two clusters.

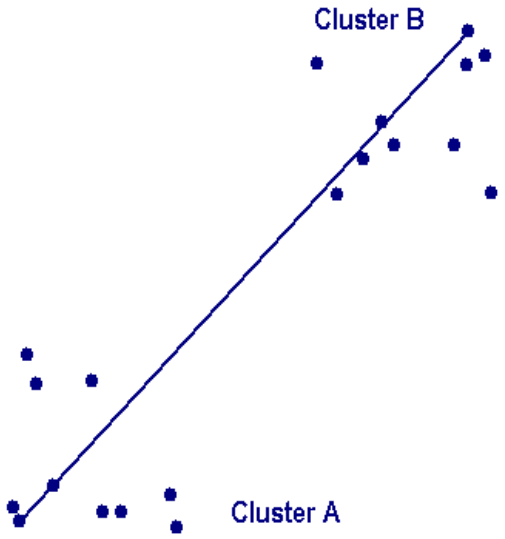

Figure 3.2: Complete Linkage Algorithm [15]

3.3.3 Ward's Method

Ward's Method is also hierarchical method but designed to optimize the minimum variance within clusters. Ward's Method tends to join cluster with a small number of data points and strongly biased towards producing cluster with roughly the same size. Given the desired number of clusters, the detailed algorithm is as follows:

1. Initialize each data point in its own cluster
2. For each pair of clusters C_i and C_j, compute

$$D(C_i, C_j) = \sum_{x \in A \cup B} \|x - \frac{1}{|A \cup B|} \sum_{x \in A \cup B} x\|^2$$

3. Repeat from step 2 until the number of clusters equals to k:

(a) Determine the pair C_i and C_j such that $D(C_i, C_j)$ is minimal

(b) Merge cluster C_i and C_j

(c) Update each $D(C_i, C_j)$ accordingly.

3.3.4 K-means

In this work, we implemented Lloyd's method for K-means clustering. Lloyd's K-means Clustering algorithm was proposed by S. Lloyd. Given a number k, the algorithm separates all data into k disjoint clusters, each of which has a center that acts as a representative. There are iterations that reset these centers then reassign each point in the data set to the closest center. The next iteration repeats until the centers do not move. The detailed algorithm is as follows:

1. Select k random points from the data set as the initial centers: $c_1, .., c_k$
2. Assign each data point to the cluster C_i corresponding to the closest cluster center $c_i (1 \leq i \leq k)$
3. After the assignment of all data points, recalculate new cluster centers as $c_i = \frac{1}{|C_i|} \sum_{j=1}^{|C_i|} x_j \quad \forall x_j \in C_i$

Figure 3.3 illustrates how the K-means works.

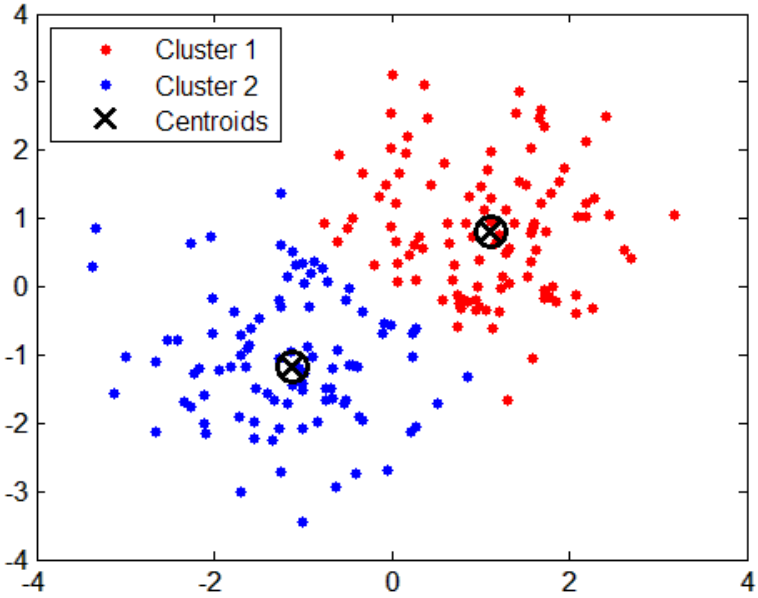

Figure 3.3: KMeans Algorithm [8]

K-means can work well for compact and hyperspherical clusters [24]. It is well known by the community that there is no efficient and universal method for selecting initial centers as well as the number of clusters. The convergence of the algorithm varies depending on the initial centers.

3.3.5 Nearest Neighbor

Nearest Neighbor [3] can be considered as one of the most simple clustering algorithms: grouping the points according to its closest neighbor. Given a threshold T and the number of clusters k, the algorithm is as follows:

1. Set $i = 1$. Create the first empty cluster and assign x_i to it.
2. Set $i = i + 1$. Find the nearest neighbor of x_i among the points that have already been assigned to clusters. Let d_m denote the distance from x_i to its nearest neighbor. Suppose its nearest neighbor is in the cluster m.
3. If d_m is less than or equal to the threshold T and the current number of clusters is less than k, assign x_i to C_m
4. Else if d_m is greater the threshold T and the current number of clusters is less than k, create a new cluster for that record.

5. Else if the current number of clusters is equal to k, assign x_i to its nearest neighbor's cluster.

6. If all points have been assigned, stop; otherwise, continue step 2.

Chapter 4

The Impurity Measure

Evaluating clustering quality has been a challenge to researchers. Since there is a scarcity of ground-truth clustering data, researchers often have to rely on classification data to evaluate the quality of clustering algorithms. A common approach to evaluating clustering algorithms makes use of labeled data by viewing the classes defined by the data labels as the correct clusters. We argue that this approach is fundamentally inaccurate. Labels focus on a single feature of the data, while a clustering may combine multiple features to create a meaningful partition.

Recall the car pricing example from the introduction, where labels correspond to a car's market value. While we might expect that similar cars will have similar market value, there may be very different cars that fall within the same price range.

We propose a new measure that accounts for the possibility that a valid clustering has more clusters than there are labels in the data. While points with distinct labels should be assigned to different clusters, there may be natural clusters that subdivide the classes defined by the labels. In particular, we view the number of labels only as a lower bound on the true number of clusters.

We formally define a measure of clustering quality that is based on labeling information. Let $\mathcal{L}$ represent the set of labels. Let $\ell(x)$ be the label of point x.

The *label of a cluster* is the label of the majority of points in that cluster, denoted by

$$\ell(C_i) = argmax_{b \in \mathcal{L}} |\{x \in C_i \mid \ell(x) = b\}|.$$

The *impurity of a clustering* is the proportion of points whose label differs from the labels of their cluster.

Definition 4.1 (Impurity of a clustering)**.** *The* impurity of a clustering C *is*

$$impurity(C) = \frac{1}{|X|} \sum_{i=1}^{|C|} |\{x \in C_i \mid \ell(x) \neq \ell(C_i)\}|$$

(where X is the domain set of the clustering).

The *impurity of a clustering algorithm*, for a specific data set and number of clusters k, refers to the impurity of the k-clustering that it produces on that data set.

Observe that a clustering where the number of clusters is equal to the number of points in the data will have 0 impurity. The objective is to find a clustering with low impurity and non-trivial cluster sizes. To this end, we examine the impurity as a function of the number of clusters.

It is easy to see that, for hierarchical algorithms, impurity decreases as the number of clusters increases. However, it might not always be the case for non-hierarchical algorithms. For example, consider a situation in which users want to apply K-means to a data set generated by two well-separated Gaussians, one of which is labeled with "+" while the other is labeled with "-". When the number of clusters is 2, K-means is likely to return a close-to-perfect clustering, which results in small impurity. However, if k is 3, the impurity might increase since K-means will partition the data set into 3 clusters, one of which contains points from both Gaussians. Now, if $k = 4$, K-means is likely to split each Gaussian group into half and obtain a better impurity than that of $k = 3$. The above example illustrates that, in general, for non-hierarchical algorithms, the impurity of k is not always larger than that of $k + 1$.

Our impurity measure was motivated by the need of selecting a good clustering algorithm together with an appropriate value of k. Since our experiments were performed without setting preference to any specific value of k, we evaluate clustering quality by considering the impurity trend over several values of k. Given two algorithms, we say the better algorithm is the one that has its impurity trend below that of the other. However, it is possible to have two impurity trends crossing each other. In such cases, we suggest to select the algorithm that has lower impurity at k of interest.

Chapter 5

Variety of Clustering Algorithms

Given the large available collection of clustering algorithms, each claims to achieve something better than some other algorithms. As the result, users often choose clustering algorithms in a very adhoc manner. This is one of the fundamental motivations for developing tools for clustering algorithm selection since different clustering algorithms tend to output very different clusterings. This phenomenon can be observed on many different data sets. We demonstrate it here with a few examples: some data sets with selected value of k and one data set with various values of k.

We performed a pairwise comparison of the output of clustering algorithms on real data sets from the UCI Machine Learning Repository [10]. Figures 5.1, 5.2, and 5.4 present the pairwise comparison of the algorithms on the Transfusion, Ionosphere, and Dermatology data sets with 15 clusters. We also illustrate this trend over different values of k on the Dermatology data, Figure 5.3 shows the results for $k = 10$ and Figure 5.5 for $k = 20$. Note that the values presented on these tables are normalized VI.

5.1 Data sets with selected value of k

Table 5.1: Transfusion with $k = 15$

	AL	**CL**	**NN**	**WM**	**KM**
AL	0	0.5204	0.5752	0.5119	0.5711
CL	0.5204	0	0.6407	0.4969	0.6253
NN	0.5752	0.6407	0	0.5502	0.5380
WM	0.5119	0.4969	0.5502	0	0.4326
KM	0.5711	0.6253	0.5380	0.4326	0

Table 5.1 presents pairwise comparison of all clustering algorithms on Transfusion data set, which consists of 748 points, with $k = 15$. Please refer to Figure 6.9 for the impurity reference. From the table, we could see that the resulted clusterings are very different from each other. The closest pair is Ward's Method and Complete Linkage; which is approximately 50% different from each other. On the other end, Nearest Neighbor and Complete Linkage are the most different from each other among the rest: 64%.

Table 5.2: Ionosphere with $k = 15$

	KM	**NN**	**WM**	**CL**	**AL**
KM	0	0.6101	0.5785	0.6079	0.8291
NN	0.6101	0	0.7484	0.7417	0.8036
WM	0.5785	0.7484	0	0.6733	0.8296
CL	0.6079	0.7417	0.6733	0	0.6619
AL	0.8291	0.8036	0.8296	0.6619	0

The difference among clusterings is demonstrated even more clearly in table 5.2. Please refer to 6.7 for the performance of the algorithms on this data set. On Ionosphere, which has 354 points, with $k = 15$, Ward's Method and K-means are the most similar among all clusterings but they are 57% difference from each other. Average Linkage and Ward's Method are the most different from each other with almost 83%.

5.2 Data set with various values of k

Having shown how different algorithms result in very different clusterings on individual data with selected value of k, we now demonstrate that phenomenon with various values of k on one single data set. In particular, we present three pair-wise comparison tables on Dermatology with $k = 10, 15, 20$.

Table 5.3: Dermatology with $k = 10$

	KM	**AL**	**CL**	**NN**	**WM**
KM	0	0.2680	0.3099	0.4054	0.2848
AL	0.2680	0	0.2957	0.3001	0.2677
CL	0.3099	0.2957	0	0.3802	0.3324
NN	0.4054	0.3001	0.3802	0	0.3229
WM	0.2848	0.2677	0.3324	0.3229	0

Table 5.4: Dermatology with $k = 15$

	KM	**NN**	**WM**	**CL**	**AL**
KM	0	0.5486	0.5050	0.5993	0.5905
NN	0.5486	0	0.2599	0.5115	0.4287
WM	0.5050	0.2599	0	0.4417	0.4203
CL	0.5993	0.5115	0.4417	0	0.5211
AL	0.5905	0.4287	0.4203	0.5211	0

Table 5.5: Dermatology with $k = 20$

	AL	**CL**	**NN**	**WM**	**KM**
AL	0	0.3031	0.369	0.5628	0.3820
CL	0.3031	0	0.4981	0.5470	0.4550
NN	0.369	0.4981	0	0.5373	0.4447
WM	0.5628	0.5470	0.5373	0	0.56
KM	0.3820	0.4550	0.4447	0.56	0

The tables 5.3, 5.4, and 5.5 present the pair-wise comparison of all algorithms on Dermatology data set with the number of clusters is set to 10, 15 and 20. For

impurity reference on this data set, please refer to Figure 6.5 . In all three tables, we can clearly see the clusterings are substantially different from each other. In fact, the difference ranges from 26% to 41%, 26% to 60%, and 30% to 56% with $k = 10, 15\, and\, 20$ respectively.

5.3 Summary

As shown in the above tables, different algorithms output substantially different clusterings on the same data. The trend has been demonstrated by both individual data sets with selected value of k and a data set with various values of k. For $k = 15$, the average distance between clusterings is 71% for the Ionosphere data, 55% for Transfusion, and 48% for Dermatology. The Variation of Information between clusterings on the same data set is often above 50%, going as high as 83%. Although in this work, we only choose one distance metric, Variation of Information, to compare clustering algorithms, the result does not bound to any specific metric. The same conclusion can be drawn if using other distance metrics such as Rand index or Jaccard index.

Chapter 6

Significance of Each Algorithm

Many empirical studies on clustering aim to identify superior clustering algorithms. When novel clustering techniques are proposed they are most often accompanied by experimental results illustrating that on some data the new algorithms find better clusterings than alternatives. Likewise, comparative studies tend to focus on identifying algorithms that perform better than their counterparts.

We argue that the utility of an algorithm depends on the specific data considered and that it is unlikely that any clustering algorithm would consistently outperform all others. We study five popular clustering algorithms: Lloyd's method (commonly refereed to as the K-means algorithm), Average Linkage, Complete Linkage, Ward's Method, and Nearest Neighbor. The definitions of these algorithms appear in Chapter 3. In this chapter, we show for each of these algorithms, there are data where it performs better than the other algorithms.

In this study we use classification data from the UCI Machine Learning Repository[10]. For every data set, we graph the impurity of the clusterings produced by each algorithm, ranging over different numbers of clusters. Specifically, we begin by plotting the impurity of every algorithm when the number of clusters equals the number of labels, we then plot the impurity of the algorithms with k equaling the number of labels rounded to the closest multiple of 5, and continue to plot the impurity in intervals of five until we reach 25 clusters.

We say that a clustering algorithm outperforms another clustering algorithm on a specific data set when the former has a lower impurity score for most values of k. Notably, a clear winner usually emerges when the number of clusters is between 5 to 10 larger than the number of labels, reinforcing the hypothesis that the true number of clusters is often larger than the number of labels.

6.1 Average Linkage

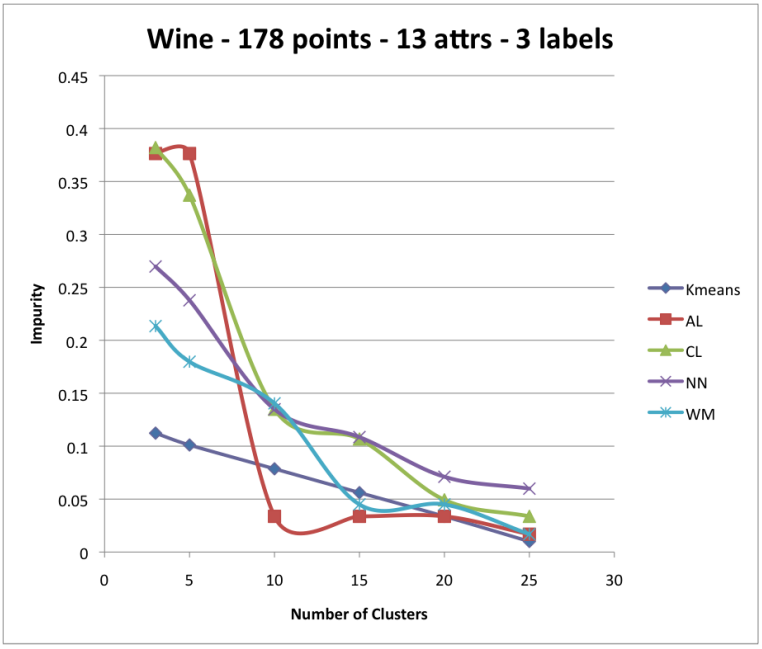

Figure 6.1: Wine - Average Linkage performs the best

The Wine data set consists of 178 elements, 13 attributes, and 3 labels. As is shown in Figure 6.1, the impurity of Average Linkage is notably better than that of other algorithms around 10 clusters. Since this data set has 178 elements, at $k = 20$ the expected cluster size drops to below 9 elements. We notice here that the impurity of all algorithms begins to converge from $k = 20$. This is expected, since the impurity of any algorithm decreases as the number of clusters approaches the number of data elements.

Another data set on which Average Linkage performs better than the others is Iris. This is one of the most popular data sets for examining the performance of novel methods in pattern recognition and machine learning. Since this data set only has 150 points with 4 attributes and 3 labels, we only plot the impurity trends up $k = 15$ instead of $k = 25$ as other data sets. As shown in Figure 6.2, even when the number of clusters is as low as the number of labels, the impurity of Average Linkage is significantly better than that of others. Only when k reaches 10, K-means and Nearest Neighbor start to catch up with Average Linkage and eventually converge to similar impurity.

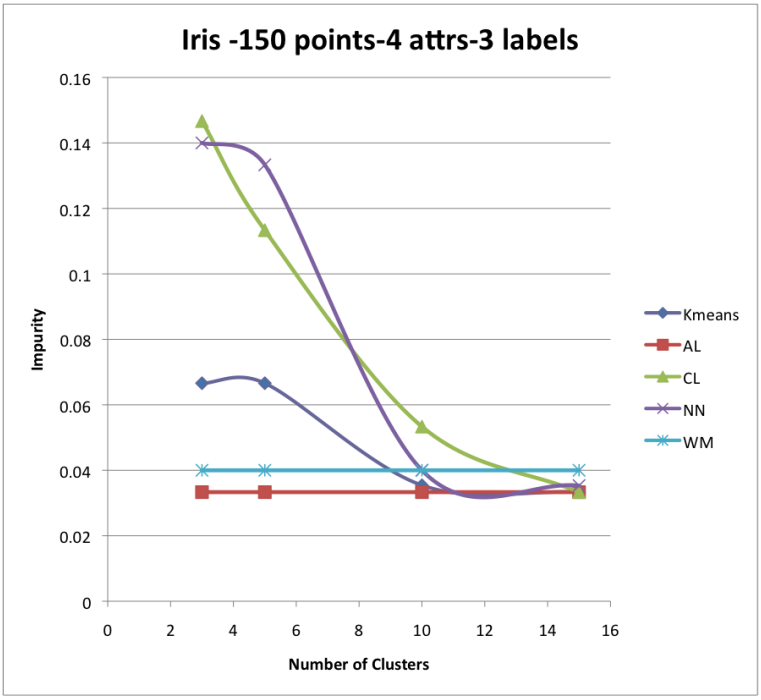

Figure 6.2: Iris - Average Linkage performs the best

6.2 Complete Linkage

Figure 6.3 shows the impurity of the algorithms on Sonar data set, which has 208 points, and each data point has 60 attributes. Until the number of clusters reaches 5, the impurity trends are indistinguishable. Complete linkage has notably lower impurity than the other algorithms from $k = 10$. As the number of clusters increases, the impurity of other algorithms decreases. K-means seems to catch up with Complete Linkage only when the number of clusters reaches 20.

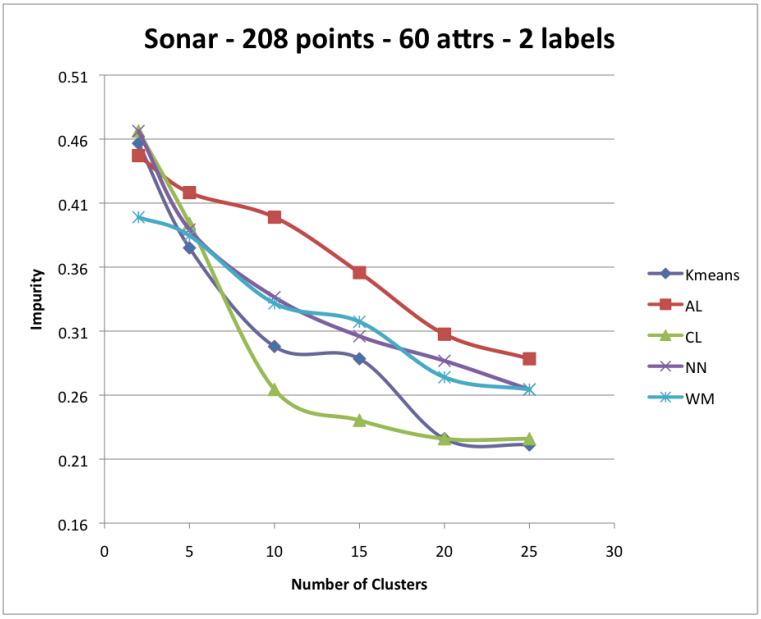

Figure 6.3: Sonar - Complete Linkage performs the best

Another data set that shows Complete Linkage performs better than other algorithms is Ecoli. This data set consists of 336 points, each of which has 7 attributes. We can see how the algorithms behave on this data set on Figure 6.4. Initially when the number of clusters equals to 8, all algorithms achieve quite similar impurity. As soon as the number of clusters reaches 10, Complete Linkage made a breakthrough and outperforms the rest.

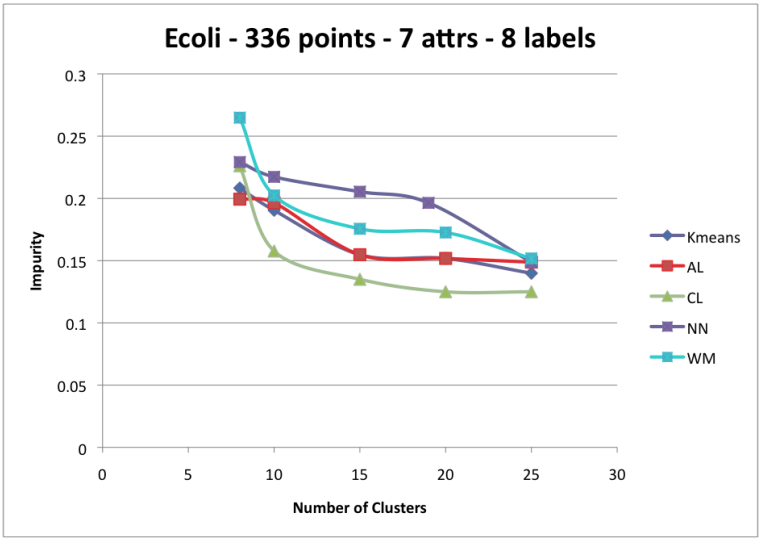

Figure 6.4: Ecoli - Complete Linkage performs the best

6.3 Ward's Method

We use Dermatology data set, which has 351 points, 34 attributes, and 6 labels, to demonstrate the need for Ward's Method. Note that as a consequence of having more labels, the impurity scores are higher than in the previous data sets. As illustrated in Figure 6.5, Ward's method outperforms the other algorithms from $k = 10$. As the number of clusters reaches 25, Average Linkage seems to catch up, but its impurity is still higher than that of Ward's Method.

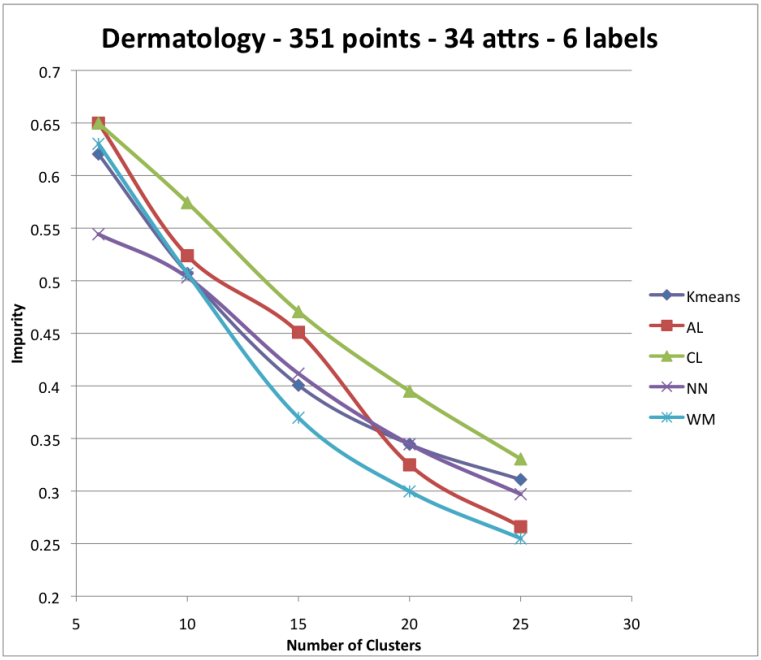

Figure 6.5: Dermatology - Ward's Method performs the best

Ward's Method also performs the best on Movement Libras data set. This data set has 360 points, each of which has 90 attributes. Similar to Dermatology data set, this one has 15 labels; therefore, we expect to see high level of impurity in Figure 6.6. Also as a consequence, we present the range for k up to 30 instead of 25 as other data sets. Starting with the lowest possible number of clusters, Ward's method has already achieved the best impurity compared to the rest. As the number of clusters reaches 30, the algorithms seem to converge.

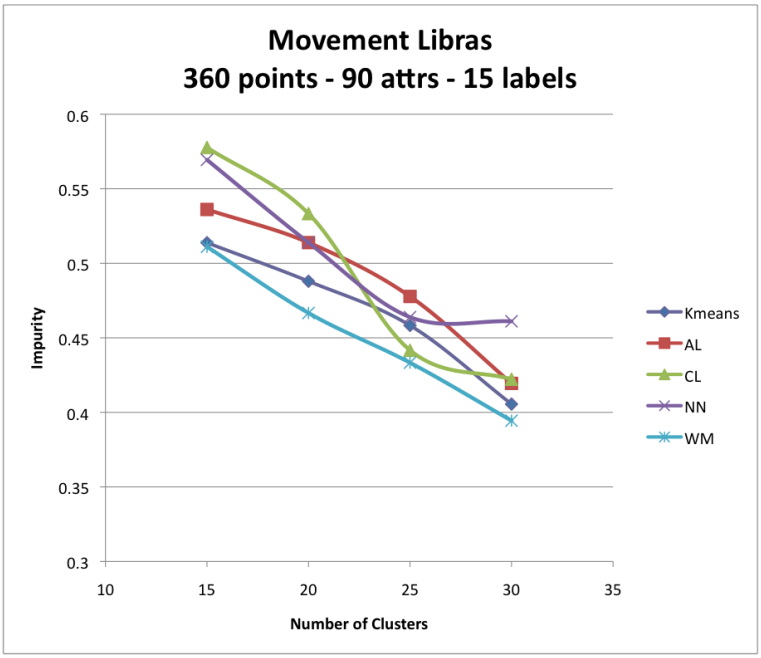

Figure 6.6: Movement Libras - Ward's Method performs the best

6.4 K-means

Figure 6.7 shows the impurity trend of five algorithms on the Ionosphere data set. This data set contains 351 data points each of which has 34 attributes, and 2 labels. The figure illustrates that K-means achieves a notably better impurity scores on the great majority of the domain. Ward's method catch up only when the number of clusters reaches 20.

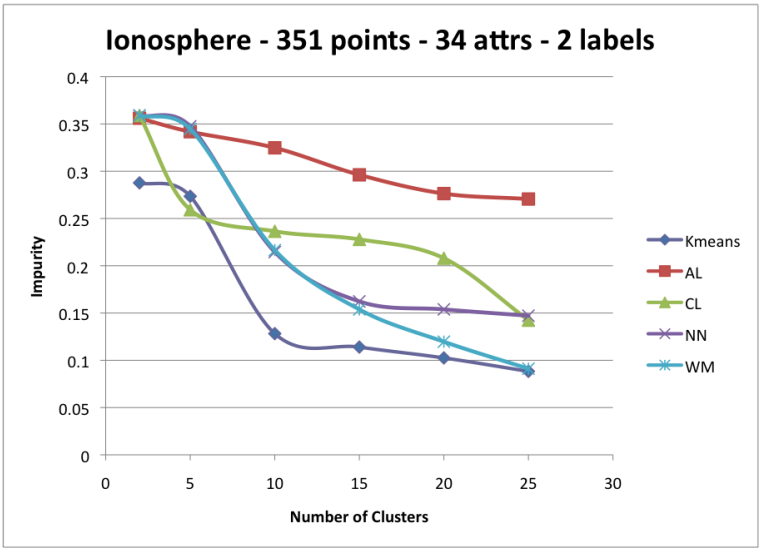

Figure 6.7: Ionosphere - K-means performs the best

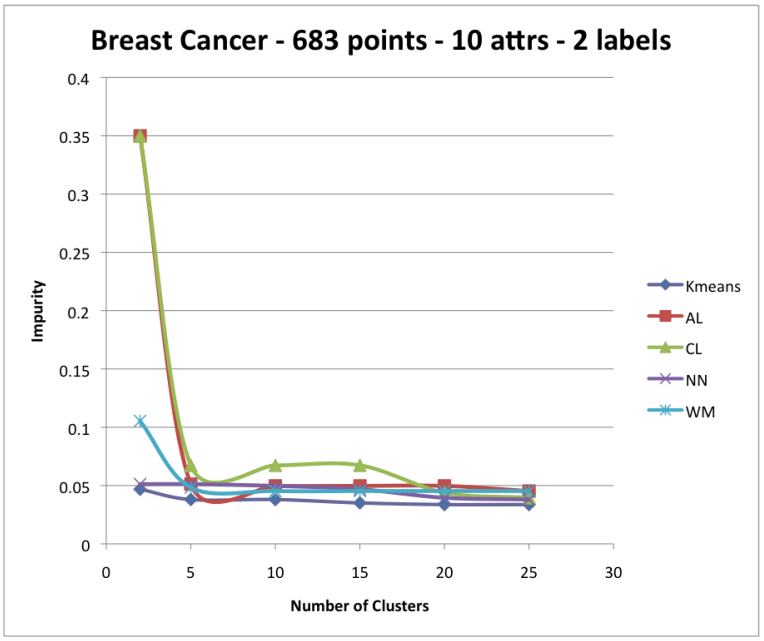

Figure 6.8: Breast Cancer - K-means performs the best

Another data set on which K-means performs better than the rest is Breast Cancer, which has 683 points, 10 attributes and 2 labels. Figure 6.8 shows the

impurity trend of the algorithms on this data set. Even when the number of clusters is set as low as 2, K-means already achieves an amazing impurity value while the others produce much worse clusterings. As the number of allowed clusters increases, the other algorithms starts to catch up, especially Average Linkage and Complete Linkage, but K-means still has the best impurity compared to the others'.

6.5 Nearest Neighbor

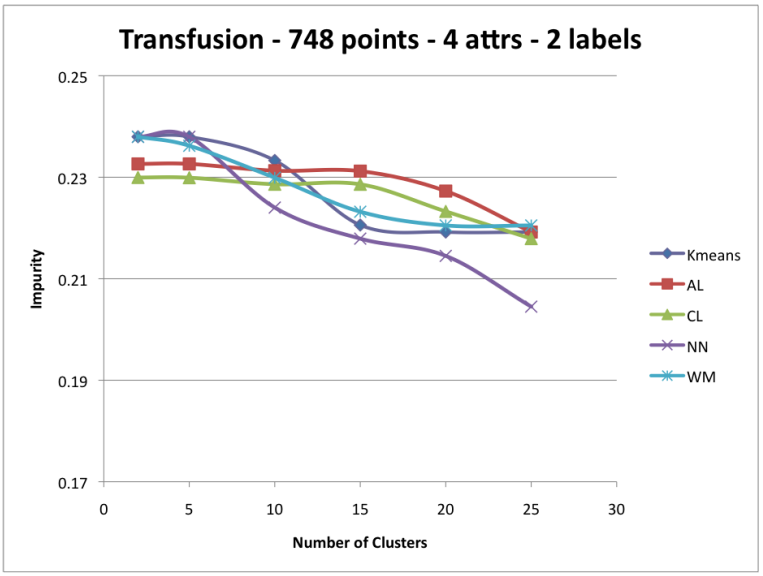

Figure 6.9: Transfusion - Nearest Neighbor performs the best

Just like the other clustering algorithms, there is also a need for Nearest Neighbor. Transfusion data set, which has 748 points and 4 attributes, is the one on which Nearest Neighbor performs better than the rest. As Figure 6.9 shows, the impurity of Neareast Neighbor is not the best when the number of clusters is as small as 2. However, as the number of clusters increases, Nearest Neighbor is the one that picks up the fastest and starts to achieve a much better impurity for the number of clusters greater than or equal to 10.

6.6 Summary

We have presented above for each clustering algorithm we consider at least one data set on which it performs the best. While many empirical studies try to show certain algorithm performs better than some other algorithms by testing them on a few data sets, our message is that each algorithm has its own advantage, and there are tasks in which each of them is needed. That also implies: there is no universal order of which some clustering algorithms always perform better than others. This result aims to motivate researchers to suggest a tool for selecting the appropriate clustering algorithm for a specific task instead of introducing many more new clustering algorithms.

Chapter 7

Practical Approach to Clustering Algorithm Selection

Motivated by the results in previous two chapters, we suggest a practical approach to clustering algorithm selection. Our impurity measure makes use of data labels. However, in most clustering applications no labels are available. We present an approach for evaluating clustering algorithms that does not require labels. This new approach is supported through a comparison to the impurity measure on labeled data.

A good clustering is usually composed of well-separated clusters. Given an algorithm that produces a good clustering on some data, we expect that if we slightly perturb the data, the algorithm will find a similar clustering. In particular, if an algorithm performs well on a data set, the difference between the clustering resulting from the original data and the one resulting from the slightly perturbed data is expected to be small.

In this section we illustrate that our impurity measure is positively correlated with sensitivity to data perturbation. We provide evidence that by selecting the algorithm which yields the most robust clustering one selects the algorithm with lowest impurity. As sensitivity to data perturbation requires no labels, and is easy (i.e. computationally efficient) to evaluate, this can serve as a realistic approach to clustering algorithm selection.

This section is organized as follows. We begin by formalizing the notion of sensitivity to data perturbation. We then discuss how cluster balance is related to our measure of sensitivity. Next, we apply this approach to evaluate clustering algorithms on data sets from the UCI Machine Learning Repository[10]. We

demonstrate a positive correlation between impurity and sensitivity to data perturbation on these data sets for a fixed number of clusters. We then study the relationship between these two measures for different values of k on the Ecoli data set.

7.1 Defining sensitivity to data perturbation

We define sensitivity for a fixed level of perturbation. To measure the distance between clusterings, we use the Variation of Information measure discussed in Section 3.2.

An *ϵ-perturbation* of data set X is obtained by adding white Gaussian noise on the elements of X. That is:

$$X' = X + G(0, \epsilon \mathbf{I})$$

. where $\mathbf{I}$ is the identity matrix.

The ϵ-sensitivity of an algorithm measures the effect that an ϵ-perturbation has on the output of an algorithm.

Definition 7.1 (ϵ-sensitivity)**.** *The ϵ-sensitivity of an algorithm A for data set X and $1 \leq k \leq |X|$ is*

$$\mathbb{E}_{X' \sim X + G(0, \epsilon \boldsymbol{I})}[VI(A(X,k), A(X',k))].$$

Selecting ϵ: Since our goal is to select the most suitable algorithm for a specific data set, we are interested in the ranking of the algorithms by their sensitivity. Since perturbation by ϵ is randomized, we select ϵ in a manner that ensures that the sensitivity ranking is persistent over different ϵ-perturbations of the data. Simultaneously, we need to select a sufficiently high perturbation, as otherwise all algorithms may be highly robust (note that every algorithm has sensitivity 0 for sufficiency small perturbations). As such, we have chosen to assign ϵ to the maximum value that yields a stable sensitivity ranking over the algorithms. When is sensitivity ranking of the algorithms stable? The same level of perturbation on a given data set can result in different perturbed data sets. Therefore, we say the sensitivity ranking of the algorithms is stable only if the order of the algorithms in terms of sensitivity does not change while given different perturbed data sets of the same level of perturbation.

We select the maximal value of ϵ that yields the same sensitivity ranking over the algorithms on three different ϵ-perturbations of the data. We estimate the sensitivity of an algorithm (for a specific data set and number of clusters) using the average variation of information of the clustering obtained on the original data with the clusterings on the three perturbed data sets.

7.2 Clusterings with many small clusters

Stability may not be a good indication of quality when the algorithm produces clusterings which have many small but only a few large clusters. In many cases, small clusters are the results of detecting outliers. Outliers are points that deviate markedly from the remaining data, which makes them insensitive to small perturbation.

This observation implies that before we can trust stability as an indication of quality, we need to examine the cluster sizes that the algorithms produce. We say a balance clustering is one in which most clusters are sufficiently large. Given a data set on n points partitioned into k clusters, the expected cluster size of a balance clustering is n/k. We say that a cluster is small if it has less than $\sqrt{n/k}$ elements.

As mentioned above, small clusters can be the result of detecting outliers, and outliers are insensitive to noise regardless how good the clustering is. To be on a safe side, we decided not to consider clusterings where at least half the clusters are small. In our initial investigation, we have also included Single Linkage. However, we consistently notice that when the number of clusters is small, Single Linkage tends to detect outliers, which are robust to small perturbation. Therefore, we find its results not interesting, and more importantly, not a reliable indication of quality. On the other hand, when algorithms find clusterings where the majority of clusters have at least $\sqrt{n/k}$ points, we consistently observe a positive correlation between impurity and sensitivity.

7.3 Correlation between impurity and sensitivity across different data sets

We study the correlation between sensitivity and impurity across different data sets, for a fixed number of clusters. We chose the number of clusters to be the number of labels plus 3; however, any other value of k where the expected cluster size is not

too small could have been used. Since Variation of Information (see Section 3.2) ranges from 0 to $\log |X|$, the same range applies for sensitivity. For interpretability across different data, we normalize sensitivity by $\log(|X|)$.

In order to identify clusterings that contain many small clusters while the other clusters are large, all diagrams in this section include a list of cluster sizes in descending order next to the name of the algorithm that produced the clustering. The number of elements, attributes, and labels in the data are indicated on top of each diagram.

7.3.1 Wine data set

In Figure 7.1, we plot the impurity versus sensitivity of the clustering algorithms on the Wine data set. Note that half the clusters produced by Average Linkage have size less than $\sqrt{n/k}$. As discussed above, this type of clusterings tend to have low sensitivity regardless of its performance, as is confirmed in this example.

The remaining algorithms which produce more balanced clusters clearly illustrate a positive correlation between impurity and sensitivity. In particular, the ranking of the algorithms in terms of sensitivity percentage is consistent with the ranking of the algorithms in terms of impurity. The one that has smallest sensitivity percentage also has the lowest impurity score.

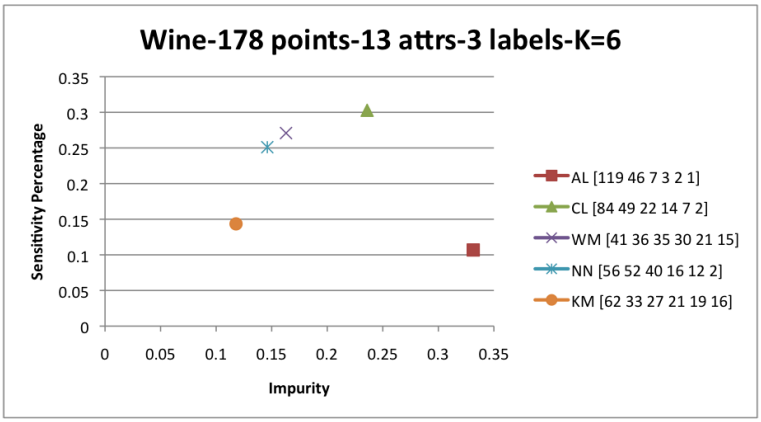

Figure 7.1: Wine - Variance = 0.00031623 - $k = 6$

7.3.2 Iris data set

Figure 7.2 illustrates a positive correlation between impurity and sensitivity on the Iris data set. On this data set, all algorithms produce quite balanced clusterings. Ward's method deviates from the curve, yet does not interfere to the positive correlation.

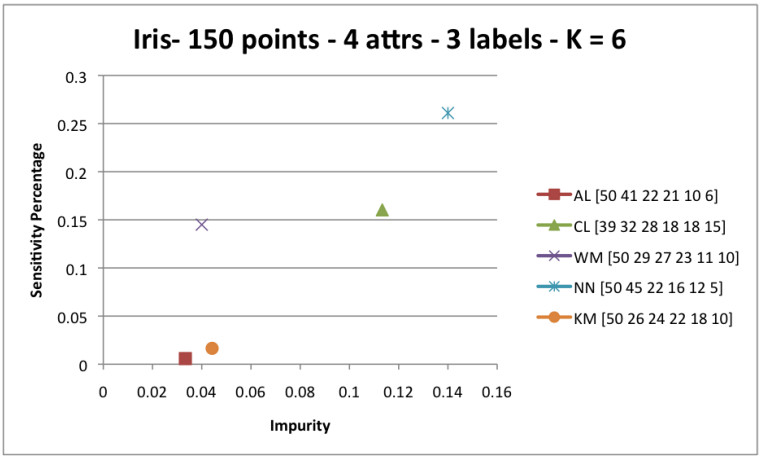

Figure 7.2: Iris - Variance = 0.0001 - $k = 6$

7.3.3 Bupa data set

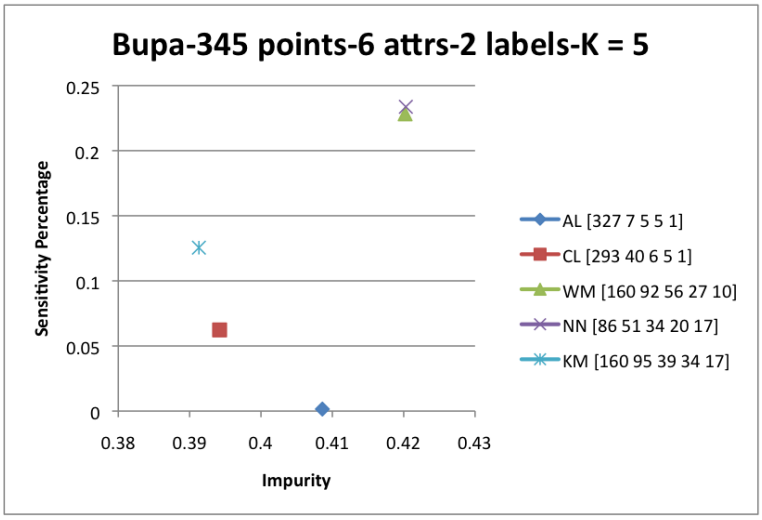

Figure 7.3: Iris - Variance = 0.0001 - $k = 6$

As Figure 7.3 illustrates, both Average Linkage and Complete Linkage produce clusterings where the majority of clusters are small (consisting of less than $\sqrt{n/k}$ elements) on the Bupa data set. Indeed, these algorithms output clusterings with a single large. As a result, both Average Linkage and Complete Linkage have low sensitivity no matter what their impurity is. The other three algorithms produce quite balance clusterings and demonstrate a positive correlation between impurity and sensitivity.

7.4 Correlation between impurity and sensitivity across different k

Having shown the positive correlation between sensitivity and impurity on individual data sets with certain value of k, we now show that such correlation holds with various values of k on a data set. We demonstrate this trend on the Ecoli data set, which has 336 points, 7 attributes, and 8 labels. Figure 7.4, 7.5, and 7.6 illustrate a comparison of impurity and sensitivity on the Ecoli data set for $k \in \{10, 15, 20\}$ respectively.

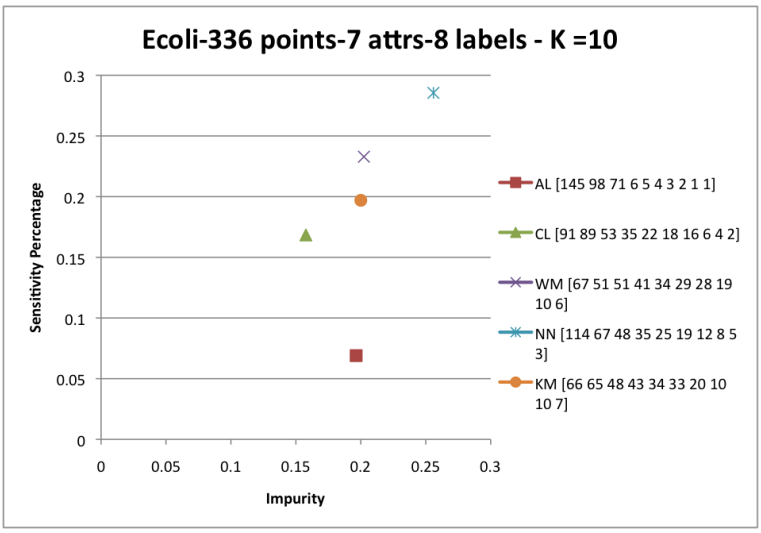

Figure 7.4: Impurity versus sensitivity on the Ecoli data set for $k = 10$

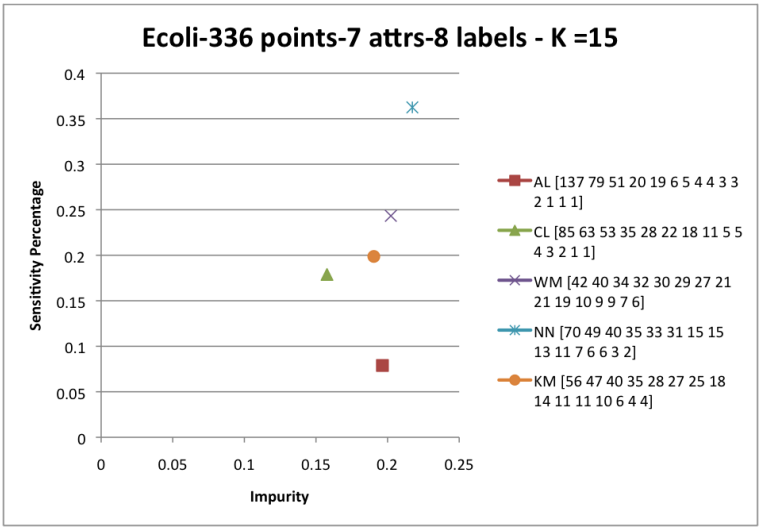

Figure 7.5: Impurity versus sensitivity on the Ecoli data set for $k = 15$

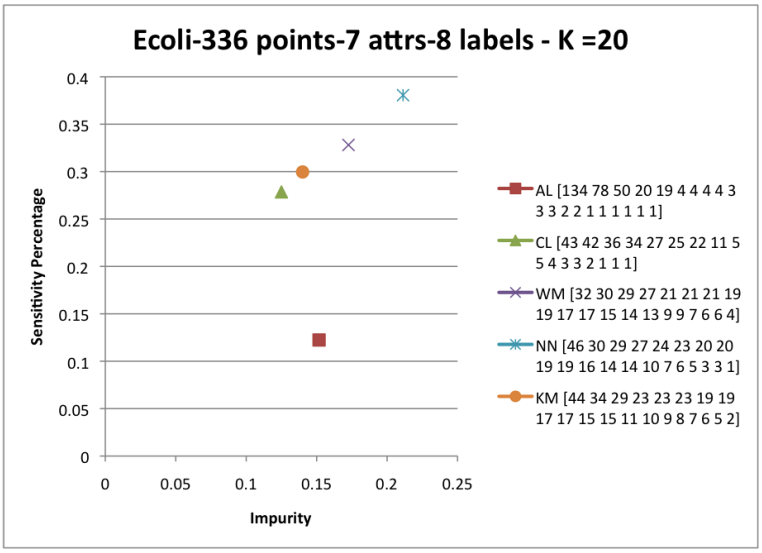

Figure 7.6: Impurity versus sensitivity on the Ecoli data set for $k = 20$

Impurity and sensitivity are positively correlated across different numbers of clusters. As we saw in the graphs above, here too Average Linkage produces clusterings with many small clusters, and as the result, its sensitivity is very low regardless of its performance. On the other hand, all other algorithms clearly support our claim: positive correlation between impurity and sensitivity. The algorithms'

ranking based on sensitivity percentage is consistent with that based on impurity. In all graphs, among the considered algorithms, Complete Linkage has the lowest sensitivity percentage as well as the smallest impurity. The result is consistent with 6.4 since Complete Linkage outperforms the others on this data set.

7.5 Summary

In this chapter, we have demonstrated a positive correlation between impurity and sensitivity to perturbation. We show that the positive correlation holds across different real data sets, and different numbers of clusters. Our experiments confirm that when the majority of clusters are small (which can be due to outliers), sensitivity is low irrespective of how the clustering relates to the data labels.

We observe that Average Linkage is prone to producing clusterings, in which there are many small clusters and the rest of clusters are large, and to a lesser degree this phenomenon also occurs for Complete Linkage. In our original investigation we have also included Single Linkage algorithm. However, we found that Single Linkage consistently output clusterings with a majority of small clusters, and has low sensitivity whether or not it performs well on the task.

On the other hand, when clustering algorithms find more balanced clusters, we repeatedly observe a positive correlation between impurity and sensitivity. Therefore sensitivity can replace impurity when no labels are available. We propose a practical approach to clustering algorithm selection, which does not make use of data labels. Given a specific task, users can easily calculate the sensitivity of the available algorithms and select the one with the lowest sensitivity. As we discussed and explained above, users should not take into consideration the algorithms that produce unbalanced clusterings since those can be the result of detecting outliers.

Chapter 8

Conclusions and Future Work

8.1 Conclusion

Clustering performance has been difficult to evaluate since there is scarcity of real life data with a "correct" ground truth clustering. This is in contrast to the situation for classification where there are many data sets labeled with their correct classifications. One solution to this issue is to view the classes defined by the data labels as the correct clusters that the clustering algorithm should detect. We have argued that this approach has serious flaws. Due to fundamental differences between classification and clustering, we cannot directly rely on classifications labels to identify clusters. Labels describe a single aspect of the data, while clustering aims to obtain a meaningful partition based on all available attributes. While it may be reasonable to assume that data with different labels belong to different clusters, elements with the same label may be meaningfully partitioned into multiple groups. Based on this observation, we develop the impurity measure for evaluating clustering quality on labeled data, and demonstrate how to utilize this measure for selecting a clustering algorithm.

We use our new impurity measure to demonstrate that the choice of clustering algorithm depends on the specific clustering task. Studying a set of well-known clustering algorithms, we show that each of them outputs the best clustering on some real data. These results demonstrate the need for a new research perspective: instead of searching for overall superior clustering algorithms, researchers should aim to provide users with tools for selecting a clustering algorithm according to their specific tasks.

We also propose a novel model-selection tool in the absence of data labels. This

tool is based on the stability of clustering over random perturbations of the input data set. Using labeled data, we show that sensitivity to data perturbation is positively correlated with our impurity measure. This indicates that the sensitivity measure can be used instead of impurity when no data labels are available. In particular, given a specific task, users can calculate the sensitivity of available algorithms and select the one that has the lowest sensitivity. In this work, we suggest not to consider the algorithms that return clusterings where at least half of the clusters are small since these clusterings may be robust to small perturbation regardless of how useful they are.

8.2 Future Work

Based on our experiments, we notice that hierarchical algorithms, especially Single Linkage and sometimes Average Linkage, produce clusterings where more than half clusters are small. Therefore, they are eliminated from our algorithm selection process. Perhaps, a small modification to their stopping criteria might resolve the issue. In particular, instead of the trivial criteria that is being used: the algorithm stops when k reaches the desired number of clusters K, we could let the algorithm run until either $k = K$ or K of the current clusters has at least $\sqrt{n/k}$ points.

How to select k is one of the very important questions that have been raised in literature. Unfortunately, it is still not fully answered. As one of our future research directions, we would like to investigate the relation between k and sensitivity. Since sensitivity can be evaluated without access to labels, the answer to this might provide users with practical tools to k selection.

K-means is known to be sensitive to initial choice of centroids. Often, users will have to run K-means several times, and perhaps pick the clustering with the lowest cost. By doing that, users seem to ignore the information that the other clusterings might contain. We are planning to investigate more on K-means' runs and hope to introduce an aggregation scheme which makes use of the information contained in the majority of the runs and returns a single clustering.

References

[1] A. Ben-Hur, A. Elisseeff, and I. Guyon. A stability based method for discovering structure in clustered data. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pages 6–17, 2002.

[2] Chen, G. Jaradat, S. A. Banerjee, N. Tanaka, T. S. Ko, M. S. H. Zhang, and M. Q. Evaluation and comparison of clustering algorithms in analyzing es cell gene expression data. *STATISTICA SINICA*, 2002.

[3] M. M. Erene. *Nearest Neighbor Clustering*, 1999. http://www.cse.iitb.ac.in/dbms/Data/Courses/CS632/1999/clustering/node21.html.

[4] G. Fung. A comprehensive overview of basic clustering algorithms, 2001.

[5] K. Hammouda and F. Karray. A comparative study of data clustering techniques. Technical report, University of Waterloo, Ontario, Canada.

[6] B. King. Step-wise clustering procedures. *Journal of the American Statistical Association*, 62:86–101, 1967.

[7] S. Lloyd. Least squares quantixation in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[8] The MathWorks Inc. *K-Means Algorithm Image.* http://www.mathworks.com/ help/toolbox/stats/kmeans.html.

[9] M. Meila. Comparing clusterings by the variation of information. In *Learning Theory and Kernel Machines*, volume 2777 of *Lecture Notes in Computer Science*, pages 173–187. Springer Berlin / Heidelberg, 2003.

[10] P. M. Murphy and D. W. Aha. *UCI Machine Learning Repository.* UC Irvine. http://archive.ics.uci.edu/ml/.

[11] S. Phillips. Acceleration of k-means and related clustering algorithms. In D. Mount and C. Stein, editors, *Algorithm Engineering and Experiments*, volume 2409 of *Lecture Notes in Computer Science*, pages 61–62. Springer Berlin / Heidelberg, 2002.

[12] G. Punj and D. W. Stewart. Cluster analysis in marketing research: Review and suggestions for application. *Journal of Marketing Research*, 20:134–148, 1983.

[13] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

[14] A. Rauber, J. Paralic, and E. Pampalk. Empirical evaluation of clustering algorithms. *Journal of Information and Organizational Sciences (JIOS*, 24:2000, 2000.

[15] Resampling Stats Inc. *Hierarchical Clustering.* http://www.resample.com/xlminer/help/HClst/HClst_intro.htm.

[16] R. R. Sokal and C. D. Michener. A statistical method for evaluating systematic relationships. *University of Kansas Scientific Bulletin*, 28:1409–1438, 1958.

[17] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques, 2000.

[18] W. Stuetzle. Estimating the cluster tree of a density by analyzing the minimum spanning tree of a sample. *Classification*, 20(5):25–47, 2003.

[19] W. Stuetzle and R. Nugent. A generalized single linkage method for estimating the cluster tree of a density. *Computational and Graphical Statistics*, 19(2):397–418, 2010.

[20] C. Tselikis, S. Mitropoulos, C. Douligeris, E. Ladis, K. Georgouleas, C. Vangelatos, and N. Komninos. Empirical study of clustering algorithms for wireless ad hoc networks. In *Systems, Signals and Image Processing, 2009. IWSSIP 2009. 16th International Conference on*, 2009.

[21] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl. Constrained k-means clustering with background knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 577–584. Morgan Kaufmann Publishers Inc., 2001.

[22] J. H. Ward. Hierarchical grouping to optimize an objective function. *American Statistical Association*, 58(301):236–244, 1963.

[23] G. A. Wilkin and X. Huang. A practical comparison of two k-means clustering algorithms. 2008.

[24] R. Xu and D. W. II. Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*, 16(3):645–678, 2005.

[25] M. Zat and H. Messatfa. A comparative study of clustering methods. *Future Generation Computer Systems*, 13(2-3):149 – 159, 1997.

[26] Y. Zhao, G. Karypis, and U. Fayyad. Hierarchical clustering algorithms for document datasets. *Data Mining and Knowledge Discovery*, 10:141–168, 2005.