HIERARICAL CLUSTERING AND DATA CORRELATION FOR CHEMICAL COMPOUNDS

Jennifer Garner, Supervisor: Dr. Daniel Ashlock
MATH*4600 PROJECT, WINTER 2016 University of Guelph

Table of Contents

Abstract 1
Motivation for Project and Background 2
Hierarchical Clustering Algorithms 2
The Omni-Correlator 7
Data Used 9
Results for the two Hierarchical Clustering Methods 11
Experiment 1 11
Experiment 2 12
Experiment 3 12
Experiment 4 13
Results for the Omni-Correlator 14
Experiment 5 14
Experiment 6 15
Conclusions and Next Steps 16
References 17

Abstract

The first half of this paper describes the use of two hierarchical clustering algorithms - one uses a neighbour-joining, join-the-dots algorithm and the other uses the new “Bubble Clustering” algorithm – that can be used to generate dendograms or trees. These trees are a visual aid to determine the relatedness of chemical compounds with certain features that can be collected either from chemical models or experimental methods. The main benefit of the Bubble Associator is increased stability when changing the number of features or compounds being compared. The second half of this paper describes the use of the Omni-Correlator code, which uses an evolutionary algorithm and function stacks to generate a correlation matrix. This matrix can suggest how one feature may be predicted using other features, and the various relationships between the features for a chemical dataset. It was tested using a chemical dataset where the

features were tied together, and for a dataset that was mostly experimental; in both cases the correlation matrix produced was meaningful and made sense from a chemical standpoint.

Motivation for Project and Background

The development of chemical models is often performed using an equation of state (EOS) and builds upon concepts such as the ideal gas law (PV=nRT, where P=pressure, V=volume, n=moles, R=ideal gas constant, and T=temperature). Data from a representative, easily-accessible compound of well-characterized purity and behaviour can be used to determine qualities of similar compounds. For example, experimental PVT data at the critical point can be unreliable, and so a model was proposed in [1] using liquid oxygen data to determine the critical temperature and critical density data for the liquid and vapour phases of hydrogen, fluorine, and neon. Compressibility, density, and viscosity for natural gases (sweet, sour, and gas condensates) was determined using an empirical model in [2]. This work is an example of using a data set to generate a model that does not require the use of an EOS.

Since the estimation of most chemical data requires a proper reference chemical compound, the first step in generating chemical models should be to determine which compounds are related and for which properties. For example, in [3] the author was able to show a correlation between the noble gases Ne, Ar, Kr, and Xe using the critical constants for PT and Vm (molar volume or V/n).

Hierarchical Clustering Algorithms

One method of determining important relationships is through a hierarchical clustering. Hierarchical clustering methods have been used previously by various

groups to visually determine the relationships between datasets and to group components of a dataset, such as transactional elements [4], fatty acids in unknown microorganisms [5], normal and cirrhotic liver tissue [6], macromolecular crystal structures [7], samples of *Curcuma longa* L. oil for quality assessment [8], and principal components [9]. There has not been any hierarchical clustering directly of chemical compounds for the sake of categorizing them.

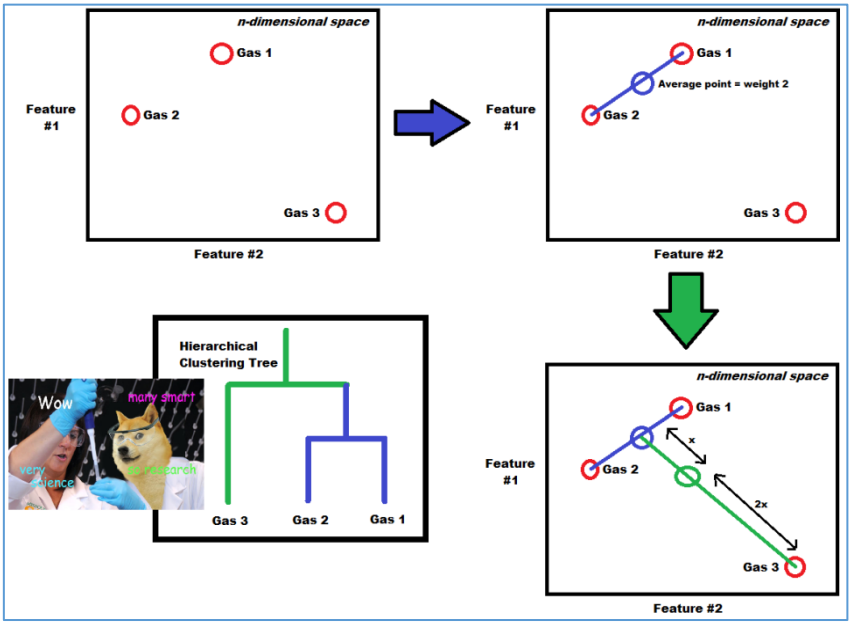
A neighbour-joining algorithm is a specific example of hierarchical clustering. It is best suited for data that has a common ancestry – for example an evolutionary relationship among biological species [10]. The first experiments that were run for this project used the neighbour-joining algorithm to compare chemical compounds. How to generate dendograms (or trees) using this algorithm is henceforth explained. This method is referred to as the join-the-dots (JTD) method.

First, with a dataset containing n features (modes for comparison) and x compounds, normalize the dataset such that all features are within the range [0,1] using the following formula:

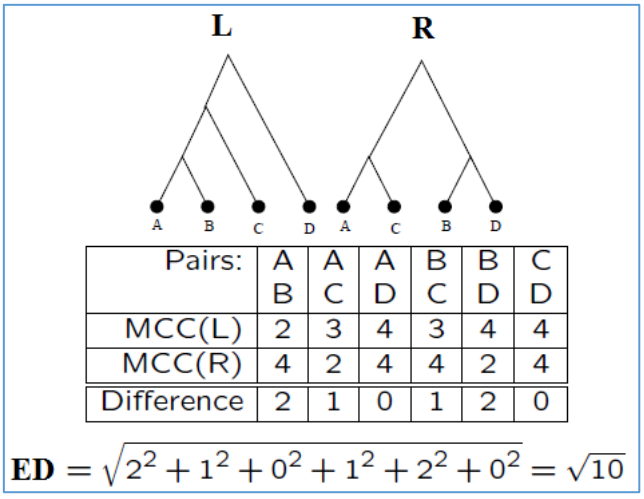
$Normalized\ Value = \frac{value\ of\ feature\ for\ compound\ x - minimum\ value\ for\ feature}{Maximum\ Value\ for\ feature - minimum\ value\ for\ feature}$. Then, generate an n-dimensional vector space containing the feature vectors. A feature vector is an n-dimensional vector of numerical features that represents an object (in this case a chemical compound). Using the Euclidean distance metric, connect the two closest points in n-space. The Euclidean distance (ED) between two points $p = (p_1, p_2, \dots, p_n)$ and $q = (q_1, q_2, \dots, q_n)$ is defined by the Pythagorean formula:

$$ED = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

The next step is to join this newly generated point, now with a weight of two, to the next closest point. This second point is at a weighted average distance between points 1+2 and 3. The following graphic explains these steps.



Note that the vertical distance between points is an indication of how far away the points are in space relative to one another. The method uses an agglomerative or bottom-up approach; the points begin as their own cluster, and then they are clustered with their neighbours until all points are connected. The dendogram, or tree, is built starting with the last two connected points, and branches off until all individual points are obtained. A benefit of this type of algorithm is that the data is easy to store in a computer through the use of parentheses, which note where the branches split off.



Pairs:	A B	A C	A D	B C	B D	C D
MCC(L)	2	3	4	3	4	4
MCC(R)	4	2	4	4	2	4
Difference	2	1	0	1	2	0

$$\mathbf{ED} = \sqrt{2^2 + 1^2 + 0^2 + 1^2 + 2^2 + 0^2} = \sqrt{10}$$

Previous work by the Ashlock group has shown that neighbour-joining algorithms producing trees with randomly-generated datasets can be unstable when features are removed [11]. For a tree, the minimal containing clade (MCC) vector (whose dimension is the number of possible point pairs in the set) is

represented by each point pair having a value that corresponds to the number of leaves or taxa that are needed to connect that point pair. An example of two trees and their MCC and the ED_{MCC} is shown.

To calculate the instability, two sets of trees were used: (1) trees generated using all of the data where each taxa was "snipped" out individually ($T_{snipped}$) and (2) trees generated using the data without each taxa ($T_{rebuilt}$). The trees from [11] were compared using the following instability measure, where K is the set containing the taxa and x are the members of that set:

$$Instability = \frac{1}{|K|}\sum_{x \in K} ED_{MCC}(T_{snipped}, T_{rebuilt})$$

Thus, when lots of re-arrangement of a tree occurs after the removal of a datapoint (or taxa), the calculated instability will be high.

Since chemical data does not necessarily belong to a common ancestor, and since not all chemical data is obtainable for all compounds, it was necessary to devise a new hierarchical clustering method that would not cause trees to fall apart upon the application or removal of new data or new compounds.

An associator is defined as any measurement that can indicate similarity between objects, and k-means clustering is an example of how to generate an associator [12]. When data points are associated strongly (i.e. found within a clustering), the associator can award a quality measure to the data points to improve the given association between those data points. The "Bubble Associator" or Bubble Clustering algorithm (BCA) is the proposed method for improving tree stability between chemical datasets. Using a normalized dataset as input, an associator matrix is generated, which can be used to make a tree. The following algorithm explains the Bubble Associator code:

Find the minimum ED, m, and the maximum ED, M, between the data points.

Repeat:

Randomly generate radius R in the range [m,M]

A point in the space, p, is chosen uniformly at random

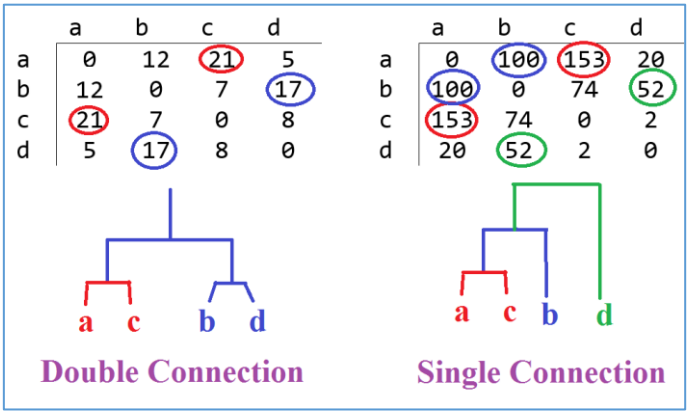
The number of points, N, within R of p is computed

For each point q within R of p:

A reward of 1/N is given to each point-pair in R of p

Until (Sample Number)

The sample number can be changed by the user, and gives a better tree for higher iterations; however, there will be a point where increasing the sample number does not make any significant changes to the association matrix used to generate the tree. For the datasets that were used in this paper, more than 1M samples did not change the tree, and above 10M samples, my computer crashed. Note that the more points that are found within an n-dimensional hypersphere or "bubble" (within R of p), the lower the reward, meaning that the best association reward given is when two data points are found to be clustered together.



The way in which the Bubble Associator generates the tree is different from the JTD method because it begins by connecting the two most associated points. From there, the next point(s) is/are connected that most strongly correlate(s) with this initial pairing. A benefit of this method is thus the immediate ability to point out the most correlated pair of points in the n-dimensional space. A graphic showing how to generate a Bubble tree from the associator matrix is shown.

The Omni-Correlator

The next part of the data analysis was done using the Omni-Correlator (OC), which provides a means to do multi-feature analysis on a dataset. The motivation is not just to predict one feature from others, but to see possible relationships between these features. For instance, now that chemical compounds can be clustered according to some hierarchy (as indicated by the BCA), one might want to take a section of that hierarchy and use one of its members to generate a model of predicting features for other members of that cluster.

To use the OC, we first need a dataset in the form of a .dat file that specifies the number of features and compounds to be compared. This dataset is then normalized between [0,1] as before, and the code outputs a file called "normalizingFactors.dat" that provides the maximum and minimum values for each feature. This file can be used to check that the input data file was correctly formatted, and is useful for spotting strange outliers that may be present in the dataset.

Once the dataset has been inputted, the OC performs multiple evolutionary runs that try to predict each feature from the values for the other features using function stacks. A function stack is an array of nodes, and the default number of nodes is 12, where each node contains a ephemeral constant in the range $-E \leq c \leq E$ (E=5 by default). For example, feature 1 can be predicted from the other features by: $f_1 = Q(f_2, f_3, \ldots, f_n)$.

Note that a feature cannot be used to predict itself. The types of function stacks that the OC can generate are: negation, scaling by a constant, square-root, sine, cosine, and arctangent (for single arguments), and addition,

subtraction, protected division, max(arguments), min(arguments), and weighted average (for 2 arguments).

The OC uses tournament selection to replace the two worst members with the two best members within the tournament, where high values (~20) favour early convergence to a model and low numbers (~4) favour exploration. The OC also uses two-point crossover and a mutation operator, where the maximum number of mutations (MNM) is the limit to how many mutations can be applied to the model (range 1-MNM). In this paper, all OC runs were performed with tournament size 7, MNM equal to 3, initial population size 10, maximum population size 10,000, and a minimum of 30 runs for each feature. A small study was done to determine the most important variable to improve data quality and the number of mating events was found to be highly significant above initial population size, MNM, and tournament size.

The fitness of the evolutionary algorithm is the mean squared error over x data points (how different are the predicted values from the actual values?) multiplied by a small logarithmic penalty determined by the number of variables used, n. This penalty encourages the OC to generate relationships that use a lower number of variables.

$$Fitness = \sqrt{\sum_{x} \frac{(predicted\ value - actual\ value)^2}{x}} * (Ln(n) + 1)$$

	a	b	c	d
a	0	0	10	80
b	0	0	3	8
c	571	25	0	9
d	82	7	52	0

At the end of each run, the measure "importance/error" is used to reward each feature that was useful in predicting the current target feature. These values can be used to generate a correlation matrix. This matrix has zeros down the diagonal indicating that a feature cannot predict itself, and is read starting from the left-most column.

Three possible relationships for model building can be found using the correlation matrix:

1. Features are unrelated. Example above: a cannot predict b and b cannot predict a.
2. One feature is asymmetrically useful in predicting another, which suggests that other features are involved in the prediction. Example above: a can predict c well (571), but c cannot predict a well (10).
3. Two features are symmetrically useful in predicting one another. Example above: a and d are useful to predict one another.

One feature of the OC is its generation of the time-of-last-innovation file (TLI.dat), which gives the mating event for each feature for each run at which the last significant improvement took place in terms of lowering the error. To determine what error improvement is significant, the Error_values.dat file can be screened; for example the significant error improvement measure was set to 0.01 for the two datasets compared in this paper as errors were generally in the range 0.01-0.9. The maximum allowed error for this research was set to 100. For datasets with limited correlation, high error values are produced coupled with low mating event values for the significant TLI.

Data Used

Data was gathered mainly from SigmaAldrich SDS catalogues [13], the Air Liquide Gas Encyclopedia [14], the PubChem Database [15], the Lenntech Database [16], [17], [18] and [19].

The **compression factor**, Z, is the molar volume of a gas (Vm) divided by the molar volume of a perfect gas (Vm^{o}) (from PV=nRT). We can re-write the compression

factor expression to give: $PV_m = RTZ$. Since gases approach ideality at high molar volumes and high temperatures, Z is considered as the first term in a series:

$$PV_m = RT\left(1 + \frac{B}{V_m} + \frac{C}{V_m^2} + \cdots\right)$$

This expansion is known as the **virial EOS**. B and C are **virial coefficients**, and can be calculated from experimental data regarding a gas' compression factor. In the datasets that were used, a, b, and c are constants that can be used to predict the first virial coefficient B, which is usually the most important of the virial coefficients [20] [17]. B can also be related to the **critical compression factor Z_c**, which can be determined using the critical values (P_c, T_c, and Vm_c) for a compound and the ideal gas constant, R [17].

The critical constants can be derived from a phase diagram for a given compound (plot of temperature versus pressure). Below the critical temperature, a gas can be condensed to form a liquid, as separated by a defined surface. At or above the critical temperature, a compressed gas will form a dense supercritical fluid.

Toxicity measures were incorporated for the later datasets to try and improve the yes (value=1) and no (value=0) measure. The easiest measure of toxicity to find that covered the most compounds was the **Immediately Dangerous to Life or Health Concentrations (IDLH)**, given in parts-per-million (ppm). The values used were only dependent on toxic effects and not the lower-explosive limit (LEL). The definition for IDLH is: "An atmospheric concentration of any toxic, corrosive or asphyxiant substance that poses an immediate threat to life or would cause irreversible or delayed adverse health effects or would interfere with an individual's ability to escape from a dangerous atmosphere" [18].

Other data includes: molar mass (g/mol of substance), the index of hydrogen deficiency (IHD), melting point, boiling point at 1atm of pressure (**normal boiling point**), specific gravity, heat of combustion, enthalpy of vaporization, flash point, and auto-ignition temperature. The **IHD** is the number of double bonds or rings present in an organic compound. The **specific gravity** (SG) is the density of a liquid divided by the density of water at 4°C.

Heat of combustion is the energy released per amount (mass or moles) of a substance when it has undergone complete combustion with oxygen. **Enthalpy of vaporization** is the amount of energy in joules required to transform a given amount of substance into a gas. The **flashpoint** is the temperature at which the gas at the surface of a liquid will ignite in air when exposed to a source of ignition. The **auto-ignition temperature** is the temperature at which the compound will spontaneously ignite without source of ignition. For non-flammable materials, the value given in the dataset was 20x the highest flash point and/or auto-ignition temperature from within the dataset.

Results for the two Hierarchical Clustering Methods

All trees generated from Experiments 1-4 can be found in the supporting information file folder that I have attached with my report.

Experiment 1

This experiment compared P_c, Vm_c, T_c, a, b, c, Z_c, molar mass, and IDH for a total of 76 chemical compounds, which included various gases, solvents/organics, and a few solids. The purpose of this experiment was to compare as many compound and properties as possible to see how the BCA versus the JTD method could predict data clusters.

The BCA does not indicate how far apart correlations between branches of the tree are as does the JTD method; to add a horizontal distance metric to the tree would be a next step for improving its readability and usefulness. Bubble clustering shows fewer small branches and lots of large branches, suggesting lower levels of clustering than for the JTD tree. Bubble clustering improves the data analysis by only considering significant relationships and these pairwise relationships are a lot easier to see directly from the tree. A problem with the JTD method is that it could be interpreted to show relationships that do not really exist. We can also derive the two most closely related points from the Bubble clustering tree, but we cannot do this as easily with the JTD tree because it generates the tree starting from the last two points that were connected in n-space.

Experiment 2

The purpose of this experiment was to see how the BCA and the JTD method compared for small datasets, and so only the normal boiling point, the molar mass, and the density (at 0˚C and 1atm) were compared for 11 gases. The trees were essentially identical except the BCA did not pair-up Xenon with Krypton.

Experiment 3

This experiment was performed to determine the relative stabilities of the two hierarchical clustering methods. In Experiment 3a, the P_c, Vm_c, T_c, a, b, c, and M were used to compare 24 compounds, mainly consisting of elements and inorganic compounds. For Experiment 3b, these same values were used except data was added relating to colour, odour, and toxicity (using the simple measure of yes=1, no=0), and physical state at SATP (0,0.5,1 for gas, liquid, and solid respectively).

The results from this experiment were significant in proving the increased stability for the trees generated using the BCA. Comparing BCA tree 3a with 3b indicates that the oxygen-argon relationship is still present after data addition, whereas the JTD method removes that association and puts the two compounds in completely different sections of the tree. The two other important relationships that were maintained between the two BCA trees were the ((He,D_2)Ne) tri-cluster and the HCl-NH_3 pair; in the JTD trees these two clusters were destroyed.

Experiment 4

These trees were generated using the data collected during the latter half of the semester, given in the table below.

Feature Number	Feature Name	Minimum Value	Maximum Value	Units
0	Latent Heat of Vaporization	1.67	13.7	x10^5 J/kg
1	Heat of Combustion	-460.13	0	x10^5 J/kg
2	Specific Gravity of liquid	0.59	3.12	Unitless
3	Auto-ignition Temperature	180	15400	°C
4	Flash Point	-191	1580	°C
5	Critical Temperature	132.9	748.5	K
6	Critical Pressure	2.49	11.35	MPa
7	Molar Mass	17.03	159.81	g/mol
8	Normal Boiling Point	-191.5	218	°C
9	Melting Point	-220	79.5	°C
10	IDLH	2	100,000	ppm

The purpose of this experiment was to show that, by having an extreme outlying value, data can be clustered by the tree-generator to observe two specific groups. In this case, the non-flammable compounds are much further away in space from the flammable compounds due to the high values for flashpoint and auto-ignition temperature that were used. This is an example of using domain knowledge to generate targeted clustering.

One section of both trees includes all the non-flammable gases except for bromine, which is unusual because it is a liquid at SATP, and has a high specific gravity. Another outlier, propane (probably due to its high heat of combustion), can be easily seen on the BCA tree. However, propane and bromine are more distinctly separate in the BCA tree than the JTD tree. Another difference between the trees is that the BCA does not directly associate chloroform and trichlorofluoromethane as does the join-the-dots method. Since these two compounds differ only by H swapped with F, their properties should be reviewed individually to determine whether they are actually closely related or not.

Results for the Omni-Correlator

All relevant TLI plots generated from Experiments 5 and 6 can be found in the supporting information file folder that I have attached with my report.

Experiment 5

The correlation matrix for this experiment was generated using the properties from the table below for 76 compounds as found in the trees for Experiment 1:

Feature Number	Feature Name	Minimum Value	Maximum Value	Units
0	Critical Pressure	0.23	22.06	MPa
1	Critical Molar Volume	41.7	250	cm^3/mol
2	Critical Temperature	5.19	647.1	K
3	Critical Compression Factor	0.228	0.312	Unitless
4	Constant a	15.9	540.5	na
5	Constant b	3.37	380.9	na
6	Constant c	3.245	1928.2	na
7	Molar Mass	2.02	352.02	g/mol

The following is the correlation matrix generated from Experiment 5. This dataset is “rigged” because it contains correlations and functions that are already known between the variables. The error values that were found (0.01 magnitude) were

therefore lower than in Experiment 6 (0.1 magnitude), where experimental values were used. The evolutionary algorithm used to generate the matrix was set to all of the default parameters with the number of mating events (mevs) set to 10M. From the TLI file, the errors tend to stop improving with 0.01 significance measure after about 5M mevs.

The OC does a good job in predicting an asymmetrical relationship between Z_c and (P_c, Vm_c), where the critical pressure and critical molar volume can be used to predict the critical compression factor, but not the other way around. This correlation makes a lot of sense based on the equation shown in the correlation matrix for Z_c. T_c was not as useful in predicting Z_c compared to P_c and Vm_c, probably because it is redundant to have all three critical constants to predict Z_c. The OC also shows that there are fairly good symmetric correlations between the PVT critical constants, which would probably be the first area to explore with making an actual model from the matrix.

	P_c	V_{cm}	T_c	$\frac{P_c V_{cm}}{RT_c}$	a	b	c	M
P_c	0.000	3780.968	6259.302	2176.851	3724.068	189.619	6282.531	195.530
V_{cm}	13375.280	0.000	2093.194	8444.437	7903.859	3347.673	5574.444	2103.491
T_c	9227.504	6594.496	0.000	590.960	1163.940	1930.306	4814.955	265.311
Z_c	833.400	240.273	345.950	0.000	90.004	1145.928	477.388	0.000
a	457.209	9123.915	735.517	718.717	0.000	7359.809	292.770	361.578
b	70.587	2240.844	1120.965	8885.299	11235.223	0.000	777.592	1266.004
c	9545.028	4519.595	4159.125	170.382	389.531	4792.814	0.000	369.020
M	171.710	2253.933	1093.796	42.078	789.264	576.281	100.400	0.000

Experiment 6

The dataset from Experiment 4 was used for this OC experiment. The TLI data file suggested that no more significant improvements (measure 0.01) could be made by increasing the mevs beyond 100,000 for at least two features (4,11). Increasing

the mevs to 5M and looking at the TLI file indicates that most significant decreases in error happen at or before 500,000 mevs. However, the matrices generated from the 10M and 100,000 mevs retain a lot of similar zeros and so the matrix from 100,000 mevs is shown below.

	1 Latent Heat of Vapor.	2 Heat of Comb.	3 Specific Gravity	4 Auto-ignition Temp	5 Flash Point	6 T_c	7 P_c	8 M	9 B.P.	10 M.P.	11 IDLH
1	0.000	5.889	85.516	79.019	25.388	42.279	126.336	81.105	46.960	0.000	0.000
2	0.000	0.000	23.060	6.483	229.962	33.603	208.916	26.131	12.419	0.000	0.000
3	33.373	67.732	0.000	26.219	453.919	0.000	436.116	500.460	18.067	15.923	0.000
4	27.380	0.000	394.594	0.000	651.865	0.000	646.656	399.716	194.526	635.457	218.524
5	960.109	2205.480	2012.338	3096.319	0.000	576.774	218.890	1610.708	278.981	131.323	57.173
6	51.863	0.000	234.537	0.000	0.000	0.000	711.440	637.201	1376.786	321.285	0.000
7	271.607	93.179	281.831	0.000	0.000	0.000	0.000	29.684	35.283	0.000	0.000
8	655.422	0.000	165.645	0.000	30.110	202.680	50.736	0.000	68.559	351.768	9.202
9	232.165	31.155	0.000	31.623	0.000	1638.184	1073.594	0.000	0.000	0.000	0.000
10	0.000	0.000	0.000	0.000	0.000	313.929	0.000	0.000	0.000	0.000	0.000
11	0.000	61.012	104.715	12.642	0.000	0.000	0.000	0.000	0.000	27.605	0.000

The OC matrix above indicates an asymmetric relationship between 5 and (2,3,4), where the flash point can be predicted with high correlation using the heat of combustion, specific gravity, and auto-ignition temperature. Investigation of this relationship for modelling data would also be of interest. This matrix also predicts a symmetrical relationship between the critical temperature and the boiling point, which is a sensible correlation.

Conclusions and Next Steps

This paper has compared two different hierarchical clustering algorithms, the Bubble Clustering Algorithm (BCA) and the Join-The-Dots (JTD) Algorithm, on various chemical data sets of varying sizes and features. The BCA is an improvement from the JTD method because it has higher stability when chemical data is added or removed. The datasets used to generate the trees can be tailored

by giving one or two values an obvious space for clustering, and the example given was with flammable vs non-flammable materials. Improvements to be made for the BCA include generation of a horizontal distance measure, similar to the JTD trees, that shows how close or far apart in value the data is from the associator matrix for neighbouring branches.

This paper has also explained how to use the Omni-correlator code to generate correlation matrices of significant importance and it has been used to satisfactorily show relationships within trivial and non-trivial (experimental) data sets for chemical compounds. The next steps would be to take these significant relationships and to try and build a model with the dataset. Improvements for the OC would be to make the program more user friendly by allowing the code to accept different formats of input file. Also, it may be helpful to generate multiple TLI files for one run of the OC.

References

[1] R. D. Goodwin, "Estimation of Critical Constants Tc, pc from the p(T) and T(p) Relations at Coexistence," *Journal of Research of the National Bureau of Standards - A. Physics and Chemistry,* vol. 74A2, no. 2, pp. 221-227, 1970.

[2] A. M. Elsharkawy, "Efficient methods for calculations of compressibility, density and viscosity of natural gases," *Fluid Phase Equilibria,* vol. 218, pp. 1-13, 2004.

[3] P. Molyneux, "Novel correlations between the critical constants of the noble gases," *Fluid Phase Equilibria,* vol. 279, pp. 41-46, 2009.

[4] M. Vranic, D. Pintar and D. Gamberger, "Adapting hierarchical clustering distance measures for improved presentation of relationships between transaction elements.," *Journal of Information and Organizational Sciences,* vol. 36, no. 1, pp. 69-86, 2012.

[5] T. Li, L. Dai, L. Li, X. Hu, L. Dong, J. Li, S. K. Salim, J. Fu and H. Zhong, "Typing of unknown microorganisms based on quantitative analysis of fatty acids by mass spectrometry and hierarchical clustering," *Analytica Chimica Acta,* vol. 684, pp. 112-120, 2011.

[6] J. Laursen, N. Milman, N. Pind, H. Pedersen and G. Mulvad, "The association between content of the elements S, Cl, K, Fe, Cu, Zn and Br in normal and cirrhotic liver tissue from Danes and Greenlandic Inuit examined by dual hierarchical clustering analysis," *Journal of Trace Elements in Medicine and Biology,*

vol. 28, pp. 50-55, 2014.

[7] A. E. Bruno, A. M. Ruby, J. R. Luft, T. D. Grant, J. Seetharaman, G. T. Montelione, J. F. Hunt and E. H. Snell, "Comparing Chemistry to Outcome: The Development of a Chemical Distance Metric, Coupled with Clustering and Hierarchal Visualization Applied to Macromolecular Crystallography," *PLoS ONE,* vol. 9, no. 6, pp. 1-19, 2014.

[8] M. Li, X. Zhou, Y. Zhao, D.-P. Wang and X.-N. Hu, "Quality Assessment of Curcuma longa L. by Gas Chromatography-Mass Spectrometry Fingerprint, Principle Components Analysis and Hierarchical Clustering Analysis," *Bulletin of the Korean Chemical Society,* vol. 30, no. 10, pp. 2287-2293, 2009.

[9] M. Arguelles, C. Benavides and I. Fernadez, "A new approach to the identification of regional clusters: hierarchical clustering on principal components," *Applied Economics,* vol. 46, no. 21, pp. 2511-2519, 2014.

[10] S. Al Mamun, "Load Balancing Issues with Constructing Phylogenetic Trees using Neighbour-Joining Algorithm," *Journal of Physics: Conference Series,* vol. 341, pp. 1-4, 2012.

[11] D. Ashlock, T. vonKonigslow and J. Schonfeld, "Breaking a Hierarchical Clustering Algorithm with an Evolutionary Algorithm," *Intelligent Engineering Systems Through Artificial Neural Networks,* pp. 197-204, 2009.

[12] E. Kim, S. Kim, D. Ashlock and D. Nam, "Multi-k: accurate classification of microarray subtypes using ensemble k-means clustering.," *BMC Bioinformatics,* vol. 10, no. 260, pp. 1-12, 2009.

[13] SigmaAldrich, "Sigma-Aldrich: A Part of Millipore Sigma," 2016. [Online]. Available: http://www.sigmaaldrich.com/canada-english.html. [Accessed January 2016].

[14] Air Liquide, "Gas Encyclopedia," Air Liquide, 2016. [Online]. Available: http://encyclopedia.airliquide.com/encyclopedia.asp. [Accessed January 2016].

[15] National Institute of Health, "PubChem Search," PubChem NLM, NIH, HHS, 2016. [Online]. Available: https://pubchem.ncbi.nlm.nih.gov/. [Accessed January 2016].

[16] Lenntech, "Lenntech Periodic Table, Elements," Lenntech, 2016. [Online]. Available: http://www.lenntech.com/periodic/elements/index.htm. [Accessed January 2016].

[17] D. Ambrose, M. Ewing and M. McGlashan, "3.5 Critical constants and second virial coefficients of gases," National Physical Laboratory: Kaye & Laby Tables of Physical & Chemical Constants, 2015. [Online]. Available: http://www.kayelaby.npl.co.uk/chemistry/3_5/3_5.html. [Accessed 30 January 2016].

[18] National Institute for Occupational Safety and Health, "NIOSH Immediately Dangerous To Life or Health (IDLH)," Centers for Disease Control and Prevention: The National Institute for Occupational Safety and Health (NIOSH), 4 December 2014. [Online]. Available: http://www.cdc.gov/niosh/idlh/idlhintr.html. [Accessed 10 March 2016].

[19] N. P. Cheremisinoff, Handbook of Hazardous Chemical Properties, MA: Butterworth-Heinemann, 2000.

[20] P. Atkins and J. De Paula, Physical Chemistry: Ninth Edition, Great Britain: Oxford University Press, 2010.