COALA : A Novel Approach for the Extraction of an Alternate Clustering of High Quality and High Dissimilarity

Eric Bae and James Bailey
NICTA Victoria Laboratory
Department of Computer Science and Software Engineering
University of Melbourne, Australia
{kheb,jbailey}@csse.unimelb.edu.au

Abstract

Cluster analysis has long been a fundamental task in data mining and machine learning. However, traditional clustering methods concentrate on producing a single solution, even though multiple alternative clusterings may exist. It is thus difficult for the user to validate whether the given solution is in fact appropriate, particularly for large and complex datasets. In this paper we explore the critical requirements for systematically finding a new clustering, given that an already known clustering is available and we also propose a novel algorithm, COALA, to discover this new clustering. Our approach is driven by two important factors; dissimilarity and quality. These are especially important for finding a new clustering which is highly informative about the underlying structure of data, but is at the same time distinctively different from the provided clustering. We undertake an experimental analysis and show that our method is able to outperform existing techniques, for both synthetic and real datasets.

1. Introduction

As a fundamental data mining task, cluster analysis is extremely important. However, traditional clustering techniques focus on producing only a single solution, even though multiple alternate clusterings[1] may exist. It is thus difficult for the user to validate whether the given solution is in fact appropriate, particularly if the dataset is large and complex, or if the user has limited knowledge about the clustering algorithm being used. In this case, it is highly desirable to provide another, alternative clustering solution, which is high quality, yet different from the original solution. We illustrate the idea using two examples.

Example A *: Consider a mining task where multiple sources of data are combined, such as the merging of several protein datasets. Suppose a clustering exists for each data source. After merging, it is possible that several alternative clusterings might be present, each high quality, yet dissimilar to the others. Using a standard algorithm, it would be difficult, if not impossible, to extract more than one of these clusterings directly from the integrated data.*

Example B *: When searching for documents, a typical search engine may return a single clustering in which documents are organized by their topical differences. However, this may not provide the correct groups for the task. If a search engine allows its users to 'cluster again', by providing them a new clustering which categorizes documents differently, users may find their answer.*

These examples highlight the attraction of gaining different perspectives of the data, which may then lead to providing deeper insight of the data.

Challenges : The main difficulty of discovering high quality and dissimilar alternate clusterings stems from the unsupervised nature of cluster analysis and that there exists no easy definition of what exactly a cluster is. This naturally leads to clustering solutions being highly dependent on the similarity function implemented by the particular algorithm used [16]. As a result, if one is trying to find multiple clusterings by just naively applying a number of different clustering algorithms [22], the following difficulties present themselves :

- An inability to know which algorithms to apply and how many, hence a risk of clustering overload
- A risk of collecting highly similar clusterings
- The requirement of a compulsory post analysis to select the appropriate clusterings.

[1] A *clustering* is a set of clusters

- A difficulty in quantitatively evaluating the degree of (dis)similarity/quality for the candidate solutions.
- The inefficiency of running algorithms multiple times.

Indeed, naively trying different clustering algorithms is crude and far from systematic, if the user is expecting to gain different types of knowledge from the data. It may exhibit random and unpredictable behaviour, where the extraction process cannot be parameterized in a meaningful way to control the outcome. Furthermore, we have found that it is not just the naive approach which has drawbacks. Even a current state-of-the-art technique [11] does not always produce convincing results for this problem.

In this paper, we propose a systematic technique called COALA[2], to retrieve a new clustering which is distinctively different with respect to a pre-defined clustering that is provided as background knowledge. Our approach emphasizes the twin objectives of quality and dissimilarity. We experimentally show it can produce more accurate results than the most recent work in the area.

1.1. Overview of Our Approach

We now overview our approach in COALA, looking first at the dissimilarity requirement. We believe that the 'uniqueness' of each clustering is vital, if two or more clusterings are to be shown to the user. This leads us to our first requirement, the 'dissimilarity requirement'.

Dissimilarity requirement *: Given two clusterings C and S, they can be presented as solutions if they are as dissimilar from one another as possible.*

Our algorithm addresses this requirement via the use of instance-based 'cannot-link' constraints. This type of constraint has been proposed in *constraint clustering* [24]. In essence, given an existing clustering, our algorithm derives 'cannot-link' constraints and uses them to guide the generation of a new, dissimilar clustering. While the dissimilarity requirement addresses the issue of difference, presenting them is meaningless if they are not of high quality. Therefore, we impose a second requirement concerning the clustering quality.

Quality Requirement *: Given two clusterings C and S, they can be considered as solutions if they are both high quality clusterings.*

With our approach, the quality requirement is implicitly dependent on the distance function used by COALA to aggregate the closest objects together. Quality is governed by a pre-specified 'quality threshold', denoted by ω, which defines a numerical minimum bound on the quality required. For our purposes, the quality of a clustering can be quantitatively measured by use of the Dunn Index [7].

It is important to note that the two requirements can exhibit an inverse relationship. Suppose C is the pre-defined clustering, then if the quality of the new clustering S is increased, the dissimilarity between C and S may decrease and vice versa. For such a situation, the quality threshold ω plays an important role in balancing the trade-off between the two factors. Its influence on the two requirements will be discussed further in section 5.4.

With the two requirements in mind, we can now specify the target problem of our work as follows :

Problem definition *: Given a clustering C (provided as pre-defined class labels) with r clusters, find a second clustering S with r clusters, having high dissimilarity to C, but also satisfying the quality requirement threshold ω.*

We illustrate our overall objective with respect to these requirements in Fig. 1. Assume that the Fig. 1(a) was provided as background knowledge. If two alternate clusterings 1(b) and 1(c) were to be presented by COALA, then according to our problem definition, Fig. 1(c) would be selected as a preferred solution since it has higher quality (calculated by Dunn index) while it is also more dissimilar to clustering 1(a) than the clustering 1(c) is to 1(a). Of course, our problem definition can be extended to be more general and this is discussed in the future work section 6.

Contributions : Overall, our main contributions in this paper are as follows :

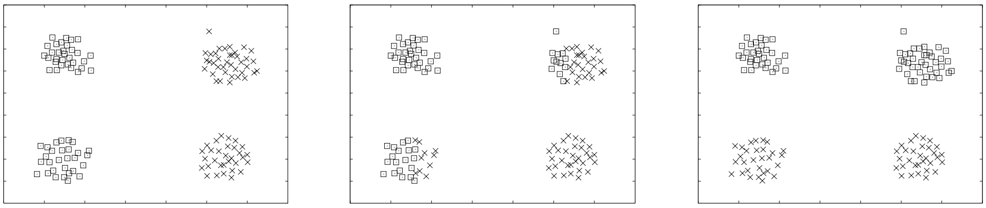
- We develop a novel algorithm, COALA, which incorporates automatically generated constraints to extract a new clustering with respect to a given clustering. This algorithm addresses both dissimilarity and quality requirements for the new clustering. We experimentally show it can outperform the state-of-the-art technique called CIB [11]. Furthermore, unlike [11], it does not require knowledge of a joint distribution for the data.[3]
- We offer the first (to our knowledge) combined quantitative measure of both quality and dissimilarity. This can used to give an overall score for the new clustering compared to the pre-defined one.

2. Related Work

Conditional Information Bottleneck : The most relevant work in retrieving dissimilar clusterings is called *con-*

[2] **Constrained Orthogonal Average Link Algorithm**, where the term 'orthogonal' refers to dissimilarity.

[3] Note that the extension to COALA, COALACat which handles categorical attributes actually requires the full dataset (much like CIB clustering in [11]). See section 4.1 for details.)



(a) Known clustering

(b) Extracted clustering A

(c) Extracted clustering B

Figure 1. Two possible alternative clusterings shown in 1(b) and 1(c), given the clustering 1(a) as background information. All clusterings have 2 clusters.

ditional information bottleneck (CIB) [11]. The technique uses the pre-defined class labels as additional information with which an alternate clustering is found. The underlying principal of this technique is based on *information bottleneck* (IB) in [21]. The general idea of IB is that given two variables (i.e. X representing objects, Y representing the features), the shared information between these two variables are maximized while one variable is compressed through another variable (i.e. C for clusters).

In [11], IB is extended by an introduction of another variable (i.e. Z representing the pre-defined class labels) in which the new objective is to find the optimal assignment of X to C while preserving as much information about Y conditioned on the information provided by the Z.

However, both IB and CIB methods must have joint distribution information for each variable which may not be available all the time. Moreover, a lack of related techniques has led to a limited comparison of the CIB method with any alternatives techniques.

Our method, on the other hand, is fundamentally different in its approach and ability to discover the second clustering. While in both techniques the background information is provided, COALA automatically generates constraints from the background knowledge provided while in [11], the specific usage and processing of the additional information are not specified. The CIB method can also be viewed as performing 'local-refinements' to the provided clustering and then merging the refined clusters to form a dissimilar clustering [11]. In contrast, we use a cannot-link constraint set to explicitly guide the clustering generation, giving more accurate results, while allowing users to balance between dissimilarity and quality through the quality threshold ω.

Clustering with background knowledge : A number of techniques have also utilized background knowledge to guide their clustering process. In constraint clustering [6, 24], knowledge is expressed as 'must-link' and 'cannot-link' constraints to produce more efficient and accurate clusters. In [3, 13], negative information about undesired structures or features is provided to ensure that clustering process avoids these information and focusing on the clusterings in 'positive' data. However, unlike COALA's automatic generation of constraints, these negative information is presumed to be provided by a manual process.

Ensemble Clustering : Generating multiple clusterings and merging them to offer a final consensus clustering is the objective of *ensemble clustering* [10] which we briefly described in the section 1.1. Ensemble clustering adopts several clustering generation methods which all can be considered as a naive method. These clusterings are typically generated by a) applying many algorithms, b) changing initial conditions of an algorithm and c) random samples of data. For the reasons already stated in section 1.1, however, these methods are unlikely to be effective in extracting high quality, dissimilar clusterings.

Feature Selection and Subspace Clustering : Finally, we note that feature-based methods, such as selecting certain features or applying dimension reduction methods are not practical. As explained in [12], such an attempt may cause useful information to be omitted and it is difficult to select associated features in a very high dimensional space. Furthermore, our method suggested here is different to the idea of subspace clustering [18]. Although subspace clustering uncovers a number of clusters from varying projections of features, the key difference is that we are discovering a completely new *clustering*, rather than just individual clusters. While it might be possible to create clusterings from subspace clusters, it is not at all obvious how to deal with problems such as subspace cluster overlap and duplication and we believe it to be a separate research issue.

3. Notations

Let $D = \{x_1, x_2, .., x_n\}$ be a set of n objects. Let C and S represent two clusterings, each partitioning D into r clusters ($C = \{c_1, c_2, ..c_r\}$ and $S = \{s_1, s_2, .., s_r\}$). We will typically use C to denote as the existing clustering that is provided as background knowledge and S as the new clustering retrieved by our technique with respect to C. Further, we denote $d(c_i, c_j)$ to be the distance between clusters c_i and c_j.

Cannot-link Constraints : A cannot-link constraint is a pair of distinct data objects (x_i, x_j) where $i \neq j$. For a clustering S to satisfy this type of constraint, the objects x_i and x_j *must not* be in the same cluster.

In our method, a set of cannot-link constraints, L, is automatically generated from the provided clustering C, prior to the actual clustering process of COALA (refer to Algorithm 1). These constraints are used to ensure that given two clusters s_i and s_j of S, they cannot be merged if they contain any pairs of objects which were from the same cluster in C.

4. COALA

Underlying model : COALA is built upon an agglomerative hierarchical clustering algorithm. This kind of algorithm typically starts by treating each object as a single cluster and then iteratively merges a pair of clusters which exhibit the strongest similarity. Upon each merge, the pairwise similarity between the newly formed cluster and each of the remaining clusters is then re-calculated.

Distance (similarity) function : The similarity function of hierarchical algorithms can be one of many different types (i.e. Euclidean distance, density, entropy) and methods (i.e. average distance, mutual information). Although many of them are effective, we use the *average-linkage* (AL) [23] algorithm to calculate the distance, because of its accuracy and robustness. The AL technique determines the similarity between clusters by calculating the average distance of all pairwise objects between clusters.

Preliminary process : An important component of COALA is the use of 'cannot-link' constraints to ensure that the second clustering S is dissimilar from the given clustering C. These constraints are generated prior to the actual clustering process and described in Algorithm 1). The algorithm creates one cannot-link constraint, per pair of objects which are in the same cluster in C[4]. We describe the role of these constraints in the next part of COALA.

Algorithm 1 GenerateConstraints

```
Require: clustering $C = \{c_1, c_2, .., c_r\}$, constraint set $L = \{\}$
for $i = 0$ to $r$ do
    for $j = 0$ to $|c_i|$ do
        for $k = j + 1$ to $|c_i|$ do
            $L = L \cup addConstraint(x_j, x_k)$ {add object pair $(x_j, x_k)$ to the set $L$, where $x_j, x_k \in c_i$}
        end for
    end for
end for
```

Merge candidate generation : Once the preliminary step is complete, COALA proceeds in an agglomerative fashion (algorithm 2), by first creating n clusters, with each cluster c_i containing a single object x_i. The algorithm then iterates until all objects are grouped together into one cluster (line 2). At each iteration, COALA finds two candidate pairs of clusters for a possible merge, one denoted as (q_1, q_2) (line 3), which we call a 'qualitative pair' and the other denoted as (o_1, o_2) (line 4), which we call a 'dissimilar pair'. The qualitative pair is the one with the minimum distance over all the pairs of clusters (ensuring the highest quality clusters when merged). The dissimilar pair has the minimum distance over all the pairs of clusters *that also satisfy the cannot-link constraints* (these pairs may be the same). COALA will select just one of these pairs to merge (discussed shortly).

The purpose of finding the dissimilar pair, is to achieve the dissimilarity of the clustering S, with respect to C, at each merge step. If we assume that the points in the qualitative pair (q_1, q_2) were in the same cluster in C, then by merging the dissimilar pair (o_1, o_2), we are avoiding the same grouping structures as C and constructing a dissimilar clustering S. Hence, if we continue to prefer the dissimilar pair over the qualitative pair, then we are 'progressively' building a clustering S which is dissimilar from C.

However, it is actually infeasible to always merge the dissimilar pair and at some point in the agglomerative process, we reach a point where no clusters actually satisfy the cannot-link constraints [5] in line 4. From this point on, we proceed with merges of the qualitative pairs.

Merge determination : It is not ideal, however, to always select the dissimilar pairs for merging, as this ignores the quality component. The ideal scenario would be to merge a dissimilar pair, whose quality (or in our case, distance calculated by AL algorithm), is also reasonable. On the other hand, regardless of how dissimilar the two clusters are, if their quality is poor, then selecting the qualitative pair would be more appropriate for the merge.

The quantitative determination of selecting which pair to merge, is made by comparing the distance between the qualitative pair $d(q_1, q_2)$ against the distance between the dissimilar pair $d(o_1, o_2)$ and comparing the ratio against the quality threshold ω (line 5) (which is a value between 0 and 1, and provided as an initial parameter to the algorithm). The rationale behind this comparison is to assure that the distance for the dissimilar pair is *at least* ω *close* to the distance for the qualitative pair. Therefore, if the dissimilar pair is too far apart with respect to ω, then it is more appropriate to retain quality by merging the qualitative pair. In fact, we can define two types of merges given two candidates (q_1, q_2) and (o_1, o_2) :

Qualitative merge : *qualitative merge is performed if*

[4] Through an efficient use of data structures and set functions available, one does not actually need to implement the algorithm exactly as written.

Algorithm 2 COALA

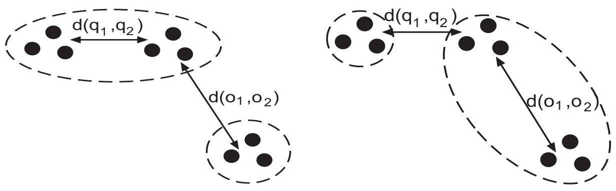
```
Require: dataset $D = \{x_1, x_2, .., x_n\}$, quality threshold $\omega$, constraint set $L$, $d(c_i, c_j)$ returns a distance between clusters $c_i$ and $c_j$
$c_i = \{x_i\}, \forall i$ where $1 \leq i \leq n$, $c_i \in C$ {$C$ is a set of clusters}
for $k = |D|$ to 1 do
    $(q_1, q_2) = minDist(c_i, c_j) \forall i, j$
    where $1 \leq i, j \leq n$
    {$minDist$ finds a pair of clusters with minimum average distance, $(q_1, q_2)$ is a qualitative pair}
    $(o_1, o_2) = minDist(c_i, c_j)$
    such that $satisfyConstraint(c_i, c_j, L)$
    {$(o_1, o_2)$ is a dissimilar pair}
    if $\frac{d(q_1,q_2)}{d(o_1,o_2)} \geq \omega$ then
        $c_{q_1} = merge(c_{q_1}, c_{q_2})$ {move all objects in $c_{q_1}$ to $c_{q_2}$}
        $remove(c_{q_2})$ {$c_{q_2}$ is now redundant}
    else
        $c_{o_1} = merge(c_{o_1}, c_{o_2})$
        $remove(c_{o_2})$
    end if
end for

function $satisfyConstraint(c_i, c_j, L)$
$satisfy = true$
for $k = 0$ to $|c_i|$ do
    for $l = 0$ to $|c_j|$ do
        if $(x_k, x_l) \in L$ where $x_k \in c_i$, $x_l \in c_j$ then
            $satisfy = false$
            $break$
        end if
    end for
end for
return $satisfy$
```

$\frac{d_q}{d_o} < \omega$. This means that merging (o_1, o_2) is expected to degrade the quality of S far more in relation to any dissimilarity gained. Therefore, it is better to merge (q_1, q_2) in order to retain quality of S.

***Dissimilar merge** : dissimilar merge is performed if $\frac{d_q}{d_o} \geq \omega$. This means that merging (o_1, o_2) is expected to retain the quality of S, while at the same time achieving dissimilarity from C.*

Therefore, by specifying different values of ω, we can control the degree of dissimilarity and quality. We illustrate the above two types of merges in Fig. 2. Assume that the current iteration step has generated two merge candidates - (q_1, q_2) and (o_1, o_2) - where the first pair has the closest distance (the qualitative pair), while the latter is the closest distance pair which also satisfy the cannot-link constraints (the dissimilar pair). We assume that $d(q_1, q_2) \leq d(o_1, o_2)$ since a qualitative pair is likely to have shorter distance than the dissimilar pair whose distance depends on the constraints.



(a) qualitative merge (b) dissimilar merge

Figure 2. Comparing 'qualitative merge' and 'dissimilar merge'. Figure 2(a) emphasizes the similarity between two clusters with a high ω value, while Fig.2(b) highlights merging dissimilar clusters led by a low ω value.

If ω was set to a relatively high value, then we are effectively emphasizing quality for the clustering S more than the dissimilarity. This means that in Fig. 2, $\frac{d(q_1,q_2)}{d(o_1,o_2)}$ may not be greater than ω and therefore COALA does not select (o_1, o_2) to merge. In other words, although this pair satisfies the dissimilarity requirement via cannot-link constraints, with respect to ω, the quality of the clustering would be degraded too much. Thus COALA proceeds with the pair (q_1, q_2) (qualitative merge in Fig. 2(a)).

In contrast, setting ω to a low value relaxes the quality requirement and focuses more on the dissimilarity requirement. Therefore, the dissimilar pair (o_1, o_2) is more likely to be chosen for the merge. We later investigate the influence of the threshold value ω, in our experimental analysis.

Overall, the behaviour COALA can be 'tuned' by the user through different values of the threshold. Indeed, by applying various ω values and by using quantitative measures of dissimilarity and quality, users can actually learn whether the new clustering found is of any value. Of course it is also quite reasonable for the user to use a default value for ω. In fact, this is what we do in practice for our experiments, setting $\omega = 0.6$.

4.1. COALACat

Although similarity functions based on geometric distances work well for numerical data, it is often true that datasets contain categorical attributes, whose values cannot be naturally ordered in a metric space. Therefore, cluster analysis of categorical values has been studied extensively and there are numerous methods to handle the problem [4, 14]. The COALA algorithm faces a similar problem with categorical attributes and in this section we show how to modify our algorithm to handle this type of data.

The extended algorithm called, COALACat (COALA-Categorical), and implements a similarity measure based on the entropy of clusters, as introduced by the ACE algorithm [4]. It was shown in [4], that ACE surpasses many of the traditional categorical clustering algorithms in accuracy. Moreover, as the algorithm uses a bottom-up hierarchical method and has a simple, intuitive evaluation of cluster entropy values, it is a good choice to use for extending COALA.

The merge procedure of COALACat is identical to that of COALA in Algorithm 2. However, its similarity function is based on the overall expected information or entropy of clusters. Extending the notation mentioned in section 3, let us now consider dataset $D = \{x_1, x_2, .., x_n\}$ of n objects described by r attributes $\{a_1, a_2, .., a_r\}$. Let $x_i\,[a_j]$ refer to the value of object x_i on attribute a_j. Assume that for each attribute value a_j (where $1 \leq j \leq r$), a_j takes a value from its domain A_j. There are a finite number of distinct categorical values in domain A_j and the number of distinct values is denoted as $|A_j|$. Let $p(x_i\,[a_j] = v), v \in A_j$, represent the probability of $x_i\,[a_j] = v$. The classical definition of entropy for such the sample set D is as follows:

$$H(c) = -\sum_{j=1}^{d} \sum_{v \in A_j} p(x_i\,[a_j] = v|D) log_2 p(x_i\,[a_j] = v|D) \tag{1}$$

which calculates the amount of 'expected information' contained in cluster C where we assume that $x_i \in c$. The intuition for the above function is that if two clusters have similar distributions of categorical values, then merging them together will not distort the $H(c)$ dramatically, but combining two dissimilar distributions causes higher distortion. Hence at each iteration, a pair of clusters which has the minimum amount of 'information distortion' is merged.

$$min \Delta H(c_{new}) = H(c_{i,j}) - H(c_i) \tag{2}$$

The rest of the COALACat algorithm to find a high quality, dissimilar clustering with constraints using threshold ω is identical to that of COALA. Finally, as mentioned in section 1.1, COALACat requires the original dataset to measure the entropy values (much like CIB clustering in [11]) while COALA functions straight from the similarity association matrix (for numerical attributes).

4.2. Quantitative Evaluation

Once a new clustering is found, it is important to evaluate the dissimilarity and quality in a quantitative manner, and we provide these measures in this section.

Dissimilarity : A number of measures exist for comparing similarity/dissimilarity between two clusterings. We have chosen to use the Jaccard index [19], which is a well known measure based on 'pair-counting' technique, that observes object-to-cluster assignments between two clusterings. It is defined by the function below :

$$J(C, S) = \frac{N_{11}}{N_{11} + N_{01} + N_{10}} \tag{3}$$

where N_{11} is the number of pairs of points in the same cluster for both C and S and N_{00} measures the number of pairs that are in different clusters in C and S. N_{01} and N_{10} are the number of pairs where a pair belongs to the same cluster in one clustering, but not the other. This effectively measures a ratio between the 'agreement' and 'disagreement' between clusterings. Jaccard returns a value between 0 and 1, where a higher value indicates higher dissimilarity.

Quality : To quantitatively measure the quality of a clustering, we employed a generalized Dunn index [7], which has proved to be an effective measure in Bezdek's experiments [2] compared to others. It is defined as follows:

Dunn index *: Let $C = \{c_1, .., c_k\}$ be a clustering, $\delta : C \times C \rightarrow R_0^+$ be a cluster-to-cluster distance and $\Delta : C \rightarrow R_0^+$ be a cluster diameter measure, then Dunn index is*

$$DI(C) = \frac{min_{i \neq j}\,\{\delta(c_i, c_j)\}}{max_{1 \leq l \leq k}\,\{\Delta(c_l)\}} \tag{4}$$

Higher values of the index indicate higher quality. For COALACat which takes categorical values, we have calculated the average information distortion of each clustering, instead of the Dunn index.

Overall Clustering Score : We also propose to combine the two measures and provide an overall score on the clustering generated. As mentioned earlier, the quality and dissimilarity requirements may share an inverse relationship which leads us to adapt a widely used metric called F-Measure [15]. This has been traditionally applied to information retrieval systems, to coalesce the precision and recall values and calculate the harmonic mean to act as an overall score. This measure has also been used in [8], for other clustering contexts. Following the definition of F-Measure [15], we define the overall DQ-Measure as below.

$$DQ(C, S) = \frac{2J(C, S)DI(C, S)}{J(C, S) + DI(C, S)} \tag{5}$$

where J corresponds to Jaccard index (dissimilarity) and DI refers to Dunn index (quality). The $DQ(C, S)$ works well for our purpose, capturing the inverse relationship between two variables effectively and in our case, indicates the validity of the clustering in terms of both dissimilarity and quality requirements.

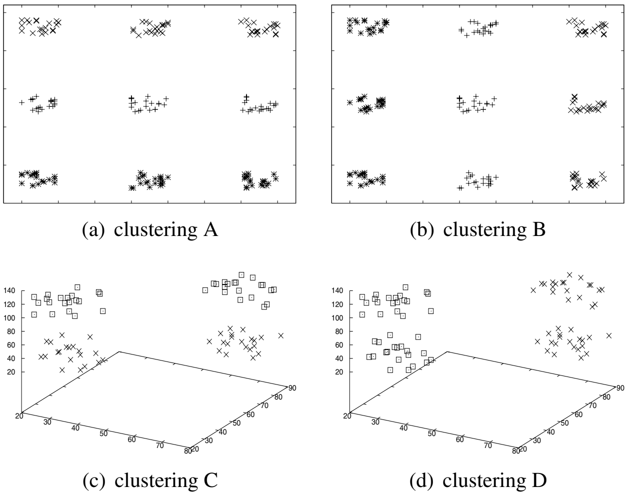
5. Experiments

For our experimental analysis, we have implemented two competing approaches, the naive method and CIB [11], as well as our COALA approach (and COALACat). The naive method takes the clustering generation approach used in ensemble clustering [10] which we described in section 1.1. For our purpose, the behaviour of the naive technique is to apply the k-means algorithm multiple times to the dataset, with each application parameterized by different random initial points. It is worth noting, however, that this naive method does not exactly solve the problem of finding a new clustering, given an existing one. Rather, it separately generates two clusterings from scratch. When comparing the algorithms, we used Jaccard index, Dunn index and DQ-Measure to compute dissimilarity, quality and overall score respectively, for the new clustering retrieved.

Our experiments are organized as follows. Firstly, we tested numerical datasets (synthetic and real world), to compare COALA with the other two methods. Secondly, we clustered categorical datasets with COALACat. Thirdly, we investigated the inverse relationship between the dissimilarity and quality. In all experiments, a default quality threshold of $\omega = 0.6$ was used, a value which we have found gives effective results.

5.1. Synthetic Datasets

Four synthetic datasets were prepared for the experiments. We constructed each dataset to contain two different clusterings (similar to Fig. 1). The datasets, however, differed in their dimensionality and also in the number of clusters they contained (refer to Fig. 3).



(a) clustering A

(b) clustering B

(c) clustering C

(d) clustering D

Figure 3. Visual representation of some of the synthetic data used.

For the naive method, we applied k-means algorithm three times using different initial points (therefore generating three clusterings). By comparing pairwise clusterings, we selected the two clusterings which returned the highest DQ-Measure score. Since the naive method does not use the background information, a clustering which is of higher quality was selected as a 'known' clustering. Our objective in this experiment was to validate whether the algorithms are able to correctly extract the two clusterings included in the dataset. The result of the experiment is shown in Fig. 4.

It can be seen that CIB and COALA all perform well by finding the correct clustering when applied to these datasets, with COALA performing slightly better, since a few points were not clustered as expected with CIB. On the other hand, the results clearly highlight the nature of naive method, which does not correctly deduce hidden structures. The only datasets for which it was able to find the expected clustering were A and C. For dataset B, an incorrect clustering was retrieved, while for D it retrieved the exactly same clustering. This arises because the naive method is unable to control the extraction process.

5.2. Real World Data

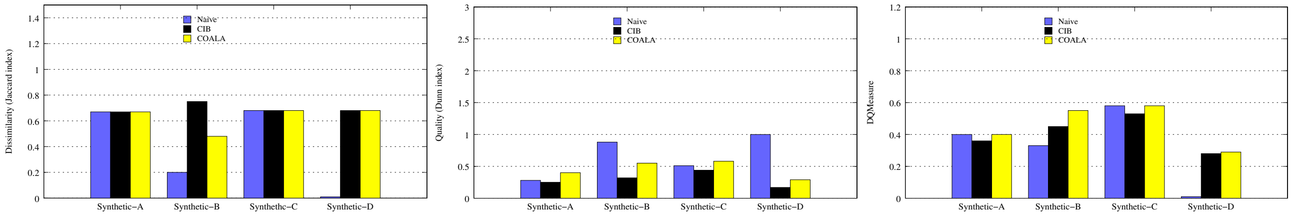
We also examined the performance of the three approaches on a number of real world data sets. Figure 5 shows the comparisons with four datasets (ESL, glass, vehicle, ionosphere). These datasets already have pre-defined class labels, which were supplied to COALA and CIB as the existing clustering C to generate an alternative clustering S.

Figure 5 clearly shows that COALA outperforms its rivals in all cases in terms of the overall DQ-Measure. For the dissimilarity and quality of the retrieved clusterings (Fig. 5(a) and 5(b)), COALA extracts high quality clusterings while its dissimilarity is also relatively high. The overall DQ-Measure in Fig. 5(c) clearly indicates the superior performance of COALA over the naive method and CIB.

The naive method again gives unstable results, for some datasets it retrieves a clustering of reasonable dissimilarity and quality (i.e. ESL) yet in other cases, it does not extract any new information from data (i.e. ionosphere, where the dissimilarity is actually zero).

Furthermore, when we studied further the new clusterings returned by COALA, it was interesting and unexpected to discover that in nearly all datasets, COALA actually extracted a clustering which was of higher quality than the pre-defined clustering provided. Such a clustering can be extremely valuable as it offers not only new information, but better grouping structures underlying the data.

Finally, we can consider the new clustering found by COALA as an additional way to group the data objects. For example, in the dataset vehicle, the given class labels organize vehicles into either 'OPEL', 'SAAB', 'BUS' and 'VAN' classes. However, these labels cannot highlight common properties shared by vehicles across different classes. The additional clustering offered by COALA consists of

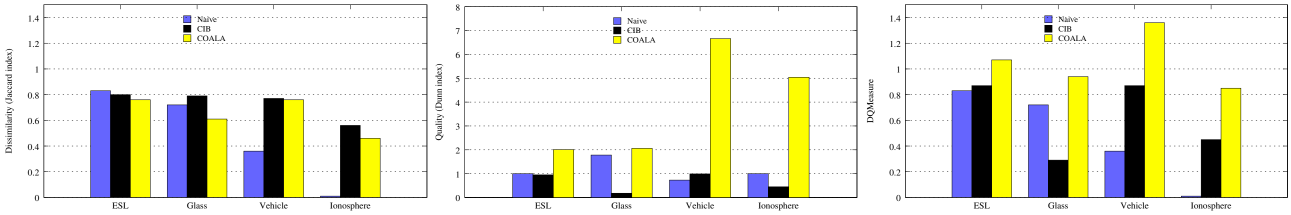


(a) Dissimilarity comparison

(b) Quality comparison

(c) DQ-Measure comparison

Figure 4. Comparison of three algorithms applied to four synthetic datasets.



(a) Dissimilarity comparison

(b) Quality comparison

(c) DQ-Measure comparison

Figure 5. Comparison of three algorithms applied to four real-world datasets.

new groups of vehicles which are somewhat at the 'secondary level', yet this still contains valuable information. This can be also viewed as effectively emphasizing another subset of features giving rise to a different clustering which normally would not surface, because of a more dominant clustering present on other features.

5.3. Experiments for COALACat

For COALACat, we used two categorical datasets - 'vote' and 'breast-cancer'. Unlike the Dunn index, a low value in the entropy-based quality measure indicates a good clustering. Therefore, this value was appropriately normalized ($\frac{|max+\Delta Q|}{max}$), where max is the upper bound and ΔQ is the difference in average entropy values. The results are shown in Fig. 6 and show that even for these categorical datasets, COALACat is providing clusterings which are highly dissimilar and of good quality, giving the better results overall compared to naive and CIB.

5.4. Impact of Quality Threshold ω

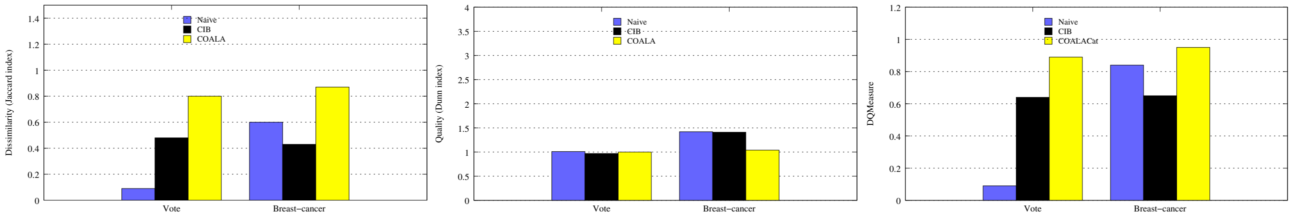
We described in sections 1.1 and 4 the inverse relationship between the two requirements which is controlled by the quality threshold ω. We now study the influence of this threshold in more detail. We applied COALA to the real world datasets - 'ESL', 'glass' and 'vehicle - varying the ω threshold value from 0 to 1. Figure 7(a) shows the increase in the quality of clustering retrieved, as the threshold increases, while Fig. 7(b) shows the decrease in the dissimilarity between two clusterings as the threshold increases. These figures support our definitions of qualitative merge and dissimilar merge stated in section 4, where emphasizing one requirement effectively degrades the other. This inverse relationship is also supported by the decreasing slope of 'dissimilarity vs. quality' graph displayed in Fig. 7(d), 7(e) and 7(f). While it is possible to try varying values of threshold, we have found that in practice, setting $\omega = 0.6$ offered the best results for both dissimilarity and quality.

Finally, as we will discuss in the following section, the quality and dissimilarity measures themselves have limitations and this could explain the occasional inconsistent 'rises' and 'falls' of the values in these graphs.

6. Discussions and Future Work

Advantages of COALA : Our experimental results indicate that COALA is more effective than other approaches. Also, while the current state-of-the-art technique CIB requires the joint distribution, COALA only requires a similarity function between pairs of points and uses a readily available agglomerative hierarchical algorithm. Moreover, with cannot-link constraints and the quality threshold, COALA is able to derive a clustering of high quality and at the same time dissimilar from the existing clustering. We have also extended COALA to handle categorical attributes in COALACat which also produced more accurate results over other methods tested.

Performance of COALA : Despite the flexible and intuitive clustering process, hierarchical algorithms are characterized by a high complexity [17]. In COALA, generating cannot-link constraints (algorithm 1) takes $O(n^2)$. The it-

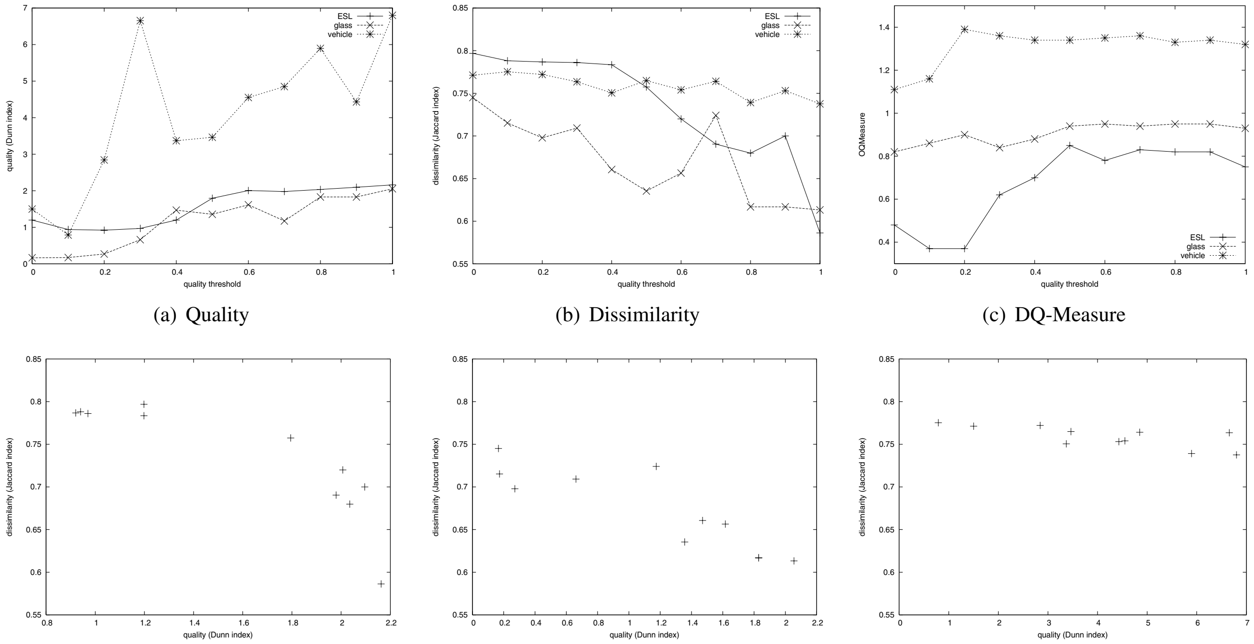


(a) Dissimilarity comparison

(b) Quality comparison

(c) DQ-Measure comparison

Figure 6. Comparison of three algorithms applied to two real-world categorical datasets.



(a) Quality

(b) Dissimilarity

(c) DQ-Measure

(d) ESL - Quality & Dissimilarity.

(e) glass - Quality & Dissimilarity.

(f) vehicle - Quality & Dissimilarity.

Figure 7. Dissimilarity, quality and overall score given by Jaccard index, Dunn index and DQ-Measure respectively for three datasets as the quality threshold ω value changes.

eration and merge steps typically take $O(dn^2 + n^2 \log n)$ where d is the cost of calculating the distance function. Calculation of the distance function takes $O(n^2)$ complexity followed by $O(n^2)$ selection steps, each having cost of $O(\log n)$. Validating whether a pair of clusters satisfy constraints also takes $O(N^2)$ time. In a typical setting, where we consider d to be constant, or very small w.r.t. $\log n$, we can simplify the overall process of COALA to $O(n^2 \log n)$.

To overcome this high complexity, a number of extensions have been proposed, such as using a new data structure called quad tree [9] or applying a parallel clustering technique [20]. Furthermore, employing other more efficient clustering models (i.e. partitioning algorithms - k-means) may also enhance the performance and accuracy of COALA. Moreover, selecting an appropriate distance function remains as a non-trivial task. One might consider more sophisticated methods which combine multiple functions [16], or applying techniques to learn about distance functions through various means [25].

Measuring Dissimilarity and Quality : In the experimental section, we have used the Jaccard and Dunn indices to evaluate the dissimilarity and quality of clusterings, but these measures also have their drawbacks.

The Jaccard index is limited in only considering the point-to-cluster assignments, while there are other factors that could differentiate clusterings, such as cluster centroids and density profiles [26]. Therefore, utilizing these various factors can lead to more accurate comparisons.

Dunn index has been effective for measuring quality, but it is known to be overly sensitive to outliers and prefers compact and well-separated clusters [1]. In fact, we have seen some inconsistencies in Fig. 7, where the increase in the quality as the ω value increases is, sometimes not continuous. For future work, we would like to investigate other measures for validating quality. Lastly, in our experiments, we assumed that the alternate clustering has the same number of clusters as the pre-defined clustering. However, in future, we would like to extend COALA to handle varying numbers of clusters between the new and old clusterings.

Extraction of Multiple Clusterings : In this paper we only considered a task of extracting a single alternate clustering S, with respect to the given clustering C. However, it is certainly possible for several clusterings to be present in data and therefore, it would be useful to discover multiple alternate clusterings. Intuitively, extracting multiple clusterings would be carried out by recursively applying COALA while accumulating cannot-link constraints generated at each iteration. Of course at some point it would not be ideal to continue generating new clusterings, as their quality may dramatically decrease after all important relationships are exhausted from data.

7. Conclusion

We have described a new system, COALA, which generates a new clustering, with respect to a pre-defined, existing clustering. This is an important problem which has not been dealt systematically in previous work.

COALA addresses both dissimilarity and quality requirements for the new clustering and we have experimentally shown it outperforms other techniques. We also offered a combined quantitative measure of both quality and dissimilarity and showed that it is a reasonable and effective way to evaluate clusterings. We described some limitations of the current COALA system and identified a number of interesting avenues for extension.

Acknowledgement This work is partially supported by National ICT Australia. National ICT Australia is funded by the Australian Government's Backing Australia's Ability initiativem in part through the Australian Research Council.

References

[1] J. Bezdek and N. Pal. Some new indexes of cluster validity. *Sys., Man and Cybernetics*, 28(3):301–315, 1998.

[2] J. Bezdek, L. Wanquing, Y. Attikiouzel, and M. Windham. A geometric approach to cluster validity for normal mixtures. *Soft Computing*, pages 166–179, 1997.

[3] G. Chechik and N. Tishby. Extracting relevant structures with side information. In *Advances in Neural Info. Processing Systems*, pages 857–864. MIT Press, 2003.

[4] K. Chen and L. Liu. The "best k" for entropy-based categorical data clustering. In *Inter. Conf. on Scien. and Stat. Database Management*, pages 253–262, 2005.

[5] I. Davidson and S. Ravi. Clustering with constraints: Feasibility issues and the k-means algorithm. In *SIAM Intern. Conf. on Data Mining*, 2005.

[6] I. Davidson and S. Ravi. Identifying and generating easy sets of constraints for clustering. In *AAAI Conference*, 2006.

[7] J. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. In *Journal of Cybernetics*, pages 32–57, 1974.

[8] S. Eissen and B. Stein. Analysis of clustering algorithms for web-based search. In *International Conference on Practical Aspects of Knowledge Management*, pages 168–178, 2002.

[9] D. Eppstein. Fast hierarchical clustering and other applications of dynamic closest pairs. In *SODA: ACM-SIAM Symposium on Discrete Algorithms*, page 1, 1998.

[10] A. Fred. Finding consistent clusters in data partitions. In *Multiple Classifier Systems*, pages 309–318, 2001.

[11] D. Gondek and T. Hofmann. Conditional information bottleneck clustering. *Intern. Conf. on Data Mining*, 2003.

[12] D. Gondek and T. Hofmann. Non-redundant clustering with conditional ensembles. *International Conference on Knowledge Discovery and Data Mining*, pages 70–77, 2005.

[13] D. Gondek, S. Vaithyanathan, and A. Garg. Clustering with model-level constraints. In *SIAM International Conference on Data Mining*, 2005.

[14] S. Guha, R. Rastogi, and K. Shim. ROCK: A robust clustering algorithm for categorical attributes. *Information Systems*, 25:345–366, 2000.

[15] B. Larsen and C. Aone. Fast and effective text mining using linear-time document clustering. In *Intern. Conf. on Knowledge Discovery and Data Mining*, pages 16–22, 1999.

[16] M. Law, A. Topchy, and A. Jain. Multiobjective data clustering. In *Computer Society Conference on Computer Vision and Pattern Recognition*, pages 424–430, 2004.

[17] M. Nanni. Speeding-up hierarchical agglomerative clustering in presence of expensive metrics. In *PAKDD*, pages 378–387, 2005.

[18] L. Parsons, E. Haque, and H. Liu. Subspace clustering for high dimensional data: a review. *SIGKDD Explor. Newsl.*, 6(1):90–105, 2004.

[19] A. Patrikainen. Methods for comparing subspace clusterings, master's thesis, www.cis.hut.fi/ annep/lisuri.pdf, 2005.

[20] S. Rajasekaran. Efficient parallel hierarchical clustering algorithms. *Parallel Distrib. Syst.*, 16(6):497–502, 2005.

[21] N. Tishby, F. Pereira, and W. Bialek. The information bottleneck method. *Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999.

[22] A. Topchy, H. Martin, C. Law, A. Jain, and A. Fred. Analysis of consensus partition in cluster ensemble. In *Intern. Conf. on Data Mining*, pages 225–232, 2004.

[23] A. Voorhees. Implementing agglomerative hierarchical clustering algorithms for use in document retrieval. *Info. Processing and Management*, pages 465–476, 1986.

[24] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained k-means clustering with background knowledge. In *Intern. Conf. on Machine Learning*, pages 577–584, 2001.

[25] E. Xing, A. Ng, M. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. *Advances in NIPS*, 15:505–512, 2002.

[26] D. Zhou, J. Li, and H. Zha. A new mallows distance based metric for comparing clusterings. *International Conference on Machine Learning*, pages 1028–1035, 2005.