Abstract—Clustering is one of the basic operations in data analysis, and the cluster structure of a dataset often has a marked effect on observed patterns in data. Testing whether a data mining result is implied by the cluster structure can give substantial information on the formation of the dataset.

We propose a new method for empirically testing the statistical significance of patterns in real-valued data in relation to the cluster structure. The method relies on principal component analysis and is based on the general idea of decomposing the data for the purpose of isolating the null model. We evaluate the performance of the method and the information it provides on various real datasets. Our results show that the proposed method is robust and provides nontrivial information about the origin of patterns in data, such as the source of classification accuracy and the observed correlations between attributes.

I. INTRODUCTION

Significance testing is an important concept in data mining. Analytical significance tests often cannot be employed to assess data mining results, which has led research on empirical significance testing methods. Randomization is a basic method for empirical significance testing which has been successfully used in many contexts [1], [2]. A randomization method generates a large sample of random datasets that conform to a null model formulated from a null hypothesis to be tested. A measure for a pattern of interest is then calculated from both the original data and the random samples and compared. The pattern in the original data is declared significant if the original significantly differs from the samples in terms of the empirical p-value.

A fundamental task in significance testing is the formulation of the pattern to be tested and the null hypothesis. When using randomization, we must convert the null hypothesis into a null model of the dataset that the randomization then attempts to preserve while perturbing all other features. Despite the common use of null models in analytical tests, empirical randomization-based tests often ignore the null model, leading to overly optimistic significance results.

There are two approaches to building a null model. We may create an exact model that defines an explicit null distribution for the datasets or a property model where we randomly draw datasets that share the required properties of the original dataset. While exact models are obviously preferable in terms of ease of use, such models are often not available for complex null hypotheses, despite some recent advances [3]. On the other hand, property models can be used to form models of arbitrary complexity, but the difficulty of sampling the search space with a desired distribution (often uniform) can range from easy to impossible.

An approach to solving the sampling problem of property models is to express the data in a special form or structure, where the properties selected for conservation are separated as carefully as possible from the rest. In Diagram (1), the dataset $\mathcal{D}$ is converted into a form or structure $\mathcal{S}$ where the null model has been separated. Next, the parts of $\mathcal{S}$ that are unrelated to the null model are randomized. Finally, the randomized data sample $\mathcal{D'}$ is reconstructed from the randomized structure $\mathcal{S'}$. The conversion $(a)$ in Diagram (1) must be invertible so that samples $\mathcal{D'}$ can be obtained from randomized $\mathcal{S'}$ in phase $(c)$.

\begin{equation}
\begin{array}{c}
\text{Original} \\
\downarrow \\
\text{Randomized}
\end{array}
\begin{array}{c}
\mathcal{D} \xrightarrow{(a)} \mathcal{S} \\
\downarrow \\
\mathcal{D'} \xleftarrow{(c)} \mathcal{S'}
\end{array}
\end{equation}

We employ this approach to the case of cluster structure by observing that the cluster structure of a dataset is effectively separated from its other features by means of principal component analysis (PCA). Based on this separation, we introduce a randomization method for testing the statistical significance of patterns relative to a (property) null model that includes the cluster structure. Our experiments show that the proposed method can be used to study, for example, the relevance of cluster structure in classification accuracy. We also give an example of mining a dataset for correlations and determining their dependence on the cluster structure.

The rest of the paper is organized as follows. Section I-A surveys previous work related to our topic. Section II concentrates in analyzing the concept of cluster structure and introduces our suggested method called PCARand. Section II-C presents other methods that might be used on the same problem. Several different types of experiments are conducted in Section III. Finally Section IV gives concluding remarks on the work and explores the concept of representation-based randomization further from the main points of this paper.

A. Related Work

Using randomization for significance testing has been widely studied. Good sources for general information about...
significance testing and randomization are [1] and [4]. Preserving certain properties of datasets during randomization has been discussed and solutions offered in [2], [5] and [3]. In [6] authors show how to do incremental data mining with 0-1 datasets. This means iteratively including new requirements to the null model one by one and doing significance testing on how model requirements interact with each other and with the data pattern under research. Preserving the cluster structure of a 0-1 dataset is discussed in [6].

Research on various null models has been done on many different application fields. In ecology, a wide array of data properties have been incorporated into null models [7]. The transformation approach of Diagram (1) has been successfully in analyzing time series and especially fMRI data through randomizing the wavelet or Fourier representation of data [8]. Spectrum preserving randomization for graphs is studied in [9]. However, no work has been done on incorporating the cluster structure of data to a null model. Most null models that have been suggested and their related sampling methods relate closely to the targeted application and have limited use outside of it.

II. CLUSTER STRUCTURE PRESERVING RANDOMIZATION

A. Cluster Structure in Data

Cluster structure is a basic characteristic of a dataset and it has wide-spreading influence on many data mining and machine learning tasks. Thus it is natural to consider whether data analysis results are implied by the cluster structure alone or due to something else hidden in the data. Recently there have been several reports on the connection between PCA and the standard k-means clustering method, see e.g. [10]. In [10] the authors show that k-means for two clusters is essentially equivalent to partitioning the data based on its first principal component. This connection is also generalized to an arbitrary number of clusters.

Another way to connect the cluster structure and the PCA representation of data is through spectral graph theory. First we create a similarity matrix \( W = DD^T \) from the inner products of the data points. As is known from spectral graph theory, the eigenvectors of \( W \) determine the cluster structure of the graph drawn with \( W \) as adjacency matrix [11]. The eigenvectors of the largest eigenvalues determine large components and the eigenvectors of the smallest eigenvalues determine small-scale congestions in the graph.

For ease of presentation, we assume here that the data has been transformed so that its mean is the zero vector. Let the singular value decomposition of \( \mathbf{D} \) be \( \mathbf{D} = \mathbf{U} \Sigma \mathbf{V}^T \), where \( \mathbf{U} \) and \( \mathbf{V} \) are unitary matrices and \( \Sigma \) is a diagonal matrix of the singular values of \( \mathbf{D} \). The singular values of \( \mathbf{D} \) are also called the spectrum of \( \mathbf{D} \). Taking the product \( \mathbf{P} = \mathbf{U} \Sigma \) the PCA decomposition of \( \mathbf{D} \) can be written as \( \mathbf{D} = \mathbf{P} \mathbf{V}^T \). We call \( \mathbf{P} \) the mixing matrix and \( \mathbf{V} \) the basis change matrix.

Algorithm 1. The PCARand algorithm.

<table>
<thead>
<tr>
<th>Input: Dataset ( \mathbf{D} )</th>
<th>Output: Randomized data sample ( \hat{\mathbf{D}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: ( \mathbf{P}, \mathbf{V} \leftarrow \text{pca}(\mathbf{D}) ) ( \Rightarrow \mathbf{D} = \mathbf{P} \mathbf{V}^T, \mathbf{V} ) is unitary</td>
<td></td>
</tr>
<tr>
<td>2: ( \mathbf{P}_N \leftarrow \text{normalize}(\mathbf{P}) )</td>
<td></td>
</tr>
<tr>
<td>3: ( \hat{\mathbf{P}}_N \leftarrow \text{Margins}(\mathbf{P}_N) )</td>
<td></td>
</tr>
<tr>
<td>4: ( \hat{\mathbf{P}} \leftarrow \text{denormalize}(\hat{\mathbf{P}}_N) )</td>
<td></td>
</tr>
<tr>
<td>5: ( \hat{\mathbf{D}} \leftarrow \hat{\mathbf{P}} \mathbf{V}^T )</td>
<td></td>
</tr>
<tr>
<td>6: return ( \hat{\mathbf{D}} )</td>
<td></td>
</tr>
</tbody>
</table>

In [10] it is shown how the (k-means compatible) cluster structure of data \( \mathbf{D} \) is laid out by the mixing matrix \( \mathbf{P} \). More specifically, the cluster structure is dependent on the singular values of \( \mathbf{D} \) and on the distribution of element values in each of the principal components, that is, in columns of \( \mathbf{P} \). The work in [10] concentrates only in standard k-means, but the same dependencies exist between kernel PCA and kernel k-means methods, see e.g. [12]. Also the work of this paper can be trivially extended to arbitrary cluster structure definitions through kernel methods.

B. The PCARand Algorithm

We now formulate a randomization method for the null hypothesis “The observed pattern is implied by the cluster structure in the data”. By Section II-A the associated null model is captured by the singular values of the data matrix \( \mathbf{D} \) and the distribution of values in the columns of \( \mathbf{P} \).

There are several ways to implement the invertible isolation of phase (a) in Diagram (1). One way would be to randomize only the basis change matrix \( \mathbf{V} \) into \( \hat{\mathbf{V}} \), and then reconstruct the randomized sample as \( \hat{\mathbf{D}} = \mathbf{P} \hat{\mathbf{V}}^T \). This would break the unitarity of \( \mathbf{V} \) since otherwise we would only rotate the data. However, giving up unitarity would cause the original principal components to mix uncontrollably, which we wish to avoid. Another way to implement phase (a) is to randomize the matrix \( \mathbf{P} \) with a method introduced in [5] that preserves the distributions of values in rows and columns. We shall call this method the Margins method and use it also for reference in our experiments.

PCARand, our proposed solution for cluster structure preserving randomization starts by decomposing the data with PCA. The columns of the mixing matrix \( \mathbf{P} \) are normalized using the \( L_\infty \)-norm and then randomized with the Margins method. The randomized mixing matrix is denormalized and combined with the original PCA basis change matrix to form the final randomized data sample. These three steps make up the three phases (a), (b) and (c) of Diagram (1). Algorithm 1 describes PCARand in pseudocode. In detail, the Margins method runs a Markov chain Monte Carlo (MCMC) method of local operations on the matrix. The Metropolis-Hastings [13] algorithm is used to generate samples that with high probability have the least errors in the row and column distributions compared to the original data.
The randomized samples must be exchangeable with the original data so that the empirical p-values are well-defined. Put otherwise, the original and the randomized data must be comparable samples of the sample space. For this, we employ the method of Besag and Clifford [4], [14] that first runs the MCMC chain backwards for N steps to reach state \( \mathcal{R} \). The actual samples are then generated by running the chain forwards for N steps from state \( \mathcal{R} \). This guarantees the correctness of the empirical p-values even if the chain covers only a small part of the sample space.

The main idea in the PCA Rand algorithm is the change of basis to the space of principal components. As shown in Section II-A, even this simple modification in the randomization process has a significant effect on the character and the results of randomization. This raises some questions on the prevalent concept of randomizing data by its rows and columns. While single data points do relate to sensible units of data, the collective set of attributes of a dataset might not be the best way for understanding the data. Feature extraction and selection might end up suggesting sets of composite attributes greatly differing from the original ones.

C. Other Approaches to Preserving the Cluster Structure

The result of k-means clustering is a model of k circular high-density areas within the input data. This model can be approximated with a Gaussian mixture model and used for randomization. Such a model follows the principle “Given the clusters, the variables are independent”. This leads us to the Mixture method described in Algorithm 2. There are several downsides with this approach. Choosing a too small k causes the randomized samples to be too vague and generic, while too high k starts overfitting the data. Therefore an initial model selection phase is needed, using e.g. cross-validation or Bayesian Information Criterion (BIC). Still, Mixture is capable of retaining only the major Gaussian structure of the dataset, removing any intra-cluster or inter-cluster information from it. Additionally, Mixture forgets the identities of the data points, preventing randomization of only a part of the attributes, e.g., when we would like to randomize only the input or the output variables of data.

Another method, PreserveNComponents, runs a simple randomization of the mixing matrix \( \mathbf{P} \), where the strongest

\[ \text{Input: Dataset } \mathbf{D} \text{ of size } m \times n, \]
\[ \text{number of mixture components } k \]
\[ \text{Output: Randomized data sample } \tilde{\mathbf{D}} \]

\begin{algorithm}
  1: \( \mathcal{M} \leftarrow \text{fit mixture model}(\mathbf{D}, k) \)
  2: \( \tilde{\mathbf{D}} \leftarrow \text{sample}(\mathcal{M}, m) \)
  3: return \( \tilde{\mathbf{D}} \)
\end{algorithm}

Algorithm 2. The Mixture algorithm.

We use the local operation Addd of [5] that perturbates single elements of the matrix by small amounts. We control the error in the row and column distributions through histograms of 20 bins with the GM-CdHist method of [5].

III. Experiments

We compare our PCA Rand method to Mixture and PreserveNComponents that also preserve the cluster structure and to Margins that preserves only the value distributions of rows and columns in data. The three datasets, GLASS, LETTER, and OCR, that we use in the experiments are all taken from the UCI Machine Learning Repository [15].

A. Data Spectrum and Clustering Error in Randomization

Preserving cluster structure requires preserving also the singular values of data. This requirement is directly included in PCA Rand and PreserveNComponents, but Mixture preserves the singular values only indirectly. Figure 1 illustrates the results for the GLASS dataset, the results are similar for other datasets. We observe that apart from Margins, the original spectrum is essentially retained. Rotating a dataset, that is, expressing a dataset in a different basis, preserves the data spectrum exactly. However, randomizing data with PCA Rand is not just a rotation. Rotating \( \mathbf{D} \) with a unitary matrix \( \mathbf{R} \) results in \( \tilde{\mathbf{D}} = \mathbf{DR} = \mathbf{P}(\mathbf{V}^\top \mathbf{R}) \), which means that any rotation of \( \mathbf{D} \) keeps the PCA mixing matrix \( \mathbf{P} \) unchanged. Therefore controlling sufficient perturbation in Margins-randomization of \( \mathbf{P} \) ensures that PCA Rand differs from a simple rotation of the data.

Figure 2 displays the k-means clustering error before and after randomization with various methods and values of k for the OCR dataset. Increasing the value of N with PreserveNComponents decreases the error, but the decrease
notably stops when \( N \geq 8 \), suggesting a limit in the amount of cluster structure in the data. The relative performance of PCARand improves when increasing \( k \). With higher \( k \) also the lesser principal components start to have an impact on the clustering error requiring \( \text{PreserveNComponents} \) to keep more principal components untouched, whereas PCARand randomizes the data uniformly in all the components.

### B. Exploring Sources of Classification Accuracy

Supervised classification of data points often uses the cluster structure of the dataset to its advantage. Modern classification algorithms are very good at finding hidden structure in data, but partitioning the data into clusters can be a good starting point for classification, especially when using kernel methods. We use PCARand for testing the relevance of the cluster structure in classification of two different optical character recognition datasets, OCR and LETTER. The attributes in the OCR dataset are essentially the average grey values in different parts of the drawing area for the characters. The classes in the dataset are the ten decimal digits from 0 to 9. The attributes in the LETTER dataset are different statistical measures of distorted characters. The classes of the dataset are the 26 different letters of the English language. As we will see, despite the similarity of the classification tasks, there is a strong difference in the relevance of cluster structure in classification.

We compare the classification accuracy of a support vector machine (SVM) [16] and the \( k \)-means algorithm for the original and randomized data. We use the LibSVM software package [17] and a standard C-SVM with a Gaussian kernel optimized with five-fold cross-validation. Multi-class classification is done with the one-against-one method. In \( k \)-means classification we first run \( k \)-means and assign each cluster with the predominant class label in the cluster. New data points are classified to the class of the cluster with the closest centroid. We use 75\% of the data for training and the remaining 25\% for testing.

Tables I and II display the classification accuracies before and after randomization using the SVM and \( k \)-means classifiers. All results are averaged over 100 randomizations. To ease the discussion of the results, we refer to the entries on Tables I and II using the labels displayed in Table III. The datasets are clearly different in their character. Both classification methods reach comparable accuracies on the original OCR data, but not with the LETTER dataset, where \( k \)-means performs poorly when compared with SVM. In particular, the attributes of the OCR dataset expose the underlying classification structure in a form which \( k \)-means is able to recognize. In contrast, the \( k \)-means algorithm cannot find the structure in the LETTER dataset.

The difference between entries (1) and (4) shows the amount of extra information a SVM classifier can extract from the data in addition to its cluster structure. Differences (2)-(5) and (3)-(6) are indicative of the structure relevant to classification other than the cluster structure, as retained by PCARand and Margins, respectively. The advantage of preserving the cluster structure in randomization can be seen from differences (2)-(3) and (5)-(6). The first one of these

---

**Table I**

<table>
<thead>
<tr>
<th>Randomization</th>
<th>SVM accuracy</th>
<th>( k )-means accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original data</td>
<td>99.4 %</td>
<td>94.4 %</td>
</tr>
<tr>
<td>PCARand</td>
<td>90.0 %</td>
<td>70.8 %</td>
</tr>
<tr>
<td>Margins</td>
<td>69.9 %</td>
<td>57.0 %</td>
</tr>
</tbody>
</table>

**Table II**

<table>
<thead>
<tr>
<th>Randomization</th>
<th>SVM accuracy</th>
<th>( k )-means accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original data</td>
<td>97.9 %</td>
<td>28.5 %</td>
</tr>
<tr>
<td>PCARand</td>
<td>50.2 %</td>
<td>21.6 %</td>
</tr>
<tr>
<td>Margins</td>
<td>32.7 %</td>
<td>19.0 %</td>
</tr>
</tbody>
</table>

**Table III**

<table>
<thead>
<tr>
<th>Randomization</th>
<th>SVM accuracy</th>
<th>( k )-means accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original data</td>
<td>(1)</td>
<td>(4)</td>
</tr>
<tr>
<td>PCARand</td>
<td>(2)</td>
<td>(5)</td>
</tr>
<tr>
<td>Margins</td>
<td>(3)</td>
<td>(6)</td>
</tr>
</tbody>
</table>
PCARand approximately retains the dataset. The preservation of cluster structure is clear. Margins preserves the row and column structure better than the k-means method. Differences are larger because PCARand preserves cluster structure on all scales irrespectively of k, whereas the k-means classification results can extract structure from only one scale and one value of k.

Comparing the two datasets, the difference (1)-(4) highlights the importance of (k-means compatible) cluster structure for the OCR dataset. Let us now compare the entries (4) and (2) for the LETTER dataset. The PCARand method destroys all other features from the data apart from its cluster structure. However, the SVM classifier is able to reach a higher accuracy after PCARand than the k-means classifier for the original data. This suggests that k-means clustering cannot capture all the cluster structure in the data. We can arrive at the same conclusion through differences (2)-(3) and (5)-(6). Indeed, in the case of LETTER the first difference is clearly larger than the second difference, showing the amount of cluster structure existing, but not found by k-means. In the case of OCR this effect is much smaller.

C. Mining Correlations Not Implied by the Cluster Structure

In Figure 3 a histogram of correlations in the OCR dataset gives an example of how preserving clusters relates to correlations in data. The difference between Margins and PCARand is clear. Margins preserves the row and column distributions, but pays no attention to the joint distribution of column pairs, while PCARand separates the original columns to uncorrelated groups (principal components). Running Margins on these groups approximately retains their joint distributions and hence correlations.

However, correlations are not always determined by the cluster structure or vice versa. If we wish to explore different aspects of our dataset and correlations in particular, we need to separate the effect of clusters from the correlations. Otherwise one analysis may assign too high significance to a feature already discovered in a different analysis. Furthermore, often the cluster-dependent correlations drown out the weaker correlations that are independent of the clusters.

We study correlations in the GLASS dataset and assess their significance and dependence on the cluster structure. The data contains 214 samples of glass pieces from which the refractive index and the concentrations of 8 different elements are measured. Hence there are 36 pairwise correlations to be tested for significance. We use 9999 randomized samples for computing the empirical p-values with Margins, PCARand, PreserveNComponents, Mixture and the permutation method. Results are corrected for multiple hypothesis testing by controlling the False Discovery Rate (FDR) at level 0.05 with the Benjamini-Hochberg method [18]. Each randomization method corresponds to a different null hypothesis. For example, running Margins corresponds to assuming the null hypothesis “any structure in data is implied by the row and column distributions” and running PreserveNComponents corresponds to “any structure in data is implied by the N strongest principal components”.

Table IV shows statistical significance results for correlations in the GLASS data. The permutation method is, true to form, overly optimistic and finds a high amount of significance from the data. The Margins method shares this problem as its null model does not take cluster structure into account. Therefore it reports 16 significantly high correlations, although most of them result from sets of points residing in the same area of space, that is, in clusters. PCARand is more successful in culling high correlations that exist only due to the cluster structure. It asserts only the high correlation of 0.327 between sodium and barium concentrations as significant. The same correlation value 0.182 in the randomized samples shows that the two elements co-occur moderately within the different clusters of the dataset. However, the two elements are correlated also across different clusters, a structure that PCARand destroys. Therefore this correlation is only partly explained by the cluster structure and has also another independent cause. The permutation and the Margins method give this correlation a p-value of 0.0001 versus the p-value 0.05 given by PCARand, indicating the dual nature of the correlation.

Neither Mixture nor PreserveNComponents found any significant correlations in the data. Indeed, among all parameter values from 1 to 10, no p-value was below 0.4. PreserveNComponents does not perturb the data enough.
leading to too conservative significance test results. Also the Mixture method fits the original data too closely. To conclude, the majority of the pairwise correlations in the GLASS dataset appear to result from the cluster structure. There are, however, some correlations found by the PCARand method that exist independent of the structure and may give a complementary view into the structure of the dataset.

IV. Conclusion

A. Cluster Structure Preserving Randomization

We proposed a randomization procedure that preserves the cluster structure of a dataset. The motivation for such a procedure is to test for the null hypothesis “the observed pattern is implied by the cluster structure in the data”. Our proposed solution PCARand utilizes the nature of PCA to isolate the cluster structure of data. The method randomizes the PCA mixing matrix by approximately preserving its row and column distributions, thus retaining the cluster structure, but replacing any other data features with features typical of a random dataset with the given cluster structure.

Experiments confirm the validity of our approach in comparison with other algorithms and also in terms of the clustering error before and after randomization. A particular advantage of our algorithm over other methods is its lack of parameters, which avoids the bias inherent in parametrized approaches. To demonstrate applications in assessing data mining results, we used PCARand to study the classification structure of a dataset in relation to its cluster structure. We also mined the correlations in a real dataset and analyzed which correlations are implied by the cluster structure and which exist independently of the clusters.

B. Representation-based Randomization

The approach of Diagram (1) constitutes a basic template for designing randomization based on invertible representations of the data. Indeed, any well-defined decomposition \( D = f(A, B, C) \) of the data can be used to randomize data \( D \) by randomizing any part or parts of the representation and then reconstructing a randomized data sample \( \tilde{D} \).

Such “representation-based” randomization has its advantages. Compared with model-based randomization, there is less bias from assumptions or opinions. Representation-based randomization enables incremental data mining; that is, we first randomize only \( A \), then \( A \) and \( B \) and so on, making incremental changes to our null model and gaining more detailed knowledge about the dataset. Also, many useful representations of data are robust against elementary transformations. For example, the PCA decomposition remains invariant under affine transformations (dilation, rotation and translation), which is often desirable to avoid bias.

As an example of using representations in randomization, the wavelet transform is a common tool for resampling fMRI data, see e.g. [8]. Nonnegative Matrix Factorization (NMF) can be used to decompose text or relational data. A connection between NMF, kernel k-means and spectral clustering is found in [12]. Also Independent Component Analysis (ICA) [19] with its statistically independent components and Linear Discriminant Analysis (LDA) may prove useful for null model formulation in the future.

References