Explainable Clustering and Application to Wealth Management Compliance

Enguerrand Horel  
Stanford University  
ehorel@stanford.edu

Kay Giesecke  
Stanford University  
giesecke@stanford.edu

Victor Storchan  
J.P. Morgan  
victor.storchan@jpmchase.com

Naren Chittar  
J.P. Morgan  
naren.chittar@jpmchase.com

Abstract

Many applications from the financial industry successfully leverage clustering algorithms to reveal meaningful patterns among a vast amount of unstructured financial data. However, these algorithms suffer from a lack of interpretability that is required both at a business and regulatory level. In order to overcome this issue, we propose a novel two-steps method to explain clusters. A classifier is first trained to predict the clusters labels, then the Single Feature Introduction Test (SFIT) method is run on the model to identify the statistically significant features that characterise each cluster. We describe a real wealth management compliance use-case that highlights the necessity of such an interpretable clustering method. We illustrate the performance of our method through an experiment on financial ratios of U.S. companies.

1 Introduction and Related Work

The large amount of data and diversity of sources (financial time series, customers characteristics, financial statements, etc.) from the financial industry, create many applications where unsupervised learning techniques can be leveraged successfully. Clustering methods especially, can find meaningful patterns among this unstructured data and provide support for decision making. Applications of clustering techniques to financial cases can be found in risk analysis [7], credit scoring [12], financial time series analysis [4], [3], portfolio management [8] and financial statements’ anomalies detection [9]. In all these applications, being able to interpret the obtained clusters and explain the rational behind their construction is necessary to trust them and provide transparency to regulators.

Some previous works tackle the problem of interpreting clusters by visualizing them across two or three dimensions typically found by a PCA analysis [11]. This has the disadvantage of restricting the number of dimensions used to explain and in addition, the principal components are no more directly interpretable. Another group of methods uses the centroid or a selected set of points to represent the cluster [10]. These methods, while successful in some cases, are very sensitive to the geometry of the clusters. Distinct from these previous methods that directly interpret the clusters, two-steps methods explain the clusters through an interpretable model that learns how to classify them. The cluster assignment of each point can be used to label the data and train a classifier on them. Classification trees [2] are often used in practice [8]. Because the model has to be interpretable, this prevent the use of a larger class of models such as deep neural networks that could potentially provide better classification accuracy. Another work proposes to directly generate interpretable tree-based clustering models [1]. It has the inconvenient of restricting the type of clustering algorithm that can be used.

Preprint. Under review.
To overcome these limitations, we propose to interpret a classifier trained on the clusters by using the SFIT method \[6\]. This method identifies the statistically significant features of the model as well as feature interactions of any order in a hierarchical manner. It can be applied to any classification model and is computationally efficient. Hence, by combining a two-steps method and this general model interpretability method, we do not have to restrict the choice of clustering technique nor the choice of classifier to predict the cluster assignment. This provides a general interpretability framework that can be applied to any clustering algorithm and type of data.

The structure of this paper is as follows. In section 2, we present a real use-case from the financial industry that poses the problem of interpreting clusters. After explaining our method, we describe in section 3 its key component: the SFIT method used to interpret the cluster classifier. Because the data of the described use-case are highly sensitive, we could not use them directly. As a replacement, we illustrate our method in section 4 on a dataset of U.S. companies clustered using their financial ratios.

## 2 Explainable clustering for compliance monitoring

### 2.1 An overview of the business case

In banks, Wealth Management teams help customers to meet their financial goals by managing their financial assets. An account is associated to each client and an investment strategy is designed by the account manager according to the risk aversion of the client. Using financial performance metrics as features, the efficiency of an account’s strategy can be monitored. Comparison with benchmarks or indices are examples of such performance metrics.

When an account is underperforming based on these features, it is directed to the Compliance team. As a matter of fact, the Compliance team has to spend a considerable amount of time manually reviewing accounts across all monitoring activities and across all global regions to catch the problematic ones. They typically use criteria learned from previous cases to identify the accounts to escalate for further investigation. This is often done using very few features in a not very principled way. As a consequence, systematic explanation of why an account was flagged as problematic is missing in most cases.

In order to automatise this procedure and cover all the different cases that result in an underperforming behavior, a clustering algorithm is implemented. Clustering has the advantage of taking into account a large number of different risk factors. This enables to group accounts that behave similarly and bring understanding on the underlying reasons of poor performance. Clustering accounts can also be used to sample from the cluster distribution and estimate the frequency of one particular type of underperformance. However, the clustering method has to be fully understandable by the Wealth Management Compliance team so that they can explain what are the significant features that characterise each cluster and ensure that it is aligned with their expertise and domain knowledge.

### 2.2 A proposal to mitigate the compliance monitoring challenge

As mentioned in the previous section, our goal is to design an interpretable clustering capability for Compliance Monitoring teams that can then be used to select underperforming accounts.

**The Clustering Step.** A clustering algorithm (such as K-means, DBSCAN, or agglomerative clustering) is run over the set of all underperforming accounts.

**The Explaining Step.** A label is assigned to every cluster which allows to label the whole set of accounts. We can then train a classifier in a supervised way using this dataset. This classifier learns to predict the cluster of a given account. The SFIT procedure can now be applied on the trained classifier. A single model has been trained to classify all clusters, but we ultimately want to interpret each cluster independently. To interpret a specific cluster, the SFIT method is run using only the accounts belonging to this cluster. This allows us to provide for each cluster, a set of features that are significantly characterising it.
3 Presentation of the SFIT method

3.1 Introduction

SFIT [6], is a method that assesses the statistical significance and importance of features of machine learning models. It is based on a novel application of a forward-selection approach. Given a trained model and one of its features, it compares the predictive performance of the model that uses only the intercept with the model that uses both the intercept and the feature. The performance difference captures the intrinsic contribution of the feature in isolation which leads to an informative notion of feature importance. This approach has the advantage of being robust to correlation among features. Other advantages of our method include: (1) it does not assume any assumptions on the distribution of the data nor assumptions on the specification of the model; (2) it can be applied to both continuous and categorical types of features; (3) in addition to assessing the contribution of individual features, it can also identify higher order interactions among features in a hierarchical manner.

Formally, a set of \( n \) i.i.d. accounts \( Z_1, ..., Z_n \sim \mathcal{P} \) with \( Z_i = (X_i, Y_i) \). \( X_i \) is a vector of size \((p + 1)\) that contains the \( p \) features measuring the performance of the account plus an intercept at the first coordinate. The features can be a mix of continuous and categorical variable with the latter assumed to be binary variables through one-hot encoding. \( Y_i \) represents the index of the cluster \( \{1, 2, ..., C\} \) where \( C \) represents the total number of clusters. We randomly split the accounts \( \{1, ..., n\} \) into two subsets \( Z_1 \) and \( Z_2 \) and denote the two corresponding split of the data as \( \mathcal{D}_k = \{(X_i, Y_i) : i \in I_k\} \), \( k = 1, 2 \).

We denote by \( \hat{\mu} \), the cluster classifier trained on \( \mathcal{D}_1 \). To evaluate the contribution of the individual feature \( j \) removed from the potential interaction that it could have with the remaining features, we define the transformed input vector \( X_i^{(1,j)} \) which is obtained from \( X_i \) where all entries except for the \( j^{th} \) coordinate and the intercept are replaced with 0. Similarly, \( X_i^{(1)} \) is the transformed input vector where all entries except for the intercept are set to zero. This transformed input prevents us from having to re-fit a new model for each input. Then, given the loss function \( L(Y, \hat{\mu}(X)) \) used to train the classifier (like the cross entropy loss for instance), we can define

\[
\Delta_j = \Delta_j(X_i, Y_i) = (1 - \beta)L(Y_i, \hat{\mu}(X_i^{(1)})) - L(Y_i, \hat{\mu}(X_i^{(1,j)}))
\]

the difference between \((1 - \beta)\%\) of the prediction loss from the model using the intercept term only and the loss from the model using the intercept plus the feature \( j \). \((1 - \beta)\) times the baseline loss and not the loss value itself is considered to make this test more robust to non-informative variables and control its type-I error. More details about this parameter and how to optimally select can be found in [6]. Let’s now define \( \hat{m}_j \), the metric that is used to assess the significance of feature \( j \):

\[
\hat{m}_j = \text{median}_{i \in \mathcal{D}_2} \Delta_j
\]

for \( 2 \leq j \leq p + 1 \). \( \hat{m}_j \) is defined as the median over the inference set \( \mathcal{D}_2 \) of the differences of predictive performance. Intuitively, \( \hat{m}_j \) represents the predictive power of variable \( j \) compared to a baseline model that only has an intercept term, the bigger it is, the more predictive power the variable contains.

Using a standard sign test, it is possible to obtain finite sample confidence interval for \( m_j \) and to perform the following one-sided hypothesis test of significance:

\[
H_0 : m_j \leq 0 \text{ versus } H_1 : m_j > 0
\]

using the statistic \( \hat{m}_j \).

This method can be generalized to higher-order interactions between features as explained in more details in [6].

4 Experiment

4.1 Data

To illustrate our explainable clustering method, we use the Financial Ratios Firm Level dataset from Wharton Research Data Services (WRDS). This dataset provides, for all U.S. companies, 71 commonly used financial ratios grouped into the following seven categories: capitalization, efficiency,
financial soundness/solvency, liquidity, profitability and valuation. From this database, we extract a subset of 682 companies which corresponds to all the unique companies listed over the last 10 years. In addition, we have for each company, its NAICS (North American Industry Classification System) and description. The data are centered and scale to unit variance as a pre-processing step.

4.2 Results

We cluster our dataset of companies into 5 clusters using an agglomerative hierarchical clustering algorithm. This algorithm works in a bottom-up fashion: each observation starts in its own cluster, and then, clusters are successively merged together. We use a Ward linkage that minimizes the sum of squared differences within all clusters, and euclidean distance. We obtain the following clusters:

- cluster 1: 201 samples (manufacturing, retail),
- cluster 2: 277 samples,
- cluster 3: 139 samples (energy, resources),
- cluster 4: 60 samples (telecommunication, technology),
- cluster 5: 5 samples.

Because cluster 5 does not have a significant enough size to perform meaningful analysis, we choose to discard it. By looking at the industry code of the companies of each cluster, we notice that the largest cluster contains a mix of various industries while the 3 remaining clusters are fairly specialized as listed above.

We then label each company using its cluster assignment. We train a 3 hidden layers fully connected neural network to perform classification on the clusters. The dataset is split into three parts, a training set of size 480, a validation set of size 125 and a test set of size 77. We optimize the architecture of the network through random search over the validation set. We end up using ReLU as activation function and a first hidden size of 100, a second hidden size of 50 and a third of 25. The network is trained for at most 50 epochs using Adam optimizer and early stopping. We obtain a classification accuracy of 0.88 on the test set.

We finally run on the trained network one SFIT method per cluster, i.e. by only using the data of this cluster, and returns the five most important features. For the first cluster, the first five features are: gross profit margin, asset turnover, long term debt, current debt and net profit margin, the value of their corresponding test statistic along with their 95% confidence interval can be found in Table 1. For the second cluster, they are: book/market, free cash flow/operating cash flow, pre-tax return on total earning assets, sales/invested capital, total debt/ebitda as displayed in Table 2. For the third cluster: total debt/ebitda, operating profit margin before depreciation, pre-tax return on total earning assets, free cash flow/operating cash flow, return on assets as shown in Table 3. And for the last cluster: research and development/sales, cash balance/total liabilities, gross profit margin, cash ratio, operating cf/current liabilities as displayed in Table 4. From these, we can see that the most predictive features are not the same for each cluster. They capture what makes a cluster distinct from the others. This is an efficient way to interpret and explain what are the intrinsic characteristics of a cluster. These results are also consistent with the main types of industries present in each cluster. For instance, the research and development/sales ratio is the most significant feature of the cluster that mainly contains telecommunication and technology companies.

| Variable   | Median | 95%-CI lower bound | 95%-CI upper bound |
|------------|--------|-------------------|-------------------|
| gpm        | 0.755  | 0.657             | 0.835             |
| at_turn    | 0.677  | 0.554             | 0.850             |
| lt_debt    | 0.556  | 0.366             | 0.615             |
| curr_debt  | 0.542  | 0.391             | 0.728             |
| npm        | 0.532  | 0.509             | 0.568             |

Table 1: Cluster 1 top 5 most significant variables along with their corresponding test statistics and confidence intervals.
Table 2: Cluster 2 top 5 most significant variables along with their corresponding test statistics and confidence intervals.

| Variable | Median | 95%-CI lower bound | 95%-CI upper bound |
|----------|--------|---------------------|---------------------|
| bm       | 0.114  | 0.102               | 0.126               |
| fcf_ocf  | 0.057  | 0.055               | 0.059               |
| pretret_earnat | 0.043 | 0.024               | 0.061               |
| sale_invcap | 0.037 | 0.035               | 0.040               |
| debt_ebitda | 0.031 | 0.021               | 0.039               |

Table 3: Cluster 3 top 5 most significant variables along with their corresponding test statistics and confidence intervals.

| Variable | Median | 95%-CI lower bound | 95%-CI upper bound |
|----------|--------|---------------------|---------------------|
| debt_ebitda | 0.856  | 0.746               | 0.987               |
| opmbd    | 0.692  | 0.360               | 0.903               |
| pretret_earnat | 0.518 | 0.489               | 0.545               |
| fcf_ocf  | 0.419  | 0.235               | 0.656               |
| roa      | 0.382  | 0.337               | 0.422               |

Table 4: Cluster 4 top 5 most significant variables along with their corresponding test statistics and confidence intervals.

| Variable | Median | 95%-CI lower bound | 95%-CI upper bound |
|----------|--------|---------------------|---------------------|
| rd_sale  | 1.89   | 1.70               | 2.02                |
| cash_lt  | 1.23   | 1.09               | 1.54                |
| gpm      | 0.718  | 0.564              | 0.796               |
| cash_ratio | 0.449 | 0.333              | 0.527               |
| ocf_lct  | 0.283  | 0.119              | 0.598               |

We propose in this paper a novel two-steps method that can interpret any clustering algorithms. For each cluster, this method identifies the statistically significant features that characterise it as well as feature interactions. We justify the necessity of such a method by describing a Wealth Management Compliance use-case that requires explaining clusters of underperforming accounts. We demonstrate its effectiveness on a dataset of financial ratios of U.S. companies.

Acknowledgments

The authors would like to thank the Wealth Management Technology team at J.P. Morgan and especially Amish Seth for their collaboration and help to gain insight on the business case.

References

[1] Dimitris Bertsimas, Agni Orfanoudaki, and Holly Wiberg. Interpretable clustering via optimal trees. *arXiv preprint arXiv:1812.00539*, 2018.

[2] Leo Breiman. *Classification and regression trees*. Routledge, 2017.

[3] José G Dias, Jeroen K Vermunt, and Sofia Ramos. Clustering financial time series: New insights from an extended hidden markov model. *European Journal of Operational Research*, 243(3):852–864, 2015.

[4] Martin Gavrilov, Dragomir Anguelov, Piotr Indyk, and Rajeev Motwani. Mining the stock market: Which measure is best. In *Proc. of the 6th ACM SIGKDD*, pages 487–496, 2000.
[5] TP Hancock, DH Coomans, and YL Everingham. Supervised hierarchical clustering using cart. In *Proceedings of MODSIM 2003 International Congress on Modelling and Simulation*, pages 1880–1885. Citeseer, 2003.

[6] Enguerrand Horel and Kay Giesecke. Computationally efficient feature significance and importance for machine learning models. *arXiv preprint arXiv:1905.09849*, 2019.

[7] Gang Kou, Yi Peng, and Guoxun Wang. Evaluation of clustering algorithms for financial risk analysis using mcdm methods. *Information Sciences*, 275:1–12, 2014.

[8] Victoria Lemieux, Payam S Rahmdel, Rick Walker, BL Wong, and Mark Flood. Clustering techniques and their effect on portfolio formation and risk analysis. In *Proceedings of the International Workshop on Data Science for Macro-Modeling*, pages 1–6. ACM, 2014.

[9] Samir Omanovic, Zikrija Avdagic, and Samim Konjicija. On-line evolving clustering for financial statements’ anomalies detection. In *2009 XXII International Symposium on Information, Communication and Automation Technologies*, pages 1–4. IEEE, 2009.

[10] Dragomir R Radev, Hongyan Jing, Małgorzata Styś, and Daniel Tam. Centroid-based summarization of multiple documents. *Information Processing & Management*, 40(6):919–938, 2004.

[11] C Radhakrishna Rao. The use and interpretation of principal component analysis in applied research. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 329–358, 1964.

[12] Hongshan Xiao, Zhi Xiao, and Yu Wang. Ensemble classification based on supervised clustering for credit scoring. *Applied Soft Computing*, 43:73–86, 2016.