

Correlation Clustering for Learning Mixtures of Canonical Correlation Models

X. Z. Fern* C. E. Brodley† M. A. Friedl‡

Abstract

This paper addresses the task of analyzing the correlation between two related domains X and Y . Our research is motivated by an Earth Science task that studies the relationship between vegetation and precipitation. A standard statistical technique for such problems is Canonical Correlation Analysis (CCA). A critical limitation of CCA is that it can only detect linear correlation between the two domains that is globally valid throughout both data sets. Our approach addresses this limitation by constructing a mixture of local linear CCA models through a process we name *correlation clustering*. In correlation clustering, both data sets are clustered simultaneously according to the data's correlation structure such that, within a cluster, domain X and domain Y are linearly correlated in the same way. Each cluster is then analyzed using the traditional CCA to construct local linear correlation models. We present results on both artificial data sets and Earth Science data sets to demonstrate that the proposed approach can detect useful correlation patterns, which traditional CCA fails to discover.

1 Introduction

In Earth science applications, researchers are often interested in studying the correlation structure between two domains in order to understand the nature of the relationship between them. The inputs to our correlation analysis task can be considered as two data sets X and Y whose instances are described by feature vectors \vec{x} and \vec{y} respectively. The dimension of \vec{x} and that of \vec{y} do not need to be the same, although there must be a one-to-one mapping between instances of X and instances of Y . Thus, it is often more convenient to consider these two data sets as one compound data set whose instances are described by two feature vectors \vec{x} and \vec{y} . Indeed, throughout the remainder of this paper, we will refer to the input of our task as one data set, and the goal is to study how the two sets of features are correlated to each other.

Canonical Correlation Analysis (CCA) [4, 6] is a

multivariate statistical technique commonly used to identify and quantify the correlation between two sets of random variables. Given a compound data set described by feature vectors \vec{x} and \vec{y} , CCA seeks to find a linear transformation of \vec{x} and a linear transformation of \vec{y} such that the resulting two new variables are maximally correlated.

In Earth science research, CCA has been often applied to examine whether there is a cause-and-effect relationship between two domains or to predict the behavior of one domain based on another. For example, in [13] CCA was used to analyze the relationship between the monthly mean sea-level pressure (SLP) and sea-surface temperature (SST) over the North Atlantic in the months of December, January and February. This analysis confirmed the hypothesis that atmospheric SLP anomalies cause SST anomalies.

Because CCA is based on *linear* transformations, the scope of its applications is necessarily limited. One way to tackle this limitation is to use nonlinear canonical correlation analysis (NLCCA) [5, 8]. NLCCA applies nonlinear functions to the original variables in order to extract correlated components from the two sets of variables. Although promising results have been achieved by NLCCA in some Earth science applications, it tends to be difficult to apply such techniques because of the complexity of the model and the lack of robustness due to overfitting [5].

In this paper we propose to use a mixture of local linear correlation models to capture the correlation structure between two sets of random variables (features). Mixtures of local linear models not only provide an alternative solution to capturing nonlinear correlations, but also have the potential to detect correlation patterns that are significant only in a part (a local region) of the data. The philosophy of using multiple local linear models to model global nonlinearity has been successfully applied to other statistical approaches with similar linearity limitations such as principal component analysis [12] and linear regression [7]. Our approach uses a two-step procedure. Given a compound data set, we propose to first solve a clustering problem that partitions the data set into clusters such that each cluster contains instances whose \vec{x} features and \vec{y} features are linearly correlated. We then independently apply CCA

*School of Elec. and Comp. Eng., Purdue University, West Lafayette, IN 47907, USA

†Dept. of Comp. Sci., Tufts University, Medford, MA 02155, USA

‡Dept. of Geography, Boston University, Boston, MA, USA

to each cluster to form a mixture of correlation models that are locally linear.

In designing this two-step process, we need address the following two critical questions.

1. Assume we are informed *a priori* that we can model the correlation structure using k local linear CCA models. *How should we cluster the data in the context of correlation analysis?*
2. In real-world applications, we are rarely equipped with knowledge of k . *How can we decide how many clusters there are in the data or whether a global linear structure will suffice?*

Note that the goal of clustering in the context of correlation analysis is different from traditional clustering. In traditional clustering, the goal is to group instances that are similar (as measured by certain distance or similarity metric) together. In contrast, here we need to group instances based on how their \vec{x} features and \vec{y} features correlate to each other, i.e., instances that share similar correlation structure between the two sets of features should be clustered together. To differentiate this clustering task from traditional clustering, we name it *correlation clustering*¹ and, in Section 3 we propose an iterative greedy k -means style algorithm for this task.

To address the second question, we apply the technique of cluster ensembles [2] to our correlation clustering algorithm, which provides a user with a visualization of the results that can be used to determine the proper number of clusters in the data. Note that our correlation clustering algorithm is a k -means style algorithm and as such may have many locally optimal solutions—different initializations may lead to significantly different clustering results. By using cluster ensembles, we can also address the local optima problem of our clustering algorithm and find a stable clustering solution.

To demonstrate the efficacy of our approach, we apply it to both artificial data sets and real world Earth science data sets. Our results on the artificial data sets show that (1) the proposed correlation clustering algorithm is capable of finding a good partition of the data when the correct k is used and (2) cluster ensembles provide an effective tool for finding k . When applied to the Earth science data sets, our technique detected significantly different correlation patterns in comparison to what was found via traditional CCA. These results led our domain expert to highly interesting hypotheses that merit further investigation.

¹Note that the term *correlation clustering* has also been used by [1] as the name of a technique for traditional clustering.

The remainder of the paper is arranged as follows. In Section 2, we review the basics of CCA. Section 3 introduces the intuitions behind our correlation clustering algorithm and formally describes the algorithm, which is then applied to artificially constructed data sets to demonstrate its efficacy in finding correlation clusters from the data. Section 4 demonstrates how cluster ensemble techniques can be used to determine the number of clusters in the data and address the local optima problem of the k -means style correlation clustering algorithm. Section 5 explains our motivating application, presents results, and describes how our domain expert interprets the results. Finally, in Section 6 we conclude the paper and discuss future directions.

2 Basics of CCA

Given a data set whose instances are described by two feature vectors \vec{x} and \vec{y} , the goal of CCA is to find linear transformations of \vec{x} and linear transformations of \vec{y} such that the resulting new variables are maximally correlated.

In particular, CCA constructs a sequence of pairs of strongly correlated variables $(u_1, v_1), (u_2, v_2), \dots, (u_d, v_d)$ through linear transformations, where d is the minimum dimension of \vec{x} and \vec{y} . These new variables u_i 's and v_i 's, named *canonical variates* (sometimes referred to as canonical factors). They are similar to principal components in the sense that principal components are linear combinations of the original variables that capture the most variance in the data and in contrast canonical variates are linear combinations of the original variables that capture the most correlation between two sets of variables.

To construct these canonical covariates, CCA first seeks to transform \vec{x} and \vec{y} into a pair of new variables u_1 and v_1 by the linear transformations:

$$u_1 = (\vec{a}_1)^T \vec{x}, \quad \text{and} \quad v_1 = (\vec{b}_1)^T \vec{y}$$

where the transformation vectors \vec{a}_1 and \vec{b}_1 are defined such that $\text{corr}(u_1, v_1)$ is maximized subject to the constraint that both u_1 and v_1 have unit variance.² Once $\vec{a}_1, \vec{b}_1; \dots; \vec{a}_i, \vec{b}_i$ are determined, we then find the next pair of transformations \vec{a}_{i+1} and \vec{b}_{i+1} such that the correlation between $(\vec{a}_{i+1})^T \vec{x}$ and $(\vec{b}_{i+1})^T \vec{y}$ is maximized with the constraint that the resulting u_{i+1} and v_{i+1} are uncorrelated with all previous canonical variates.³ Note that the correlation between u_i and v_i becomes weaker as i increases. Let r_i represent the correlation between the i th pair of canonical variates, we have $r_i \geq r_{i+1}$.

²This constraint ensures unique solutions.

³This constraint ensures that the extracted canonical variates contain no redundant information.

It can be shown that to find the projection vectors for canonical variates, we only need to find the eigenvectors of the following matrices:

$$M_x = (\Sigma_{xx})^{-1}\Sigma_{xy}(\Sigma_{yy})^{-1}\Sigma_{yx}$$

and

$$M_y = (\Sigma_{yy})^{-1}\Sigma_{yx}(\Sigma_{xx})^{-1}\Sigma_{xy}$$

The eigenvectors of M_x , ordered according to decreasing eigenvalues, are the transformation vectors $\vec{a}_1, \vec{a}_2, \dots, \vec{a}_d$ and the eigenvectors of M_y are $\vec{b}_1, \vec{b}_2, \dots, \vec{b}_d$. In addition, the eigenvalues of these two matrices are identical and the square-root of the i -th eigenvalue $\sqrt{\lambda_i} = r_i$, i.e., the correlation between the i -th pair of canonical variates u_i and v_i . Note that in most applications, only the first few most significant pairs of canonical variates are of real interest. Assume that we are interested in the first d pairs of variates, we can represent all the useful information of the linear correlation structure as a model M , defined as

$$M = \{(u_j, v_j), r_j, (\vec{a}_j, \vec{b}_j) : j = 1 \dots d\}$$

where (u_j, v_j) represent the j th pair of canonical variates, r_j is the correlation between them and (\vec{a}_j, \vec{b}_j) represent the projection vectors for generating them. We refer to M as a CCA model.

Once a CCA model is constructed, the next step is for the domain experts to examine the variates as well as the transformation vectors in order to understand the relationship between the two domains. This can be done in different ways depending on the application. In our motivating Earth science task, the results of CCA can be visualized as colored maps and interpreted by Earth scientist. We explain this process in Section 5.

3 Correlation Clustering

In this section, we first explain the basic intuitions that led to our algorithm and formally present our k -means style correlation clustering algorithm. We then apply the proposed algorithm to artificially constructed data sets and analyze the results.

3.1 Algorithm Description Given a data set described by two sets of features \vec{x} and \vec{y} , and the prior knowledge that the correlation structure of the data can be modeled by k local linear models, the goal of correlation clustering is to partition the data into k clusters such that *for instances in the same cluster the features of \vec{x} and \vec{y} are linearly correlated in the same way*. The critical question is how should we cluster the data to reach this goal. Our answer is based on the following important intuitions.

Table 1: A correlation clustering algorithm

Input:	a data set of n instances, each described by two random vectors \vec{x} and \vec{y}
	k , the desired number of clusters
Output:	k clusters and k linear CCA models, one for each cluster
Algorithm:	<ol style="list-style-type: none"> 1. Randomly assign instances to the k clusters. 2. For $i = 1 \dots k$, apply CCA to cluster i to build $M^i = \{(u_j, v_j), r_j, (a_j, b_j) : j = 1 \dots d\}$, i.e., the top d pairs of canonical variates, the correlation r between each pair, and the corresponding d pairs of projection vectors. 3. Reassign each instance to a cluster based on its \vec{x} and \vec{y} features and the k CCA models. 4. If no assignment has changed from previous iteration, return the current clusters and CCA models. Otherwise, go to step 2.

Intuition 1: If a given set of instances contains multiple correlation structures, applying CCA to this instance set will not detect a strong linear correlation.

This is because when we put instances that have different correlation structure together, the original correlation patterns will be weakened because they are now only valid in part of the data. Conversely, if CCA detects strong correlation in a cluster, it is likely that the instances in the cluster share the same correlation structure. This suggests that we can use the strength of the correlation between the canonical variates extracted by CCA to measure the quality of a cluster. Note that it is computationally intractable to evaluate all possible clustering solutions in order to select the optimal one. This motivates us to examine a k -means style algorithm. Starting from a random clustering solution, in each iteration, we build a CCA model for each cluster and then reassign each instance to its most appropriate cluster according to its \vec{x} and \vec{y} features and the CCA models. In Table 1, we describe the basic steps of such a generic correlation clustering procedure.

The remaining question is how to assign instances to their clusters. Note that in traditional k -means clustering, each iteration reassigns instances to clusters according to the distance between instances and cluster centers. For correlation clustering, minimizing the

Table 2: Procedure of assigning instances to clusters

1. For each cluster i and its CCA model M^i , described as $\{(u_j^i, v_j^i), r_j^i, (\vec{a}_j^i, \vec{b}_j^i) : j = 1 \cdots d\}$, construct d linear regression models $v_j^i = \beta_j^i * u_j^i + \alpha_j^i, j = 1 \cdots d$, one for each pair of canonical variates.
2. Given an instance (\vec{x}, \vec{y}) , for each cluster i , compute the instance's canonical variates under M^i as $u_j = (\vec{a}_j^i)^T \vec{x}$ and $v_j = (\vec{b}_j^i)^T \vec{y}, j = 1 \cdots d$, and calculate \hat{v}_j as $\hat{v}_j = \beta_j^i * u_j + \alpha_j^i, j = 1 \cdots d$, and the weighted err^i $err^i = \sum_{j=1}^d \frac{r_j^i}{r_1^i} * (v_j - \hat{v}_j)^2$, where $\frac{r_j^i}{r_1^i}$ is the weight for the j th prediction error.
3. Assign instance (\vec{x}, \vec{y}) to the cluster minimizing err^i .

distance between instances and their cluster centers is no longer our goal. Instead, our instance reassignment is performed based on the intuition described below.

Intuition 2: If CCA detects strong a correlation pattern in a cluster, i.e., the canonical variates u and v are highly correlated, we expect to be able to predict the value of v from u (or vice versa) using a linear regression model.

This is demonstrated in Figure 1, where we plot a pair of canonical variates with correlation 0.9. Shown as a solid line is the linear regression model constructed to predict one variate from the other. Intuition 2 suggests that, for each cluster, we can compute its most significant pair of canonical variates (u_1, v_1) and construct a linear regression model to predict v_1 from u_1 . To assign an instance to its proper cluster, we can simply select the cluster whose regression model best predicts the instance's variate v_1 from it's variate u_1 . In some cases, we are interested in the first few pairs of canonical variates rather than only the first pair. It is thus intuitive to construct one linear regression model for each pair, and assign instances to clusters based on the combined prediction error. Note that because the correlation r_i between variate v_i, u_i decreases as i increase, we set the weight for the i^{th} error to be $\frac{r_i}{r_1}$. In this manner, the weight for the prediction error between u_1 and v_1 is always one, whereas the weights for the ensuing ones will be smaller depending on the strength

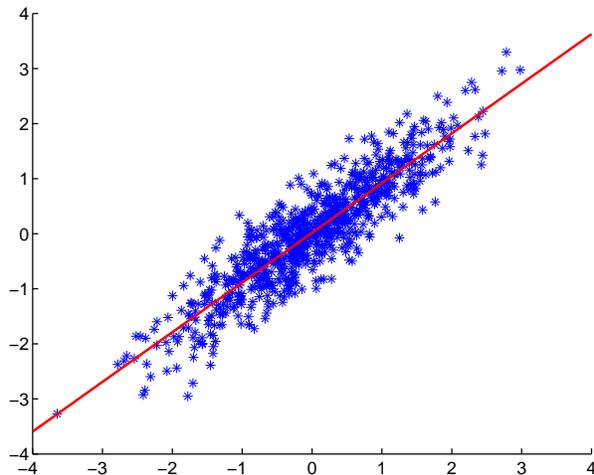


Figure 1: Scatter plot of a pair of canonical variates ($r = 0.9$) and the linear regression model constructed to predict one variate from another.

of the correlations. This ensures that more focus is put on the canonical variates that are more strongly correlated. In Table 2, we describe the exact procedure for reassigning instances to clusters.

Tables 1 and 2 complete the description of our correlation clustering algorithm. To apply this algorithm, the user needs to specify d , the number of pairs of canonical variates that are used in computing the prediction errors and reassigning the instances. Based on our empirical observations with both artificial and real-world datasets, we recommend that d be set to be the same as or slightly larger than the total number of variates that bear interest in the application. In our application, our domain expert is interested in only the top two or three pairs of canonical variates, consequently we used $d = 4$ as the default choice for our experiments.

The proposed correlation clustering algorithm is a greedy iterative algorithm. We want to point out that it is not guaranteed to converge. Specifically, after reassigning instances to clusters at each iteration, there is no guarantee that the resulting new clusters will have more strongly correlated variates. In our experiments, we did observe fluctuations in the objective function, i.e., the weighted prediction error. But, fluctuations typically occur only after an initial period in which the error computed by the objective function quickly decreases. Moreover, after this rapid initial convergence, the ensuing fluctuations are relatively small. Thus we recommend that one specify a maximum number of iterations, and in our experiments we set this to be 200 iterations.

Table 3: An artificial data set and results

	Data Sets		Global CCA	Mixture of CCA	
	D_1	D_2		clust. 1	clust. 2
r_1	0.85	0.9	0.521	0.856(.001)	0.904(.001)
r_2	0.6	0.7	0.462	0.619(.001)	0.685(.004)
r_3	0.3	0.4	0.302	0.346(.003)	0.436(.003)

3.2 Experiments on Artificial Data Sets To examine the efficacy of the proposed correlation clustering algorithm, we apply it to artificially generated data sets that have pre-specified nonlinear correlation structures. We generate such data by first separately generating multiple component data sets, each with a different linear correlation structure, and then mixing these component data sets together to form a composite data set. Obviously the resulting data set’s correlation structure is no longer globally linear. However, a properly constructed mixture of local linear models should be able to separate the data set into the original component data sets and recover the correlation patterns in each part. Therefore, we are interested in (1) testing whether our correlation clustering algorithm can find the correct partition of the data, and (2) testing whether it can recover the original correlation patterns represented as the canonical variates, and (3) comparing its results to the results of global CCA on the composite data set.

In Table 3, we present the results of our correlation clustering algorithm and traditional CCA on a composite data set formed by two component data sets, each of which contains 1000 instances. We generate each component data set as follows.⁴ Given the desired correlation values r_1 , r_2 , and r_3 , we first create a multivariate Gaussian distribution with six random variables $u_1, u_2, u_3, v_1, v_2, v_3$, where u_i and v_i are intended to be the i th pair of canonical variates. We set the covariance matrix to be:

$$\begin{pmatrix} 1 & 0 & 0 & r_1 & 0 & 0 \\ 0 & 1 & 0 & 0 & r_2 & 0 \\ 0 & 0 & 1 & 0 & 0 & r_3 \\ r_1 & 0 & 0 & 1 & 0 & 0 \\ 0 & r_2 & 0 & 0 & 1 & 0 \\ 0 & 0 & r_3 & 0 & 0 & 1 \end{pmatrix}$$

This ensures that $\text{corr}(u_j, v_j) = r_j$, for $j = 1, 2, 3$ and $\text{corr}(u_i, u_j) = \text{corr}(v_i, v_j) = \text{corr}(u_i, v_j) = 0$ for $i \neq j$. We then randomly sample 1000 points from this joint Gaussian distribution and form the final vector of \vec{x} using linear combinations of u_j ’s and the vector of \vec{y} using linear combinations of v_j ’s.

⁴The matlab code for generating a component data set is available at <http://www.ecn.purdue.edu/~xz>

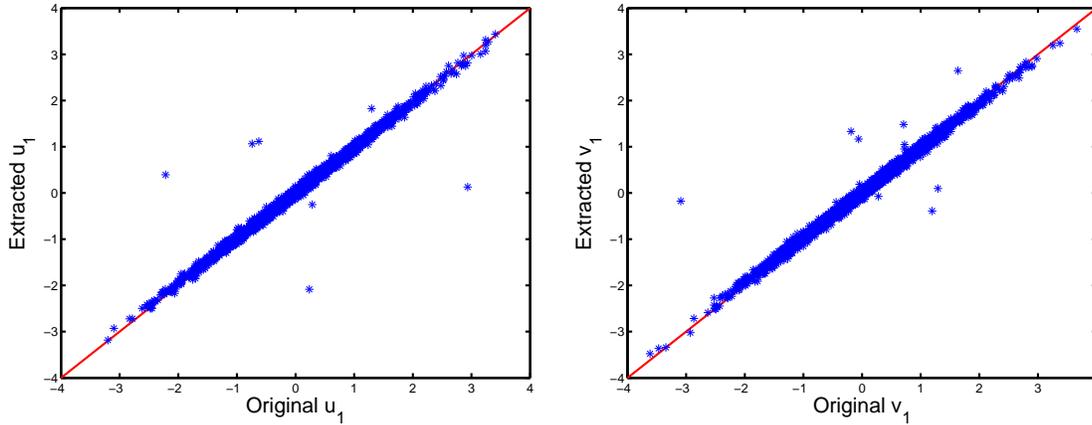
Columns 2 and 3 of Table 3 specify the correlation between the first three pairs of canonical variates of each of the constructed datasets, D_1 and D_2 . These are the values that were used to generate the data. We applied the traditional CCA to the composite data set (D_1 and D_2 combined together) and we report the top three detected canonical correlations in Column 4. We see from the results that, as expected, global CCA is unable to extract the true correlation structure from the data.

The last two columns of Table 3 show the results of applying the proposed correlation clustering algorithm to the composite data set with $k = 2$ and $d = 4$. The results, shown in Columns 5 and 6 are the average over ten runs with different random initializations (the standard deviations are shown in parentheses). We observe that the detected canonical correlations are similar to the true values. In Figure 2, We plot the canonical variates extracted by our algorithm (y axis) versus the true canonical variates (x axis) and the plots of the first two pairs of variates are shown. We observe that the first pair of variates extracted by our algorithm are very similar to the original variates. This can be seen by noticing that for both u_1 and v_1 most points lie on or are close to the line of unit slope (shown as a red line). For the second pair, we see more deviation from the red line. This is possibly because our algorithm put less focus on the second pair of variates during clustering. Finally, we observe that the clusters formed by our algorithm correspond nicely to the original component data sets. On average, only 2.5% of the 2000 instances were assigned to the wrong cluster.

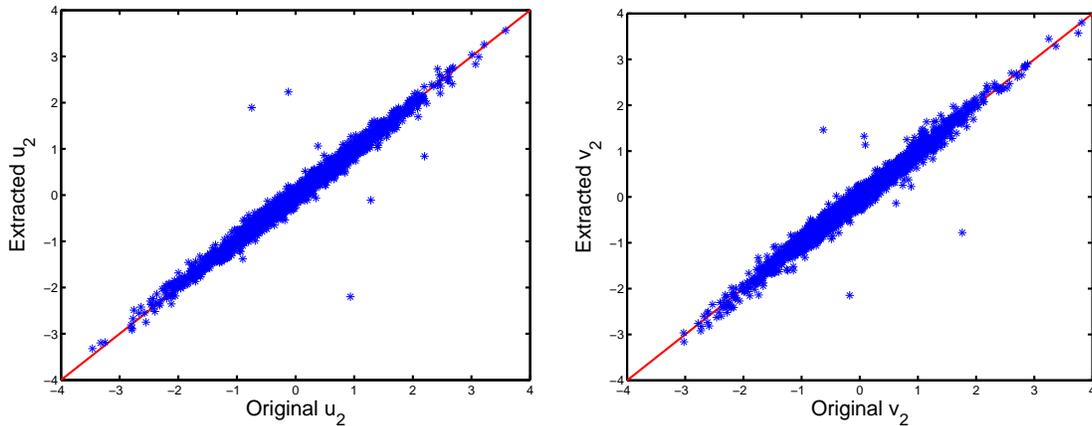
These results show that our correlation clustering algorithm can discover local linear correlation patterns given prior knowledge of k , the true number of clusters in the data. Our algorithm performs consistently well on artificially constructed data sets. This is in part due to the fact that these data sets are highly simplified examples of nonlinearly correlated data. In real applications, the nonlinear correlation structure is often more complex. Indeed, when applied to our Earth science data sets, we observe greater instability of our algorithm—different initializations lead to different clustering solutions. We conjecture that this is because our clustering algorithm is a k -means style greedy algorithm and has large number of locally optimal solutions.

4 Cluster Ensembles for Correlation Clustering

In this section we address a problem in the practical application of the proposed correlation clustering algorithm—identification of the number of clusters in the data. A complicating factor is that because we are



(a). The first pair of canonical variates



(b). The second pair of canonical variates

Figure 2: Comparing the first two pairs of canonical variates extracted by our mixture of CCA algorithm and the original canonical variates.

dealing with a k -means style greedy algorithm there may be many locally optimal solutions. In particular, different initializations may lead to different clusters. In this section we show how to apply cluster ensemble techniques to address these issues.

The concept of cluster ensembles has recently seen increasing popularity in the clustering community [11, 2, 10, 3], in part because it can be applied to any type of clustering as a generic tool for boosting clustering performance. The basic idea is to generate an ensemble of different clustering solutions, each capturing some structure of the data. The anticipated result is that by combining the ensemble of clustering solutions, a better final clustering solutions can be obtained. Cluster ensembles have been successfully applied to determine the number of clusters [10] and to improve clustering performance for traditional clustering tasks [11, 2, 3].

Although our clustering tasks are significantly different from traditional clustering in terms of the goal, we believe similar benefits can be achieved by using cluster ensembles.

To generate a cluster ensemble, we run our correlation clustering algorithm on a given data set with $k=2$ for r times, each run starting from a different initial assignment, where r is the size of the ensemble. We then combine these different clustering solutions into a $n \times n$ matrix S , which describes for each pair of instances the frequency with which they are clustered together (n is the total number of instances in the data set.) As defined, each element of S is a number between 0 and 1. We refer to it as a similarity matrix because $S(i, j)$ can be considered as the similarity (correlation similarity instead of the conventional similarity) between instances i and j .

After the similarity matrix is constructed, we can then visualize the matrix using a technique introduced by [10] to help determine how many clusters there are in the data. This visualization technique has two steps. First, it orders the instances such that instances that are similar to each other are arranged to be next to each other. It then maps the 0-1 range of the similarity values to a gray-scale such that 0 corresponds to white and 1 corresponds to black. The similarity matrix is then displayed as an image, in which darker areas indicate strong similarity and lighter areas indicate little to no similarity. For example, if all clustering solutions in the ensemble agree with one another perfectly, the similarity matrix S will have similarity value 1 for these pairs of instances that are from the same cluster and similarity value 0 for those from different clusters. Because the instances are ordered such that similar instances are arranged next to each other, the visualization will produce black squares along the diagonal of the image. For a detailed description of the visualization technique, please refer to [10].

To demonstrate the effect of cluster ensembles on our correlation clustering, we generate three artificial data sets using the same procedure as described in Section 3.2. These three data sets contain one, two, and three correlation clusters respectively. We apply our correlation clustering algorithm 20 times with different initializations and construct a similarity matrix for each data set. In Figure 3 we show the images of the resulting similarity matrices for these three data sets and make following observations.

- For the one-cluster data set, shown in Figure 3 (a), the produced similarity matrix does not show any clear clustering pattern. This is because our correlation clustering algorithm splits the data randomly in each run—by combining the random runs through the similarity matrix, we can easily reach the conclusion that the given data set contains only one correlation cluster.
- For the two-cluster data set, shown in Figure 3 (b), First, we see two dark squares along the diagonal, indicating there are two correlation clusters in the data. This shows that, as we expect, the similarity matrix constructed via cluster ensembles reveal information about the true number of clusters in the data.

In addition to the two dark diagonal squares, we also see small gray areas in the image, indicating that some of the clustering solutions in the ensemble disagree with each other on some instances. This is because different initializations sometimes lead to different local optimal solutions. Further,

we argue that these different solutions sometimes make different mistakes—combining them can potentially correct some of the mistakes and produce a better solution.⁵ Indeed, our experiments show that, for this particular two-cluster data set, applying the average-link agglomerative clustering to the resulting similarity matrix reduces the clustering error rate from 2.0% (the average error rate of the 20 clustering runs) to 1.1%. In this case, cluster ensembles corrected for the local optima problem of our correlation clustering algorithm. Cluster ensembles have been shown to boost the clustering performance for traditional clustering tasks, here we confirm that correlation clustering can also benefit from cluster ensembles.

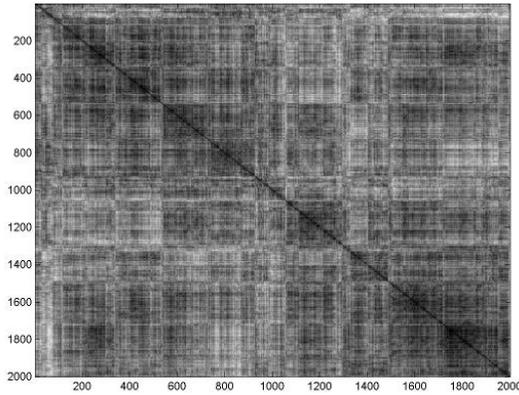
- For the last data set, shown in Figure 3 (c), we see three dark squares along the diagonal, indicating that there are three correlation clusters in the data.

Comparing to the two-cluster case, we see significantly larger areas of gray. In this case, our correlation clustering algorithm was asked to partition the data into two parts although the data actually contains three clusters. Therefore, it is not surprising that many of the clustering solutions don't agree with each other because they may split or merge clusters in many different ways when different initializations are used, resulting in much larger chance for disagreement. However, this does not stop us from finding the correct number of clusters from the similarity matrix. Indeed, by combining multiple solutions, these random splits and merges tend to cancel out each other and the true structure of the data emerges.

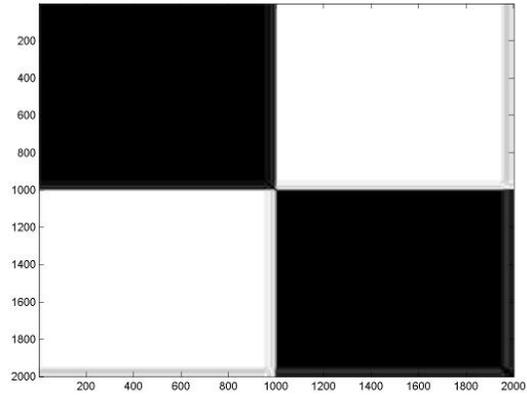
With the help of the similarity matrix, now we know there are three clusters in the last data set. We then constructed another cluster ensemble for this data set, but this time we set $k=3$ for each clustering run. The resulting similarity matrix S' is shown in Figure 3 (d). In this case, the average error rate achieved by the individual clustering solutions in the ensemble is 7.5% and the average-link agglomerative clustering algorithm applied on S' reduces the error rate to 6.8%.

To conclude, cluster ensembles help to achieve two goals. First, they provide information about the true structure of the data. Second, they help improve clustering performance of our correlation clustering algorithm.

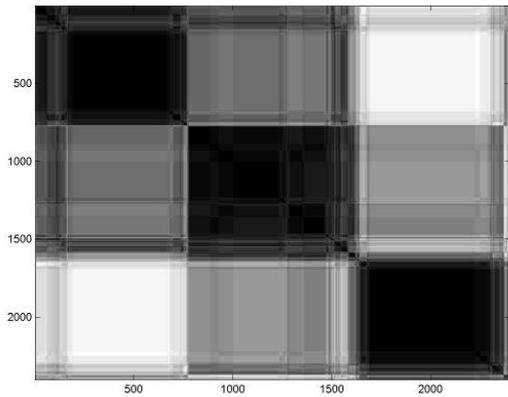
⁵It should be noted that if the different solutions make the same mistakes, these mistakes will not be corrected by using cluster ensembles.



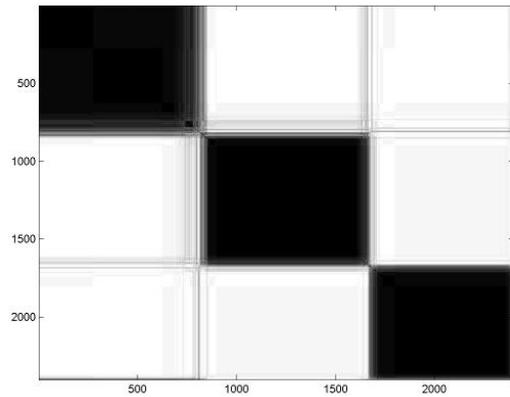
(a)



(b)



(c)



(d)

Figure 3: Visualization of similarity matrices: (a). S for the one-cluster data set; (b). S for the two-cluster data set ; (c). S for the three-cluster data set, and (d). S' for the three-cluster data set

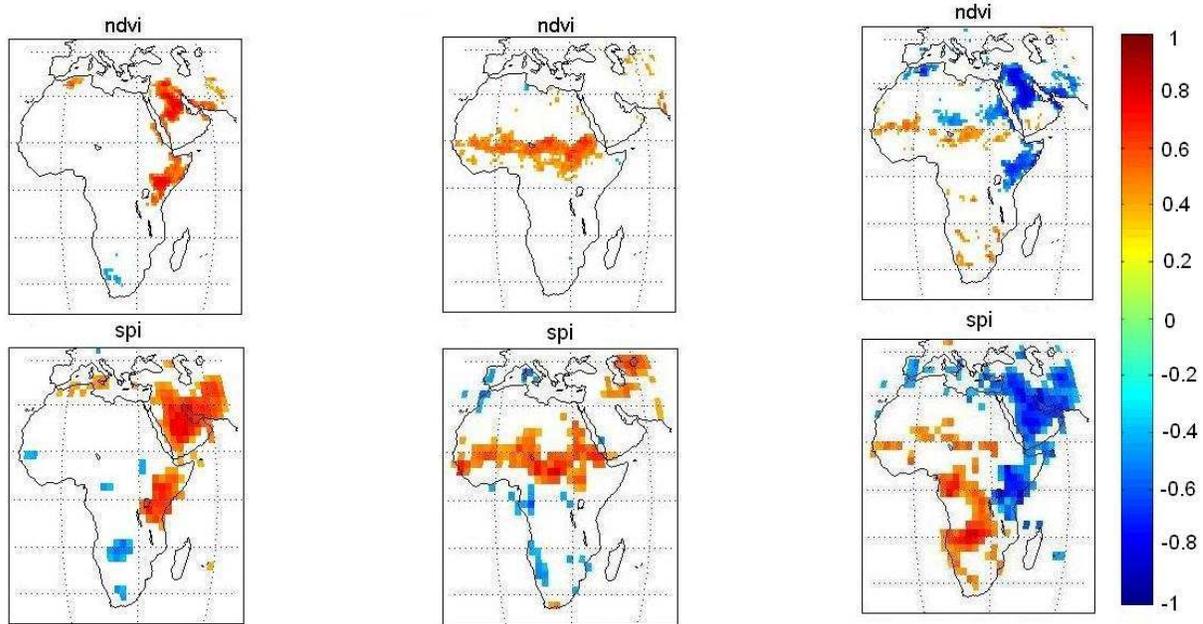
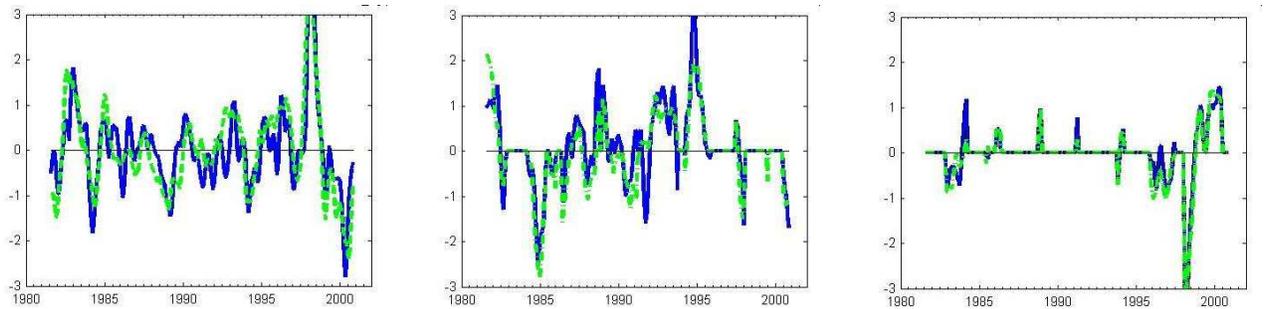
5 Experiments on Earth Science Data Sets

We have demonstrated on artificial data sets that our correlation algorithm is capable of finding locally linear correlation patterns in the data. In this section, we apply our techniques to Earth science data sets. The task is to investigate the relationship between the variability in precipitation and the dynamics of vegetation. Below, we briefly introduce the data sets and then compare our technique to traditional CCA.

In this study, the standardized precipitation index (SPI) is used to describe the precipitation domain and the normalized difference vegetation index (NDVI) is used to describe the vegetation domain [9]. The data for both domains are collected and aligned at monthly time intervals from July 1981 to October 2000 (232 months). Our analysis is performed at continental level for the

continents of North America, South America, Australia and Africa. For each of these continents, we form a data set whose instances correspond to time points. For a particular continent, the feature vector \vec{x} records the SPI value at each grid location of that continent, thus the dimension of \vec{x} equals the number of grid locations of that continent. Similarly, \vec{y} records the NDVI values. Note that the dimensions of \vec{x} and \vec{y} are not equal because different grid resolutions are used to collect the data. The effect of applying our technique to the data is to cluster the data points in time. This is motivated by the hypothesis that during different time periods the relationship between vegetation and precipitation may vary.

For our application, a standard way to visualize CCA results is to use colored map. In particular, to



(a) Conventional CCA

(b) Cluster 1 of MCCA

(c) Cluster 2 of MCCA

Figure 4: The results of conventional CCA and Mixture of CCA (MCCA) for Africa. Top panel shows the NDVI and SPI canonical variates (time series). Middle and bottom panel show the NDVI and SPI maps.

analyze a pair of canonical variates, which are in this case a pair of correlated time series, one for SPI and one for NDVI. We produce one map for SPI and one map for NDVI. For example, to produce a map for SPI, we take the correlation between the time series of the SPI canonical variate and the SPI time series of each grid point, generating a value between -1 (negative correlation) and 1 (positive correlation) for each grid point. We then display these values on the map via color coding. Areas of red (blue) color are positively (negatively) correlated with the SPI canonical variate. Considered together, the NDVI map and SPI map identify regions where SPI correlates with NDVI. Since our technique produces local CCA models, we can visualize each cluster using the same technique.

Note that an exact geophysical interpretation of the produced maps is beyond the scope of this paper. To do so, familiarity with the geoscience terminologies and concepts is required from our audience. Instead, we will present the maps produced by traditional CCA and the maps produced by our technique, as well as plots of the time series of the SPI and NDVI canonical variates. Finally, a high level interpretation of the results is provided by our domain expert. For brevity, the rest of our discussion will focus on the continent of Africa, which is a representative example where our method finds patterns of interest that were not discovered by traditional CCA.

We apply our technique to the data set of Africa by setting $k=2$ and constructing a cluster ensemble of

size 200.⁶ The final two clusters were obtained using the average-link agglomerative algorithm applied to the similarity matrix.

Figure 4 (a) shows the maps and the NDVI and SPI time series generated by traditional CCA. Figures 4 (b) and (c) show the maps and the time series for each of the two clusters. Note that each of the maps is associated with the first pair of canonical variates for that dataset/cluster. Inspection of the time series and the spatial patterns that are associated with the canonical variates for each cluster demonstrates that the mixture of CCA approach provides information that is clearly different from results produced by conventional CCA. For Africa, the interannual dynamics in precipitation are strongly influenced by a complex set of dynamics that depend on El-Nino and La Nina, and on the resulting sea surface temperature regimes in Indian Ocean and southern Atlantic ocean off the coast of west Africa. Although exact interpretation of these results requires more study, the maps of Figures 4 (b) and (c) show that the proposed approach was able to isolate important quasi-independent modes of precipitation-vegetation covariability that linear methods are unable to identify. As shown in [9], conventional CCA is effective in isolating precipitation and vegetation anomalies in eastern Africa associated with El-Nino, but less successful in isolating similar patterns in the Sahelian region of western Africa. In contrast, Figures 4 (b) and (c) show that the mixture of CCA technique isolates the pattern in eastern Africa, and additionally identifies a mode of covariability in the Sahel that is probably related to ocean-atmosphere dynamics in the southern Atlantic ocean.

6 Conclusions and Future Work

This paper presented a method for constructing mixtures of local CCA models in attempt to address the limitations of the conventional CCA approach. We developed a correlation clustering algorithm, which partitions a given data set according to the correlation between two sets of features. We further demonstrated that cluster ensembles can be used to identify the number of clusters in the data and ameliorate the local optima problem of the proposed clustering algorithm. We applied our technique to Earth science data sets. In comparison to traditional CCA, our technique led to interesting and encouraging new discoveries in the data.

As an ongoing effort, we will closely work with our domain expert to verify our findings in the data from

a geoscience viewpoint. For future work, we would also like to apply our technique to more artificial and real-world data sets that have complex nonlinear correlation structure. Finally, we are developing a probabilistic approach to learning mixture of CCA models.

References

- [1] N. Bansal, A. Blum, and S. Chawla. Correlation clustering. *Machine Learning*, 56:89–113, 2004.
- [2] X. Z. Fern and C. E. Brodley. Random projection for high dimensional data clustering: A cluster ensemble approach. In *Proceedings of the Twentieth International Conference on Machine Learning*, 2003.
- [3] X. Z. Fern and C. E. Brodley. Solving cluster ensemble problems by bipartite graph partitioning. In *Proceedings of the Twenty First International Conference on Machine Learning*, pages 281–288, 2004.
- [4] H. Hotelling. Relations between two sets of variates. *Biometrika*, 28:321–377, 1936.
- [5] W. Hsieh. Nonlinear canonical correlation analysis by neural networks. *Neural Networks*, 13:1095–1105, 2000.
- [6] R.A. Johnson and D. W. Wichern. *Applied multivariate statistical analysis*. Prentice Hall, 1992.
- [7] M. Jordan and R. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6:181–214, 1994.
- [8] P. L. Lai and C. Fyfe. Kernel and nonlinear canonical correlation analysis. *International Journal of Neural Systems*, 10(5):365–377, 2000.
- [9] A. Lotsch and M. Friedl. Coupled vegetation-precipitation variability observed from satellite and climate record. *Geophysical research letter*, In submission.
- [10] S. Monti, P. Tamayo, J. Mesirov, and T. Golub. Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, 52:91–118, 2003.
- [11] A. Strehl and J. Ghosh. Cluster ensembles - A knowledge reuse framework for combining multiple partitions. *Machine Learning Research*, 3:583–417, 2002.
- [12] M. Tipping and C. Bishop. Mixtures of probabilistic principal component analysers. *Neural Computation*, 11, 1999.
- [13] E. Zorita, V. Kharin, and H. von Storch. The atmospheric circulation and sea surface temperature in the north atlantic area in winter: Their interaction and relevance for iberian precipitation. *Journal of Climate*, 5:1097–1108, 1992.

⁶We use large ensemble sizes for the Earth science data sets because they contain a small number of instances, making it computationally feasible and also larger ensemble sizes ensure that the clusters we found in the data are not obtained by chance.