

A study of SOM clustering software implementations

A. B. Adeyemo
Computer Science Department
University of Ibadan
Nigeria
+2348052107367
sesanadeyemo@gmail.com

ABSTRACT

Clustering algorithms generally suffer from some well-known problems for which the Self Organizing Maps (SOM) algorithms are adept at handling. While there are many variants of the SOM algorithm, software programmes that implement the SOM algorithms have tended to give varying results even when tested on the same data sets. This can have serious implications when the goal of the clustering is novelty detection. In this study a comparison of the performance of some SOM clustering software was carried out and results presented.

CCS Concepts

• General and reference → -computing tools and techniques
→ Empirical studies

Keywords

Comparative Analysis; Clustering; Self Organizing Maps.

1. INTRODUCTION

In the clustering process data is grouped in such a way that the intra-cluster similarity is maximized while the inter-cluster similarity is minimized. Data can be described by either categorical or numeric features. Due to the differences in the characteristics of these two kinds of data, attempts to develop criteria functions for mixed data have not been very successful [15]. There are two widely used clustering methods: the hierarchical and the nonhierarchical (partitional) methods. The hierarchical clustering process can be categorized as divisive when a large data set is divided into several small groups and, agglomerative when a small data set are put together to create a larger cluster. Self-Organizing Maps (SOM) are competitive networks that provide a "topological" mapping from the input space to the clusters [4]. The SOM was inspired by the way in which various human sensory impressions are neurologically mapped into the brain such that spatial or other relations among stimuli correspond to spatial relations among the neurons.

In a SOM, the neurons (clusters) are organized into a grid which is usually two-dimensional, but sometimes one-

dimensional (or (rarely) three or more-dimensions). The reason for using one- and two dimensional grids is that space structures of higher dimensionality cause problems with data display and cannot be displayed on the monitor. The SOM working algorithm is a variant of multidimensional vectors clustering of which the Kmeans clustering algorithm is an example of this type of algorithm [9].

The SOM neural network uses a competitive learning algorithm and is a method for unsupervised learning, based on a grid of artificial neurons whose weights are adapted to match input vectors in a training set. The SOM algorithm is fed with feature vectors, which can be of any dimension. The algorithm for the training of the SOM [4] is explained easily in terms of a set of artificial neurons, each having its own physical location on the output map, which take part in a winner-take-all process where a node with its weight vector closest to the vector of inputs is declared the winner and its weights are adjusted making them closer to the input vector.

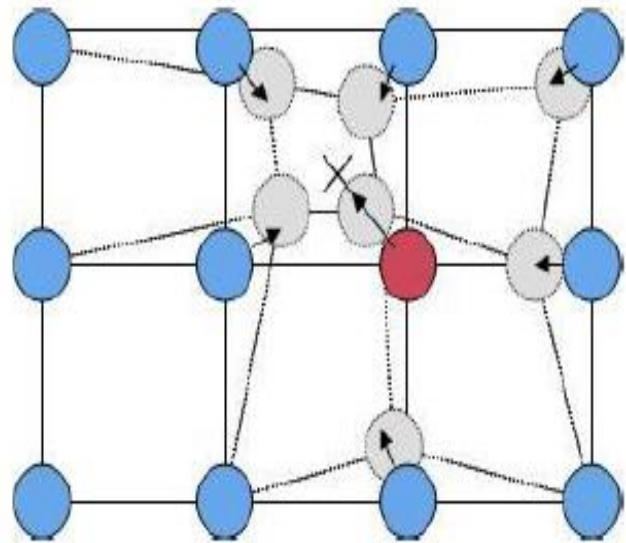


Figure 1: Illustration of the updating of the Best Matching Unit (BMU) of a SOM grid and its neighbors

In each training step, one sample vector 'x' from the input data set is chosen randomly and a similarity measure is calculated between it and all the weight vectors of the map. The Best-Matching Unit (BMU), denoted as 'c', is the unit whose weight vector has the greatest similarity with the input sample 'x' (figure 1). The similarity is usually defined by means of a distance measure, usually the Euclidian distance. The BMU is

defined mathematically as the processing element for which the expression:

$$d = \|x - v_c\| = \min_i (\|x - v_i\|) \dots\dots\dots 1$$

where d is the distance measure.

Each node has a set of neighbors. When a node wins a competition, the neighbor's weights are also changed but not as much as that of the winning node. The further the neighbor is from the winner, the smaller its weight change. The SOM update rule for the weight vector of the unit i is given mathematically as:

$$v_i(t+1) = v_i(t) + h_{c(x),i}(x(t) - v_i(t)) \dots\dots\dots 2$$

where

t represents the sample index for each presentation of a sample 'x'
 $h_{c(x),i}$ represents the neighborhood function around the winner unit 'c', with neighborhood radius r(t).

The neighborhood function is like a smoothing kernel that is time-variable. It is a decreasing function of the distance between the the ith and cth reference vectors on the map grid. The neighborhood function is usually expressed as the Gaussian function which can be expressed mathematically as:

$$h_{c(x),i} = \alpha(t) \exp\left(-\frac{\|r_i - r_c\|^2}{2\sigma^2(t)}\right) \dots\dots\dots 3$$

where

$\alpha(t)$ represents the learning rate factor and takes values $0 < \alpha(t) < 1$

$\sigma(t)$ represents the width of the neighborhood function which decreases monotonically with the regression steps.

A simpler definition of the neighbourhood function given by Kohonen [4] is:

$$h_{c(x),i} = \sigma(t) \dots\dots\dots 4$$

If $\|r_i - r_c\|$ is smaller than a given radius around node 'c' and the radius is also a monotonically decreasing function of the regression steps, but otherwise $h_{c(x),i} = 0$. $\sigma(t)$ is a diminishing function of time. At the beginning of the learning procedure it is fairly large, but it is made to gradually shrink during learning. Towards the end of learning a single winning processing element is trained. A linear diminishing function of time is usually used. The learning process consisting of winner selection by Equation (1) and adaptation of the synaptic weights by Equation (2). This process is repeated for each input vector, usually for a large number of cycles with different inputs producing different winners. The network therefore associates output nodes with groups or patterns in the input data set. The SOM algorithm is very simple and allows for many subtle adaptations.

There are some visual displays that are used to "determine" where the natural cluster boundaries are in the SOM. Some of the visual tools that can be used are Histograms [6], Component Plane displays [3], U-matrix, P-matrix and U* matrix displays [10], [11], [12], [13]. An important concept in interpreting these displays is the interaction of the two properties of the SOM. These are the

neighborhood relationship, and the density mapping. Neighboring neurons in the SOM cannot be too far away from each other (in order to maintain their similarity) but the SOM also wants to place more neurons in areas of high input density (for example, logical clusters). Because of this, there will be neurons that will be placed in areas between natural clusters which are typically low input density areas (so that the map can "stretch" between clusters).

The standard SOM algorithm uses numeric type variables and the Euclidean distance function. The arithmetic operations used during the learning phase for the update of the feature vectors cannot be used with categorical values. The SOM was not directly designed to work with categorical variables due to the limitation of learning laws. The method usually adopted is to translate categories to numeric numbers during data pre-processing before training using the transformed data using standard SOM algorithm [2]. The Kohonen SOM clustering algorithm has also been used for classification purposes with remarkable results. There is a fundamental difference between the clustering process and the classification process. Clustering is an unsupervised process while classification is supervised. Usually data clustering is used as a pre-processor for classification purposes [8].

A rich variety of versions of the basic SOM algorithm have been proposed. Some of the variants aim at improving the preservation of topology by using more flexible map structures instead of the fixed grid. Some of these methods however cannot be used for visualization as easily as the regular grid. Some variants aim at reducing the computational complexity of the SOM [3]. Experiments using different distance measures, map topologies, training parameters such as the learning rate and neighbourhood function can be carried out.

Using identical settings, training of a SOM map over different iterations can lead to different mappings, because of the random initialisation. Yet it has been shown that the conclusions drawn from the map remain remarkably consistent, which makes it a very useful tool in many different circumstances [14]. Some of the desirable features that good SOM clustering software should have include:

1. Being able to set the neighborhood kernel function and to set the start value for the neighborhood function (learning radius): The neighborhood function determines how strongly the processing elements are connected to each other. Neighborhoods of different sizes in different neuron configurations (e.g. rectangular and hexagonal lattices) can be used. The simplest neighborhood function is the bubble (winner-takes-all): it is constant (or 1) over the whole neighborhood of the winner unit and zero elsewhere. Usually the neighbourhood function is expressed as a Gaussian function and as expected using the winner-takes-all function retrieves less clusters than the Gaussian function.
2. Being able to set the activation function and weight initialization methods: Before the training, initial values are given to the prototype vectors of the SOM. The SOM is very robust with respect to the initialization process, however, when properly accomplished it allows the algorithm to converge faster to a good

solution. Initialization procedures that have been used are: Random initialization, where the weight vectors are initialized with small random values; Sample initialization, where the weight vectors are initialized with random samples drawn from the input data set; Linear initialization, where the weight vectors are initialized in an orderly fashion along the linear subspace spanned by the two principal eigenvectors of the input data set.

3. Being able to set the choice of cooling strategy during training: for example linear or exponential.
4. Being able to set the distance measure to be used, for example, Euclidean, Manhattan and Maximum value: It is noted that the distance measure between data points is an important component of a clustering algorithm. If the components of the data instance vectors are all in the same physical units then it is possible to use the simple Euclidean distance metric to successfully group similar data elements. The Euclidean distance in a two or three-dimensional space measures is the actual geometric distance between objects in the space. However, it has been observed that even the Euclidean distance can sometimes be misleading, because of the way the mathematical formula used to combine the distances between the single components of the data feature vectors into a unique distance measure that can be used for clustering purposes is computed. Different formulas lead to different clustering's. Therefore, domain knowledge must be used to guide the formulation of a suitable distance measure for each particular application.
5. Being able to set the scaling technique to be used: for example z-transform, (0,1) transform, (1,-1) transform or none, depending on the clustering goal and data set.
6. Being able to set the starting and stopping learning rate: The learning rate is a decreasing function of time between [0,1]. The learning rate can be expressed as a linear function and as a function inversely proportional to time. Using the inverse function ensures that all input samples have approximately equal influence on the training result. Some learning rate functions that have been implemented are the linear, inverse-of-time, and as a power ser.
7. Being able to set the training algorithm to be used: for example batch, on-line, hybrid etc. The batch algorithm has been shown to be faster [4] than the normal sequential algorithm (and the results are just as good or even better).
8. Good data visualization options: for example histograms, hinton charts, weight charts (maps), U-Matrix, P-Matrix etc. Good result analysis and presentation functions: computation of vital statistics for evaluating the quality of the clustering for example, mean, standard deviation (or variance), correlation coefficient, t-test etc.

This work presents a comparative study of the performance some SOM clustering software when tested on the same data set. Results were presented and reasons for the observed variations presented. The study also presents the desirable features that standard SOM software should have.

2. MATERIALS AND METHODS

Agro metrological data for FRIN headquarters, Ibadan, Nigeria was used. The data set had 254 records and the attributes in the data set were: Year (numeric), Month (text), Total Rainfall in millimeters (numeric), Minimum Temperature in Celsius (numeric), Maximum Temperature in Celsius (numeric), Relative Humidity and Fire Danger Index (numeric). The SOM software used were: NNClust, Pittnet Neural Network Educational Software and RapidMiner Studio.

The NNClust software was programmed to use only the Gaussian neighbourhood function and the Euclidean distance measure. The user can input the learning rate and starting neighbourhood size. The software automatically normalizes the input data between -1 and 1 and has features for generating data/result statistics and data visualization such as weight maps and radar charts. The Pittnet software also uses the Gaussian neighbourhood function and Euclidean distance metrics. The user also defines the starting learning rate and it also automatically normalizes the data between 0 and 1. It is a DOS based program that saves its result in a text file and has no data analysis or data visualization ability. RapidMiner studio (Community Edition) has facilities for selecting parameters for defining the learning rate, neighbourhood radius and can choose either to normalize the data or not. It also has an array of tools for statistical data analysis and data visualization.

Using the three software's clusters was generated. The arithmetic mean of each cluster group was also computed. The arithmetic mean is a measure of central tendency which describes the central location of data and is usually used with other statistical measures such as the standard deviation because it can be affected by extreme values in the data set and therefore be biased. The standard deviation describes the spread of the data and is a popular measure of dispersion. It measures the average distance between a single observation and its mean.

3. RESULTS AND DISCUSSION

The meteorological data was clustered using NNClust SOM clustering software with a starting learning rate of 0.9 and was trained over 100 epochs. The software accepts only numeric values. Non numeric values are treated as missing values which are replaced by the column mean. The software was set to identify a maximum of ten clusters, however only eight clusters were generated. The software uses the number of clusters specified to create the SOM grid. The mean and standard deviation of the eight clusters were computed. Increasing the training cycle did not improve the results. Table 1 presents the summary of the eight clusters, while figure 2 presents the chart of the cluster means.

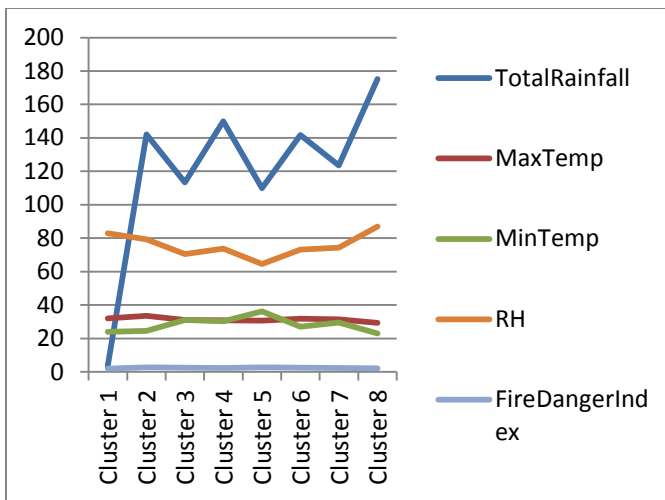


Figure 2: Chart of NNClust cluster means

The meteorological data was trained using the Pittnet software with a starting learning rate of 0.9 and was set to train for 100 epochs, although the software stops training as soon as the maximum number of clusters have been generated. The software requires the user to specify the number of clusters expected apriori. This number is used in conjunction with the number of input signals (attributes) to determine the SOM grid size. Expected number of clusters was set to ten. The software identified only four clusters. The mean and standard deviation of the clusters were computed. Table 2 presents the summary of the clusters, while figure 3 presents the chart of the cluster means.

TheRapidMiner Studio software was used to cluster the meteorological data set using a starting learning rate of 0.9 and was trained over a 100 epochs. The expected number of clusters was set at ten and the software generated ten clusters. Table 3 presents the summary of the cluster means with their standard deviations while figure 4 presents a chart of their cluster means.

3.1 Discussion of Results

The quality of the clusters identified in the data by the three software's can be inferred from a comparison of the mean and standard deviation of the clusters. If the value of the standard deviation is low, then the clustered records are within the same range. However if the value is high this suggests the presence of outliers in the clustered data records. For example table 4 presents the clustered records for cluster 2 (table1) for the NNClust software which is representative of the trend observed in the clusters identified by the software. Interpreting the cluster is indecisive when the values in the Total Rainfall fields are considered. The field has a mean of 142.05 and a standard deviation of 136.011711.

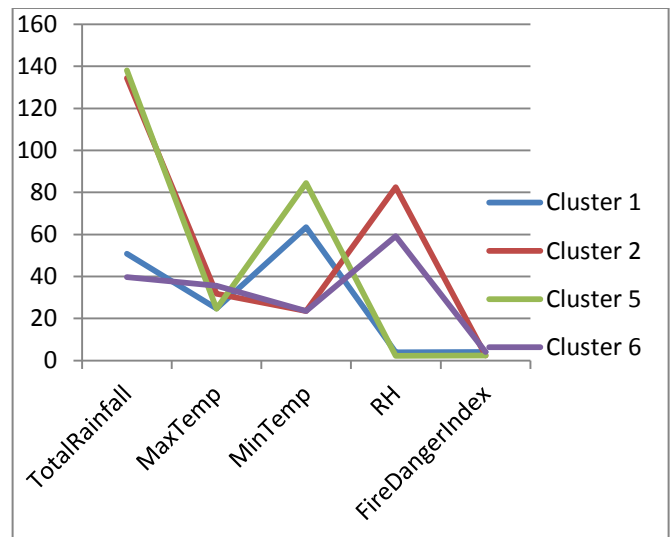


Figure 3: Chart of Pittnet software cluster means

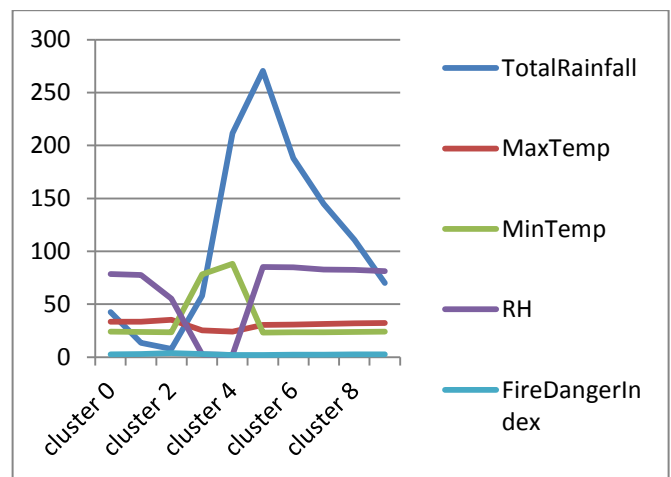


Figure 4: Chart of Rapid Miner Studio cluster means

Similarly considering the clusters identified by the Pittnet software in table 2 the same trend is observed. Table 5 presents the records for cluster 4 (table 2) for the Pittnet software cluster results. It can be observed that the cluster consists of data records which have the same value for the FireDangerIndex attribute. However, considering the Total Rainfall field which has a mean value of 39.74444 and a standard deviation of 43.34732. The high standard deviation value implies that there are outlier data values in the clustered records.

The clusters identified by the RapidMiner software presented in table 3 were easier to interpret. They followed the expected rainfall pattern which is known for the region where the data was collected [5]. Cluster 2 (table 3) contained records with only a high FireDangerIndex of 4 as presented in table 6, while cluster 5 (table 3) contains records with the highest recorded Rainfall level in the data set. The other clusters also contained data records which can be categorized by the Rainfall level pattern of the region.

4. ACKNOWLEDGMENTS

Some of the problems found in the literature about clustering algorithms are: Most clustering techniques are based on distance calculations which are very sensitive to ranges of variables, therefore the values have to be normalized. Normalization however is a subjective function, and these transformations cannot be carried out without creating biases; The presence of outliers in data sets create problems in data clustering based on distance calculations when they have not been identified and removed from the data set; Handling categorical variables (non-numeric data, non-numeric variables, categorical data, nominal data, or nominal variables) are a problem for most clustering algorithms, and even when data encoding methods are used they can introduce extra biases due to the number of values which the encoding introduces in the categorical variables; The selection of variables also has a large influence on clustering results, while the assigning of different weights for variables and categorical values can be used, when many variables and categorical values are involved, it can affect the clustering quality; Capturing patterns (or behaviors) hidden inside time-varying variables and modeling them is another problem and most clustering techniques do not possess this predictive modeling capability; Most clustering techniques were developed for laboratory generated simple data sets consisting of a few to several numerical variables; hence they can't be used for large data analyses that consist of many categorical complex data.

Most common implementation of data clustering algorithms suffer from these problems, however, SOM's are very robust and are adept at handling these problems but this depends also on the goal of the algorithm's implementation (programming). Applications programmed for demonstration purposes cannot be used for large scale projects and some implementations are not flexible and do not give users much options. However if the various implementations of the conventional SOM algorithm (which are usually focused on the goals of the programmer) provides enough options to the user, it is still a very robust algorithm that can be used for both numerical, categorical and mixed data sets. Further work in this study is focused on the development of an open flexible SOM clustering tool with adequate features that can be used for research purposes.

5. REFERENCES

[1]. Chang C., Ding Z., (2004), "Categorical data visualization and clustering using subjective factors", *Data & Knowledge Engineering*, Published by Elsevier B.V.
[2]. Chen N. and Marques N. C., (2005), "An Extension of Self-Organizing Maps to Categorical Data", *Proceedings of the 12th*

Portuguese conference on progress in Artificial Intelligence , pp 304 - 313, (Springer-Verlag Berlin, Heidelberg ©2005)
[3]. Kaski S., (1997), "Data exploration using self-organizing maps", *Acta Polytechnica Scandinavica, Mathematics, Computing and Management in Engineering Series No. 82*, Espoo 1997.
[4]. Kohonen T, (1999), "The Self-Organizing Map (SOM)", Helsinki University of Technology, Laboratory of Computer and Information Science, Neural Networks Research Centre, Quinquennial Report (1994-1998), (Downloaded from <http://www.cis.hut.fi/research/reports/quinquennial/> January 2006).
[5]. Nigeria Climate Review, 2010, Nigerian Meteorological Agency, www.nimetng.org
[6]. Pampalk E, Rauber A, Merkl D, (2002), "Using Smoothed Data Histograms for Cluster Visualization in Self Organizing Maps", Technical Report OeFAI-TR-2002-29, extended version published in *Proceedings of the International Conference on Artificial Neural Networks, Springer Lecture Notes in Computer Science, Madrid, Spain, 2002*.
[7]. Pelczer I. J. and Cisneros H. L., (2008), "Identification of rainfall patterns over the Valley of Mexico", 11th International Conference on Urban Drainage, Edinburgh, Scotland, UK, 2008
[8]. Principe J. C., Euliano N. R. Lefebvre W. C. (2000), *Neural and Adaptive Systems: Fundamentals Through Simulations*, John Wiley and Sons Inc, ISBN 0-471-35167-9, pp 656.
[9]. Statsoft Electronic Statistics Textbook, (2002), Copyright, 1984-2003, (<http://www.statsoftinc.com/txtbook/glosd.html#Data Mining>), Downloaded June 2002.
[10]. Ultsch, A., (1999), *Data Mining and Knowledge Discovery with Emergent Self-Organizing Feature Maps for Multivariate Time Series*, In Kohonen Maps, (1999), pp. 33-46.
[11]. Ultsch A, (2003a), Maps for the Visualization of high-dimensional Data Spaces, *Proc. Workshop on Self Organizing Maps*, pp 225 - 230, Kyushu, Japan, 2003.
[12]. Ultsch A, (2003b), U*-Matrix: a Tool to visualize Clusters in high dimensional Data, Technical Report No. 36, Computer Science Department, University of Marburg, Germany, 2003.
[13]. Ultsch A., Moerchen F, (2005), ESOM-Maps: tools for clustering, visualization, and classification with Emergent SOM, Technical Report No. 46, Dept. of Mathematics and Computer Science, University of Marburg, Germany, 2005.
[14]. Wehrens R. Buydens L. M. C., 2007, Self- and Super-organizing Maps in R: The kohonen Package, *Journal of Statistical Software*, published by the American Statistical Association, Vol. 21, Issue 5
[15]. Zengyou He, Xiaofe I Fe, Shengchun Deng, (2003), "Clustering Mixed Categorical and Numeric Data", Department of Computer Science and Engineering, Harbin Institute of Technology, Harbin 150001, P. R. China

Table 1: Summary of NNClust clusters

		TotalRainfall	MaxTemp	MinTemp	RH	FireDangerIndex
Cluster 1	Mean	3.7	32	24	83	2
	SD	0	0	0	0	0
Cluster 2	Mean	142.05	33.5	24.5	79.33333	2.666666667
	SD	2.61629509	22.627417	16.9706	4.501851	0.516397779
Cluster 3	Mean	113.313158	31.1236842	31.0605	70.54737	2.5
	SD	69.9895185	15.4557389	11.4404	45.62364	1.246560403
Cluster 4	Mean	149.99	30.8333333	30.2967	73.75333	2.333333333
	SD	98.1425436	3.53058883	20.0499	25.41582	0.546672274
Cluster 5	Mean	109.891667	30.6333333	36.1667	64.64444	2.638888889
	SD	92.1210985	4.02073199	24.3938	34.37646	0.723198364
Cluster 6	Mean	141.621277	31.7574468	27.0617	73.1617	2.617021277
	SD	97.0359995	2.63056819	13.7078	20.8623	0.644481304
Cluster 7	Mean	123.545794	31.4411215	29.4963	74.41028	2.411214953
	SD	81.8137003	2.96536463	18.4077	24.4239	0.531165877
Cluster 8	Mean	175.268966	29.3793103	23.069	86.89655	2.068965517
	SD	85.4901878	1.49794605	1.06674	4.312315	0.257880715

Table 2: Summary of the Pitnett software clusters

		TotalRainfall	MaxTemp	MinTemp	RH	FireDangerIndex
Cluster 1	Mean	50.850001	24.75	63.5	3.9	4
	SD	31.32483	0.070709	12.0208153	0.141421356	0
Cluster 2	Mean	134.3332	31.7082	23.5984375	82.4218728	2.3828125
	SD	91.137324	2.254123	1.06439596	6.908488013	0.487025284
Cluster 3	Mean	138.05185	24.64815	84.4074074	2.196296296	2.407407407
	SD	45.668999	15.90804	27.2370968	39.48311832	1.836329785
Cluster 4	Mean	39.744444	35.55556	23.5555556	59.22222133	4
	SD	43.347321	1.333333	1.74005108	7.120003363	0

Table 3: Summary of Rapid miner Studio clusters

		TotalRainfall	MaxTemp	MinTemp	RH	FireDangerIndex
cluster 0	Mean	42.35385	33.41154	23.99615	78.46153846	2.730769231
	SD	8.192056	2.308823	0.911913	7.798619207	0.603833905
cluster 1	Mean	13.50513	33.47179	23.80769	77.43589744	2.820512821
	SD	9.379343	2.342845	1.280909	6.302860135	0.451418517
cluster 2	Mean	7.64	35.36	23.42	55.2	3.8
	SD	16.15873	17.96476	13.16786	40.93966268	1.299899072
cluster 3	Mean	57.94667	25.35333	78.13333	2.726666667	2.933333333
	SD	13.23034	15.63488	11.11308	32.15964741	1.361648053

cluster 4	Mean	211.4214	23.90714	88.14286	1.871428571	2.071428571
	SD	46.93198	1.320527	4.24005	0.299816794	0.267261242
cluster 5	Mean	270.4346	30.36154	23.21923	85.19230769	2.115384615
	SD	42.68863	1.395814	0.859101	5.129837598	0.322602539
cluster 6	Mean	188.0463	30.77805	23.31463	84.90243902	2.146341463
	SD	15.90989	1.518801	0.887288	5.180757078	0.357839043
cluster 7	Mean	144.6971	31.42	23.47429	82.85714286	2.342857143
	SD	10.84353	1.991127	0.995089	7.6855206	0.481593992
cluster 8	Mean	110.85	31.84474	23.72105	82.31578947	2.473684211
	SD	9.73158	2.332462	1.076822	6.794692934	0.603451429
cluster 9	Mean	70.05862	32.27241	24.04828	81.31034483	2.482758621
	SD	8.635041	2.37936	1.180684	9.043953972	0.508547628

Table 4: Sample NNClust software cluster result

Year	Months	TotalRainfall	MaxTemp	MinTemp	RH	FireDangerIndex
1980	Feb.	60	35	27	75	3
1987	Aug.	357.1	30	23	86	2
1987	Nov.	10	35	24	80	3
1989	Mar.	57	35	25	77	3
1991	Apr.	108.9	32	24	83	2
1998	Sept.	259.3	34	24	75	3
Mean		142.05	33.5	24.5	79.33333	2.666667
SD		136.0117	2.073644	1.378405	4.501851	0.516398

Table 5: Sample Pittnet software cluster result

Year	Months	TotalRainfall	MaxTemp	MinTemp	RH	FireDangerIndex
1989	Feb.	18.4	35	22	51	4
1990	Feb.	40.3	35	23	64	4
1990	Mar.	11.7	37	25	69	4
1994	Jan.	1.3	33	20	45	4
1997	Mar.	122.2	35	23	62	4
1998	Feb.	2	36	25	60	4
2000	Mar.	48.8	37	25	62	4
2001	Mar.	15	37	25	60	4
2001	Apr.	98	35	24	60	4
Mean		39.74444	35.55556	23.55556	59.22222	4
SD		43.34732	1.333333	1.740051	7.120003	0

Table 6: Sample Rapidminer software cluster result

Year	Months	TotalRainfall	MaxTemp	MinTemp	RH	FireDangerIndex
1989	Feb.	18.4	35	22	51	4
1994	Jan.	1.3	33	20	45	4
1998	Feb.	2	36	25	60	4
2001	Mar.	15	37	25	60	4
2004	Mar.	1.5	35.8	25.1	60	3
Mean		7.64	35.36	23.42	55.2	3.8
SD		8.361399	1.499333	2.319914	6.906519	0.447214

Table 7: Sample RapidMiner software cluster result

Year	Months	TotalRainfall	MaxTemp	MinTemp	RH	FireDangerIndex
1979	Jul.	291.2	29	23	85	2
1979	Sept.	269	29	23	86	2
1979	Oct.	223.6	31	24	86	2
1979	Nov.	261.4	32	24	83	2
1980	Jun	306	31	23	82	2
1980	Aug.	427.4	28	23	88	2
1980	Sept.	333.5	29	23	90	2
1981	Sept.	233.9	30	23	86	2
1981	Oct.	225.1	31	24	83	2
1983	May	250.7	31	24	85	2
1984	May	223	32	23	86	2
1984	Jun	233.6	30	22	82	2
1985	Jul.	307.2	30	23	86	2
1985	Aug.	232.2	30	23	89	2
1986	Jun	312.9	31	23	83	2
1986	Sept.	374.1	29	22	84	2
1987	Jul.	246.8	30	23	85	2
1987	Aug.	357.1	30	23	86	2
1987	Sept.	252.5	31	23	87	2
1988	Jun	242.9	30	22	82	2
1988	Jul.	240.9	29	23	84	2
1988	Sept.	225.1	30	23	87	2
1989	May	259.2	32	23	83	2
1989	Jun	338.7	31	23	86	2
1989	Aug.	275	29	22	88	2
1990	Apr.	233.8	33	24	82	3
1990	Jul.	293.6	29	23	90	2

1990	Oct.	255.4	31	23	85	2
1991	May	258.2	32	24	84	2
1991	Jul.	306.6	29	23	90	2
1992	Sept.	275.4	29	23	90	2
1992	Oct.	276.3	31	23	88	2
1993	Jul.	261	29	27	87	2
1993	Aug.	237.7	29	23	90	2
1993	Sept.	255.5	30	23	86	2
1994	Sept.	236	30	23	89	2
1995	May	334.3	31	24	81	2
1995	Aug.	304.2	29	23	91	2
1996	Aug.	224.7	30	23	89	2
1996	Sept.	304.1	29	22	90	2
1997	Apr.	261.7	32	24	70	3
1998	May	245.4	34	25	70	3
1998	Sept.	259.3	34	24	75	3
2000	Jul.	220.4	29	23	73	3
2000	Aug.	263.8	29	23	85	2
2001	May	265	33	24	74	3
2001	Sept.	275.2	29	22	90	2
2002	Oct.	265	29	24	87	2
2003	Jun	275.3	30.6	24.5	92	2
2003	Sept.	226	30.8	22.4	92	2
2003	Oct.	254.9	32	23.2	92	2
2006	Sept.	250.8	30.4	22.3	86	2
Mean		270.4346	30.36154	23.21923	85.19231	2.115385
SD		42.68863	1.395814	0.859101	5.129838	0.322603