Predictive Analytics and Futurism

April 2018
Issue 17

Why Consider a Delphi Study?

By Ben Wolzenski

In the December 2017 *Predictive Analytics and Futurism Newsletter*, author and recent Predictive Analytics and Futurism (PAF) Section Council member Bryon Robidoux wrote about the TED talk, "The Human Insights Missing from Big Data," by Tricia Wang. I highly recommend that article, which also contains a link to access the TED talk. It provides a perfect preface to this article about an old futurism tool in the new world of predictive analytics: the Delphi study. Both articles support the idea of supplementing the results of a model with data from alternative sources to help validate the model. A more scientific way than relying on yourself or a co-worker for insight is to use a Delphi study.

Like predictive analytics, the Delphi method is used for forecasting. But there they diverge; instead of tools and data, the Delphi employs a panel of experts ("panelists") to address specific questions or issues. But unlike a roundtable discussion or a mere survey, the Delphi technique gathers responses from panelists anonymously, and sends all those separate responses (again, anonymously) to each panelist. The panelists are asked to reconsider and possibly refine their responses based on the information gleaned from the responses of all the others. These "rounds" of questions and answers are repeated until the respondents stop making material changes to their answers. The result may be a consensus, or convergence around two or more points of view.

The Delphi method is most useful when other forecasting techniques, especially those that use past data to estimate future outcomes, appear to have limited value. Or when the forecaster simply feels the need for a second opinion, derived by other means. The Delphi method has been around since the 1950s, but was almost unused by the actuarial profession until 2005, when the Society of Actuaries (SOA) published "A Study of the Use of the Delphi Method, A Futures Research Technique For Forecasting Selected U.S. Economic Variables and Determining Rationales for Judgments." That landmark study was as much (or more) about how to perform a Delphi study as it was about predicting economic variables in 2024 (and the rationales for those predictions).

Then, in 2009, the SOA published "Blue Ocean Strategies in Technology for Business Acquisition by the Life Insurance Industry." In three rounds of narrative questions and panelists' responses, a series of strategies were identified and refined. Here are two examples:

- Strategy #5: Your Way Insurance Company—"Prospects custom-design coverage online"
- Strategy #8: Holistic Insurance Company—"Risk 'agents' help mitigate all risks"

The next major Delphi study by the SOA was spearheaded by the Long Term Care Think Tank and published in 2014: "Land This Plane," with the goal of arriving at a consensus on solutions to the nation's long-term care financing challenges. There were widely different views about the roles of government and the insurance industry among the long-term care experts recruited to be panelists. Despite these differences, the final report identified a series of principles upon which there was general (albeit not unanimous) agreement.

And even as this article was written, the SOA has launched a second Delphi study regarding economic variables, with a focus on methods and assumptions for financial projection models. With an ever-greater world of data at our disposal, the comprehensive training of actuaries gives us an advantage in applying human insight—and the Delphi method can provide a means to derive value from that insight.

Ben Wolzenski, FSA, MAAA, is managing member at Actuarial Innovations, LLC in St. Louis. He can be reached at *bwolzenski@gmail.com*.

Hierarchical Clustering: A Recommendation From a Nonhierarchical Manager

By Dave Snell

Figure 1
Example Where *k*-Means (Where k=4) Works Well[1]

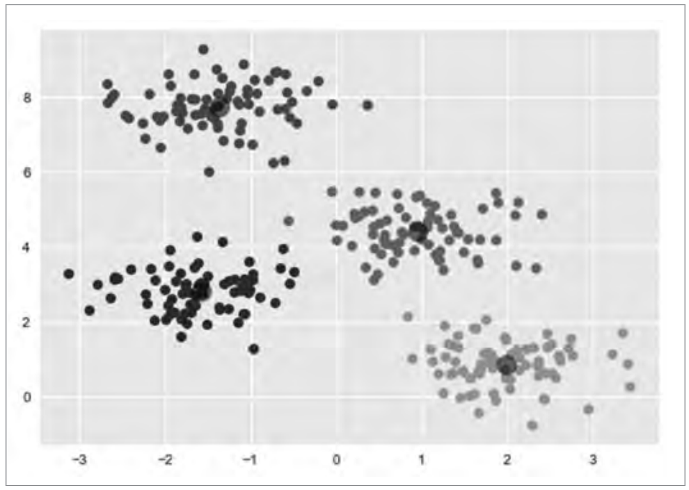
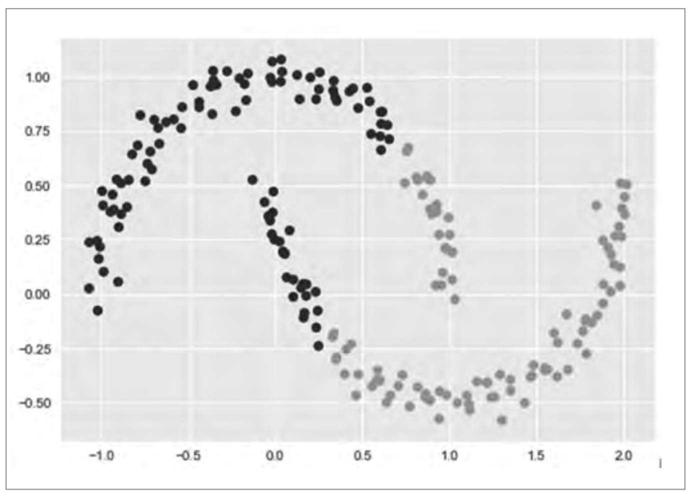


Figure 2
Example Data Where *k*-Means Does Not Work Well
(consider an affinity method instead)



Most of the people who know me well are aware that I'm not a big fan of hierarchical management. Back when I was VP over a fairly large area I used to value highly the direct reports who felt comfortable challenging my ideas; and the collaborative outcomes from our discussions were often far better than my original thoughts.

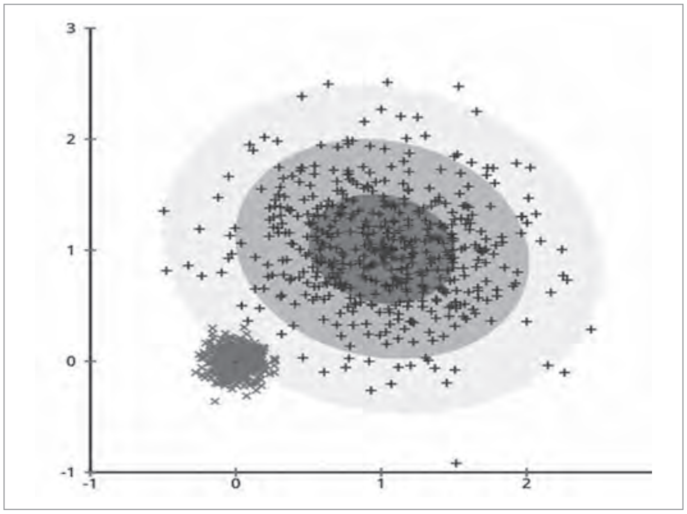
So, it might seem strange that my first choice on an article to describe clustering is about the benefits of hierarchical clustering as opposed to the more commonly used nonhierarchical techniques such as *k*-means clustering. Both categories are usually unsupervised machine learning techniques (techniques where you do not know the outcomes or labels ahead of time); but *k*-means clustering intuitively appeals to mathematicians because it is easy to conceptualize (but not visualize) in several dimensions.

In *k*-means clustering, you just pick a *k* (the desired number of clusters), assume *k* random points in your data as the initial centers of the clusters, assign each data point to one of the clusters based on their distances from those *k* centers, and then compute new centers for each cluster based on the distance metrics. Since the initial choices were random, it is likely they were wrong. At the next round of point assignments, some points are reassigned to another cluster based on closeness to the new centers you calculated. Again, the cluster centers are recalculated and the process continues until points stop changing from one cluster to another. This method is computationally efficient, easily accommodates several dimensions of factors, and, again, it appeals to mathematicians.

Unfortunately, it is not always the most appropriate clustering technique. As you can see in Figure 1, *k*-means can do a good job if the underlying data clusters are distinct (not overlapping), and the underlying clusters are somewhat spherical in nature and of similar density. If the data is donut-shaped, or follows a specific curve, or is radial in nature, as in Figures 2 and 3 (pg. 16), it does not give a good result.

Beyond this, *k*-means clustering requires you to choose the number of clusters (*k*) ahead of time. If you are doing an exploratory analysis of a large set of data, you may not know the appropriate *k* ahead of time. Granted, you can try several different values of *k* and see where the sweet spots seem to be on an elbow curve; you can do a silhouette analysis; and you can measure the purity of each cluster; but these tests can introduce complexity rather than clarity.

Figure 3
Example Data Where *k*-Means Does Not Work Well (consider a Gaussian mixture model instead)



Most of all, though, the *k*-means approach is not as easy to explain to nonmathematicians, and once you get to higher dimensions, where scatter plots may not be appropriate, it lacks a visually intuitive presentation mechanism.

In cases of higher dimensionality,[2] such as four or more, you may wish to consider a hierarchical clustering approach. Even three-dimensional clusters can be very misleading when shown in two dimensions. A famous anamorphic creation by the artist Michael Murphy titled "Perceptual Shift" shows this vividly. Looking at it from the front, it appears to be a human eye; but from the side it is a cone of seemingly scattered balls.[3] The most recognizable pattern of stars in the northern hemisphere, the Big Dipper (actually part of the constellation Ursa Major) looks like a flattened ladle from Earth; but Mirza, the closest star of the seven, is 78 light years away from us while Dubhe, the farthest, is 123 light years away! Seen from another galaxy, this group of stars looks nothing like a dipper.

A hierarchical clustering approach starts with the assumption that every data point is its own cluster. Then, it computes the distance between each pair of clusters and starts grouping them accordingly.

In order for the algorithms to work, there are four distance rules we have to specify:

1. Distance cannot be negative: $d_{i\,j} > 0$ when $j \neq i$ (i.e., the distance from cluster i to a different cluster j is positive).
2. Distance from any cluster to itself is zero: $d_{i\,i} = 0$.
3. Distance is symmetric: $d_{i\,j} = d_{j\,i}$ (i.e., the distance from cluster i to cluster j is the same as the distance from cluster j to cluster i).
4. A triangular inequality holds: $d_{i\,j} + d_{j\,k} >= d_{i\,k}$.

Given these rules, we can choose any of a number of different metrics for "distance." Some common choices are shown in Figure 4.

Figure 4
Commonly Used Distance Metrics for Hierarchical Clustering[4]

Names	Formula
Euclidean distance	$\|a-b\|_2 = \sqrt{\sum_i (a_i - b_i)^2}$
Squared Euclidean distance	$\|a-b\|_2^2 = \sum_i (a_i - b_i)^2$
Manhattan distance	$\|a-b\|_1 = \sum_i \lvert a_i - b_i \rvert$
Maximum distance	$\|a-b\|_\infty = \max_i \lvert a_i - b_i \rvert$
Mahalanobis distance	$\sqrt{(a-b)^\top S^{-1}(a-b)}$ where *S* is the Convariance matrix

Figures 5 and 6 give an idea of what this process looks like visually. Initially, let's assume that we had only six data points. We start out assuming each is its own cluster. Alternatively, if you feel this is too trivial an example, we might wish to say that Figure 5 is the result of previous clustering of a large number of points already; and we are now down to six clusters.

We see in Figure 5 that clusters *b* and *c* are very close to each other, as are clusters *d* and *e*. This is reflected in Figure 6, as the number of clusters is reduced in Round 1 to four: clusters *a*, *bc*, *de* and *f*.

In the next round we note that cluster *de* is closer to cluster *f* than to any other cluster so they are combined into cluster *def*. Next, *def* is combined with cluster *bc* to obtain cluster *bcdef*. Finally, cluster *a* is combined with *bcdef* to form the single cluster *abcdef*. Usually, hierarchical clustering methods are also called agglomerative methods,[5] and you can see why here. Eventually, you end up with just one cluster.

At this point, you might be wondering where I am going with this discussion. Why is the lumping together of all the data into just one cluster of any use to us?

The usage comes into play via a special sort of tree diagram, called a dendrogram. A dendrogram of the clustering process we did for our example is shown in Figure 7. Note that this is a visual way of showing how the clusters are combined and also the relative dissimilarity between the clusters. The taller the height before two clusters are combined, the more dissimilar they are. We see that cluster *a* was most different from all of the other clusters, while *d* and *e* were relatively close.

Let's consider a more practical example of how hierarchical clustering can be useful.

Assume your daughter (or son or niece or nephew or friend) is a junior or senior in high school and wants to apply to a university with the intent of a double major—in actuarial science and data science. You want to help in this project, so you compile a list of 40 or so universities that offer both of these majors. The parameters for selection may include items such as student population, ratio of students to faculty, percentage of scholarships available, distance from home (far enough away for autonomy

Figure 5
Six Clusters Prior to Hierarchical Clustering

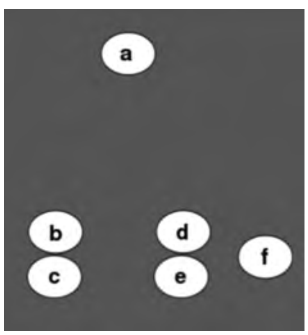
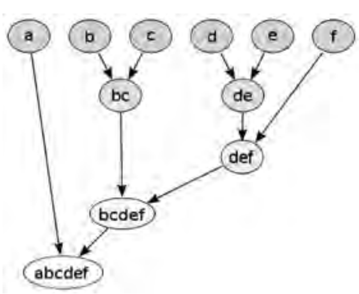


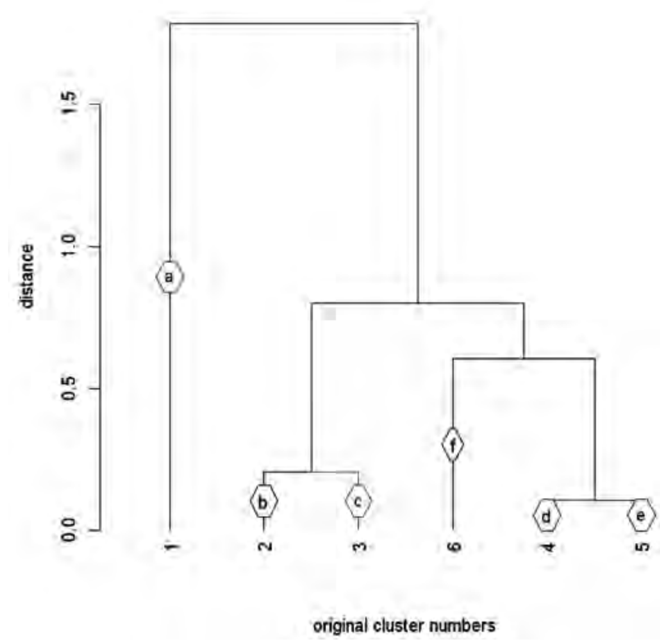
Figure 6
Traditional Representation of Hierarchical Clustering



Another benefit of hierarchical clustering is repeatability.

Unlike k-means clustering, which can result in different answers based on different starting values for the randomly chosen first set of center points, hierarchical clustering is repeatable. As long as your data has not changed, and you use the same distance metric, you will always get the same result.

Figure 7
Dendrogram of Six Clusters[6]



and close enough to bring laundry home), housing costs and tuition, number of Nobel Laureates teaching classes, median SAT and ACT scores of incoming students, median compensation of graduates after five years, athletic team performances, cultural opportunities, male-female student ratio, international student ratio, cafeteria selections, average temperature range, proximity to the ocean or the mountains, population of nearby city, Centers of Actuarial Excellence (CAE) status, data science rating and perhaps several other criteria.

You don't want to risk applying to only one university, since you can't predict how selective they may be. Perhaps the admissions officer at the interview will be impressed by her initiative and creativity to make an interview video while juggling on a skateboard to show multitasking ability. But what if the interviewer considers this an indicator of a frivolous nature? On the other hand, each application is expensive both in dollars and in the time spent visiting the campus and researching the overall school environment. It would be nice to be able to say with some confidence that a specific subset, or group within these 40 schools, is most similar to this student's interests and abilities. This can be an ideal problem for a hierarchical clustering solution. You have many dimensions and it is not obvious how to group the schools into logical clusters.

It will be necessary to convert the categorical factors, such as CAE status and cultural opportunities to numeric values—often via dummy variables. Then there is the issue that some of these numeric parameters have wide ranges relative to others. For example, the number of students might be just a few hundred, or many thousands. Expenses and distance from home may also have wide ranges. Compare those to the number of Nobel Laureates, where 0 to 5 might cover every one of the schools. In order to avoid having the wide-range items completely overshadow the importance of short-range ones, we would employ statistical techniques to standardize and normalize our values. One such technique might be to substitute each value x_i with $(x_i - x_{mean})/x_{standard\ deviation}$, which would work fine for a mix of all numeric parameters, but still tends to have higher weight than the categorical surrogates that range from 0 to 1. In a mixed parameter environment, it might be better to map x_i to $(x_i - x_{minimum})/(x_{maximum} - x_{minimum})$, thus ensuring all the items have the range 0 to 1.

Once you have your values normalized, both Python and R have packages that can do all the heavy-lifting work of creating the dendrogram for you. R, in particular, has a package *dendroextras* that allows you to label and color your clusters:

```
if (!is.element('dendroextras',
  installed.packages()[,1]))
  install.packages("dendroextras",
    repos='http://cran.us.r-project.org')
```

I don't have all those parameters available for my hypothetical problem, but I did find a ranking of world university rankings on Kaggle at *https://www.kaggle.com/mylesoneill/world-university-rankings* that I will use for a very quick demonstration of how to generate a dendrogram of the universities. In this demonstration, I'll keep it simple and use the built-in hierarchical clustering in R:

```
# file from Kaggle site in text
input <- read.csv('cwurData.csv')
tail(input)
```

that produces Figure 8.

Figure 8
Sample of Kaggle University Rankings (Kaggle dataset has 1,000 universities in this dataset)

	world_rank	institution	country	national_rank	quality_of_education	alumni_employment	quality_of_faculty	publications	influence	citations	broad_
2195	995	King Abdulaziz University	Saudi Arabia	4	367	449	218	595	430	645	
2196	996	University of the Algarve	Portugal	7	367	567	218	926	845	812	
2197	997	Alexandria University	Egypt	4	236	566	218	997	908	645	
2198	998	Federal University of Ceará	Brazil	18	367	549	218	830	823	812	
2199	999	University of A Coruña	Spain	40	367	567	218	886	974	812	
2200	1000	China Pharmaceutical University	China	83	367	567	218	861	991	812	

Now, we generate the dendrogram:

```
# just take the top 40 for this example
uniRatings <- input[1:40,c(2,1,4:10)]
# exclude university name and normalize
normalizedRatings <- scale(uniRatings[,2:9])
distance <- dist(normalizedRatings,
  method='euclidean')
clus <- hclust(distance, method='complete')
plot(clus,hang=-1) # display the dendogram
# cut the dendogram into 5 clusters
groups <- cutree(clus, k=5)
rect.hclust(clus, k=5, border='red')
# output is Figure 9
```

I then add a new column that denotes group number to the data frame:

```
uniRatings$group <- groups
uniRatings[1:8]
# Output is Figure 10
```

Figure 9
Top 40 World Universities in Five Clusters

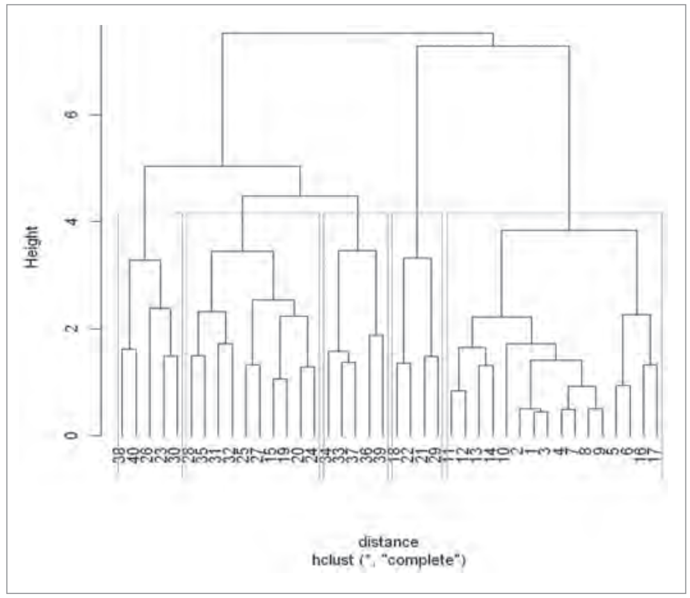


Figure 10
Section of Group 1 of the Top Universities

institution	world_rank	national_rank	quality_of_education	alumni_employment	quality_of_faculty	publications	influence	citations	group
Harvard University	1	1	7	9	1	1	1	1	1
Massachusetts Institute of Technology	2	2	9	17	3	12	4	4	1
Stanford University	3	3	17	11	5	4	2	2	1
University of Cambridge	4	1	10	24	4	16	16	11	1
California Institute of Technology	5	4	2	29	7	37	22	22	1
Princeton University	6	5	8	14	2	53	33	26	1
University of Oxford	7	2	13	28	9	15	13	19	1
Yale University	8	6	14	31	12	14	6	15	1

Our top group is probably no big surprise. The highest-rated universities are in the same group.

But later, we find some surprises, as the 10 universities shown in Figure 11 are all ranked very similarly (31 through 40), but they are not that much alike when you consider all of the parameters. In fact, University College London is more like Osaka University or University of Toronto than it is like Northwestern or Washington University in St. Louis. Of course, different criteria, such as my hypothetical ones, would group all these universities differently, but that is part of the beauty of hierarchical clustering: You get to decide what features are important, and the similarity grouping is based only upon them.

```
uniRatings[31:40,]
# output is Figure 11
```

In this article, I expressed my opinion that hierarchical clustering can provide advantages over *k*-means clustering when the number of dimensions, *n*, is too high for a scatter plot.[7] The dendrogram is a convenient way to show both the clusters and the relative dissimilarity between them. It also lets you choose a cut point (number of clusters) after construction of the dendrogram so you can see logical groupings by extent of dissimilarity before you do more calculations. I hope you find the examples using R useful. Python has very similar capabilities. Whichever programming language you prefer, I think it is worth investigating this underutilized technique for clustering. ■

Dave Snell, ASA, ACS, ARA, ChFC, CLU, FALU, FLMI, MAAA, MCP, teaches AI Machine Learning at Maryville University in St. Louis. He can be reached at *dave@ActuariesAndTechnology.com*.

ENDNOTES

1 Just because a scatter plot looks good in two dimensions does not mean it actually represents the data arrangement. See a detailed description of the anamorphic creation by Michael Murphy, "Perceptual Shift," at *https://mymodernmet.com/michael-murphy-perceptual-shift/*.

2 Although hierarchical clustering is good for *n* dimensions, where *n* is often > 3 and beyond those we can readily graph, it involves the computation and storage of an *n* by *n* matrix, which can be a strain on computing and storage resources.

3 *Supra*, note 1.

4 Figures 4, 5, and 6 are derived from Wikipedia. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled GNU Free Documentation License.

5 Actually, hierarchical clustering can be agglomerative (the usual case) where you start with *n* points and keep combining them until you have only one cluster; or they can be divisive, where you start with one cluster, then keep subdividing it.

6 Figure 7 was generated by the author using the R package *dendroextras*.

7 *Supra*, note 2.

Figure 11
Another Section of the Top University Rankings, Showing Varying Groupings

	institution	world_rank	national_rank	quality_of_education	alumni_employment	quality_of_faculty	publications	influence	citations	group
31	University College London	31	4	35	101	45	27	23	33	2
32	Osaka University	32	3	77	101	44	39	44	51	2
33	Northwestern University	33	23	101	32	101	24	25	20	5
34	University of Michigan, Ann Arbor	34	24	68	60	101	2	17	7	5
35	University of Toronto	35	1	101	101	34	7	14	18	2
36	University of North Carolina at Chapel Hill	36	25	101	86	56	31	29	31	5
37	Washington University in St. Louis	37	26	74	62	101	32	18	30	5
38	University of Utah	38	27	92	101	41	74	52	67	4
39	University of Washington - Seattle	39	28	101	101	40	5	7	5	5
40	University of California, Santa Barbara	40	29	101	101	28	68	72	36	4