Finding Alternative Clusterings Using Constraints

Ian Davidson Zijie Qi

Abstract

[1]

The aim of data mining is to find novel and actionable insights. However, most algorithms typically just find a single explanation of the data even though alternatives could exist. In this work, we explore a general purpose approach to find an alternative clustering of the data with the aid of must-link and cannot-link constraints. This problem has received little attention in the literature and since our approach can be incorporated into many clustering algorithm that uses a distance function, compares favorably with existing work.

1. Introduction and Motivation

Consider the following situation practitioners typically find themselves in: you have a collection of data to cluster and use your favorite algorithm, A, which you know tries to optimize function f. Algorithm A finds clustering π whose objective function value is $f(\pi) = x$. Upon examining π the clustering makes sense, for example, when clustering face image data a natural clustering arises along the lines of gender. However, a valid question is then: "Does there exist another clustering which is different to π but has an objective function value approximately equal to x?" That is, does there exist an alternative but equally good clustering?

Conversely, you may also find the clustering found by A not particularly useful and actionable and wish to find an alternative to it. Consider clustering loan applications to determine a method to identify bad loans but the clusters fall along racial lines. You may wish to find another alternative but equally good clustering. Searching for alternative clusterings, is quite reasonable since data in high dimensional space may have many alternative clusterings that exploit a different subset of features. Similarly, the underlying phenomena in the data could be quite complex and the data chosen to represent it insufficient to justify one single explanation. In both these and other possible situations, the objective function f is not uni-modal and our aim is to find alternative modes/clusterings. We should emphasize that we do not seek slight variations of existing clusterings, rather completely different clusterings.

We can formulate the above objective as the Generalized Alternative Clustering Problem (GA-CP).

Problem 1 ***Generalized Alternative Clustering Problem (GA-CP)*** *Given an algorithm A with an objective function f, an **existing** clustering π such that $f(\pi) = x$. Does there exists another clustering π' that is different to π and where $f(\pi') \approx f(\pi)$?*

One potential approach to the GA-CP is to construct an algorithm with a dual objective function that simultaneously attempts to search for a different and good clusterings. This approach was taken by Bae and Bailey [1] and Gondek and Hoffman [11] with considerable success. However, the dual objective function approach ties their approaches to a particular algorithm which may or may not be suitable for the task at hand. Another approach is to generate many clusterings and then sort out which are truly different as is the case with Meta-Clustering [12] but this is not efficient for large data sets. Still a third alternative is to remember previous clusterings and steer a sampling/search algorithm away from them [7], but this requires creation of a Markov chain Monte Carlo (MCMC) sampler which will require considerable time to reach equilibrium. Finally, a fourth approach is to project the data into an alternative **sub-space** [5] but as we shall see this also has limitations such as not being appropriate for lower dimensional data sets such as spatial data.

We propose a general purpose and efficient method of addressing GA-CP that is algorithm independent and in particular is appropriate for low dimensional data. We propose using instance level constraints [15] to characterize the existing clustering, learn a distance function from those constraints and then performing a singular value decomposition of this function so as to transform the data to rule out the previously founding clustering, but maintain the inherent structure in the data. Instance level constraints state two instances must be in the same cluster (must-link) or must not be in the same cluster (cannot-link).

Formally, our aim is to create an approach that:

- Is general purpose and can address the GA-CP prob-

[1] To Appear in the IEEE ICDM 2008 Conference, Bari Italy

lem for a variety of distance based clustering algorithms

- Can specifically identify what properties/parts of the clustering to find an alternative to
- Is efficient and easily implementable
- Is usable for low dimensional (i.e. spatial) data sets

We begin this paper by describing in detail our approach in section 2. We then discuss its properties in section 3. We show its usefulness in section 4 on several UCI data sets comparing the approach to several others for non-hierarchical clustering. We then illustrate the general purposeness of our approach by addressing the novel problem of finding alternative clusterings for agglomerative algorithms in section 5. To our knowledge finding alternative clusterings for agglomerative clustering has not been addressed. We then verify our approach produces intuitive results for several real world problems including finding alternative spatial clusterings of pandemic simulation data in section 6.2. We then summarize and compare our approach to related work in section 7 and then conclude.

2 The Alternative Distance Function Transformation (ADFT) Approach

To achieve our aim of finding an alternative clustering in a general purpose manner we explore a data transformation approach that converts the original data X to a new space X' using a distance function D'. We note that either the existing data X can be transformed to a new space X' by $X' = D'^T X$ and the clustering algorithm uses the Euclidean distance function or the old data X can be used but the algorithm uses the distance function D'. At a high level our approach can be summarized as the following steps:

1. *Initial Application Step:* Apply a clustering algorithm A to X and obtain clustering π.
2. *Characterizing Step Using Constraints:* Identify and specify composition properties of π using a set of must-link and cannot-link constraints (C). For example, we can specify must-link (cannot-link) constraints between all points in the same (different) cluster(s) and then learn a distance function D_π from C. We focus on constraint based function learning techniques as they allow us the flexibility to focus on different properties of the clustering.
3. *Alternative Calculation Step:* Find an alternative distance function D'_π from D_π.
4. *Transformation Step:* Transform the data set according to D'_π.
5. *Re-clustering Step:* Re-cluster the data using the newly transformed data.

.

We now provide the details for the more complicated steps.

2.1 The Mathematical Formulation

The input for a clustering algorithm A is a set of points X typically in the Euclidean space. The clustering algorithm then returns a set partition π of those points which can be interpreted as that *points in the same cluster are similar and those in different clusters dissimilar*. This partitioning of points could be interpreted as a transformation on the points $X \rightarrow X_\pi$ such that those points in the same cluster are similar (close together) and those in different clusters are different (far apart).

Our aim is to quantify this transformation that the clustering implicitly performs and then construct a **alternative transformation** which can be applied to the data points and then reapply the clustering algorithm to the newly transformed data set.

The Characteristic Step. We can characterize the transformation that π represents using any number of approaches with each emphasizing a different characteristic. For example, we could use linear discriminant analysis (LDA) to learn a transformation with each instance in a given cluster having its own label. However, in this paper we shall explore converting the characteristics/composition of π to a set of instance level constraints (must-link and cannot-link) [15] and then learning a distance function from the constraints. A valid question is then why not just cluster under the "flipped" constraints for the same value of k to find an alternative clustering. That is if $\pi = \{(a, b), (c, d)\}$ then this clustering can be uniquely represented as the constraints must-link(a,b), must-link(c,d), cannot-link(b,d) and cannot-link(b,c) (not all entail constraints are provided for clarity). However, flipping these constraints for even this simple four point data set produces cannot-link(a,b), cannot-link(c,d), must-link(b,d) and must-link(b,c) for which no clustering exists that satisfies all constraints. For similar reasons it is not desirable to learn a distance function from the flipped constraints due to the many inconsistent constraints that flipping could generate. Furthermore, even if a set of non-contradictory constraints could be generated, then trying to find just a single clustering to satisfy them is known to be **NP**-complete [8] for any constraint type combination involving cannot-link constraints. This is a large hurdle since we most certainly wish to generate must-link constraints from points in the same cluster and when flipping them will produce the undesirable cannot-link constraints.

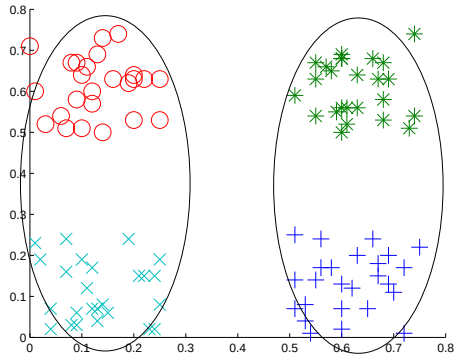

Figure 1. The data and clustering π in the original space.

Instead we can use a variety of algorithms that learn a distance function from constraints [17]. Throughout this work we use the approach of Xing and collaborators [16] to learn our distance functions from the complete set of constraints generated from the clustering π. We leave other variations to future work such as generating constraints only from the undesirable parts of the clustering.

To illustrate how a distance function D_π is representative of π consider Figure 1. The original data consists of four sub-populations from which the algorithm finds the vertical clustering shown by the ellipses. This clustering can be represented by the transformation matrix D_π below since it transform the data so that the points in the same cluster are together and those in different cluster are far apart. See the left image in Figure 2 to see the effect of transforming the data using this matrix.

$$D_\pi = \begin{bmatrix} 1 & 0 \\ 3 & 1 \end{bmatrix}$$

The Alternative Calculation Step. Given the characteristic distance function D_π interpreting what this transformation is and how to find an alternative transformation is difficult. To help our understanding we can decompose D_π using singular value decomposition (SVD) such that $D_\pi = HSA$ where H is the hanger matrix, S the stretcher matrix and A the aligner matrix. For example using our example we find:

$$D_\pi = \begin{bmatrix} 1 & 0 \\ 3 & 1 \end{bmatrix} = HSA$$

$$\approx \begin{bmatrix} 0.96 & -0.29 \\ 0.29 & 0.96 \end{bmatrix} \begin{bmatrix} 3.3 & 0 \\ 0 & 0.3 \end{bmatrix} \begin{bmatrix} 0.29 & -0.96 \\ 0.96 & 0.29 \end{bmatrix}$$

Although SVD is used extensively in areas such as latent semantic index by the mining community the geometric interpretation of the results are often overlooked and we now explain them. The transformation that D_π performs can be decomposed into three separate transformations that are applied one after the other reading right to left. The aligner matrix (A) creates a new orthonormal basis with each basis being written as a row vector in A. In our example the data is re-aligned nearly 90 degrees clock-wise and we can visualize the new basis by tilting our heads to the right when looking at the image in Figure 1. The stretcher matrix (S) is a diagonal matrix with the entry $s_{i,i}$ stretching (if the entry is greater than 1) or compresses (if the entry is less than 1) along the i^{th} dimension in the **new basis**. In our example it stretches about 3 times along the new x dimension and compressing the data nearly a third along the new y dimension. Finally the hanger matrices comprises of column vectors that rotate the data in the direction of the vectors.

Given this decomposition of D_π we now wish to create an alternative distance function to cluster under. When creating this new function we wish to make sure the algorithm does not rediscover the clustering π. To achieve this we wish to make those points that were close together in the same cluster far apart and those far apart close together since we generated both must-link and cannot-link constraints from π. However, in performing this transformation it is important not to completely destroy the inherent structure in the data, otherwise the resultant clustering would be not particularly useful. To achieve the above we keep the same alignment and rotation but can "flip" the stretching and the compressing by using the inverse of S. Therefore, we can state that the new transformation is:

$$D'_\pi = HS^{-1}A \tag{1}$$

We shall see that this computation has an interpretation in a system of linear equations context in section 3. We can then transform the data to new positions using $X' = D'^T_\pi X$. In our example the new positions of the points are shown in Figure 2 (middle) and we can see that the points in the same clusters are now far apart and vice-versa.

A valid question is why not use the orthogonal projection (defined as $D' = I - D(D^TD)^{-1}D^T$) as used by others [5] with the substitution $D = D_\pi$. We see that when doing this the orthogonal projection (Figure 2 right) is less efficient at "reversing" the transformation that represents π. Furthermore, when we do learn a distance function from the vertical clustering the typical function returned is:

$$D_\pi = \begin{bmatrix} 3 & 0 \\ 0 & \frac{1}{3} \end{bmatrix}$$

which stretches the data three times along the $x - axis$ and compresses the data by one third along the $y - axis$. The transformation shown by our approach and the orthogonal

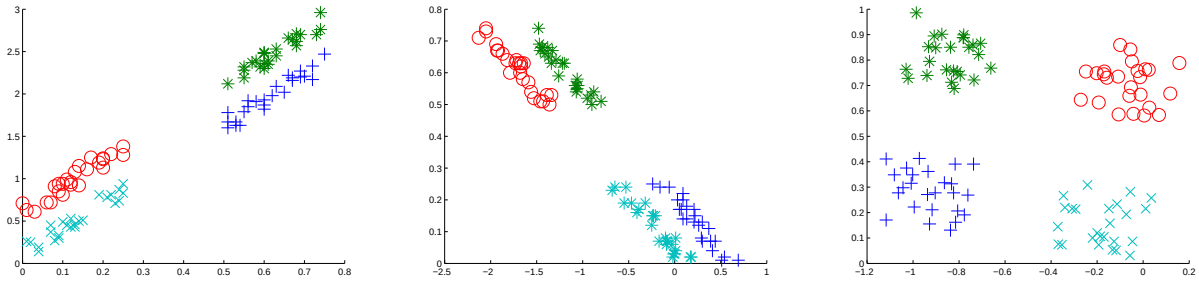

Figure 2. For the matrix $D_\pi = \{1\ \ 0; 3\ \ 1\}$, the data mapped into a space using D_π (left), mapped using D'_π (middle) and mapped using the orthogonal projection [5] (right). Note all images are scaled so they are comparable.

projection are shown in Figure 3. As we can see the orthogonal projection maps all points to the origin. We shall see in the related work section this is because no sub-space orthogonal to D_π exists. It is quite likely that in lower dimensional space that no or a very rudimentary orthogonal sub-space exist.

3 Properties and Advantages Of Our Approach

We now discuss several properties and advantages of our approach.

3.1 The Approach Is Applicable For A Variety of D_π

This is a straight-forward result of the SVD being defined for any square or rectangular matrix. This makes the work also applicable for dimension reduction approaches where D_π will have s rows and m columns and will reduce the space from m to s dimensions.

3.2 The Approach Performs a Linear Transformation

When transforming the instance space to find an alternative clustering our aim is two fold. Firstly, we wish to explicitly reverse the transformation representing the clustering we have already found. We can do this by encoding the properties as constraints and learn a distance function from them. In this way we explicitly allow specification of what properties are undesirable in the clustering. However, when reversing these properties we do not wish to unduly effect the inherent structure in the data, by say performing a non-linear transformation, so that we are effectively clustering some other data set that bears little relation to the data set under consideration.

This is unlikely to occur in our approach since we are effectively performing just a *linear transformation* on the data. All of our matrices returned by the SVD: H, S and A are linear transformations of the data since they represent rotation, stretching and realignment which are known to be linear. As is well known, a combination of linear transformation is also a linear transformation and being an example of an affine transformation preserves important properties such as the co-linearity of points.

3.3 The Approach Finds a Least Squares Solution

Consider our approach where X is the original set of points in a space from which a good clustering characterized by the transformation D_π exists. A valid question is then, what could be a new set of positions (X') so when D_π is applied to X', the result is a transformation back to X. That is X' represents the alternative point positions that would reverse the effect of D_π so as to transform the positions back to those in X.

This can be framed as a least squares minimization problem with X' being unknown.

$$||D_\pi X' - X||^2 \tag{2}$$

The well known Moore-Penrose pseudo-inverse gives us the solution where as before H, S, A are the hanger, stretch and aligner matrices obtained from the SVD of D_π.

$$X' = (A^T S^{-1} H^T)^T = H S^{-1} A \tag{3}$$

We note that equation 3 is precisely equation 1. A visual interpretation of this can be obtained by considering

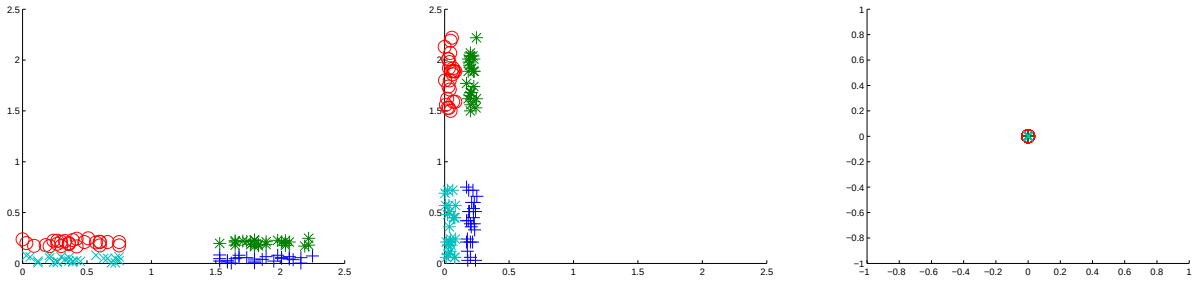

Figure 3. For the matrix $D_\pi\{3\ \ 0; 0\ \ \frac{1}{3}\}$, the data mapped into a space using D_π (left), mapped using D'_π (middle) and mapped using the orthogonal projection (right).

our previous example shown in Figure 2. Applying D_π to the data set in the center image will produce the original data set shown in Figure 1.

4 Applications to Non-Hierarchical Clustering

In this experimental section we compare our results against three algorithms (see Section 7) to illustrate that our general approach can find alternative clusterings that are of reasonable quality. We used the same data sets as used by [1].

The authors of COALA have already shown their approach outperformed the CIB approach of Gondek and Hoffman and we also compare our approach against the two approaches in [5] which deals with orthogonal sub-space projections. Cui et' al report success using the k-means algorithm and hence we also apply k-means after applying our ADFT approach. Our results are shown in Table 1. We begin by computing two measures of clustering quality, the Generalized Dunn index (DI) [1] and Vector Quantization Error (VQE) in columns 2 and 3 for the set partition (π_l) defined by the extrinsic labels. This is one experimental setup used in [1] and has the benefit of producing an algorithm independent initial clustering which the algorithms will attempt to find a good and alternative clustering to. The DI is a measure of the minimum distance between two clusters calculated as the average distance between each pair of points that are in different clusters, this is then normalized by the maximum cluster diameter [4]. We report the VQE as it is the objective function that k-means minimizes and allows us to compare our results with the work described in [5].

Then for the COALA algorithm (columns 4,5 and 6) we report the properties of the alternative clustering found after using the criterion $\omega = 0.6$ as used in their work [1]. The Jaccard index (JI) [1] is a measure of similarity between two clusters and hence unlike the DI with the JI the lower the better in our application.

We note that our approach typically finds more different (as measured by the JI) clustering that are typically more compact (as measured by VQE), but not as well separated (as measured by the DI) compared to COALA. This is to be expected as minimizing VQE is the objective of k-means and the DI is effectively the join criterion used by [1] in their average linkage algorithm. Comparing our work to Cui and collaborator's work our approach typically finds equally different clusterings but our clusterings tend to have better quality when measured by both the DI and VQE. This can be explained since these lower dimensional data sets do not necessarily contain a useful orthogonal subspace.

5 Applications to Hierarchical Clustering

In this section, we present the first results, to our knowledge, on finding alternative dendrograms. We begin by running a complete linkage algorithm on the data to obtain π. We then cut the dendrogram at the point k equals the number of extrinsic classes and learn a distance function D from the constraints generated from those clusters. We then use equation 1 to find D' and transform the data and then reapply the same algorithm to obtain π'. Our hope is that the two dendrograms will be different but need not be if an alternative structure does not exist. Table 2 shows our results. We compare dendrograms by cutting the dendrogram at the level k equals the number of extrinsic labels and then measuring the the JI, DI and VQE. Averaging these properties over all levels is undesirable as the lower levels of the dendrogram tend to dominate this calculation. We see that for these data sets there exists alternative dendrograms that are comparable in quality.

Note, these results are not comparable with Table 1 since the starting clusterings are different.

(1) **Data Set**	(2) DI (π_l)	(3) VQE (π_l)	(4) JI [1]	(5) DI [1]	(6) VQE [1]	(7) JI AFDT	(8) DI AFDT	(9) VQE AFDT	(10) JI [5] Alg 1	(11) DI [5] Alg 1	(12) VQE [5] Alg 1	13 JI [5] Alg 2	14 DI [5] Alg 2	15 VQE [5] Alg 2
Glass	0.21	911	0.26	0.83	855	0.24	0.58	505	0.27	0.50	721	0.21	0.41	801
Ion.	0.65	3086	0.54	1.21	3207	0.43	0.98	2421	0.40	0.9	2650	0.4	0.86	2630
ESL	0.38	1374	0.28	0.62	1885	0.24	0.73	1787	0.28	0.65	2454	NA	NA	NA
Veh.	0.56	2.4×10^7	0.26	1.05	5.5×10^6	0.18	0.57	5.4×10^6	0.20	0.59	1.0×10^7	0.17	0.46	2.6×10^7

Table 1. Results of Comparing Several Alternative Clustering Approaches for Non-Hierarchical Clustering. Note DI=Dunn Index, JI=Jaccard Index, VQE = Vector Quantization Error. Results are averaged over ten random restarts of each algorithm if appropriate. Note for the ESL data set $k > m$ and hence the transformation performed by algorithm 2 in [5] is undefined.

Dataset	DI (π_k)	VQE (π_k)	JI (π_k, π'_k)	DI (π'_k)	VQE (π'_k)
Glass	0.88	1244	0.68	0.63	1023
Ion.	0.79	9278	0.51	1.15	11437
ESL	0.77	594	0.35	0.81	650
Vehicle	0.64	7028	0.48	0.80	7764

Table 2. The result of applying our technique for agglomerative complete linkage algorithms.

6 Two Real World Examples

The comparison on UCI data sets yields interesting comparative insights and we now validate the work on several real world problems and attempt to understand the alternative explanations.

6.1 Hand Written Digits on a Stylus

In this section we present results on the classic Pen-Digit data set which consists of handwritten digits recorded on a pen-based tablet. Each instance corresponds to a single digit and has 16 attributes, which represent the 8 x, y positions of the pen as the digit is being written. Each pair of co-ordinates is sampled as the digit is being written. Users were free to write the digits in any form they wanted. By applying our algorithm, we hope to find alternative explanations of how people write digits with respect to how quickly they are written and the structure how the digit is created.

After running k-means algorithm for $k = 2$ on the original handwritten data set, clustering π_1 is obtained. Each group's centroids are shown in Figure 4. Although the most frequently occurring digits in each cluster do not appear to have much in common, the centroids of the 2 clusters explain two distinct ways that digits are written. The cluster on the top shows that people write digits in this cluster in a counter-clockwise way with a constant speed. Note that the way the x, y co-ordinates were recorded means that a longer distance between adjacent co-ordinates indicate a high writing speed. The clustering on the bottom explains that people can also write digits from left to right then go down, with initially a very high speed then considerably slower.

We follow our approach discussed earlier and create a new distance matrix D' using equation 1, transform the data and then reapply k-means for $k = 2$ to obtain π_2. The centroids of each cluster are shown in Figure 5. The cluster on the top shows that another two-way explanation on how people write digits can be that some are written clockwise (not counter-clockwise as before) with a smooth speed on most strokes except when writing initially and in the middle of the character. The clustering on bottom explains that people write the remaining digits from left to right down then go left down, but the writing speed is increasing gradually and dramatically. Note, this is similar in shape to the bottom cluster in Figure 4 but the **speed** of writing the digit is backwards. In the earlier clustering the digit is written very quickly initially then slowly, in this clustering the digit is written very slowly initially then quickly.

What is quite remarkable is that these are two genuinely alternative explanations of the data as Table 3 indicates. They are quite different as indicative of the low JI but have very similar quality measures.

6.2 Pandemic Simulation Data

An interesting problem posed at a recent SIAM DM workshop (`http://www.cs.dartmouth.edu/ cbk/sdm07/`) is the division of a large geographic area into distinct

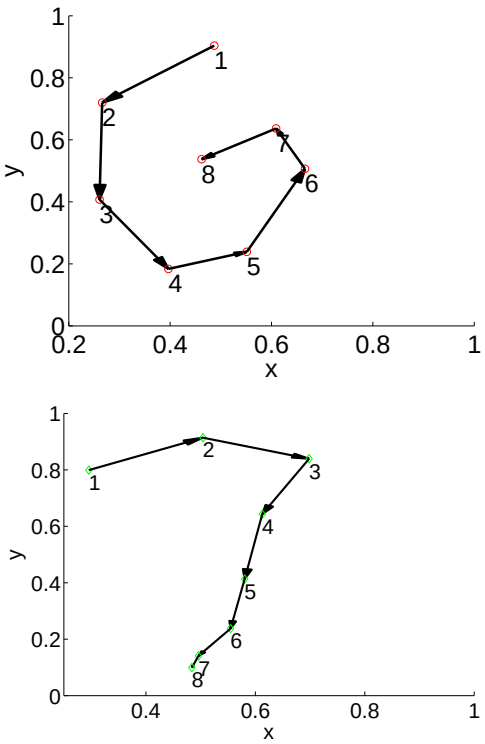

Figure 4. The centroids of the first clustering π_1 found.

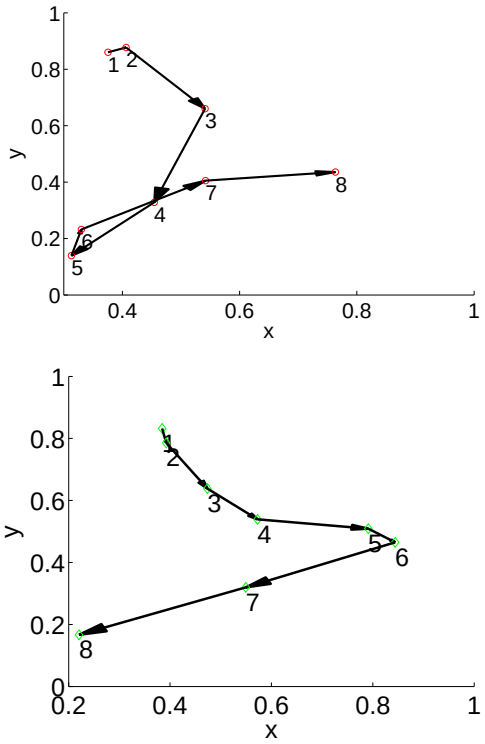

Figure 5. The centroids of the second clustering π_2 found.

DI(π_1)	DI(π_2)	VQE(π_1)	VQE(π_2)	JI(π_1, π_2)
0.83	0.83	1886	1919	0.36

Table 3. The Comparison Between Two Clusterings Found in the Pen Digit Data Set.

spatio-demographic regions on which various pandemic preparation sub-policies can be enacted. In this context, the data to be mined is the aggregated normalized household data containing information such as the x, y location of the house, income, age, household size, total number of trips and the distance, etc. Each tuple in this database is effectively the averaged snapshot of the population at the end of a pandemic simulated period over many different simulation runs. Ideally, decision regions should (1) include a collection of households that are homogeneous with respect to the infection rate (i.e., most households in the collection have a high rate of infection or a low rate of infection) and (2) be easy to isolate. At its core, the problem of finding decision regions is a clustering problem that involves breaking down the area of Portland (see Figure 6) into distinct regions.

Clustering the data with a Euclidean distance metric shows that homogeneous regions do exist. Figure 7 shows clustering of Portland households where each household is colored according to which of the twenty-five clusters it belongs to. Applying our approach we obtain an alternative clustering shown in the bottom of the same figure. When measuring the usefulness of the clustering with respect to the infection rates of each household, we find the two clustering are similar but the figures show that one clustering (the bottom) has better spatial properties in that the regions are contiguous. We note that since $k = 25$ and $m = 10$ (dimension) the second approach of Cui and collaborators [5] could not be used on this data set.

7 Related work

As discussed in the introduction there are four primary approaches to finding alternative clusterings: a) Dual-objective function algorithms, b) Generate and collect algorithms, c) MCMC samplers with memories and d) Data transformation approaches. We have mentioned that the generate and collect approach algorithms such as Meta-Clustering [12] and the MCMC sampler approaches are not

Figure 6. One of the Case Study Areas: Portland, OR. Forming decision regions will involve dividing this entire region into sub-regions.

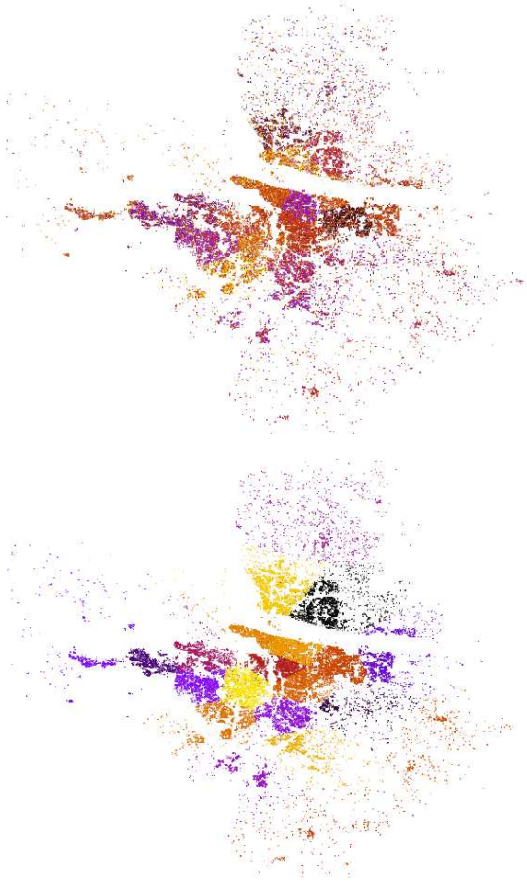

Figure 7. Two Alternative Clusterings Found of Portland Oregon with each of the 25 clusters color-coded. The two alternative clusterings are comparable when measured on the ability to differentiate infected/uninfected households but have different spatial properties.

efficient for large data sets. We now discuss related works for approaches a) and d).

The seminal paper on finding alternative clusterings was by Gondek and Hofmann [11]. In their work, they explored the idea of using the conditional information bottleneck (CIB) approach to find an alternative clustering to a given non-novel clustering. Their approach subtracts the background knowledge about the given clustering by maximizing conditional mutual information, but is limited since it requires the explicit joint distribution information between the cluster labels and the relative features which can be difficult to formalize. Though their experimental results clearly illustrated the approach can find alternative clustering to a given clustering it is limited to *information alternatively* (as defined by only the cluster labels) and for problems in the CIB framework.

In 2006, Bae and Bailey showed that Gondek's approach had several short-comings: firstly, it was limited to the conditional bottleneck approach that requires a joint distribution and that CIB often found clusterings that though different were not high quality when measured by the DI. Their COALA approach which centers around using flipped constraints to generate an alternative clustering of good quality (when measured by the Dunn index). COALA [1] is a heuristic approach using only cannot-linked constraints (two instances must not be in the same cluster) to generate a single different clustering.

These approaches have been well demonstrated to be useful but have a limited scope of application which differentiates our work. In particular, both CIB and COALA are limited to finding orthogonal partitions but our approach is applicable to most algorithms that makes use of a distance function such as: k-means, CLARANS, DBSCAN, agglomerative and divisive hierarchical algorithms etc. [13].

Cui and collaborator's [5] explore two approaches to finding orthogonal clusterings using an orthogonal sub-space projection. The orthogonal sub-space projection as defined by [5] is $D' = I - D(D^T D)^{-1} D^T$. The orthogonal projection essentially finds an orthogonal sub-space to D such that if D defines a sub-space then another sub-space D' is defined as being orthogonal if **every** vector in D is orthogonal to **every** vector in D'. This is a very strong requirement and as we saw in Figures 2 and 3 can lead to undesirable properties in lower-dimensional space.

In their first method the instances in a cluster are projected onto a sub-space orthogonal to the centroid of that

cluster, this means that D will be a column vector (the centroid's location). In their second approach the matrix D in their work is learnt by extracting the k principal components from the all the cluster centroid vectors of the clustering that has already been found. This essentially determines the features that are most different/variable between the centroids and is written as k column vectors which defines a $m \times k$ matrix. We can now better explain the unusual results in Figure 2 and 3. Consider in four dimensional space if the PCA of the centroids is:

$$D = \begin{bmatrix} 1 & 0 & 0.5 \\ 0 & 1 & 0 \\ 0 & 0 & 0.5 \\ 0 & 0 & 0 \end{bmatrix}$$

Since PCA is attempting to find the orthogonal projections with most variance then the matrix D need not be sparse. Then the only sub-space alternative to these 3 vectors is the one dimensional space whose transformation matrix is below.

$$D' = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Therefore, though being useful in high dimensional space the approach of orthogonal projection may not be appropriate for lower dimensional space as our experimental results in Table 1 indicate since orthogonal sub-spaces may not be very meaningful. Furthermore, if $k > m$ where k is the number of clusters and m is the number of dimensions, then their second approach cannot be applied. This is so since each column vector attempts to represent a ortho-normal basis (having being obtained from PCA). If there are k such ortho-normal basic vectors then there exists no orthogonal projection onto a space of just m dimensions if $k > m$. This is unlikely to occur in high dimensional data sets like images when m is in the thousands but for spatial and other data sets it is often the case that $k > m$.

8 Conclusion

Finding alternative clusterings is an important problem in data mining which has great practical significance. Several successful approaches that are algorithm dependent [11, 1] and an approach [5] that finds orthogonal sub-spaces have recently been proposed.

We propose a general purpose approach that can be used with any number of clustering algorithms. The approach takes an existing clustering and learns a distance function from constraints extracted from the clustering. We then apply SVD to this learnt distance function and use equation 1 to obtain an alternative clustering that has the property that points that were in the same cluster in the original clustering are now far apart. The application of equation 1 also has a least squares interpretation that the transformed points are placed to "counter-act" the effect of the transformation matrix learnt from the constraints so as to restore the original positions of the points.

Our approach has several pragmatic benefits over these other techniques. Firstly, unlike the work of [11, 1] it is not limited to a particular clustering algorithm and can be applied to any algorithm that uses a distance function. It also has the advantage over the orthogonal sub-space work of [5] since it works well in lower dimensional space (as shown in Table 1) and unlike Algorithm 2 in their work is applicable if $k > m$ which is often the case in spatial data sets.

We demonstrated our approach on non-hierarchical clustering problems and showed it could find as different or more different clusterings than the COALA algorithm. When measuring the quality of the alternative clusterings using the VQE (the objective function of the $k - means$ algorithm we used) our approach was better on all four data sets, but when measuring quality on the DI (the objective function of their algorithm) their approach was better on three of the data sets. Comparing our approach with the orthogonal sub-space projection approaches [5] we found our approach found more different and better quality clusterings on these low dimensional data sets. We then showed that our approach could be used for agglomerative clustering to find alternative dendrograms which is to our knowledge a novel problem.

We applied our approach to two real world data sets: pen digit drawing on a style and pandemic simulation data and obtained interpretable results that show our approach finds alternative explanations of the data. Future work will involve more precise ways of specifying constraints so as to tag some parts of the clustering as being desirable and should be kept and others that an orthogonal clustering should be found for.

References

[1] E. Bae and J. Bailey, COALA : A Novel Approach for the Extraction of an Alternate Clustering of High Quality and High Dissimilarity, 6^{th} *IEEE International Conference on Data Mining*, 2006

[2] S. Basu, A. Banerjee, R. Mooney, Semi-supervised Clustering by Seeding, 19^{th} *International Conference on Machine Learning*, 2002.

[3] S. Basu, M. Bilenko and R. J. Mooney, Active Semi-Supervision for Pairwise Constrained Cluster-

ing, 4^{th} *SIAM International Conference on Data Mining*, 2004.

[4] J. Bezdek, L. Wanquing, Y. Attikiouzel, and M. Windham. A geometric approach to cluster validity for normal mixtures, Soft Computing, pp 166-179, 1997.

[5] Y. Cui, X. Fern and J. Dy, Non-Redundant Multi-View Clustering Via Orthogonalization, IEEE ICDM 2007.

[6] G. Chechikand and N. Tishby, "Extracting relevant structures with side information", *In Advances in Neural Information Processing Systems* 15 *(NIPS* 02*)*,2002

[7] I. Davidson, "Minimum Message Length Clustering Using Gibbs Sampling", 16^{th} Uncertainty and A.I. Conference (UAI), 2000.

[8] I. Davidson and S. S. Ravi, "The Complexity of Non-Hierarchical Clustering with Instance and Cluster Level Constraints", *Data Mining and Knowledge Discovery*, Vol. 14, No. 1, Feb. 2007, pp. 25–61.

[9] D. Hand, H. Manilla and P. Smyth, *Principles of Data Mining*, MIT Press, ISBN:0-262-08290-X

[10] S. D. Kamvar,D. Klein,C. D. Manning, "Interpreting and Extending Classical Agglomerative Clustering Algorithms using a Model-Based approach",*Proceedings of the Nineteenth International Conference on Machine Learning*, 2002

[11] D. Gondek, T. Hofmann, "Non-redundant data clustering", Fourth IEEE International Conference on Data Mining, 2004.

[12] R. Caruana, M. Elhawary, N. Nguyen and C. Smith, Meta Clustering, Sixth IEEE Conference on Data Mining, 2006.

[13] P. Tan, M. Steinbach, V. Kumar, *Introduction to Data Mining*, Addison-Wesley, 2005. ISBN : 0321321367.

[14] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method",*In Proceedings of the* 37^{th} *Annual Allerton Conference on Communication, Control and Computing*, 1999.

[15] K. Wagstaff and C. Cardie, "Clustering with Instance-Level Constraints", *International Conference on Machine Learning*, 2000.

[16] E. Xing, A. Ng, M. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. *Advances in Neural Information Processing Systems* (NIPS) 15, 2003.

[17] L. Yang and R. Jin, Distance Metric Learning: A Comprehensive Survey, Technical Report, The Department of Computer Science and Engineering Michigan State University, 2006