AFFINITY PROPAGATION: A CLUSTERING ALGORITHM FOR COMPUTER-ASSISTED BUSINESS SIMULATIONS AND EXPERIENTIAL EXERCISES

Precha Thavikulwat
Towson University
pthavikulwat@towson.edu

ABSTRACT

Affinity propagation is a low error, high speed, flexible, and remarkably simple clustering algorithm that may be used in forming teams of participants for business simulations and experiential exercises, and in organizing participants' preferences for the parameters of simulations. The four-equation algorithm is easy to encode into a computer program. An example is given and an application is described. Incorporated into GEO, an Internet-based, computer-assisted international business simulation of a global economy, the algorithm organizes policy proposals submitted by participants for simulating direct and representational democracy.

INTRODUCTION

The need to cluster is pervasive in experiential learning. In forming teams to work on exercises, instructors may seek either to maximize or minimize the diversity of each team, depending upon the educational objective. If the objective is to expose team members to the greatest range of views, maximally diverse teams are preferred, but if the objective is to minimize friction within teams, minimally diverse teams may be better. In both cases, clustering is necessary, either to select representatives from each cluster to form the teams or to form teams corresponding to the clusters that obtain.

The application of clustering in experiential learning goes beyond the selection of people. Clustering can be used to select ideas. Participants engaged in a business simulation may have ideas as to how the parameters of the simulation ought to be set. Some might prefer a lower income tax rate, others a lower fee for services, and still others a lower interest rate, among many possibilities. If the administrator is interested in involving participants in setting the parameters, thereby simulating some aspects of government, the administrator will require a means of clustering their preferences.

Clustering preferences is a simple problem when the preferences are highly correlated, so that knowing a person's preference on one idea is predictive of the person's preferences on other ideas, but this is a special case. In the general case when preferences are more or less independent, the clusters will be difficult to identify, because any two persons who agree completely on any one idea may disagree on related ideas.

In particular, the simulation of government requires that participants with similar ideas recognize each other, form political parties, and jointly promote their ideas. This is possible only when each participant knows what other participants think. They can gain that knowledge through conversation, but conversation is time-consuming. What is needed is an algorithm that can rapidly sort ideas into clusters, so that participants may know how their own ideas fit with those of their peers. Affinity propagation may be the ideal algorithm for this purpose. This paper explains the algorithm and shows how it has been incorporated into a computer-assisted business gaming simulation for the simulation of government.

CLUSTERING METHODS

Cluster analysis is generally regarded as a subfield of multivariate analysis (Johnson & Wichern, 2007) and of data mining (Hand, Mannila, & Smyth, 2001). Aldenderfer and Blashfield (1984) trace the literature of cluster analysis to Sokal and Sneath's (1963) *Principles of Numerical Taxonomy*, and lists seven major methods: hierarchical agglomerative, hierarchical divisive, iterative partitioning, density search, factor analytic, clumping, and graph theoretic. Aside from the well-known factor analytic method, ABSEL researchers have used hierarchical agglomerative methods (Burns & Banasiewicz, 1994; Sackson, 1990; Zalatan & Mayer, 1990), a hierarchical divisive method (Overby, 1994), and an iterative partitioning method (Chang, Choi, Moon, et al., 2005; Chang, Choi, Ng, et al., 2005). In all these instances, cluster analysis was used for academic research apart from the exercise. It was not incorporated into the exercise itself.

Affinity propagation is a graph theoretic clustering method recently developed by Frey and Dueck (2007), who have tested it against *k*-centers clustering, an iterative partitioning method similar to the popular *k*-means procedure that is available on SPSS 15, differing in that *k*-means clusters items around a computed central values whereas *k*-centers clusters them around exemplars, each one being the most central item of its cluster. When applied to a large database of human faces and a large database of mouse DNA segments, Frey and Dueck found that affinity propagation gave rise to smaller errors and arrived at its

solution at least two orders of magnitude faster, an important consideration because clustering data is inherently a computationally intensive problem. Thus, the number of possible ways in which 50 items may be in two clusters is 2^{50}, so a computer able to evaluate 10,000 possibilities a second will require 3,570 years of evaluate every possibility.

Moreover, unlike *k*-centers and *k*-means, affinity propagation is more flexible in two ways. First, it does not require the user to specify the number of clusters in advance. Rather, the user selects initial "self-similarity" values from a set derived from the data itself, such that lower self-similarity values give rise to a smaller number of clusters. Second, a small change in the algorithm causes it to identify outliers instead of exemplars, useful for some purposes.

The primary limitation of affinity propagation is its requirement of a large memory space. The method requires four N x N matrices, where N refers to the number of items to be clustered. Thus, if 10,000 items are to be clustered and if each item is to occupy the 8 bytes of memory needed for a double data type, about 3 gigabytes of memory are needed.

For business simulations and experiential exercises, affinity propagation's lower error, higher speed, and greater flexibility should be advantageous, whereas the required large memory space would generally be a negligible concern, because the number of items that must be clustered will generally fall below 1,000. Even so, the most compelling advantage of affinity propagation may be the simplicity of the algorithm, because a simple algorithm is easier to code and less likely to be coded incorrectly.

THE ALGORITHM

Mézard (2007) points out that affinity propagation is known in computer science as a message-passing algorithm, and suggests that the algorithm can be understood by taking an anthropomorphic viewpoint. Thus, imagine that each item being clustered sends messages to all other items informing its targets of each target's relative attractiveness to the sender. Each target then responds to all senders with a reply informing each sender of its availability to associate with the sender, given the attractiveness messages that it has received from all other senders. Senders absorb the information, and reply to the targets with messages informing each target of the target's revised relative attractiveness to the sender, given the availability messages it has received from all targets. The message-passing procedure proceeds until a consensus is reached on the best associate for each item, considering relative attractiveness and availability. The best associate for each item is that item's exemplar, and all items sharing the same exemplar are in the same cluster. Essentially, the algorithm simulates conversation in a gathering of people, where each in conversation with all others seeks to identify his or her best representative for some function.

Procedurally, the algorithm operates on three matrices: a similarity (*s*) matrix, a responsibility (*r*) matrix, and an availability (*a*) matrix. Results are contained in a criterion (*c*) matrix. These matrices are iteratively updated by four equations, where *i* and *k* refer, respectively, to the rows and columns of the associated matrix, as follows:

$$r(i,k) \leftarrow s(i,k) - \max_{k' \text{ such that } k' \neq k} \{a(i,k') + s(i,k')\}, \quad (1)$$

$$a(k,k) \leftarrow \sum_{i' \text{ such that } i' \neq k} \max\{0, r(i',k)\}, \quad (2)$$

$$a(i,k) \leftarrow \min\left\{0, r(k,k) + \sum_{i' \text{ such that } i' \notin \{i,k\}} \max\{0, r(i',k)\}\right\} \quad (3)$$

,

$$c(i,k) \leftarrow r(i,k) + a(i,k). \quad (4)$$

The similarity matrix is constructed by negating the distances between items. These distances are commonly calculated by summing the squares of the differences between variables that make up the items. For example, consider that in a business simulation or experiential exercise, participants are asked to indicate on a five-point scale their preferences with respect to a tax rate, a fee, an interest rate, a quantity limit, and a price limit. If the preferences of five participants are as given in Table 1, then the off-diagonal elements of the similarity matrix, calculated by negating the sum of the squares of the differences among the participants, are as given in Table 2. Thus, for the similarity between Alice and Bob, the sum of the squares of the differences is $(3 - 4)^2 + (4 - 3)^2 + (3 - 5)^2 + (2 - 1)^2 + (1 - 1)^2 = 7$, so the similarity value is -7. The diagonal elements of the matrix are chosen from the off-diagonal elements. The algorithm will converge around a smaller number of clusters if a smaller value is chosen, and vice versa. In the example, -22 is the smallest of the off-diagonal values, so placing this value in every one of the diagonal elements directs the algorithm to converge onto a small number of clusters. If the diagonal elements are not of identical value, the algorithm will converge preferentially on clusters around higher-value items.

The next step of the algorithm is to construct an availability matrix with all elements set to zero. Equation 1 is then applied to compute the responsibility matrix (Table 3). Thus, the responsibility of Bob (column) to Alice (row) is -1, which is the similarity of Bob to Alice (-7) minus the maximum of the remaining similarities of Alice's row (-6).

Table 1: Preferences of Five Participants

Participant	Tax Rate	Fee	Interest Rate	Quantity Limit	Price Limit
Alice	3	4	3	2	1
Bob	4	3	5	1	1
Cary	3	5	3	3	3
Doug	2	1	3	3	2
Edna	1	1	3	2	3

Table 2: Similarity Matrix

Participant	Alice	Bob	Cary	Doug	Edna
Alice	-22	-7	-6	-12	-17
Bob	-7	-22	-17	-17	-22
Cary	-6	-17	-22	-18	-21
Doug	-12	-17	-18	-22	-3
Edna	-17	-22	-21	-3	-22

Table 3: Responsibility Matrix

Participant	Alice	Bob	Cary	Doug	Edna
Alice	-16	-1	1	-6	-11
Bob	10	-15	-10	-10	-15
Cary	11	-11	-16	-12	-15
Doug	-9	-14	-15	-19	9
Edna	-14	-19	-18	14	-19

Next, Equations 2 and 3 are used to update the diagonal and off-diagonal elements, respectively, of the availability matrix (Table 4). Thus, the self-availability of Alice is the sum of the positive responsibilities of Alice's column excluding Alice's self-responsibility (10 + 11 + 0 + 0 = 21), and the availability of Bob (column) to Alice (row) is Bob's self-responsibility plus the sum of the remaining positive responsibilities of Bob's column excluding the responsibility of Bob to Alice (-15 + 0 + 0 + 0 = -15).

Table 4: Availability Matrix

Participant	Alice	Bob	Cary	Doug	Edna
Alice	21	-15	-16	-5	-10
Bob	-5	0	-15	-5	-10
Cary	-6	-15	1	-5	-10
Doug	0	-15	-15	14	-19
Edna	0	-15	-15	-19	9

Applying Equation 4 gives rise to the criterion matrix (Table 5). Thus, the criterion value of Bob (column) to Alice (row) is the sum of the responsibility and availability of Bob to Alice (-1 + -15 = -16). The column with the highest criterion value for each row identifies the exemplar for the item of that row. Rows that share the same exemplar are in the same cluster. Exemplar criterion values are bolded in Table 5. Two clusters appear. Alice, Bob, and Cary constitute one cluster; Doug and Edna constitute the second. Repeated applications of Equations 1 through 4 do not change the solution, so in this case the first solution is the convergent solution.

Table 5: Criterion Matrix

Participant	Alice	Bob	Cary	Doug	Edna
Alice	**5**	-16	-15	-11	-21
Bob	**5**	-15	-25	-15	-25
Cary	**5**	-26	-15	-17	-25
Doug	-9	-29	-30	**-5**	-10
Edna	-14	-34	-33	**-5**	-10

To identify outliers, the signs of all items in the similarity matrix are reversed, so all elements of the similarity matrix of Table 2 become positive. To minimize the number of outliers identified, the diagonal elements should be changed from 22, now the highest of the off-diagonal values, to 3, now the lowest. After three iterations, the algorithm converges by identifying a different outlier for each participant, which means that no participant is an outlier.

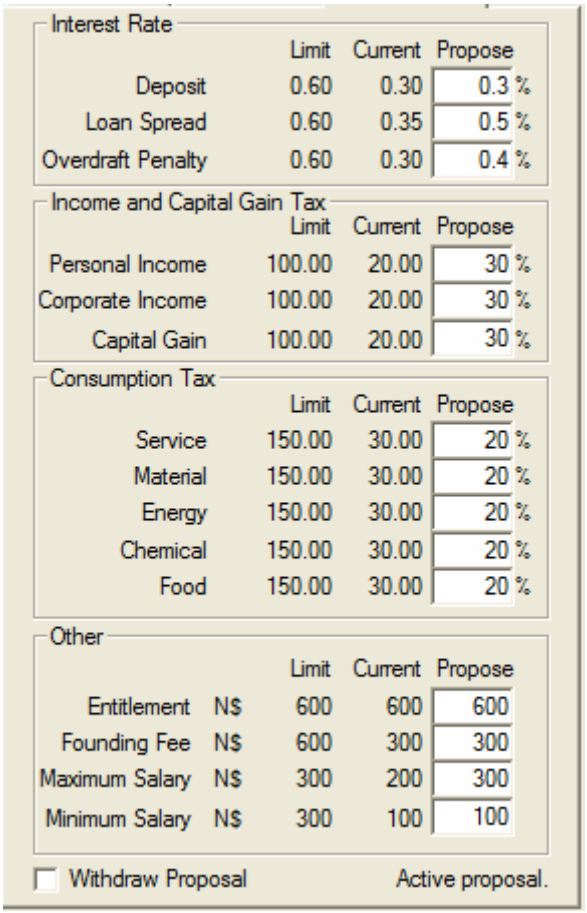
To assure convergence, Frey and Dueck (2007) suggests adding a tiny bit of random noise to the similarity matrix and damping updates of the availability and responsibility matrices by 50%. Thus, if r'_t and a'_t are the undamped updates of the responsibility and availability matrices, respectively, at iteration t, then the damped updates (r_t and a_t) are computed as follows: $r_t = .5r_{t-1} + .5r'_t$ and $a_t = .5a_{t-1} + .5a'_t$.

In this example, the variables of each item range over the same five-point scale, so they did not require standardization. In the general case when different variables are scaled differently, standardization, by converting all variables to ratios of their maximum values, by normalizing all variables to the mean of 0 and the standard deviation of 1, or by another method, will be necessary.

APPLICATION

To assist in simulating government, the affinity propagation algorithm was incorporated into GEO, an Internet-based (Pillutla, 2003), computer-assisted (Crookall, Martin, Saunders, & Coote, 1986) international business gaming simulation of a global economy. The gaming simulation tracks individual participants. Each registers and logs in with a unique user name and password. Participants are assigned to nations, and allowed to propose domestic and foreign policies for their nations. The dialog box for participants to submit domestic-policy proposals is given in Figure 1. Participants enter their proposed value for each policy variable into its associated edit box. *Limit* refers to the highest value that can be proposed for each variable, and *Current* refers to the current value of each variable.

Figure 1:
Domestic Policy Proposal Dialog Box



Submitted proposals are standardized by converting them to percentages of their maximum values. Results are presented to participants through a list box, as shown in Figure 2. On the list box, *Proposer* is the user name of the participant who submitted each proposal. *Change* is the root-mean-squared percentage difference between each proposal and the current variable settings. *To Center* is the root-mean-square percentage difference between each proposal and the central proposal, the one most similar to all participants as determined by the sum of that participant's similarity to all participants. *To Examplar* is the root-mean-square percentage difference between each proposal and its exemplar. The remaining columns of the list box give the raw variable values of each proposal.

As with many Microsoft Windows program, participants can sort the items of the list box by clicking the left mouse button on the header of the sorting column desired. Figure 3 is the same list box sorted on the Exemplar column. It shows that the proposals fall into three clusters, and that the exemplary proposals are Allison's, Gaby's, and Jung's. Participants may use the information as a basis for casting votes for the proposal they prefer. The administrator has the option of simulating either direct democracy by executing the most favored proposal directly or representation democracy by giving the author of the most favored proposal the authority to set variable values, which then might differ from what the author had proposed. As yet, no data is available on how participants respond to either form of democracy.

CONCLUSION

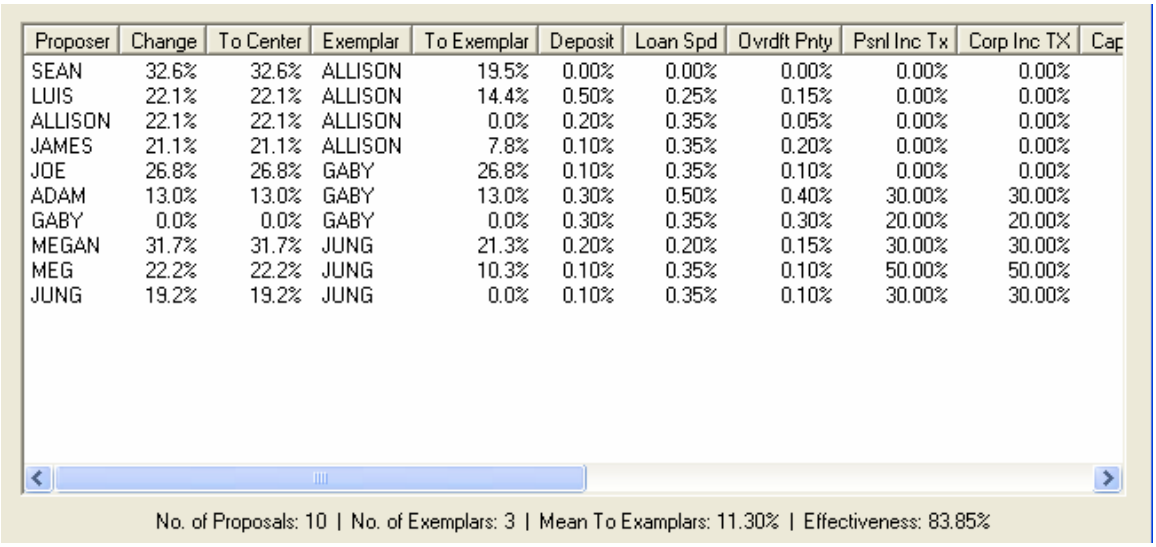
Affinity propagation is a low error, high speed, flexible, and an easy-to-code clustering algorithm that identifies clusters, exemplars, and outliers. The algorithm has been incorporated into a business simulation to organize policy proposals submitted by participants, for simulating direct and representational democracy. Besides its usefulness in simulating government, the algorithm also might be applied to assist in forming teams. Thus, teams might be formed from representative members of each cluster or the clusters themselves might constitute the teams.

Figure 2: List Box of Proposals

Proposer	Change	To Center	Exemplar	To Exemplar	Deposit	Loan Spd	Ovrdft Pnty	Psnl Inc Tx	Corp Inc TX	Cap
SEAN	32.6%	32.6%	ALLISON	19.5%	0.00%	0.00%	0.00%	0.00%	0.00%	
MEGAN	31.7%	31.7%	JUNG	21.3%	0.20%	0.20%	0.15%	30.00%	30.00%	
JOE	26.8%	26.8%	GABY	26.8%	0.10%	0.35%	0.10%	0.00%	0.00%	
MEG	22.2%	22.2%	JUNG	10.3%	0.10%	0.35%	0.10%	50.00%	50.00%	
LUIS	22.1%	22.1%	ALLISON	14.4%	0.50%	0.25%	0.15%	0.00%	0.00%	
ALLISON	22.1%	22.1%	ALLISON	0.0%	0.20%	0.35%	0.05%	0.00%	0.00%	
JAMES	21.1%	21.1%	ALLISON	7.8%	0.10%	0.35%	0.20%	0.00%	0.00%	
JUNG	19.2%	19.2%	JUNG	0.0%	0.10%	0.35%	0.10%	30.00%	30.00%	
ADAM	13.0%	13.0%	GABY	13.0%	0.30%	0.50%	0.40%	30.00%	30.00%	
GABY	0.0%	0.0%	GABY	0.0%	0.30%	0.35%	0.30%	20.00%	20.00%	

No. of Proposals: 10 | No. of Exemplars: 3 | Mean To Examplars: 11.30% | Effectiveness: 83.85%

Figure 3: Exemplar-Sorted List Box of Proposals



Proposer	Change	To Center	Exemplar	To Exemplar	Deposit	Loan Spd	Ovrdft Pnty	Psnl Inc Tx	Corp Inc TX	Cap
SEAN	32.6%	32.6%	ALLISON	19.5%	0.00%	0.00%	0.00%	0.00%	0.00%	
LUIS	22.1%	22.1%	ALLISON	14.4%	0.50%	0.25%	0.15%	0.00%	0.00%	
ALLISON	22.1%	22.1%	ALLISON	0.0%	0.20%	0.35%	0.05%	0.00%	0.00%	
JAMES	21.1%	21.1%	ALLISON	7.8%	0.10%	0.35%	0.20%	0.00%	0.00%	
JOE	26.8%	26.8%	GABY	26.8%	0.10%	0.35%	0.10%	0.00%	0.00%	
ADAM	13.0%	13.0%	GABY	13.0%	0.30%	0.50%	0.40%	30.00%	30.00%	
GABY	0.0%	0.0%	GABY	0.0%	0.30%	0.35%	0.30%	20.00%	20.00%	
MEGAN	31.7%	31.7%	JUNG	21.3%	0.20%	0.20%	0.15%	30.00%	30.00%	
MEG	22.2%	22.2%	JUNG	10.3%	0.10%	0.35%	0.10%	50.00%	50.00%	
JUNG	19.2%	19.2%	JUNG	0.0%	0.10%	0.35%	0.10%	30.00%	30.00%	

A limitation of affinity propagation is that the algorithm is too tedious to be done by hand. The algorithm must be encoded into a computer program to be of practical use. Given the ease with which this can be done and the widespread availability of computers, this limitation should not be especially restrictive.

REFERENCES

Aldenderfer, M. S., & Blashfield, R. K. (1984). *Cluster analysis*. Newbury Park, CA: Sage.

Burns, A. C., & Banasiewicz, A. (1994). The intellectual structure of ABSEL: A bibliometeic [sic] study of author cocitations over time. *Developments in Business Simulation & Experiential Exercises*, *21*, 7-12. Reprinted in the *Bernie Keys Library*, 8th ed. [http://www.absel.org].

Chang, J., Choi, K., Moon, K. K., Chan, P., Chan, T. L., & To, C. (2005). Teaching practices: A cluster analysis of students in Hong Kong. *Developments in Business Simulations and Experiential Learning*, *32*, 381-388. Reprinted in the *Bernie Keys Library*, 8th ed. [http://www.absel.org].

Chang, J., Choi, K., Ng, K., Chu, A., Hsia, P., & Kwan, R. (2005). Teaching practices: A cluster analysis of teaching staff in Hong Kong. *Developments in Business Simulations and Experiential Learning*, *32*, 381-388. Reprinted in the *Bernie Keys Library*, 8th ed. [http://www.absel.org].

Crookall, D., Martin, A., Saunders, D., & Coote, A. (1986). Human and computer involvement in simulation. *Simulation & Gaming*, *17*, 345-375.

Frey, B. J., & Dueck, D. (2007). Clustering by passing messages between data points. *Science*, *315*, 972-976.

GEO. Thavikulwat, P. (2008). http://pages.towson.edu/precha/geo. (Department of Management, Towson University, 8000 York Road, Towson, MD, USA).

Hand, D., Mannila, H., & Smyth, P. (2001). *Principles of data mining*. Cambridge, MA: MIT Press.

Johnson, R. A., & Wichern, D. W. (2007). *Applied multivariate statistical analysis.* Upper Saddle River, NJ: Pearson Prentice Hall.

Mézard, M. (2007). Where are the exemplars? *Science*, *315*, 949-951.

Overby, J. D. (1994). Cluster analyses of American universities' business core curricula structures utilized to satisfy fifteen curriculum areas. *Developments in Business Simulation & Experiential Exercises*, *21*, 109-112. Reprinted in the *Bernie Keys Library*, 8th ed. [http://www.absel.org].

Pillutla, S. (2003). Creating a Web-based simulation gaming exercise using PERL and JavaScript. *Simulation & Gaming*, *34*, 112-130.

Sackson, M. (1990). The use of cluster analysis for business game performance analysis. *Developments in Business Simulation & Experiential Exercises*, *17*, 150-154. Reprinted in the *Bernie Keys Library*, 8th ed. [http://www.absel.org].

Sokal, R. R., & Sneath, P. H. A. (1963). *Principles of numerical taxonomy*. San Francisco: Freeman.

Zalatan, K. A., & Mayer, D. F. (1999). Developing a learning culture: Assessing changes in student performance and perception. *Developments in Business Simulation & Experiential Learning*, *26*, 45-51. Reprinted in the *Bernie Keys Library*, 8th ed. [http://www.absel.org].