Machine Learning and Hedge Fund Classification using Self-Organizing Map

Satyabrota Das, CAIA & Hossein Kazemi
CAIA Association
January 27, 2019

Machine Learning (ML) and Artificial Intelligence (AI) have become ubiquitous in everyday life. Some of the applications of ML and AI affect us directly, while others subtly affect us. They are widely used in technologies associated with applications such as facial recognition, email spam and malware filtering, chatbots, etc., and in advanced applications such as robotic surgery, genomic sequencing, radiation treatment, etc. The field of finance has seen increasing use of ML and AI application, especially in the asset management industry.

Increased use of ML and AI techniques in finance can be attributed both to data from new sources, such as from credit card transactions, mobile phone location information, and from satellite images, and to easier access to highly powerful computational devices. ML and AI techniques in fraud detection and credit approval have been used for a very long time. Asset managers and asset allocators, however, have shown increased interest in the use of ML and AI tools for asset allocation, trading and generating investment ideas more recently. Applications by asset managers include analyzing credit-card data, using textual analysis on company filings, using satellite image analysis for revenue forecasting, etc.

Many of the applications of ML and AI use tools that can broadly be categorized as classification tools. They help categorize a group of data points into a small number of discrete groups containing data points with similar attributes. Whenever we are detecting fraud or making decisions on credit approval, we are essentially grouping all cases into two – good and bad or approve and not approve, and the task can be accomplished by using widely known classification techniques such as logistic regression, neural networks, classification trees and support vector machines.

In this article, we explore an application of a classification technique in finance. In particular, we examine a simple application of the Self-Organizing Map to see if the technique can be used to divide further a group of hedge funds that are already labeled as following the same hedge fund strategy. For example, while a group of hedge funds may be classified as equity long/short managers, there may be several distinct subgroups of managers with some following trend following strategies while others following fundamental strategies. Such an exploratory analysis can be useful for an investor, such as a fund of hedge funds, which is trying to construct portfolios of hedge funds that are highly diversified and do not over allocate to managers that follow the same strategy. Equally, important, the fund of funds manager would want to perform an initial evaluation of many managers to reduce the sample and therefore the due diligence cost. Rather than undertaking a full-fledged analysis of each fund, SOM can help divide the space of all acceptable funds into smaller homogenous groups, which can then be analyzed separately. Portfolio formed by choosing funds from different groups is expected to provide greater diversification benefits.

Our study is similar to the study by Harvey, Rattray, Sinclair and van Hemert (2017), who classify Equity Hedge and Macro Funds into systematic and discretionary funds and analyze their performance. They use textual analysis on each fund's description to bifurcate funds into systematic and discretionary funds and analyze the performance of the two groups, whereas we use monthly returns data to classify funds into many groups and examine the characteristics of the funds in different groups.

Our results indicate that machine learning algorithms can be used effectively to classify funds into homogenous groups. These algorithms are especially good at isolating funds that are very different from the other funds in our sample. Such funds could either be a source of alpha or a problem fund that is most likely not following the stated strategy. At the very least, machine learning tools can be used as a starting point for deeply analyzing a large number of funds.

This article is organized into several sections. We first provide a very brief introduction to the artificial neural network (ANN) and show how a simplified version of it may be used to classify managers. Next, we describe the Self-Organizing Map, which uses ANN to perform classification in a more complex and adaptive way. Then, we discuss the data used in our analysis, and finally, we show some results.

Introduction to Artificial Neural Networks (ANN)

Suppose we have historical observations on a set of economic variables such as unemployment, inflation, GDP growth and so on. Given these historical observations, we are interested to see if they can be used to predict the probability that the economy would enter a recession. We represent our historical observations by e_{it}, which is our observation of economic variable i at time t. These observations along with our observations of historical recessions can be presented in the following form

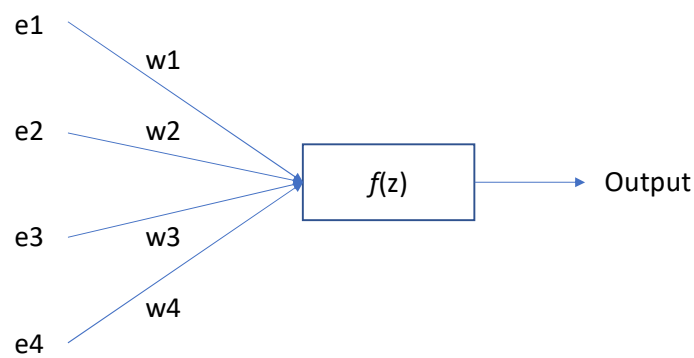
Table 1: Economic Variables and Recession

Time	Input: Economic Variables				Output: Recession
1	e_{11}	e_{21}	e_{31}	e_{41}	1
2	e_{12}	e_{22}	e_{32}	e_{42}	0
3	e_{13}	e_{23}	e_{33}	e_{43}	0
4	e_{14}	e_{24}	e_{34}	e_{44}	1

As you can see, we have collected four economic variables covering four periods. Also, we can see that recessions followed periods 1 and 4 while recessions did not follow periods 2 and 3.

Our goal is to feed these economic variables (inputs) into a function or functions where the output would be either 0 or 1 depending on whether a recession was observed. Once we have estimated a function that performs well in in-sample, we plan to use future observations of the same economic variables to obtain estimates of the likelihood that a recession could follow. Graphically, the process may look like this

Figure 1: A single Node



Here, the inputs are fed into the neuron containing the function $f(z)$, which produces an output representing the occurrence of no recession (0) or recession (1). Variable z is a function of the economic variables (e.g., average). Since these economic variables may not be equally important in influencing the outcome, it makes sense to take a weighted average of these economic variables. That is, in each date, we estimate the variable z_t as:

$$z_t = w_1 e_{1t} + w_2 e_{2t} + w_3 e_{3t} + w_4 e_{4t} + \theta$$

The parameter θ is called the bias. The variable z_t is then fed into the function

$$f(z) = \frac{1}{1+\exp(-z)}$$

This function has an interesting property that it will be between zero and 1. If z is a very large positive number then the function will approach 1, and if z is a large negative number, then the function will approach 0. The weights are selected such as the value of $f(z)$ is as close as possible to the observed outcome, recession (1) or no recession (0). The following figure displays the behavior of the function:

Figure 2: Behavior of $f(z)$

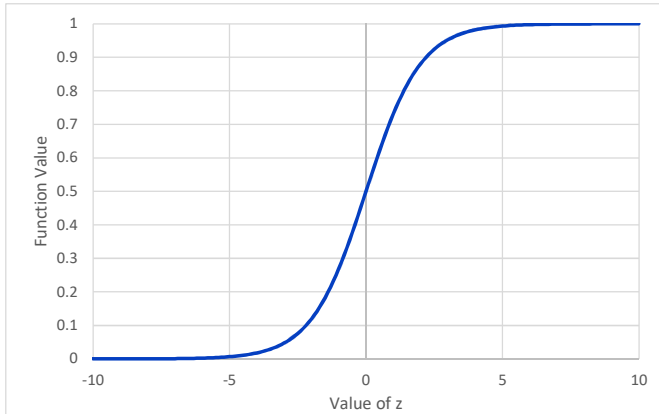
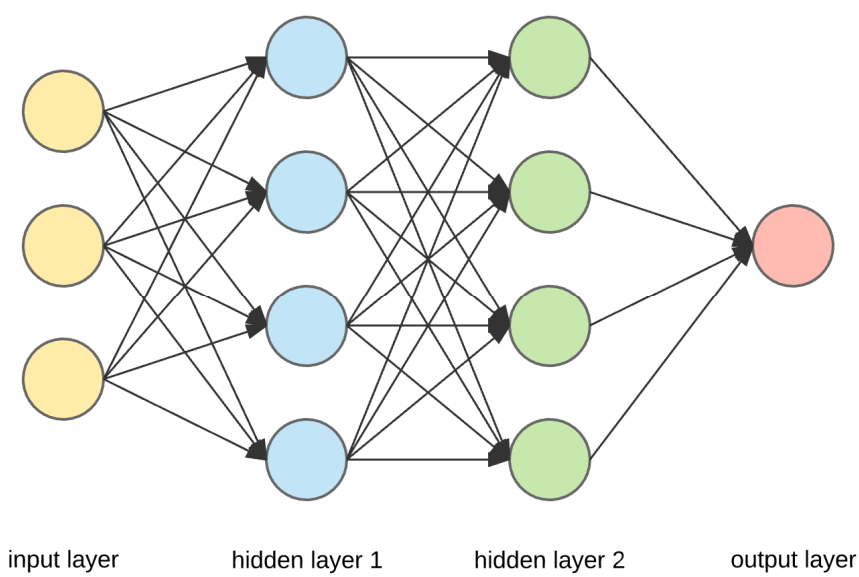


Figure 1displays a single neuron, where it receives a signal and produces an output. In a neural network, different weighted averages of the economic variables may be fed into many neurons, and the outputs of those neurons could be fed into another set of neurons. Each set of neurons is referred to as a layer. Therefore, if we think of inputs as one layer and the single neuron that produces the output as another layer, then Figure 1 has two layers. If there are layers of neurons between the input and the output layers, then they are called the hidden layers. The following figure displays an ANN with four layers.

Figure 3: A Neural Network



For each layer, a different weighted average of the outputs of the previous layer is fed into the above function (other functions may be used as well). As data is fed to the ANN, it adapts by adjusting the weights and the bias parameters to improve its performance by reducing its in-sample error until no further improvement can be made. At that point, the researcher can feed new data into the ANN to obtain a new prediction regarding the possibility of a recession in the coming period.

The process described above represents a *supervised* learning algorithm because we had "labels" for historical outcomes – recession or no recession. That is, we helped the program learn that a particular state of the economy is called a recession and another one is called an expansion.

An example of *unsupervised* learning would be to feed data about purchasing histories of many customers into an ANN, and then ask the network to determine the number of distinct groups of customers that exist. More commonly we may pre-specify the number of groups and then ask the ANN to create optimal clusters of the customers. The clustering procedure will assign each customer to one of those groups such that members of each group are as similar as possible and that each group is as dissimilar from other groups as possible.

Self-Organizing maps

A Self-Organizing Map or SOM, as it is commonly known, is a particular type of neural network that is trained using unsupervised learning to produce a low dimensional representation of a high dimensional input data. For example, suppose we have a large set of companies, and for each company, we have several pieces of information such as size, sales, ROE, leverage, and so on. If we have 50 pieces of information about each firm, there is no way we can visually inspect our sample and decide how many different types of firms are present. A SOM algorithm will reduce this 50-dimensional problem into a two-dimensional problem, which can then be visually inspected. An example can help demonstrate this point.

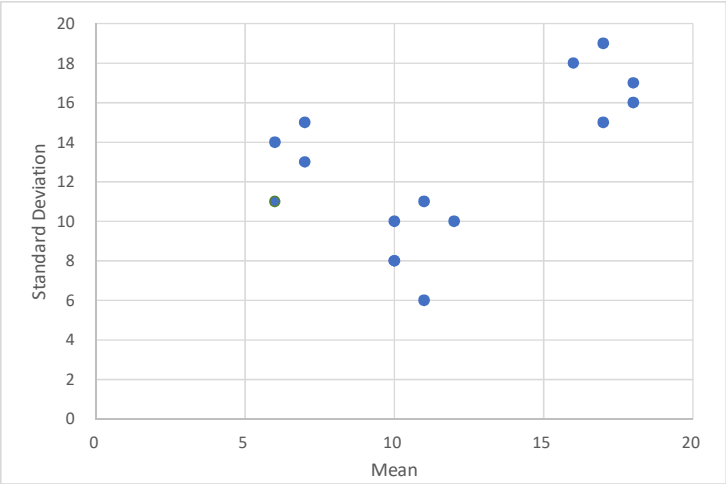
Suppose we have the following data about 15 money managers.

Table 2: Managers and Their Characteristics

	Mean	Standard Deviation	Correlation with S&P 500
	%	%	%
M1	10	10	70
M2	18	17	50
M3	11	11	20
M4	6	11	12
M5	6	14	0
M6	7	15	60
M7	7	13	50
M8	11	6	25
M9	16	18	40
M10	17	19	60
M11	12	10	75
M12	18	16	65
M13	17	15	30
M14	10	8	20
M15	10	8	10

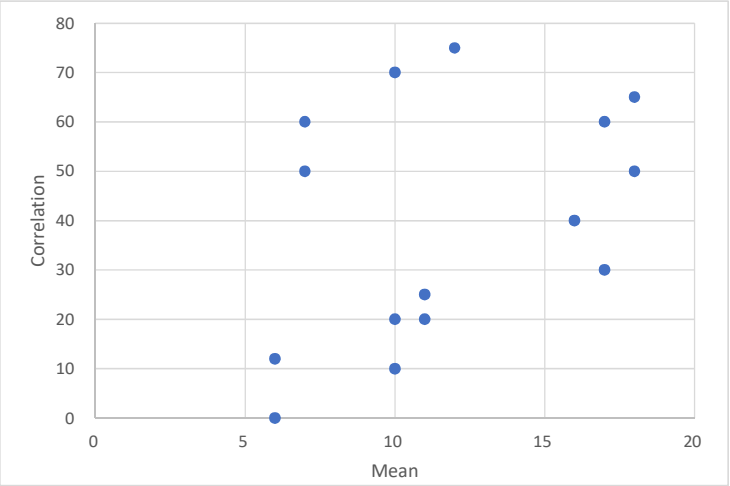
For each manager, we have historical annualized mean and standard deviation as well as correlation with the S&P 500 Index. Suppose we wish to see if these managers form 2-3 distinct groups. Just by looking at the numbers, we may not be able to accomplish this task. However, let us plot these managers in the mean-standard deviation space first.

Figure 4: Plot of Managers in Mean-Standard Deviation Space



We can see that they form three distinct groups if we were to look at their means and standard deviations. However, if we were to plot these managers in the space of mean-correlation, we would notice that there are perhaps four or five distinct groups.

Figure 5: Plot of Managers in Mean-Correlation Space

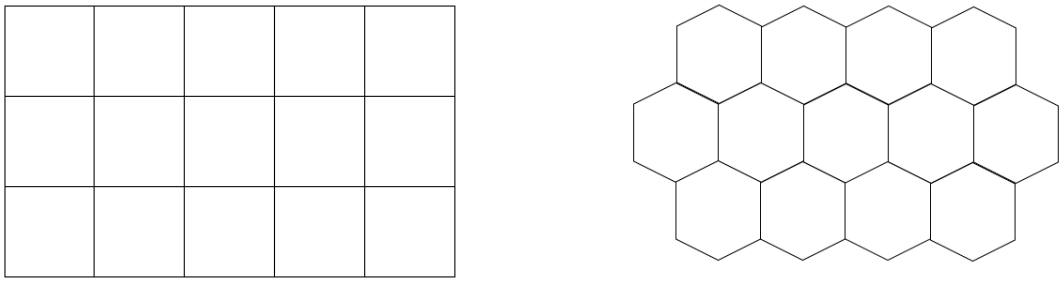


Now, suppose that we have other pieces of information such as skewness, maximum drawdown, Sortino ratio, etc. about these managers. The task of identifying distinct groups of these managers would be impossible unless we use a classification or clustering technique. We could employ the SOM algorithm to look at optimally weighted averages of these characteristics such that the managers can be put in a pre-defined number of distinct groups.

SOM aims to divide a heterogeneous group of data points into smaller homogenous sub-sets. It was introduced by Teuvo Kohonen (1982) and is also known as the Kohonen map. SOM is highly useful for visualization tasks of complex structures that would otherwise be extremely difficult for a human to recognize. An important feature of the Kohonen Map is that it attempts to preserve the relationship in the data while producing a two-dimensional output. It accomplishes this by selecting the weights that are applied to each characteristic (i.e., input) such that they are close to those characteristics. Finally, SOM is a single layer competitive process in the sense that the output nodes compete with each other to best represent the particular input sample. The success of the representation is measured using a discriminant function, where a set of input (i.e., managers) is compared with the weight vector of each output node. The particular node with its connection weights most similar to the input sample is declared the winner of the competition.

Self-Organizing Map mainly has two distinct components – a node-set data structure representing the actual map with contents and algorithms that apply to that node set. The basic principle of building a SOM is to set certain operational parameters, initialize the node set and apply its algorithms to modify its node set according to the inputs presented. The number of nodes is prespecified by the user in most instances, and the job of the algorithm is to find the position of input data in the grid. Note that SOM can only handle numeric attributes and any categorical data must be converted to a suitable scale before supplying to SOM.

Figure 6: SOM grids



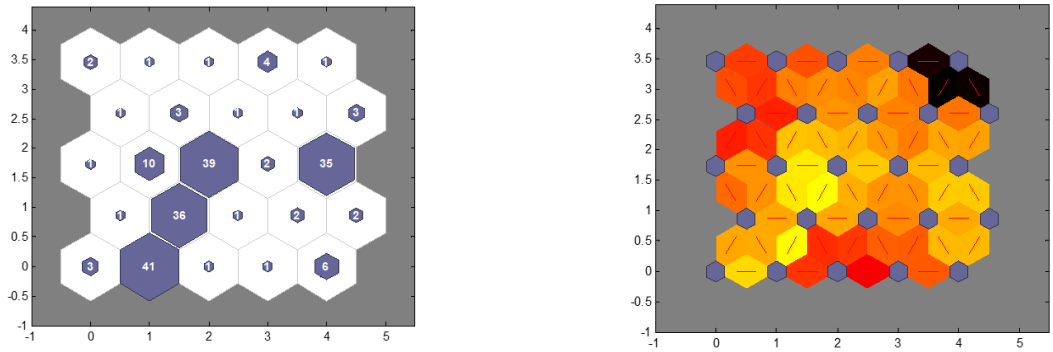
a) Rectangular grid b) Hexagonal grid

Typically, the grid in SOM is laid out either in a rectangular or in a hexagonal fashion as shown above with each cell containing a neuron. The hexagonal grid is preferred if greater variance among neighborhood size is desired, although both are equally prevalent in practice. These shapes can cause problems on the edge nodes as the neurons at the edge of the network are less central than the other neurons. To decrease the effects of edge node, spherical or geodesic structure can be used.

Some measure of distance, such as Euclidian, correlation, or direction cosine, is used on input data points to map the inputs to the output space. Initially, a set of weights are assigned to each neuron. This can be assigned randomly or by using samples from input data or by using principal components. The objective of the SOM algorithm is to minimize the distance between the input and the neurons. After each iteration, the weights of each neuron can be updated as well as input data points associated with each neuron to minimize the sum of squared distance.

Once the SOM algorithm has run through the input data and found the optimum classification, the output can be plotted as shown in Figure 7 for visual inspection. The first part of Figure 7 presents the number of input data points in each group while the second part presents the distance among the neurons. Each node can have a different number of data points associated with it as shown in the left part of Figure 7. As this is a hexagonal grid, a central node can have six distances with its neighbors. The lighter color cells on the right-hand side of Figure 7 indicate shorter distance, while the darker color cells indicate longer distance. It is apparent that although the neighbors can be adjacent, distance from one to the other can vary based on the position of the neighbors. The distance between the neighbors of the top-right corner node is greater than the distances between any other nodes. This indicates data points associated with the top-right node are very different from all other data points and warrants closer inspection.

Figure 7: Output of SOM



a) Observations in each neuron

b) Neighbor weight distance

Data Used:

We used hedge funds that are classified as long-short equity in the Morningstar CISDM Hedge Fund Database for our analysis. Three groups of funds – Global Long/Short Equity, US Long/Short Equity and US Small Cap Long/Short Equity - were used. There are 2,440 funds in these three groups, including the funds that are classified as dead. When we only considered US Dollar denominated funds, the number of funds came down to 1,948. Missing data points can cause SOM algorithms to produce erroneous results. So, we required funds with monthly return history without any missing monthly returns. After excluding funds that have missing data in any months between January 2012 and December 2017, we ended up with 195 funds. Our goal is to analyze these 195 funds over two different sub-periods – one 4-year long and the other 2-year long. We considered January 2012 to December 2015 as our first study period and January 2016 to December 2017 as our second study period.

Results and Discussions:

We, first, ran SOM with a 3 by 3 hexagonal grid and returns for the 195 funds from the January 2012 to December 2015. SOM would put the funds into 9 groups and associate each fund with a label indicating its group. Group numbers provided by SOM are meaningful to a certain extent. If two group numbers are consecutive, their position in the grid is also adjacent, and weights associated with adjacent nodes tend to be similar. Dissimilar nodes are usually placed at the edges.

Table 3 presents summary statistics for the groups produced by SOM when we ran it with monthly returns from January 2012 to December 2015 as features. Values shown in the table are a cross-sectional average of the respective statistics, i.e., value for each fund is calculated first, and then the average of the values in a group are computed.

Table 3: Summary results of different groups for the period 2012 to 2015

Group Number	Number of Funds	Average Annualized return (%)	Average Annualized Standard Deviation (%)	Average Maximum Drawdown (%)	Average Correlation with SP500	Avarage Pairwise Correlation
1	3	20.82	39.91	-42.61	0.48	0.78
2	11	-2.38	22.58	-47.75	0.51	0.46
3	1	2.32	86.19	-66.98	0.06	-
4	4	-0.97	17.52	-31.44	0.30	0.99
5	6	25.93	16.82	-16.42	0.47	0.59
6	1	19.00	33.45	-60.81	0.19	-
7	29	9.33	14.93	-21.68	0.52	0.46
8	69	9.79	11.51	-13.91	0.70	0.55
9	71	7.93	7.81	-8.41	0.35	0.16

Few things to note from Table 3. Some of the groups have few funds, and these funds seem to appear noticeably different from funds in other groups. For example, the lone fund in group 6 has very high return relative to an average fund as well as has high volatility and drawdown, while the single fund in group 3 has a low return, but extremely high volatility and high drawdown. It is also apparent that there are three major groups as the last three groups contain 169 funds out of the 195 funds.

Ideally, we would expect to find that correlation within the groups will be higher than correlation across groups as the goal of the algorithm is to bring together similar funds in the same group. To examine this hypothesis, we found the average correlation among funds from different groups. We, first, found the pairwise correlation for all different combinations of funds between two different groups and then calculated the average of the pairwise correlations. Figure 8 shows the correlation matrix among different groups.

Looking at the diagonal of Figure 8, we can see that some of the correlations are very high – even close to almost 1. Groups with such high correlation, such as groups 1 and 4 have a small number of funds in them, and these are most likely different share classes of the same fund. The interesting groups are the groups with a large number of funds. If we look at groups 7 and 8, which have 29 and 69 funds, respectively, we see that funds within the group have a high correlation, whereas these funds have low correlation with funds from other groups. This confirms our hypothesis that funds within the group tend to have higher correlation relative to funds across the group.

Figure 8: Average correlation coefficient among funds in different groups - 2012 to 2015

Groups	1	2	3	4	5	6	7	8	9
Number of Funds	**3**	**11**	**1**	**4**	**6**	**1**	**29**	**69**	**71**
1	0.78	0.43	-0.05	0.30	0.22	0.27	0.32	0.42	0.19
2	0.43	0.46	0.08	0.23	0.27	0.20	0.36	0.44	0.18
3	-0.05	0.08		0.28	0.04	0.06	0.04	0.08	0.01
4	0.30	0.23	0.28	0.99	0.17	0.18	0.34	0.32	0.19
5	0.22	0.27	0.04	0.17	0.59	0.14	0.30	0.40	0.25
6	0.27	0.20	0.06	0.18	0.14		0.09	0.16	0.10
7	0.32	0.36	0.04	0.34	0.30	0.09	0.46	0.42	0.22
8	0.42	0.44	0.08	0.32	0.40	0.16	0.42	0.55	0.27
9	0.19	0.18	0.01	0.19	0.25	0.10	0.22	0.27	0.16

We also ran the funds through SOM using data from January 2016 to January 2017. Table 4 summarizes summary statistics for the groups formed using returns from this period. As with table 1, funds with extreme values for different statistic are grouped separately from other groups. Groups with a single fund are noticeably different from other funds. These funds are quite distinct from other funds in terms of return, volatility, and drawdown.

Table 4: Summary results of different groups for the period 2016 to 2017

Group Number	Number of Funds	Average Annualized return (%)	Average Annualized Standard Deviation (%)	Average Maximum Drawdown (%)	Average Correlation with SP500	Avarage Pairwise Correlation
1	58	6.27	9.19	-9.40	0.09	0.04
2	2	27.24	34.58	-35.65	-0.22	0.99
3	1	54.58	74.42	-30.52	-0.16	-
4	28	10.07	21.56	-18.53	0.49	0.53
5	1	44.80	55.04	-41.76	0.48	-
6	1	-2.06	125.30	-80.70	0.13	-
7	101	11.33	10.99	-8.98	0.63	0.47
8	1	91.63	51.20	-27.74	0.50	-
9	2	43.05	47.93	-31.05	0.41	1.00

One significant observation from comparing Table 4 with Table 3 is that the composition of different groups changes between the two periods. While the first analysis period produced 4 groups with more than 10 funds, the second period resulted in 3 groups with more than 10 funds. Such a change in groups may indicate a change in investment style or a change in investment manager and warrants deeper examination.

To further look at how funds move from one group to the other, we created a transition matrix of the funds between the two periods. This matrix shows how the composition of groups changed between the two periods. Figure 8 displays the transition matrix. Rows display funds in different groups using the 2012 to 2015 period, while columns display funds in different groups using the

2016 to 2017 period. Most of the funds in group 9 from the first period moved to group 1 in the next period. This shows funds in this group continued to follow a similar strategy in both periods. Group 8 from the first period also shows a similar pattern by having most of the funds moving to group 7 in the second period. However, funds in group 7 in the first period were divided into 4 groups in the second period.

Figure 9: Transition matrix of funds

		Groups formed using 2016 to 2017 returns									
		1	2	3	4	5	6	7	8	9	Total
Groups formed using 2012 to 2015 returns	1					1				2	3
	2	1		1	5			3	1		11
	3						1				1
	4				4						4
	5				5			1			6
	6	1									1
	7	2	2		12			13			29
	8	4			1			64			69
	9	50			1			20			71
	Total	58	2	1	28	1	1	101	1	2	195

We, next, looked at the characteristics of the funds that were in group 7 when data from 2012 to 2015 were used and were in four different groups when data from 2016 to 2017 were used. Table 5 shows the summary statistics of these funds for the 2016 to 2017 period. Clearly, the funds in the different group look very different, indicating the effectiveness of SOM in classifying funds into different groups when underlying characteristics become different.

Table 5: Summary results of different groups for the period 2016 to 2017

Group Number	Number of Funds	Average Annualized return (%)	Average Annualized Standard Deviation (%)	Average Maximum Drawdown (%)	Average Correlation with SP500	Avarage Pairwise Correlation
1	2	1.90	8.88	-7.02	-0.09	1.00
2	2	27.24	34.58	-35.65	-0.22	0.99
4	12	12.28	21.05	-15.83	0.50	0.63
7	13	14.37	13.35	-8.90	0.45	0.38

Conclusion:

We used the Self-Organizing Map on hedge fund returns to classify hedge funds into different groups. We started with all funds categorized as long/short equity with data from January 2012 to December 2017 and found that SOM can effectively group funds into homogenous groups. SOM is especially good at separating input data points that are very different from any other data points. SOM was also able to classify funds that were similar in a particular period but were different in another.

References:

Harvey, C., Rattray, S., Sinclair, A., and Van Hemert, O., Man vs. Machine: Comparing Discretionary and Systematic Hedge Fund Performance, *The Journal of Portfolio Management*, 43(4), 55-69.

Kohonen, T., Self-organized formation of topologically correct feature maps, *Biological Cybernetics*, 43, 59–69 (1982).