The Energy Curve

Joseph S. Koopmeiners
St. John's University

August 3, 2001

Abstract

Many areas in science are very poorly understood which leads to many questions about large groups of objects that can be observed but little is known about their function. By grouping or clustering these objects together based on common traits it is possible to learn about their function. This paper discusses many different characteristics of clustering. Specifically a computer algorithm to make clusters of data and also the energy of clustered sets of data. The energy is a way of numerically representing the *tightness* of clustered data. This paper explores the reaction of the energy to both random and clustered data.

Contents

List of Tables

List of Figures

1 Introduction

Even with the substantial progress made in most scientific areas, there are still many situations where we can observe large sets of objects and still know relatively little about them. In these situations it often appears that there are certain structures in the data, unfortunately we do not know how to see these patterns. One possible way to see these patterns is to organize the data into similar groups or structures. In order to use a clustering algorithm on a computer to cluster this data there are certain things that need to be done. The data must be represented numerically, a concept of distance must be developed and the number of clusters needs to be determined. In this paper we will look at the clustering of random data in 2, 3 or more dimensions. This data will include artificially clustered data along with genetic and stock market data. Specifically this paper will give a possible clustering algorithm along with interpreting the energy of different set of data with a goal of gaining a better understanding of the clustering of data.

It is often useful to group sets together in order to see patterns developing. One way to do this is to use different characteristics as variables and to cluster them together. Clustering is a way of taking a set of points and grouping them together based on which points are "close" to one another. This process is useful for problems of all sizes since clustering can be done in any number dimensions. This project hoped to study clustering in a way that makes it easier to understand how it occurs and in turn to make it easier to apply clustering to problems in many different areas.

The idea of clustering stems from a larger concept of the Voronoi tessellation. A Voronoi tessellation is a region generated by a point in an n dimensional space such that every point in a given voronoi region is closer to its generator then to any other generator in the space. In cases where the generator is at the center of mass of the region it is called a centroidal Voronoi tessellation. Clustering is basically a centroidal Voronoi region made with only a finite number of points, where as a centroidal Voronoi region is typically used when the region is formed with an infinite amount of points. In any case where clustering or a Voronoi region is used, it is important to know how dense the regions are and in turn, how many regions to have in a space. The energy of the system is used to answer questions of this sort. This takes into account the distances from each point to its generator to develop the energy which tells how tightly clustered each region is. The lower the energy the more tightly clustered the data is.

In order to do research on clustering we first needed to develop a program that will cluster data for us. To do this we used the mathematical computing software, Matlab [1]. Matlab is able to make Voronoi regions for a group of points but it does not have a function for centroidal Voronoi region. This did not turn out to be a problem as it was ultimately easy to develop an algorithm for centroidal Voronoi tessellations and clustering.

Using what is known about Voronoi tessellations, energy and our clustering algorithm, this research hoped to study the curve made by graphing the number of clusters vs. energy. In this project we looked at many different sets of points. The two major groups we looked at were sets of random points and sets of clustered data in order to see the difference between the two energy curves. Both sets of data showed useful patterns. We were able to find an equation to describe the curve for the random data in n-dimensions. The clustered data also showed patterns which will allow us to determine how unknown data sets are clustered and in turn, what is the most efficient way to cluster them. These results will be very useful for future problems which involve clustering.

2 The Clustering Problem

There are many practical applications of the clustering problem. Basically any problem where the data needs to be put into groups can use clustering. For instance in biology it is clear that certain groups of species are related in some way. By using their different traits as variables it is possible to cluster them into groups which share the same traits. This is a situation where, for the most part, it is known that there are groups by looking at the species and seeing the similarities, on the other hand, clustering can also be used with data that nothing is known about in order to detect patterns in that data.

This application has been used in genomics because it can help initially cluster groups of genes together based on when they are expressed. These initial groupings can help lead biologists in the right direction towards determining the use of the specific genes.

Also, we have attempted to cluster stock market data in order to group stocks together that have behaved similarly in the past. This could help investors choose stocks based on whether they want stocks that behave similarly or differently.

By using clustering we hope to answer many typical questions one would encounter in a clustering situation. For example one might want to know how many clusters, or groups, to use. Also, one will want to know what is the best way to cluster the data (in the sense of what traits to cluster the data by) and also which points go into which clusters. Finally, in any clustering situation it is important to find a way to measure the effectiveness of the clustering. All of these questions can be asked for the above problems as well as any general clustering problem.

3 Voronoi Tessellation

For problems where it is important to determine if a point is closer to one of two other points a Voronoi tessellation can often be used. This is often used for problems involving the placements of depots, resource centers, or stores to determine if people are closer to one store or another as well as others problems of this sort. To define a Voronoi tessellation first let us consider a 2-dimensional region that contains n random points. From now on these points will be referred to as *generator points*. This region will also contain infinitely many other points. Each of these other points will be closer to one generator than it is to any of the other generators. If a cell is formed around each generator such that every point in the cell is closer to that generator than to any other generator the cell will be a Voronoi cell and the entire region will be a Voronoi tessellation. This is also true for n-dimensions but for visualization it is easier to look at 2-dimensions. To define this more specifically consider a set $\Omega \subset R^n$ and a set of generators $\{z_i\}_{i=1}^k$ which is in Ω. Then let,

$$V_i = \{x \in \ \Omega \ | \ ||x - z_i|| < ||x - z_j|| \ \text{ for } j = 1, \ldots, k, j \neq i\} \ i = 1, \ldots, k.$$

Each V_i would be a Voronoi cell and $\{V_i\}_{i=1}^k$ would be therefore be a Voronoi tessellation [3]. To see a more concise explanation, see Gunzberger [3].

In the general case a generator of a Voronoi cell can be at any place in the Voronoi cell. In some cases though, it may be helpful to move the generator of a Voronoi cell to its center of mass. This is usually done by placing the generator at the average of the points in its Voronoi cell. When this is done the Voronoi tessellation is called a centroidal Voronoi tessellation. Figure 1 and Figure 2 show examples of a Voronoi tessellation and a centroidal Voronoi tessellation.

The centroidal Voronoi tessellation has many different applications. The clustering problems for this project use centroidal Voronoi tessellations to develop the clusters. These clusters are basically centroidal Voronoi tessellations where the generator is a point in a finite set and not all elements are generators.

4 Energy

To explain the energy of the entire system it is easier to start by explaining the energy of one Voronoi cell. So consider a Voronoi cell from a larger region.

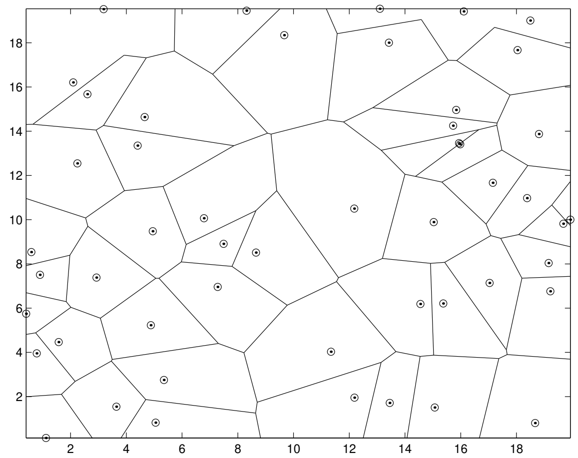

Figure 1: A Voronoi tessellation.

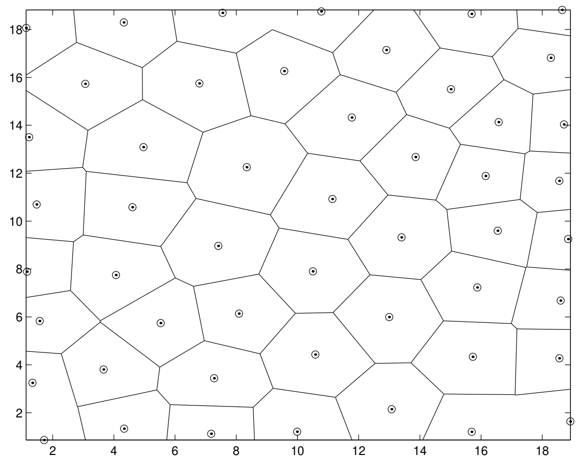

Figure 2: A centroidal Voronoi tessellation.

Suppose this cell contains exactly m points. Then we can compute the distances from each point to the generator. The sum of the squares of these distances is the energy of the cell. The sum of the energies of all the cells is the energy of the tessellation.

$$\epsilon_i = \sum_{x \in V_i} ||x - z_i||^2 \text{ and } E = \sum_{i=1}^{k} \epsilon_i$$

In order to minimize this energy, the generators for each Voronoi region should also be centroids. For a better explanation of this minimization property see, Gunzberger [2].

In order to illustrate this minimization property I have made two examples. Both examples have 20 random points and are both in 50 by 50 regions. The first picture, Figure 3, uses three randomly selected points as the generators and the energy was then computed for these generators. The second picture, Figure 4, uses our clustering algorithm to produce the generators which are centroids.

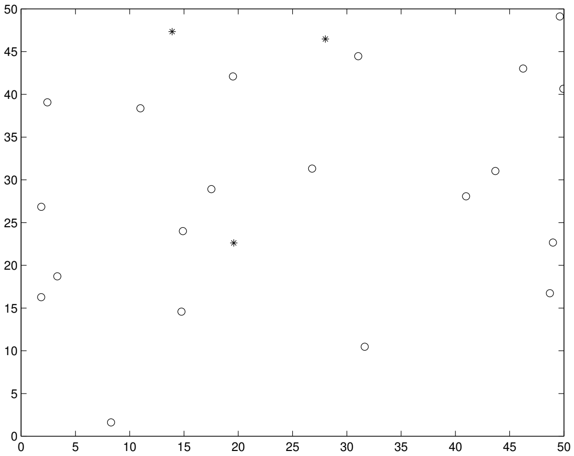

Figure 3: 20 points and 3 random generators.

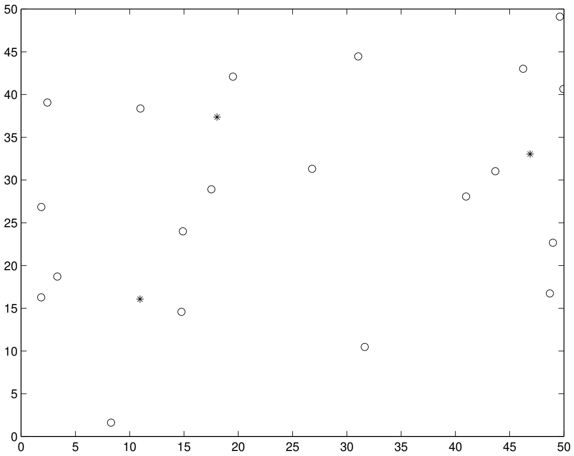

Figure 4: 20 points and 3 centroidal generators.

Using the generators and points from both examples I calculated the energy for both. Figure 3 ended with an energy of 327.9636, whereas Figure 4 had an energy of 215.0699. This illustrates the minimization property of centroids as the energy is approximately 2/3 of the original energy. Eventually these energy calculations will be used to determine different characteristics for a clustering problem.

5 The Clustering Algorithm

Using the idea of Voronoi tessellations and energy we were able to develop an algorithm for clustering sets of data. In order to do this we had to start with a set of data, in a region. This is illustrated in Figure 5.

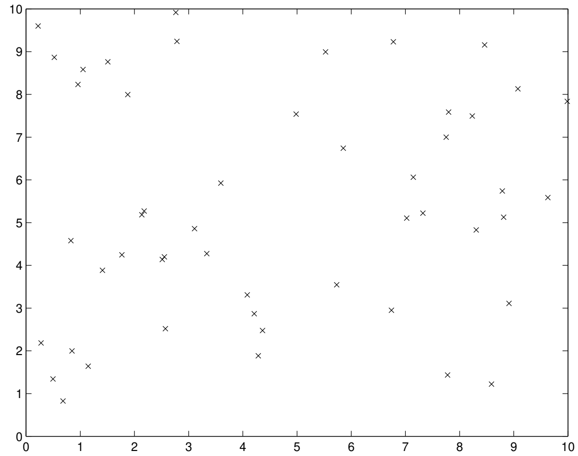

Figure 5: 50 random points.

After the set of random points is generated the next step is to pick starting points for the generators. There are two options for doing this. The first option is to pick a random point in the region and use that as the starting point for the centroids. The other option was to pick a random point from the data set and use that as the starting point for the generators. For clustered data we felt it was impractical to pick a point at random because most of the region will not have any points in it. Therefore we decided to use a random point from the data set as an initial guess for the generators. Figure 6 shows a data set with the original guesses at the centroids.

The next step is to sample the voronoi regions associated with the generators. The first step in this process is to determine which points are closest to which generators. To do this we compute the euclidean distance from each point to each generator and each random point is assigned to its nearest generator. For each generator, the set of points nearest to it are averaged. The generator is then moved to the average of these points. The average of these points, the estimated centroid, is used as the generator for the next step. This process is repeated iteratively depending on how good of an estimate for the centroid is desired. In our case we used 30. This will ultimately give us a centroidal

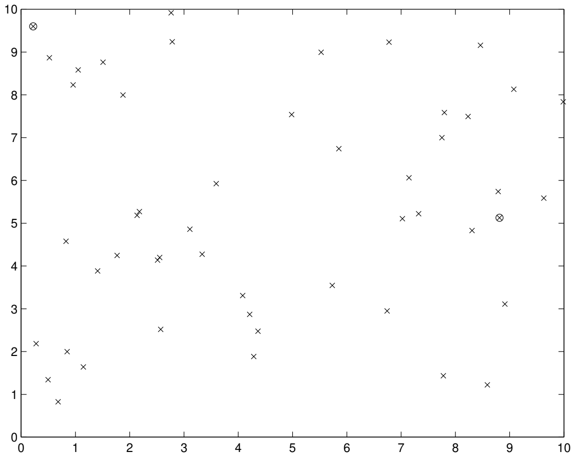

Figure 6: 50 random points and two initial generators.

Voronoi tessellation of the data. To compute the energy diagram we carried out the entire process for 1 generator, then for 2, and so on. For the 1st iteration, 1 generator is used, the second, 2; etc. An example of random points with the final generators can be seen in Figure 7.

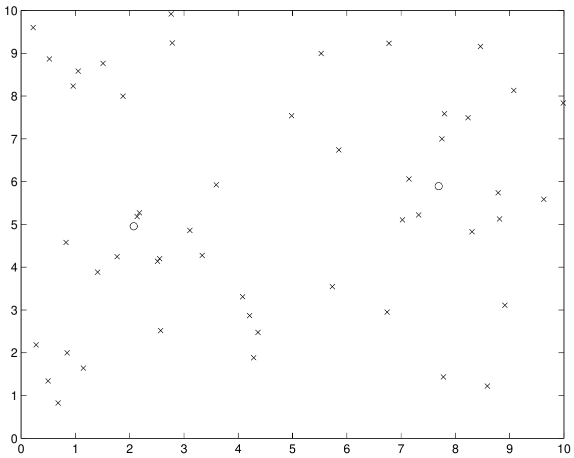

Figure 7: 50 random points with the final generators.

This algorithm worked for the most part. For the first few iterations it would in fact give us the minimized energy for the number of clusters used. Once the number of clusters got larger though, the minimum energy drops only slightly and often times our algorithm would not produce the minimum energy and the energy would actually be higher then the previous iteration. In order to eliminate this we added a 'last resort' technique. If after three tries with the original algorithm the energy did not drop, it simply takes the centroids from the previous iteration and adds another random point. Then it will find the new centroids using these points. The advantage to this is that adding an extra point can not make the energy ever go up. This way our algorithm is successful at finding the minimum energy for each number of clusters.

6 The Energy of Random Data

Ultimately, the goal is to be able to use this information to *guess* the number of clusters in a set of data in order to save time in the future. To do this we first must know how the energy curve looks for a set of random data. With this done there will be something to compare clustered and unknown data to. There are two important things to remember about the energy curve before we look at its characteristics. First of all, as the number of cluster points increases, the energy will always decrease. Secondly, if the number of cluster points is equal to the number of random points, the energy will be zero. Both of these principles can be seen in Figure 8, which is an energy curve with 40 random points.

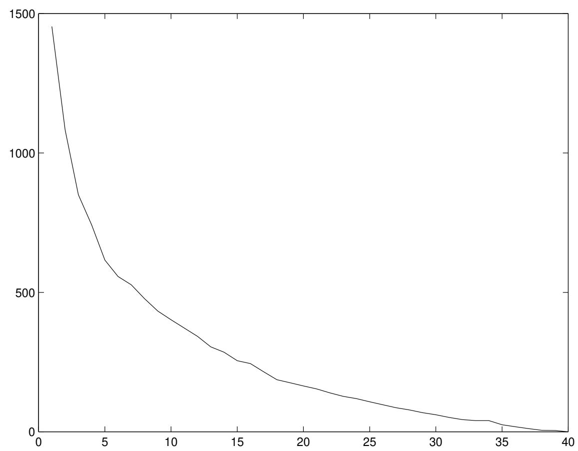

Figure 8: An Energy Curve for 40 random points.

As you can see the energy never goes up as the number of cluster points increases and when it reached 40 cluster points the energy was 0. The energy reaches 0 at this point because each random point should also be a cluster point in order to minimize the energy.

In order to first detect a pattern in the random energy curve we ran sets of data in one and two dimensions with 100 random points. In order to do future comparisons with different numbers of random points we scaled the energies so that we had a graph of cluster points vs. energy per point. After doing this we used least squares fitting and came up with these fits for the data, which can be found in table 1. To do this, we used Matlab's least squares function. We then computed R^2 for each fit which uses the distances of the actual points to the residuals in order to measure goodness of fit. The closer the R^2 is to 100 the better the fit.

data set	fit result	R^2
1d 100 pts	$y = 25.32 * x^{-1} - .30$	99.96
2d 100 pts	$y = 45.33 * x^{-.5} - 3.76$	99.86

Table 1: Fit results for the first sets of data.

From these two results we hypothesized that the general form of these curves might follow the pattern of $y = A * x^{-1/n} + B$ where n is the number of dimensions. In order to see if this was true we tested samples with different dimensions and different numbers of random points to see the effect this had on the equations. We found that as the set of random points got larger our hypothesis tended to be true. Again, in these cases we did a plot of cluster points vs. energy per point. Table 2 shows these results.

data set	fit result	R^2
1d 50 pts	$y = 25.54 * x^{-1} - .65$	99.36
1d 200 pts	$y = 25.01 * x^{-1} - .17$	99.98
1d 400 pts	$y = 24.95 * x^{-1} - .11$	99.99
2d 50 pts	$y = 50.3 * x^{-.5} - 6.95$	99.75
2d 200 pts	$y = 42.45 * x^{-.5} - 2.21$	99.93
2d 400 pts	$y = 42.61 * x^{-.5} - 1.68$	99.85
3d 50 pts	$y = 75.37 * x^{-.33} - 19.15$	98.36
3d 100 pts	$y = 65.52 * x^{-.33} - 10.86$	99.54
3d 200 pts	$y = 58.62 * x^{-.33} - 6.61$	99.95
3d 400 pts	$y = 57.40 * x^{-.33} - 4.85$	99.91
4d 100 pts	$y = 86.07 * x^{-.25} - 20.34$	99.09

Table 2: Fit results for the data.

These results seemed to confirm our hypothesis since the R^2 for these fits were all extremely high, showing that these are good fits for the data. As I commented earlier we seemed to get better fits as the size of the data set increased. In fitting this data we did in fact see this. By looking at the progression of these three 3-dimensional data sets with 50, 100 and 400 data points you can see that the fits do in fact get better as the data set gets larger.

From Figure 9, Figure 10, and Figure 11 it is clear that as the number of random points increases the equation fits the energy curve better.

Once the general form of these equations were determined our next goal was find a way to determine the variables A and B in these equation. In both of these cases we did see a pattern start to develop. It appeared that B was converging on 0 for all of these cases. Also, it seemed that for each dimension, A was also converging towards some number depending on the dimension. In

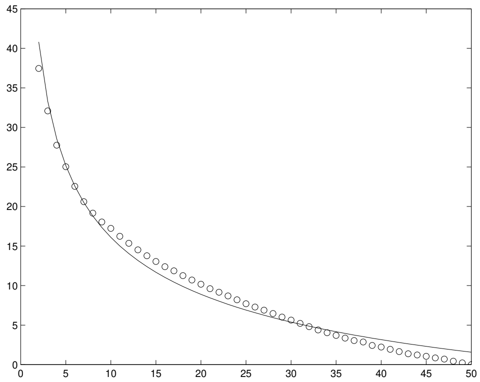

Figure 9: A 3d energy curve with 50 points.

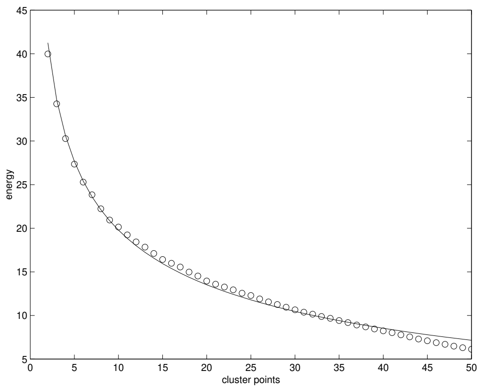

Figure 10: A 3d energy curve with 100 points.

order to check this we ran tests with larger sets of points in order to see these convergings. These results can be seen in table 3.

It seems that both A and B do in fact converge. The only problem is that as the dimension increases the size of the data set needed to find these values increases. For example, in 1d it is clear that A converges to about 25 and B is definitely moving towards 0. For 2 through 6 dimensions it seems that B is converging towards 0 but for 4, 5 and 6 dimensions there are not enough points to determine what A is converging towards. Unfortunately with our algorithm and software it was not possible to do samples with many more than 10000 points. The only other characteristic of a data set that affects the curve is the size of the region it comes from. For all of our trials we used 100 units for the size of each dimension, for example 1d was from 0 to 100. In order to account for changes in size all that needs to be done is to multiply each equation by the same number you'd multiply 100 to get the dimension. For example if the box was from 0 to 10 just multiply by .1 and from 0 to 1000 just multiply by 10.

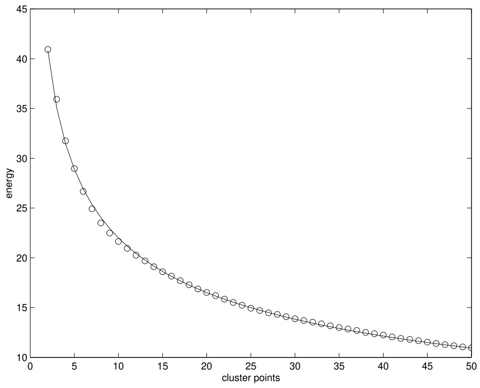

Figure 11: A 3d energy curve with 400 points.

data set	fits result	R^2
1d 10000 pts	$y = 25.01 * x^{-1} - .01$	99.99
2d 4000 pts	$y = 41.52 * x^{-.5} - .833$	99.74
2d 10000 pts	$y = 39.97 * x^{-.5} - .38$	99.71
3d 4000 pts	$y = 54.09 * x^{-.33} - 2.40$	99.79
3d 10000 pts	$y = 51.45 * x^{-.33} - 1.36$	99.51
4d 10000 pts	$y = 61.84 * x^{-.25} - 2.30$	99.20
5d 10000 pts	$y = 71.27 * x^{-.2} - 3.79$	98.87
6d 10000 pts	$y = 79.25 * x^{-1/6} - 4.75$	98.38

Table 3: Fit results for larger data sets

Using this should make it possible to predict the energy curve for random data for at least 1 and 2 dimensions, possibly more.

7 The Energy of Clustered Data

After determining how the energy curve will react to random data the next step was to see how different sets of clustered data would affect the energy curve. Initially the random and clustered data had both been plotted in the normal fashion of x, being the number of clusters, vs. the energy. In this case there was a clear difference between the random data and the clustered data but because of the curve it was difficult to understand what the differences meant. However, the graphs seemed to be behaving like $A * x^{-1/n} + B$ so a graph of $1/x^{1/n}$ versus E(x) would be a straight line. For this reason we decided to plot $1/(x^{1/n})$ instead of x so that for random data we would get a straight line. Figure 12 is an example of Figure 11 plotted in this manner.

Once the random data has been plotted in the form of a straight line it will be easier to compare it to the clustered data. The curve for clustered

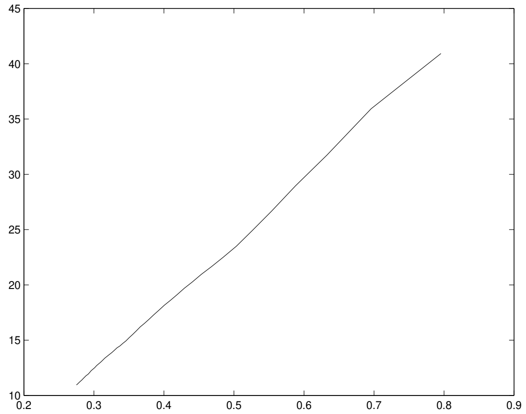

Figure 12: Figure plotted with $1/x^{.33}$.

data should always be lower then the random data because the clustered energy should always be lower then that of random data. This is because if the data is in fact clustered, by placing generator points into these clusters it should reduce the distance from each random point to its nearest generator. As long as there are less generators than clusters, the energy should be dropping quicker than random energy data. Once the number of generators is equal to the number of clusters the energy should only decrease in small increments until the number of generators is equal to the number of random points and the energy is 0.

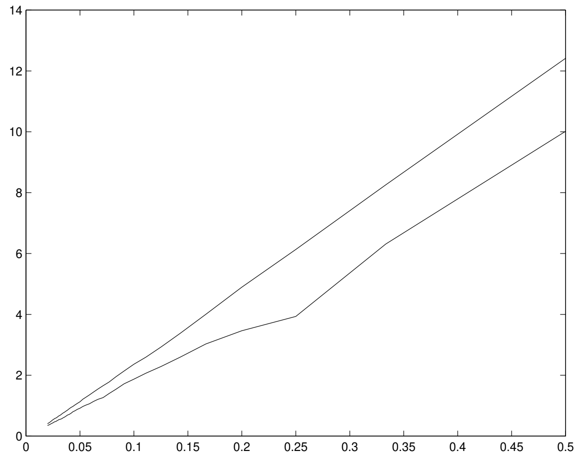

Figure 13: Comparison of random and clustered data

Figure 13, a set of random points in 1d, illustrates these points as the bottom line, which has 4 clusters, drops drastically until it reaches .25, at that point it flattens out and eventually comes back to the random data.

Using this plot of $1/(x^{1/n})$ vs. energy with unknown data should make it possible to determine whether or not the data is clustered and if so, how many clusters there are in the data. If the graph is close to a straight line it is probably random data. On the other hand if the graph of the energy shows a sharp bend this typically indicates that the data is clustered. By looking at the sharpness of the bend and where it occurs it's possible to determine how many clusters there are and how tightly they are clustered. For example if the bend is very sharp and distinct, the clusters are probably tightly clustered, on the other hand if the bend is not very sharp the data is probably not clustered tightly. Also, using the point where the bend occurs and $1/(x^{1/n})$ we can solve for the the number of clusters, x. For example, if the bend is at .25 and it is in 1-dimension there are 4 clusters.

While working on this project we looked at some DNA data which needed to be clustered. We put the data through our algorithm and came up with its energy curve. Its energy curve is illustrated in Figure 14. The curve shows a bend at about .905. The DNA data is in 14 dimensions therefore we know that if x is the number of clusters $1/x^{1/14} = .905$. We can solve for this and get $1.105^{14} = x$. This leaves us with 4.05=x but since x has to be a whole number we can assume that x=4 and therefore the optimal number of clusters for this data set is 4. Using this process the optimal number of clusters can be found for any data set.

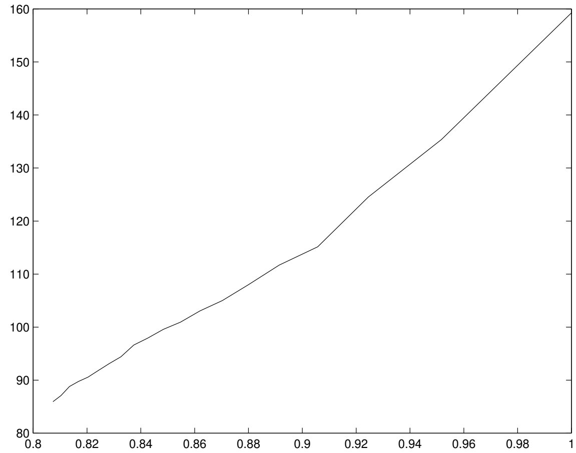

Figure 14: Energy curve for DNA data

This is just one possible way to interpret a clustered energy curve. On the other hand there are some problems with this interpretation. This graph only has 180 data points. In order to be sure that it follows $x^{-1/14}$ there should be considerably more points. With a large number of points this graph should have energy 0 at 0. With only 180 the graph should theoretically have energy 0 around .69. In reality though, the graph would appear to have energy 0 around .5. This illustrates that for this to work perfectly there needs to be a large number of points. Figure 15 illustrates this by comparing the curve to a line that passes through $(0, 0)$ and the maximum energy of the curve at $(1, 160)$.

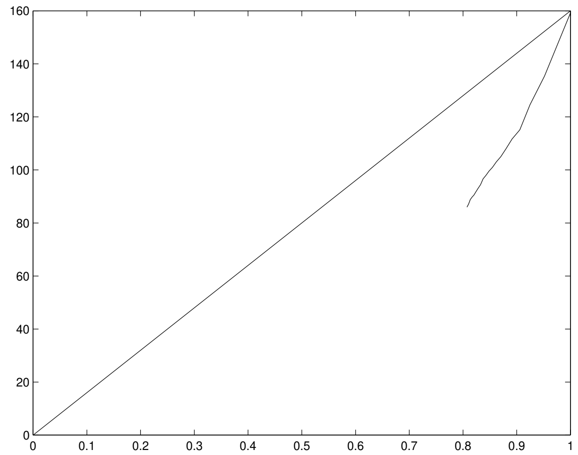

Figure 15: The DNA energy curve compared to a straight line

8 Discussion

This project has led to a better understanding of the energy curve and how it's affected by different sets of data. It seems that there is enough evidence to support the conclusion that the random energy curve does actually follow $y = A * x^{-1/n} + B$. Ultimately this did prove useful in this project as it gave some insight into the optimal clustering of different sets of data. On the other hand this does lead to several questions about this equation. Computer limitations made it impossible to estimate the coefficient A for more than 1 or 2-dimensions. Ultimately to understand what is happening with these energy curves it will be helpful to determine these coefficients. With more computing power this should certainly be possible. Once some of these coefficients are found it would be helpful if a pattern could be found as they change based on the dimension. If a pattern could be established an equation would be known for any dimension. This would make it possible to make more predictions about energy of data in larger dimensions.

Along with questions about the random energy curve, there are other questions that come from a discussion of the energy of a region. In most of these situations we have tried to determine the number of clusters in a set of data using the energy curve. Another possibility would be to try to estimate the energy of a set of data at certain numbers of energy clusters. Using different estimation techniques along with these equations it should be possible to do this.

Another possible question that this project didn't look at was the effect of different shapes of regions on the energy curves. This project only looked at situations where all dimensions were the same, squares, cubes, etc. By changing the shape to rectangles, circles as well as any other shapes it could be interesting to see how this affects the energy curve. In particular, a long thin rectangle is almost a 1d object. Hence, the energy curve for data in such a region should be "between" the 1d and 2d curves.

Since the ultimate goal of this project was to learn more about clustered data there are also many questions involving clustered data sets. One possible way to interpret the clustered energy curve was suggested earlier where you plot $x^{-1/n}$ and solve backwards for the optimal number clusters. This is just one possible way of interpreting the data. There are other possible ways to interpret the clustered energy curve using the normal clustered energy curve and finding the derivative at some places to see how it changes. This project did not look into many of these possible interpretations and ultimately they could be very helpful. Also, the method described earlier has problems when the number of clusters is very large. In these cases it can be difficult to solve for x because as the number of clusters gets larger, $x^{-1/n}$ becomes more clustered together. By looking into these problems it could make it much easier to interpret this clustered data.

References

[1] MATLAB is a software product of The MathWorks, Natick, Mass.

[2] Max Gunzberger, Qiang Du, V. Faber *Centroidal Voronoi Tessellations: Applications and Algorithms*; SIAM Review 41, 1999, 637-676.

[3] Max Gunzberger, Qiang Du, Lili Ju *Probabilistic Methods for Centroidal Voronoi Tessellations and Their Parallel Implementation*; to appear, Comput. Meths. Appl. Mech. Engrg.