British Journal of Mathematical and Statistical Psychology (2019), 72, 155–182

The British Psychological Society

www.wileyonlinelibrary.com

Expert Tutorial

Affinity propagation: An exemplar-based tool for clustering in psychological research

Michael J. Brusco*[1], Douglas Steinley[2], Jordan Stevens[2] and J. Dennis Cradit[1]

[1]Department of Business Analytics, Information Systems, and Supply Chain, Florida State University, Tallahassee, Florida, USA

[2]Department of Psychological Sciences, University of Missouri, Columbia, Missouri, USA

Affinity propagation is a message-passing-based clustering procedure that has received widespread attention in domains such as biological science, physics, and computer science. However, its implementation in psychology and related areas of social science is comparatively scant. In this paper, we describe the basic principles of affinity propagation, its relationship to other clustering problems, and the types of data for which it can be used for cluster analysis. More importantly, we identify the strengths and weaknesses of affinity propagation as a clustering tool in general and highlight potential opportunities for its use in psychological research. Numerical examples are provided to illustrate the method.

1. Introduction

Traditional hierarchical (Johnson, 1967; Ward, 1963) and non-hierarchical (Rao, 1971; Steinhaus, 1956) clustering procedures continue to play a pivotal role in the psychological sciences (see Steinley, 2006; for a review). Nevertheless, there are a variety of alternative clustering approaches that are gaining popularity, either as competitors for the traditional methods or as special-purpose techniques for specific applications. Many of these alternative approaches fall within the class of model-based clustering methods (McLachlan & Peel, 2000; Steinley & Brusco, 2011a), and include methods such as latent class analysis, latent profile analysis, and growth mixture models. However, there are also alternative methods outside the spectrum of model-based clustering that are flexible and useful, but are not especially well known in the psychological community. For example, spectral clustering (von Luxburg, 2007; Shi & Malik, 2000) and other matrix factorization approaches (Lee & Seung, 1999) have been available for at least two decades, yet are only recently gaining acceptance in psychology (Brusco, Doreian, & Steinley, 2016; Chen, Li, Liu, Xu, & Ying, 2017). Another promising method, which is the focus of this paper, is *affinity propagation*.

*Corresponding should be addressed to Michael J. Brusco, Department of Business Analytics, Information Systems, and Supply Chain, College of Business, Florida State University, Tallahassee, FL 32306-1110, USA (email: mbrusco@business.fsu.edu).

DOI:10.1111/bmsp.12136

Introduced by Frey and Dueck (2007),[1] affinity propagation is a non-hierarchical clustering method that uses a message-passing algorithm grounded in the principles of loopy belief propagation (Pearl, 1988). The algorithm takes, as input, similarities between objects and user-specified *preferences* that define the suitability of each object for serving as an *exemplar* (i.e., a representative object for a cluster). In most applications, there is no *a priori* reason for preferring one object over another as an exemplar and, accordingly, the preference value for each object is set to the same user-selected constant. However, if desired, the psychological researcher has the flexibility to vary the preferences to make it more (or less) likely that specific objects are selected as exemplars for their clusters. Upon successful optimization, the affinity propagation algorithm returns, as output, a set of exemplars and the assignment of remaining objects to the most appropriate exemplar. Affinity propagation has received widespread attention in the physical sciences. According to the Web of Science, the Frey and Dueck (2007) article has been cited more than 1,600 times since its publication. Scientific applications of affinity propagation include the analysis of fMRI data (Zhang, Li, Chen, & Fang, 2011), the exploration of protein interaction graphs (Apeltsin, Morris, Babbitt, & Ferrin, 2011; Vlasblom & Wodak, 2009; Woźniak, Tiuryn, & Dutkowski, 2010), the identification of subsets of genes that have comparable transcriptional profiles (Chang, 2012), the clustering of gene expression time series data (Kiddle *et al.*, 2010), the analysis of radio and X-ray observations from black-hole binary systems (Gallo, Miller, & Fender, 2012), and the clustering of atmospheric aerosol particles (Salimi *et al.*, 2014).

Despite its popularity in the physical sciences, applications of affinity propagation in the social sciences are relatively scant. A full text search of the string 'affinity propagation' in the PsycArticles database uncovered two hits (Brasel, 2010; Steinley, Brusco, & Hubert, 2016). Brasel (2010) indicates that affinity propagation clustering was used, but does not describe the method or reference the Frey and Dueck (2007) software program. The Steinley *et al.* (2016) article contains a reference that has 'affinity propagation' in the title, but does not discuss the method. Clearly, affinity propagation is virtually unknown to the field of psychological research. This is unfortunate, given that affinity propagation possesses clear benefits for applications to psychological research, which include: (1) the ability to handle different types of proximity data (including asymmetric data); (2) the automatic selection of the number of clusters; and (3) computational efficiency for large data sets. Moreover, the software to support such applications is freely available and easily implemented.

Psychological researchers have successfully applied traditional methods such as *K*-means and Ward's method for clustering objects (patients, examinees, respondents, etc.) based on a set of measurements on continuous variables. However, these methods are less effective, or even inapplicable, for certain types of data. For example, Brusco, Shireman, and Steinley (2017) recently found that *K*-median clustering was better than *K*-means at recovering true cluster structure when objects are measured on binary variables. In light of the similarities (discussed in Section 2.3) between *K*-median clustering and affinity propagation, the latter method is also likely to perform well for such data. Moreover, affinity propagation can be used to cluster objects measured on non-distance-based similarity measures that arise in psychology, such as those obtained from stimulus recognition or paired comparison tasks (see, for example, Hubert, Arabie, & Meulman, 2001).

[1] Software for affinity propagation is available at http://www.psi.toronto.edu/index.php?q=affinity%20propagation. We use the Matlab implementation (apcluster.m, downloaded 10 August 2017) for the analyses completed in this paper; however, the code is also available through the R software system and via Windows and Linux libraries.

One of the most vexing problems with traditional clustering procedures such as *K*-means or Ward's method is the choice of the appropriate number of clusters (Milligan & Cooper, 1985; Steinley & Brusco, 2011b). Affinity propagation does not explicitly require the number of clusters to be specified in advance. However, the number of clusters is an output of the algorithm that is implicitly determined by the user-specified preferences discussed previously.

The affinity propagation algorithm possesses the computational efficiency necessary to tackle the large data sets encountered in the physical sciences. As the number of objects has become larger and larger in some areas of psychology (such as the number of patients in psychopathology research), the availability of an algorithm that rapidly obtains a solution is likely to provide a significant benefit.

Our goal in this paper is to introduce affinity propagation to the psychological research community, clarify its relationship to other clustering methods, identify its strengths and weaknesses, and demonstrate how it might be advantageous for some clustering applications. Section 2 provides a formal description of the terminology and underlying optimization problem associated with affinity propagation. This section also clarifies how affinity propagation is related to other clustering methods such as *K*-means and *K*-median clustering. Section 3 provides a description of the affinity propagation algorithm, as well as a brief synopsis of alternative solution approaches for the problem. Section 4 highlights the advantages and disadvantages of affinity propagation, as well their implications for using the method for clustering applications in psychological research. Section 5 provides a numerical illustration of affinity propagation software using confusion data pertaining to the recognition of lipread consonants (Manning & Shofner, 1991). The paper concludes in Section 6 with a brief summary.

2. Underlying optimization model

2.1. Model formulation

The affinity propagation algorithm requires, as input, an $n \times n$ 'similarity' matrix, $\mathbf{S} = [s_{ij}]$, where n is the number of objects to be clustered and i and j are object indices. Larger values of s_{ij} indicate a greater degree of similarity between i and j than smaller values of s_{ij}. It is assumed that $s_{ii} = 0$ (for all $1 \leq i \leq n$) in the formulation. In more traditional *K*-means settings, we would be concerned with data within a vector space configured as dissimilarities. As an example, consider the problem of partitioning n objects into K clusters based on their distances as measured on a collection of v metric variables, and let the $n \times n$ matrix $\mathbf{D} = [d_{ij}]$ contain the squared Euclidean distances (measured across the v variables) between each pair of objects. The matrix $\mathbf{D}$ is a 'dissimilarity' matrix because larger d_{ij} values indicate less similarity (or greater dissimilarity) between the two objects in any given pair of objects (i and j). Therefore, one option for the input to the affinity propagation algorithm would be to set $\mathbf{S} = -\mathbf{D}$, so as to ensure a similarity rather than dissimilarity orientation of the data. There are other procedures for producing a similarity matrix from the dissimilarity matrix, such as subtracting each element in $\mathbf{D}$ from the largest element in $\mathbf{D}$.

In addition to the reliance on similarities, another important aspect of affinity propagation is the central role of *exemplars*. Analogous to the psychological usage of exemplars in object categorization (Nosofsky, 2011), affinity propagation defines one of the objects within each cluster to serve as an exemplar representing the focal point for measuring similarity within that cluster. For example, when applying affinity propagation to a set of respondents in a psychological setting, each of the resulting K clusters will have

an individual respondent (one of the n objects) serving as the exemplar for their cluster. The total similarity for each cluster is measured as the sum of the similarities between each respondent in that cluster and the respondent serving as the exemplar. The overall similarity measure for the K-cluster partition, termed *dpsim* (or 'data-point similarity': the sum of the similarities between non-exemplar data points and their exemplars) in the affinity propagation program, is the sum of the K total cluster similarity measures. The user-specified *preferences* (p_j, for $1 \leq j \leq n$, which may also be specified as the $n \times 1$ vector $\mathbf{p}$) define the suitability of each object for serving as an exemplar. The sum of the preferences for selected exemplars is referred to as *expref* (or sum of the 'exemplar preferences'). Succinctly, affinity propagation seeks to find a partition that maximizes *netsim* = *dpsim* + *expref* (where *netsim* is the 'net similarity').

A formal statement of the optimization problem that underlies affinity propagation begins with the definition of two user-provided inputs: the similarity matrix, $\mathbf{S}$; and the preference vector, $\mathbf{p}$. There are two sets of decision variables associated with the optimization problem: $y_j = 1$ if object j is selected as an exemplar and 0 otherwise, for $1 \leq j \leq n$; and $x_{ij} = 1$ if object i is assigned to the cluster for which object j serves as an exemplar and 0 otherwise, for $1 \leq i \leq n$ and $1 \leq j \leq n$. The integer linear programming formulation of the problem can then be stated as follows:

$$\text{maximize } netsim = \sum_{i=1}^{n}\sum_{j=1}^{n} s_{ij}x_{ij} + \sum_{j=1}^{n} p_j y_j, \tag{1}$$

subject to

$$\sum_{j=1}^{n} x_{ij}, \text{ for all } 1 \leq i \leq n; \tag{2}$$

$$x_{ij} \leq y_j, \text{ for all} 1 \leq i \leq n \text{ and } 1 \leq j \leq n; \tag{3}$$

$$x_{jj} = y_j, \text{ for all } 1 \leq j \leq n. \tag{4}$$

$$x_{ij} \in \{0, 1\} \text{ for all } 1 \leq i \leq n \text{ and } 1 \leq j \leq n; \tag{5}$$

$$y_j \in \{0, 1\} \text{ for all } 1 \leq j \leq n. \tag{6}$$

The objective function (equation (1)) of the optimization problem is *netsim*, and the first and second terms on the right-hand side of the equals sign are *dpsim* and *expref*, respectively. Constraint set (2) guarantees that each object is assigned to exactly one exemplar and, therefore, exactly one cluster. Constraint set (3) ensures that an object is not assigned to an object that is not selected as an exemplar. Constraint set (4) is incorporated in the affinity propagation algorithm to ensure that, if an object is selected as an exemplar, then that object must be assigned to the cluster for which it serves as the exemplar. Finally, constraint sets (5) and (6) enforce the binary restrictions on the x_{ij} and y_j variables, respectively.

2.2. Preferences and similarity data issues

It is important to clarify some issues pertaining to the preference vector (**p**), as well the input similarity data (**S**) themselves. The importance of **p** cannot be understated, as it is the key determinant of the number of clusters that will be selected. Two general guidelines for the specification of **p** are as follows. First, a uniform preference structure should be used in the absence of any information regarding differing merit for each object serving as an exemplar. That is, each element of **p** is the same constant, and in those cases, secondly, the preferences are commonly specified as the median of the similarity measures (s^M), that is, $p_j = s^M$ for $1 \leq j \leq n$. These guidelines are based on the original work of Frey and Dueck (2007), but are upheld by their implementation in applications (Kiddle *et al.*, 2010; Mukherjee & Hill, 2011). At the same time, some authors have explored alternatives for specifying **p**. For example, Wang, Qin, and Zhang (2010) used a particle swarm method to search for an optimal set of preferences.

Brusco and Köhn (2009) showed that the number of clusters could vary drastically (from just a few clusters to several hundred) as the preference values were changed. More recently, Brusco and Steinley (2015) conducted a simulation-based evaluation of different multiples of the median for setting the preference vector. The 8,640 data sets in their study were produced by manipulating seven design features and using a mixture-model generation process designed by Steinley and Henson (2005). Recovery of true cluster structure in the data sets was measured using the adjusted Rand index (ARI: Hubert & Arabie, 1985). Cluster recovery performance was very sensitive to the selection of the preferences. When using the median similarity measure, s^M, to set the preferences, the true number of clusters was generally overestimated. Increasing the preferences to $4 \times s^M$ led to an average improvement of 75 per cent in the ARI, producing the best recovery of true cluster structure.

A key assumption for affinity propagation is that the input data are in similarity form (i.e., larger elements of **S** indicate greater similarity). Nevertheless, there is considerable flexibility in the nature of **S**, including the ability to accommodate an asymmetric matrix, as we demonstrate in Section 5. The input matrix may also consist of either positive or negative elements.[2] However, it is helpful to clarify the interpretation of self-similarities (s_{ii}) and the preference vector, depending on whether the matrix elements of **S** are negative or positive.

If the elements of **S** consist of negative distances (a common condition in applications in the literature) and the median of **S** is the preference value for each object, then increasing the number of clusters (K) will worsen (or decrease) the value of *expref* because the preferences are negative and the goal is to maximize. Therefore, in the case of a negative **S**, increasing K penalizes the model for becoming more complex via the *expref* component. When the elements of **S** are non-negative, there is an interesting implication for *expref*. Assuming the preference for each object is taken as the median of **S**, increasing the number of clusters (K) will improve (or increase) the value of *expref* because the preferences are positive and the goal is to maximize. So, in this context, increasing K does not penalize the model for becoming more complex via the *expref* component, but actually improves this component of the objective function.

[2] The algorithm will also run when **S** consists of a mix of positive and negative elements. However, careful consideration of 'similarity' is required in such instances. For example, a correlation matrix may have both positive and negative elements, but it could be argued that larger absolute values of the elements (rather than larger raw values) are indicative of greater similarity.

2.3. Relationship to other clustering problems

Brusco and Steinley (2015) have recently shown that the optimization problem (equations 1–6) that serves as the foundation for affinity propagation is a maximization version of the simple plant location problem (SPLP), which is a venerable problem in the facility location literature (Balinski, 1965; Erlenkotter, 1978; Kuehn & Hamburger, 1963). In the plant location context, the *dpsim* term in the objective function is analogous to the variable costs of assigning demand centres to plants, whereas the *expref* component mirrors the fixed costs of opening plants. Brusco and Steinley (2015) provide an extensive coverage of the SPLP and discuss alternative procedures for solving the problem.

The SPLP and, therefore, affinity propagation are closely related to the *K*-median (also commonly known as the *p*-median) clustering problem, which is also rooted in location theory (Hakimi, 1964, 1965; Maranzana, 1964). The *K*-median problem differs from the SPLP and, therefore, the affinity propagation algorithm in that the following additional constraint is imposed to restrict the number of clusters to a fixed value, *K*:

$$\sum_{j=1}^{n} y_j = K. \tag{7}$$

The *K*-median problem has a long history in the clustering literature (Kaufman & Rousseeuw, 2005; Klastorin, 1985; Köhn, Steinley, & Brusco, 2010; Mulvey & Crowder, 1979; Rao, 1971; Vinod, 1969), and its similarity to affinity propagation has been widely recognized (Brusco & Köhn, 2008a,b, 2009; Dueck, 2009; Frey & Dueck, 2007, 2008; Kiddle *et al.*, 2010). Many of these more recent articles have undertaken comparative studies of affinity propagation and *K*-median clustering, and the results of these comparisons are often contradictory with respect to the relative performances of the methods (for an in-depth discussion, see Brusco & Steinley, 2015). The two methods can be implemented to optimize the same *netsim* objective function; however, comparisons are complicated by the fact that *K*-median clustering assumes that the number of clusters is prespecified, whereas affinity propagation does not. Given that affinity propagation is generally faster than most *K*-median methods, a possible strategy is to run affinity propagation first to choose the number of clusters (*K*), and then to run a *K*-median method with the chosen value of *K* to see if *netsim* can be improved. Brusco and Köhn (2008a) showed that affinity propagation commonly failed to produce the optimal value of *netsim* for some small data sets, but that running a *K*-median heuristic using the number of clusters obtained by affinity propagation produced a better (and often globally optimal) value for *netsim*. A subsequent study by Brusco and Köhn (2009) showed that, when using the value of *K* selected by the affinity propagation algorithm, a simulated annealing heuristic for *K*-median clustering produced improved *netsim* values for a variety of test problems that spanned a wide range of values of *n* and *K* (see also Brusco & Steinley, 2015).

It is also beneficial to contrast affinity propagation with *K*-means clustering, which is arguably the most popular clustering method in both research and practice (Steinley, 2006). One immediate distinction is that, like *K*-median clustering, *K*-means assumes that the number of clusters is prespecified, whereas affinity propagation does not. A second crucial difference is that the cluster centres in *K*-means clustering are *virtual* in the sense that they do not typically correspond to actual objects. In other words, the centre of a cluster in *K*-means clustering corresponds to the variable averages for the objects assigned to that cluster, whereas the centre in an affinity propagation clustering is a specific object

(i.e., an exemplar). Third, *K*-means is typically applied to dissimilarity data based on squared Euclidean distances between objects computed based on a set of metric variables. Although affinity propagation is also commonly applied to the negative squared Euclidean distances of such data, it can also accommodate more general similarity data, including asymmetric similarity data.

Finally, we contrast affinity propagation with spectral clustering (von Luxburg, 2007). Like affinity propagation, this method is broadly applicable to general similarity data and software programs are readily available (Karatzoglou, Smola, & Hornik, 2010); however, spectral clustering is not an exemplar-based method. Although there are several variants of spectral clustering, the method commonly begins with the construction of a binary similarity matrix from the raw data, which can be accomplished in a variety of different ways. This similarity matrix is subsequently transformed into a Laplacian matrix. The first *K* eigenvectors of the Laplacian matrix are identified, and a *K*-cluster partition is obtained (typically using *K*-means clustering). Because they can both be applied to the same types of data, Frey and Dueck (2006) and Dueck (2009) have reported limited comparisons of spectral clustering and affinity propagation. Their results have generally favoured affinity propagation with respect to several criteria, including computation time, solution interpretability, and *netsim* values. However, to the best of our knowledge, a rigorous comparison of the two methods in terms of true cluster structure recovery has not been undertaken, and would present a formidable, yet important, challenge for future research.

3. Algorithm and alternative methods

3.1. Affinity propagation algorithm

The affinity propagation algorithm developed by Frey and Dueck (2007) produces a solution to the problem posed by equations (1–6) using a message-passing algorithm, based on 'loopy belief propagation', a concept from the computer science literature and Bayesian belief networks (Berrou, Glavieux, & Thitimajshima, 1993; Mézard, Parisi, & Zecchina, 2002; Pearl, 1988). Belief propagation refers to an algorithm, operating on a network graph, that calculates marginal distributions for each node, conditional on the other nodes in the network. In effect, the algorithm determines a final distribution by allowing each node to 'listen' to information passed from neighbouring nodes. Adjacent nodes exchange information (i.e., 'pass messages') by telling each other how to update current probabilities ('beliefs'), based on prior information, conditional probabilities, and evidence supplied to the graphical network. When applied to graphs with tree structures, the computation is very efficient. When cycles (loops) are present, as would be the case with networks analysed by affinity propagation, the 'loopy' belief propagation algorithm is less efficient, and iterates until sufficient convergence is achieved (Ihler, Fisher, & Willsky, 2005). The affinity propagation problem is modelled as a factor graph whereby the data points (s_{ij}) are linked to cluster assignment variables, which are linked to a net similarity function (see Dueck, 2009; Figure 3.1, p. 32). Belief propagation is then performed on this factor graph via the passing of two types of messages: *responsibilities* and *availabilities*. Dueck (2009, Section 3.2.2) describes the dynamics of the algorithm.

The primary components of the affinity propagation algorithm are two matrices that contain responsibilities and availabilities, respectively. The elements of the responsibilities matrix, $\mathbf{R} = [r_{ij}]$, each correspond to a measure of the potential suitability for object j to serve as an exemplar for object i. Likewise, the elements of

the availabilities matrix, $\mathbf{A} = [a_{ij}]$, are measures of the appropriateness of object i selecting object j as its exemplar. The *dampening parameter*, λ, is also critical because it controls the degree of smoothing for the elements of **R** and **A**. User inputs to the algorithm are the similarity matrix, **S**, and preference vector, **p**. The elements of the preference vector are placed on the main diagonal of **S** by the algorithm. The steps of the algorithm are as follows:

Step 0. *Initialize the responsibilities*, *availabilities*, *cluster assignment vector* (ξ), and *dampening parameter*. Set $r_{ij} = 0$ and $a_{ij} = 0$, for all $1 \leq i, j \leq n$; and $\xi_i = 0$, for all $1 \leq i \leq n$. Set the dampening parameter (we use the default of $\lambda = 0.9$).

Step 1. *Iterate message passing until convergence*. The default setting for convergence is 100 consecutive iterations with no change in the exemplars or assignments (Dueck, 2009, p. 48).

Step 1a. Set

$$r_{ij} = \lambda r_{ij} + (1 - \lambda)\left[s_{ij} = \max_{j':j' \neq j}\{s_{ij'} + a_{ij'}\}\right].$$

Step 1b. Set

$$a_{ij} = \lambda a_{ij} + (1 - \lambda)\begin{cases} \sum_{i':i' \neq i} \max\{0, r_{i'j}\}, & \text{for} \quad j = i, \\ \min\left\{0, r_{jj} + \sum_{i':i' \notin \{i,j\}} \max\{0, r_{i'j}\}\right\}, & \text{for} \quad j \neq i. \end{cases}$$

Step 2. Return cluster memberships. For each $1 \leq i \leq n$, find $j' = \text{argmax}_j\{a_{ij} + r_{ij}\}$ and set $\xi_i = j'$ as the exemplar to which object i is assigned.

To illustrate the rationale for the algorithm, it is helpful, for the moment, to assume $\lambda = 0$. In Step 0, **R** and **A** are initialized to zero. Therefore, when Step 1a is reached in the first iteration of Step 1, the responsibility measure sent from object i to candidate exemplar object j is initialized as the similarity between object i and j (s_{ij}), minus the similarity between i and the exemplar competitor of j ($j' \neq j$) that has the greatest similarity to object i. These responsibilities are then passed to Step 1b, where the availability of object i to serve as an exemplar for object j is equal to the self-responsibility for object j, plus the sum of all positive responsibilities that object j is sent from objects other than i. The availabilities are then passed back to Step 1a to update the responsibilities. As the algorithm iterates, some objects effectively become assigned to other exemplar objects and their availabilities will become negative. As this occurs, the merit of these objects serving as an exemplar diminishes.

To avoid overreacting to the messages passed, the dampening parameter is set to $\lambda = 0.9$, such that most of the contribution to the updated values of r_{ij} and a_{ij} is from the previous values, not the new messages received. Dueck (2009) reports an analysis of the effect of the dampening parameter on algorithm performance. Parameter values on the interval $0 \leq \lambda \leq 0.4$ commonly result in a significant degree of oscillation and problems with convergence. A dampening parameter of at least $\lambda \geq 0.5$ is generally recommended. As the parameter is increased towards 1, convergence is slower, but more stable (less oscillation). Dueck (2009, p. 47) noted that 'setting the dampening factor λ to 0.9 has been sufficient in almost all cases to ensure convergence'.

The output of the algorithm includes the scalars *dpsim*, *expref*, and *netsim*, as well as the $n \times 1$ vector, ξ, which contains the exemplar to which each object is assigned. In

most instances, this algorithm converges rapidly (<1 min on current hardware platforms for problems of size $n = 2{,}000$ or smaller). However, there is the potential that, in some unusual cases, the algorithm might not converge (see Dueck, 2009, pp. 45–48; for a discussion of reasons for failure to converge). A small example is provided in the Appendix to illustrate the mechanics of the affinity propagation algorithm.

3.2. Alternatives to the affinity propagation algorithm

Upon convergence, the affinity propagation algorithm described in the previous subsection produces a feasible solution to the problem posed by equations (1–6); however, the resulting solution is not guaranteed to be a *globally optimal solution* to the problem (i.e., a solution that produces the maximum value of *netsim*). This fact, well documented in the literature (Brusco & Köhn, 2008a, 2009; Brusco & Steinley, 2015), indicates that the affinity propagation algorithm can be considered as a *heuristic algorithm* (i.e., an algorithm that does not guarantee a global optimum) for the underlying mathematical problem that it seeks to optimize. This is in contrast to *exact algorithms* that ensure a global optimum has been achieved.[3] Although exact algorithms are preferable because of their guarantee of global optimality, they are sometimes computationally infeasible for large problems.

Brusco and Steinley (2015) have shown that the problem posed by equations (1–6) can be solved using exact methods in some instances. One exact approach is to apply commercial mathematical programming software to solve the integer linear program directly. Alternatively, a combination of heuristics, Lagrangian relaxation, and branch-and-bound can be assembled to construct an exact method that is guaranteed to produce globally optimal solutions. Using this latter approach, Brusco and Steinley (2015) solved a variety of problems from the literature, ranging in size from $n = 89$ to $n = 1{,}400$ objects. It is difficult to pinpoint precisely the conditions where an exact algorithm will be feasible. Computational feasibility does depend on n to a large extent, but it also depends on the nature of the similarity data. For example, there is evidence that problems for which the similarity measures are based on Euclidean distance are easier to solve exactly than those based on non-Euclidean measures (Galvão, 2004).

Brusco and Steinley (2015) also developed a heuristic alternative to the affinity propagation algorithm. In a comparative study, they found their heuristic to be competitive with the current affinity propagation algorithm. Nevertheless, given the vast experience with the current algorithm in the literature, it is the procedure that we would recommend for applications at the present time.

4. Merits of affinity propagation for psychological research

Previous sections of the paper have raised several issues that could either promote or inhibit the use of affinity propagation in practice. Table 1 provides a summary of the advantages and disadvantages of affinity propagation. In this section, we expound on these advantages and disadvantages and their implications for the potential merits of affinity propagation in psychological research.

[3] The terms *exact* and *heuristic* algorithms are common in some disciplines (e.g., operational research) to distinguish, respectively, algorithms that are guaranteed (upon successful convergence) to produce a globally optimal solution from those that afford no such guarantee.

Table 1. Advantages and disadvantages of affinity propagation

Advantages	Disadvantages
1. The cluster centres are actual objects (exemplars)	1. There is a need to prespecify the preferences for objects to serve as exemplars
2. The method can be applied to very general similarity data, including asymmetric similarity measures	2. Although relatively rare, the algorithm can fail to converge
3. There is no need to specify the number of clusters (*K*) in advance	3. The algorithm often converges to suboptimal solutions, particularly for large similarity matrices
4. The algorithm is fast and can be applied to large similarity matrices	4. There is a potential for alternative global optima

4.1. Provision of an exemplar for each cluster

One of the key advantages of affinity propagation is that it directly provides, as part of the solution process, an exemplar as the centre for each cluster. By contrast, the cluster centres for traditional methods such as Ward's method or *K*-means clustering are averages of variables that typically do not correspond to a specific object. Thus, a key difference between a *K*-means and affinity propagation analysis is the distinction between a prototype (a *virtual* cluster centre) and an exemplar (see Köhn *et al.*, 2010; for discussion). It is certainly possible to (indirectly) obtain exemplars from a *K*-means clustering solution (or any other non-exemplar-based clustering method). For example, the observation nearest the virtual cluster centroid could be selected as the exemplar for each cluster. An even better approach, which was used by Dueck (2009) in his evaluation of competitors for affinity propagation, is to apply a one-median clustering algorithm to obtain the exemplar for each cluster. Although these types of refinement processes for selecting exemplars might be suitable for some applications, the results reported by Dueck (2009) suggested that they often produce inferior *netsim* values relative to affinity propagation.

In light of the benefits of direct selection of exemplars, affinity propagation provides a propitious approach to cluster analysis in any psychological context where the researcher desires specific exemplars to serve as representative objects for their clusters. For example, in the case of clustering organizational members based on job affect or work ties in a social network (see, for example, Totterdell, Wall, Holman, Diamond, & Epitropaki, 2004), affinity propagation could be used to select *K* individuals to serve as exemplars for further, more detailed study.

Psychological research applications of affinity propagation are not necessarily restricted to the clustering of individuals, such as organizational members, patients, or students. The method could also be used to cluster a set of stimulus items or constructs, with the goal of identifying exemplars that are more broadly representative of other constructs/items. For example, Mount, Barrick, Scullen, and Rounds (2005) applied complete-linkage hierarchical clustering (Johnson, 1967; McQuitty, 1960) to similarity data among 11 items corresponding to the big five personality traits (Digman, 1990) and the big six vocational interest types (Holland, 1985). One potential advantage of affinity propagation over hierarchical clustering for this task is the direct provision of a personality trait or vocational interest as the exemplar for each cluster. That is, affinity propagation would directly provide a trait or interest that serves as an anchor for each cluster, whereas

complete-linkage clustering did not provide such representative traits or interests. Similarly, affinity propagation would be a viable alternative for comparable applications where hierarchical clustering has been used, such as for the partitioning of mediation tactics (McLaughlin, Carnevale, & Lim, 1991), harassment questionnaire items (Wasti & Cortina, 2002), indigenous emotions (Scollon, Diener, Oishi, & Biswas-Diener, 2004), or vocational interests (Armstrong & Vogel, 2009).

4.2. Application to general similarity data

A second advantage of affinity propagation is that it can be applied to very general similarity data. That is, a critical advantage of the affinity propagation algorithm is that similarities need not be based on squared Euclidean distances. In fact, affinity propagation can be applied to data that are ordinal, non-metric, and discontinuous in nature (Dueck, 2009). Therefore, in addition to the standard Euclidean-distance-based data associated with *K*-means clustering and Ward's method, affinity propagation can also be used for similarity measures derived from paired comparison or sorting tasks. In fact, **S** need not even be symmetric. We demonstrate an affinity propagation solution to asymmetric data in our example in Section 5. As a further example, in one of the illustrations reported by Frey and Dueck (2007), **S** corresponded to air travel routeing distances between cities, which were not symmetric (i.e., $s_{ij} \neq s_{ji}$).

In light of its flexibility, it is clear that affinity propagation can be quite valuable in psychological research situations where the available similarity or dissimilarity data are inherently asymmetric. There are a variety of possible examples of such data conditions, which include: confusion matrix data obtained from a stimulus recognition task; social network ties between individuals in an organization; and co-citation matrices for psychological journals. Whereas traditional clustering methods such as *K*-means or Ward's method would generally require some pre-processing to put these data in usable form, they can be tackled directly using affinity propagation.

4.3. Automatic selection of the number of clusters

A third advantage of affinity propagation is that it does not require the number of clusters (K) to be specified in advance. This distinguishing feature of affinity propagation is noteworthy in light of the fact that the selection of the appropriate number of clusters for a *K*-means cluster analysis, or a closely related hierarchical approach such as Ward's method, is often a formidable task (see Steinley & Brusco, 2011b).

It should be noted, however, that there is a counterargument to this advantage. Although it is true that the number of clusters does not need to be prespecified, this decision is replaced by a need to choose the preference vector, **p**. This is the first disadvantage noted in Table 1. From the standpoint of the psychological researcher, the critical question is whether it is more reasonable to set a preference vector and let the affinity propagation algorithm determine the number of clusters, or to obtain solutions for different numbers of clusters and use some alternative criterion to select the best value of K. If the latter option is preferred, then it might be advantageous to make use of an alternative version of the affinity propagation algorithm that does allow users to select a specific value of K.[4] . However, this version can be time consuming because of the need to run the algorithm multiple times for

[4] Matlab software for this alternative version (`apclusterK.m`) is available at http://www.psi.toronto.edu/index.php?q=affinity%20propagation.

different preference vectors. Once solutions have been obtained for several values of K, a selection criterion, such as the silhouette index (Kaufman & Rousseeuw, 2005; Köhn *et al.*, 2010), could be used to choose the number of clusters.

4.4. Computational issues

A fourth advantage of affinity propagation is that, in most instances, it is fast and scalable for problems where the number of objects to be clustered (n) is well into the thousands. For some of these large problem instances, computational results reported by Brusco and Köhn (2009) and Brusco and Steinley (2015) have shown that the affinity propagation algorithm is commonly more efficient than alternative heuristic methods implemented using the same hardware and software platform. Another computational consideration that might be relevant to some large-scale psychological applications is the evidence that affinity propagation can be useful for classifying very large, yet sparse, similarity matrices, which might arise in a social network context. The affinity propagation website also offers a version of an algorithm customized for sparse input data (Matlab implementation: `apclusterSparse.m`).

The benefits of computational efficiency notwithstanding, there are some computational disadvantages that should be recognized by psychological researchers considering affinity propagation. The second, third, and fourth limitations identified in Table 1 are of a computational nature, pertaining to the potential for failure of the algorithm to converge, the possible suboptimality of the solution obtained, and the possibility of alternative global optima. The failure of the affinity propagation algorithm to converge appears to be relatively rare, and does not constitute a major deterrent for psychological applications. The issue of suboptimality is the most serious of the computational issues. Brusco and Köhn (2008a) showed that the affinity propagation algorithm often failed to obtain a globally optimal solution even for relatively small problem instances from the literature. Similar findings have been reported by Brusco and Köhn (2009) and Brusco and Steinley (2015). Nevertheless, it should be acknowledged that suboptimality issues have commonly plagued traditional methods such as Ward's method and K-means clustering (Brusco & Cradit, 2001; Brusco & Steinley, 2007; Steinley, 2003, 2006), yet such methods have been gainfully employed for several decades.

The fourth limitation, which is related to the third but is less severe, is that, even if the affinity propagation algorithm converges to a globally optimal partition, there is the potential for multiple global optima for some similarity matrices, and such a result would not be readily apparent to users of the algorithm. The problem of multiple global optima for affinity propagation is much less severe than it is for monotone invariant clustering methods, such as minimum diameter partitioning (see Brusco & Steinley, 2014), but perhaps slightly more common than in K-means clustering. The potential for multiple optima is also affected by the nature of the similarity data. For example, when **S** consists of negative Euclidean distances, the likelihood of multiple global optima is small. However, when **S** contains integer similarity values within a narrow range, the possibility of multiple global optima is greater.

4.5. A process diagram for implementing affinity propagation

Figure 1 presents a process flow diagram for using affinity propagation in psychological research. The first step is to obtain the $n \times n$ similarity matrix that serves as the primary input to the software program. As noted in Section 2.1, for the case of measurements for n

respondents on v metric variables, the similarity matrix is usually constructed as either the negative Euclidean or negative squared Euclidean distances between the n respondents on the v variables. If the v variables are non-metric, then any of a host of similarity measures based on matching could be used. For sorting tasks, the elements of the similarity matrix might be the percentage of times that a pair of objects were placed in the same group (see, for example, Köhn *et al.*, 2010).

The second step of the process is to specify the preference vector. A default specification is the median of the similarity measures. However, in a recent simulation study, Brusco and Steinley (2015) found that using four times the median resulted in better recovery of the known number of clusters, as well as the known cluster memberships. In the absence of any *a priori* information for specifying the preferences, we recommend, as a starting point, setting the preference values somewhere between the median and four times the median of the similarity measures. Just as exploration of different numbers of clusters is common in *K*-means and *K*-median clustering, it is typically worthwhile to run affinity propagation for different preference vectors (e.g., one, two, three, and four times the median similarity measure).

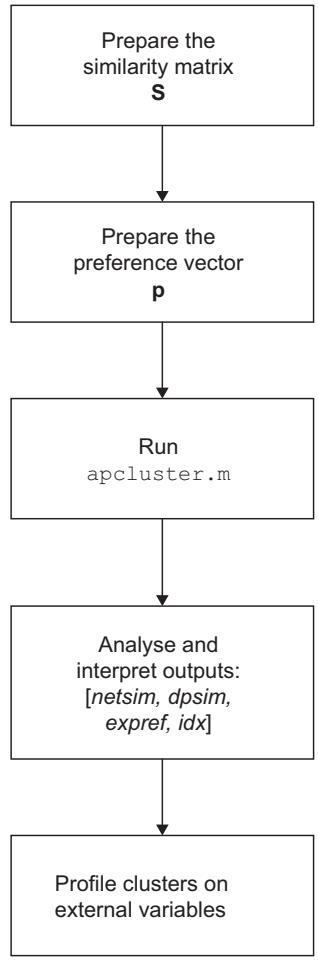

Figure 1. A flow diagram of the affinity propagation clustering process.

The third step is to run the algorithm and obtain the results. The fourth step is to analyse and interpret the clustering solution based on the exemplar of each cluster, as well as the relative sizes of each clusters. In applications where variables exogenous to the clustering process are available, a useful fifth step is to profile the clusters based on those variables.

5. Numerical illustrations

5.1. Data

To illustrate the principles of affinity propagation, we utilize a small data set pertaining to an important area of application in experimental psychology. More specifically, we use a published confusion matrix associated with the recognition of lipread consonants (Manning & Shofner, 1991, p. 596). The use of previously published data enables researchers to independently verify our results using affinity propagation software, which can help acclimate new users to the program.

The elements (s_{ij}) of the similarity matrix (**S**) represent the proportion of responses given as consonant j when consonant i was the stimulus that was actually presented. Because the percentage of mistaken responses of consonant j ($j \neq i$) when consonant i was presented is typically different than the percentage of mistaken responses of consonant i when consonant j was presented, **S** is an asymmetric matrix. Fortunately, this does not present a problem for affinity propagation. In fact, the asymmetry of **S** allows us to compare and contrast the results of three distinct applications of affinity propagation to the lipread consonant data:

1. Direct application of affinity propagation to **S** (percentage of mistaken responses of consonant j when consonant i was presented) will result in the selection of *response* exemplars. In other words, the consonants selected as exemplars are those that are most critical with respect to their role as responses to presented consonants.
2. Application of affinity propagation to $\mathbf{S}'$ (the percentage of mistaken responses of consonant i when consonant j was presented) will result in the selection of *stimulus* exemplars. That is, the consonants selected as exemplars are those that are most critical with respect to their role as stimuli for the consonant responses.
3. Application of affinity propagation to the symmetric matrix formed by $\mathbf{S} + \mathbf{S}'$ will result in the selection of exemplars based on the total confusion between pairs of consonants. In other words, the consonants selected as exemplars are those that are most critical with respect to confusability among pairs of consonants when both directions of confusion are considered.

5.2. Clustering S

The median of the 21 × 21 similarity (confusion) matrix, **S**, published by Manning and Shofner (1991, p. 596) is $s^M = .003$ and, therefore, we applied the algorithm using **S** and $p_j = .003$ for $1 \leq j \leq 21$. The solution obtained by the algorithm consisted of $K = 9$ consonants that served as cluster exemplars: {d, f, g, m, n, p, s, w, y}. These consonants, which are *response* exemplars, are positioned as the first nine columns of the similarity matrix as displayed in Table 2. Each consonant is assigned to the response exemplar for which it has maximum similarity, and the appropriate matrix elements are highlighted

using bold italic in Table 2. Summation of the bold italicized values in Table 2 yields *dpsim* = 2.515, which corresponds to the total sum of the similarities between each consonant and its exemplar. Given the identification of $K = 9$ clusters and the median similarity of $s^M = .003$, the value of *expref* is $9 \times .003 = .027$. The value of *netsim* is $2.515 + .027 = 2.542$. The *K*-median procedure developed by Brusco and Köhn (2008a) was also applied to the similarity data under the assumption of $K = 9$ clusters, and obtained the same solution as the affinity propagation algorithm.

The consonants {c, d, t, z} were assigned to the cluster with consonant 'd' serving as the response exemplar. The remaining assignments are as follows: consonants {f, v} assigned to the cluster with consonant 'f' as the response exemplar; consonants {g, j} assigned to the cluster with consonant 'g' as the response exemplar; consonants {k, m} assigned to the cluster with consonant 'm' as the response exemplar; consonants {*l*, n, r} assigned to the cluster with consonant 'n' as the response exemplar; consonants {b, p} assigned to the cluster with consonant 'p' as the response exemplar; consonants {s, x} assigned to the cluster with consonant 's' as the response exemplar; consonants {h, q, w} assigned to the cluster with consonant 'w' as the response exemplar; and consonant {y} is a singleton cluster and serves as the final response exemplar. It is also apparent that consonant 'y' could be moved to the {*l*, n, r} cluster, thus producing an eight-cluster solution with the same *netsim* value.

5.3. Clustering S′

The median of $\mathbf{S}'$ is also $s^M = .003$, and the application of the affinity propagation algorithm using $\mathbf{S}'$ and $p_j = .003$ (for $1 \leq j \leq 21$) produced a solution with $K = 6$ consonants that served as cluster exemplars: {b, j, k, q, x, z}. It is interesting to note that there is no overlap at all between these six *stimulus* exemplars and the nine response exemplars obtained for **S**. The six stimulus exemplars are positioned as the first six rows of the similarity matrix as displayed in Table 3. Each consonant is assigned to the stimulus exemplar for which it has maximum similarity, and the appropriate matrix elements are highlighted using bold italic in Table 3. Summation of the bold italicized values in Table 3 yields *dpsim* = 2.556, which corresponds to the total sum of the similarities between each consonant and its exemplar. Using $K = 6$ clusters and the median similarity of $s^M = .003$, the value of *expref* is computed as $6 \times .003 = .018$. The value of *netsim* is $2.556 + .018 = 2.574$. The *K*-median procedure developed by Brusco and Köhn (2008a) was also applied to the similarity data under the assumption of $K = 6$ clusters, and obtained the same solution.

The consonants {c, d, t, z} were assigned to the cluster with consonant 'z' serving as the stimulus exemplar. The remaining assignments are as follows: consonants {q, r, v, w} assigned to the cluster with consonant 'q' as the stimulus exemplar; consonants {g, j} assigned to the cluster with consonant 'j' as the stimulus exemplar; consonants {f, k, *l*, m, n} assigned to the cluster with consonant 'k' as the stimulus exemplar; consonants {h, n, s, x} assigned to the cluster with consonant 'x' as the stimulus exemplar; and consonants {b, p} assigned to the cluster with consonant 'b' as the stimulus exemplar.

5.4. Clustering S + S′

The median of $\mathbf{S} + \mathbf{S}'$ is $s^M = .007$, and the algorithm was applied using $\mathbf{S} + \mathbf{S}'$ and $p_j = .007$ (for $1 \leq j \leq 21$). The algorithm failed to converge (even after increasing two of the iteration limits by an order of magnitude). The algorithm returned a $K = 7$ cluster solution that yielded a value of *netsim* = 3.861; however, the *K*-median method described

Table 2. The $K = 9$ partition of **S** for the lipread consonant confusion data. The first nine columns pertain to the clusters and their exemplars. The elements in bold italics correspond to the assignments of consonants to the exemplars

	d	f	g	m	n	p	s	w	y	b	c	h	j	k	*l*	q	r	t	v	x	z
c	***.286***	.014	.020	.022	.007	.010	.007	.005	.000	.006	–	.001	.002	.002	.014	.000	.001	.180	.002	.000	.143
d	–	.014	.071	.025	.006	.007	.003	.002	.000	.002	.127	.003	.003	.000	.011	.003	.002	.204	.012	.000	.099
t	***.325***	.004	.100	.031	.019	.005	.009	.001	.000	.002	.141	.003	.012	.013	.012	.005	.002	–	.013	.002	.094
z	***.275***	.007	.024	.015	.010	.009	.005	.002	.000	.011	.245	.000	.005	.001	.014	.002	.004	.171	.014	.000	–
f	.001	–	.002	.009	.003	.000	.000	.003	.000	.000	.002	.000	.000	.000	.008	.002	.000	.002	.006	.000	.000
v	.004	***.044***	.001	.004	.000	.005	.001	.001	.001	.007	.002	.000	.003	.000	.006	.000	.003	.002	–	.001	.004
g	.005	.003	–	.006	.003	.001	.001	.000	.000	.000	.001	.025	.020	.000	.005	.005	.002	.006	.000	.000	.004
j	.004	.019	***.042***	.020	.000	.000	.001	.002	.001	.000	.000	.007	–	.000	.008	.001	.004	.003	.000	.000	.000
k	.005	.046	.002	***.244***	.070	.003	.007	.002	.007	.002	.001	.012	.007	–	.172	.000	.007	.006	.004	.004	.004
m	.002	.010	.001	–	.016	.005	.002	.009	.000	.003	.000	.001	.003	.000	.011	.001	.001	.005	.006	.001	.000
l	.002	.023	.001	.069	***.187***	.002	.034	.002	.001	.002	.003	.030	.007	.018	–	.000	.011	.000	.008	.009	.001
n	.012	.032	.015	.081	–	.006	.021	.000	.000	.004	.011	.108	.016	.046	.134	.001	.012	.004	.017	.017	.002
r	.001	.005	.002	.004	***.006***	.005	.000	.000	.001	.003	.001	.006	.000	.000	.019	.025	–	.001	.000	.003	.000
b	.003	.008	.001	.029	.003	***.440***	.001	.002	.000	–	.001	.001	.001	.000	.010	.001	.002	.002	.007	.001	.002
p	.006	.009	.005	.074	.015	–	.003	.002	.000	.424	.002	.002	.000	.002	.005	.000	.000	.005	.014	.000	.001
s	.000	.007	.001	.006	.075	.000	–	.001	.004	.000	.003	.135	.000	.012	.018	.000	.004	.001	.000	.074	.000
x	.002	.005	.003	.013	.127	.001	***.582***	.002	.003	.000	.002	.117	.007	.005	.013	.000	.001	.001	.003	–	.000
h	.001	.003	.010	.007	.004	.000	.024	***.050***	.000	.002	.000	–	.050	.000	.006	.002	.000	.002	.001	.010	.000
q	.004	.003	.013	.009	.014	.007	.001	***.034***	.001	.003	.007	.001	.008	.004	.022	–	.025	.004	.029	.000	.000
w	.001	.000	.001	.000	.003	.001	.000	–	.000	.000	.001	.001	.000	.000	.005	.001	.002	.002	.000	.000	.001
y	.000	.001	.000	.002	.003	.001	.000	.001	–	.000	.000	.000	.002	.000	.004	.000	.000	.000	.001	.000	.000

Table 3. The $K = 6$ partition of $\mathbf{S}'$ for the lipread consonant confusion data. The first six rows pertain to the clusters and their stimulus exemplars. The elements in bold italics correspond to the assignments of consonants to the exemplars

	b	p	g	j	f	k	*l*	m	y	q	r	v	w	h	n	s	x	c	d	t	z
b	–	***.440***	.001	.001	.008	.000	.010	.029	.000	.001	.002	.007	.002	.001	.003	.001	.001	.001	.003	.002	.002
j	.000	.000	***.042***	–	.019	.000	.008	.020	.001	.001	.004	.000	.002	.007	.000	.001	.000	.000	.004	.003	.000
k	.002	.003	.002	.007	***.046***	–	***.172***	***.244***	***.007***	.000	.007	.004	.002	.012	.070	.007	.004	.001	.005	.006	.004
q	.003	.007	.013	.008	.003	.004	.022	.009	.001	–	***.025***	***.029***	***.034***	.001	.014	.001	.000	.007	.004	.004	.000
x	.000	.001	.003	.007	.005	.005	.013	.013	.003	.000	.001	.003	.002	***.117***	***.127***	***.582***	–	.002	.002	.001	.000
z	.011	.009	.024	.005	.007	.001	.014	.015	.000	.002	.004	.014	.002	.000	.010	.005	.000	***.245***	***.275***	***.171***	–
c	.006	.010	.020	.002	.014	.002	.014	.022	.000	.000	.001	.002	.005	.001	.007	.007	.000	–	.286	.180	.143
d	.002	.007	.071	.003	.014	.000	.011	.025	.000	.003	.002	.012	.002	.003	.006	.003	.000	.127	–	.204	.099
f	.000	.000	.002	.000	–	.000	.008	.009	.000	.002	.000	.006	.003	.000	.003	.000	.000	.002	.001	.002	.000
g	.000	.001	–	.020	.003	.000	.005	.006	.000	.005	.002	.000	.000	.025	.003	.001	.000	.001	.005	.006	.004
h	.002	.000	.010	.050	.003	.000	.006	.007	.000	.002	.000	.001	.050	–	.004	.024	.010	.000	.001	.002	.000
l	.002	.002	.001	.007	.023	.018	–	.069	.001	.000	.011	.008	.002	.030	.187	.034	.009	.003	.002	.000	.001
m	.003	.005	.001	.003	.010	.000	.011	–	.000	.001	.001	.006	.009	.001	.016	.002	.001	.000	.002	.005	.000
n	.004	.006	.015	.016	.032	.046	.134	.081	.000	.001	.012	.017	.000	.108	–	.021	.017	.011	.012	.004	.002
p	.424	–	.005	.000	.009	.002	.005	.074	.000	.000	.000	.014	.002	.002	.015	.003	.000	.002	.006	.005	.001
r	.003	.005	.002	.000	.005	.000	.019	.004	.001	.025	–	.000	.000	.006	.006	.000	.003	.001	.001	.001	.000
s	.000	.000	.001	.000	.007	.012	.018	.006	.004	.000	.004	.000	.001	.135	.075	–	.074	.003	.000	.001	.000
t	.002	.005	.100	.012	.004	.013	.012	.031	.000	.005	.002	.013	.001	.003	.019	.009	.002	.141	.325	–	.094
v	.007	.005	.001	.003	.044	.000	.006	.004	.001	.000	.003	–	.001	.000	.000	.001	.001	.002	.004	.002	.004
w	.000	.001	.001	.000	.000	.000	.005	.000	.000	.001	.002	.000	–	.001	.003	.000	.000	.001	.001	.002	.001
y	.000	.001	.000	.002	.001	.000	.004	.002	–	.000	.000	.001	.001	.000	.003	.000	.000	.000	.000	.000	.000

by Brusco and Köhn (2008a) produced a $K = 7$ cluster solution with a slightly better value of *netsim* = 3.865 (*dpsim* = 3.816 plus *expref* = 7 × .007 = .049) and it is this solution that we interpret here. The $K = 7$ consonants that served as cluster exemplars in the solution are: {d, f, *l*, (b or p), q, s, y}. The seven exemplars are positioned as the first seven columns of the similarity matrix as displayed in Table 4. Each consonant is assigned to the exemplar to which it has maximum similarity, and the appropriate matrix elements are highlighted using bold italic in Table 4.

The consonants {c, d, g, t, z} were assigned to the cluster with consonant 'd' serving as the exemplar. The remaining assignments are as follows: consonants {f, j, v} assigned to the cluster with consonant 'f' as the exemplar; consonants {q, r, w} assigned to the cluster with consonant 'q' as the exemplar; consonants {k, *l*, m, n} assigned to the cluster with consonant '*l*' as the exemplar; consonants {h, s, x} assigned to the cluster with consonant 's' as the exemplar; consonants {b, p} assigned to the cluster with consonants 'b' or 'p' as the exemplar; and consonant {y} is a singleton cluster that serves as the final exemplar.

5.5. Comparing the partitions for S, S′, and S + S′

Table 5 facilitates a comparison of the partitions obtained for **S**, **S**′, and **S** + **S**′. For each cluster of each partition, Table 5 reports the cluster exemplar and the consonants assigned to the cluster. The orderings of the clusters were established to make comparisons across partitions as straightforward as possible. One immediate observation from Table 2 is the finer granularity of the nine-cluster partition of **S**. Six of the nine clusters in that partition have two consonants or fewer. By contrast, the partitions of **S**′ and **S** + **S**′ each have only two clusters with two or fewer consonants. Although the partitions for **S**′ and **S** + **S**′ were more similar to one another than they were to the partition for **S**, it is interesting to note that the exemplar consonants were more similar for **S** and **S** + **S**′. Four of the seven exemplars for **S** + **S**′ (d, f, s, y} were also response exemplars for **S**, one {q} was a stimulus exemplar for **S**′, one (*l*) was neither a stimulus nor response exemplar, and the remaining exemplar (b or p) could be selected from either the stimulus or response sets.

The partitions for **S**, **S**′, and **S** + **S**′ each contain the cluster {b, p}. For **S**, the *response* exemplar is 'p' because the percentage of mistaken responses of 'p' for stimulus 'b' was larger than the percentage of mistaken responses of 'b' for stimulus 'p'. However, based on the same argument, the *stimulus* exemplar is 'b' for the **S**′ partition. In the case of **S** + **S**′, where total confusion is considered, the choice between 'b' and 'p' for the exemplar is arbitrary.

The cluster {c, d, t, z}, consisting of consonants with the /i:/ sound when pronounced, emerged in the partitions for both **S** and **S**′[5]. The consonant 'd' is the response exemplar for this cluster in the partition of **S**, whereas the consonant 'z' is the stimulus exemplar for this cluster in the partition of **S**′. The explanation for this difference is based on the patterning of confusion proportions among the four consonants in the cluster. The consonant 'd' was mistakenly given as a response to the stimulus consonants 'c', 't', and 'z' more frequently than each of these consonants, respectively, were mistakenly given as a response when 'd' was presented; therefore, 'd' is the response exemplar in the cluster for the partition of **S**. Contrastingly, the stimulus constant 'z' was mistakenly identified by a

[5] The term /i:/ is linguistic notation for the 'long-e' sound. It is important to note that Manning and Shofner's (1991) experiment was conducted in the U.S., where the letter 'z' is pronounced 'zee' rather than 'zed'.

Table 4. The $K = 7$ partition of $\mathbf{S} + \mathbf{S}'$ for the lipread consonant confusion data. The first seven columns pertain to the clusters and their exemplars. The elements in bold italics correspond to the assignments of consonants to the exemplars

	d	f	*l*	p	q	s	y	b	c	g	h	j	k	m	n	r	t	v	w	x	z
c	***.413***	.016	.017	.012	.007	.010	.000	.007	–	.021	.001	.002	.003	.022	.018	.002	.321	.004	.006	.002	.388
d	–	.015	.013	.013	.007	.003	.000	.005	.413	.076	.004	.007	.005	.027	.018	.003	.529	.016	.003	.002	.374
g	***.076***	.005	.006	.006	.018	.002	.000	.001	.021	–	.035	.062	.002	.007	.018	.004	.106	.001	.001	.003	.028
t	***.529***	.006	.012	.010	.009	.010	.000	.004	.321	.106	.005	.015	.019	.036	.023	.003	–	.015	.003	.003	.265
z	***.374***	.007	.015	.010	.002	.005	.000	.013	.388	.028	.000	.005	.005	.015	.012	.004	.265	.018	.003	.000	–
f	.015	–	.031	.009	.005	.007	.001	.008	.016	.005	.003	.019	.046	.019	.035	.005	.006	.050	.003	.005	.007
j	.007	***.019***	.015	.000	.009	.001	.003	.001	.002	.062	.057	–	.007	.023	.016	.004	.015	.003	.002	.007	.005
v	.016	***.050***	.014	.019	.029	.001	.002	.014	.004	.001	.001	.003	.004	.010	.017	.003	.015	–	.001	.004	.018
k	.005	.046	***.190***	.005	.004	.019	.007	.002	.003	.002	.012	.007	–	.244	.116	.007	.019	.004	.002	.009	.005
l	.013	.031	–	.007	.022	.052	.005	.012	.017	.006	.036	.015	.190	.080	.321	.030	.012	.014	.007	.022	.015
m	.027	.019	***.080***	.079	.010	.008	.002	.032	.022	.007	.008	.023	.244	–	.097	.005	.036	.010	.009	.014	.015
n	.018	.035	***.321***	.021	.015	.096	.003	.007	.018	.018	.112	.016	.116	.097	–	.018	.023	.017	.003	.144	.012
b	.005	.008	.012	***.864***	.004	.001	.000	–	.007	.001	.003	.001	.002	.032	.007	.005	.004	.014	.002	.001	.013
p	.013	.009	.007	–	.007	.003	.001	.864	.012	.006	.002	.000	.005	.079	.021	.005	.010	.019	.003	.001	.010
q	.007	.005	.022	.007	–	.001	.001	.004	.007	.018	.003	.009	.004	.010	.015	.050	.009	.029	.035	.000	.002
r	.003	.005	.030	.005	***.050***	.004	.001	.005	.002	.004	.006	.004	.007	.005	.018	–	.003	.003	.002	.004	.004
w	.003	.003	.007	.003	***.035***	.001	.001	.002	.006	.001	.051	.002	.002	.009	.003	.002	.003	.001	–	.002	.003
h	.004	.003	.036	.002	.003	***.159***	.000	.003	.001	.035	–	.057	.012	.008	.112	.006	.005	.001	.051	.127	.000
s	.003	.007	.052	.003	.001	–	.004	.001	.010	.002	.159	.001	.019	.008	.096	.004	.010	.001	.001	.656	.005
x	.002	.005	.022	.001	.000	***.656***	.003	.001	.002	.003	.127	.007	.009	.014	.144	.004	.003	.004	.002	–	.000
y	.000	.001	.005	.001	.001	.004	–	.000	.000	.000	.000	.003	.007	.002	.003	.001	.000	.002	.001	.003	.000

Table 5. A summary of the partitions of **S**, **S**′, and **S** + **S**′ – exemplars and cluster memberships

	Partition of **S**	Partition of **S**′	Partition of **S** + **S**′
Cluster 1	Exemplar: p {b, p}	Exemplar: b {b, p}	Exemplar: b or p {b, p}
Cluster 2	Exemplar: d {c, d, t, z}	Exemplar: z {c, d, t, z}	Exemplar: d {c, d, g, t, z}
Cluster 3	Exemplar: g {g, j}	Exemplar: j {g, j}	Exemplar: f {f, j, v}
Cluster 4	Exemplar: m {k, m}	Exemplar: k {f, k, *l*, m, n}	Exemplar: *l* {k, *l*, m, n}
Cluster 5	Exemplar: w {h, q, w}	Exemplar: q {q, r, v, w}	Exemplar: q {q, r, w}
Cluster 6	Exemplar: s {s, x}	Exemplar: x {h, n, s, x}	Exemplar: s {h, s, x}
Cluster 7	Exemplar: n {*l*, n, r}		Exemplar: y {y}
Cluster 8	Exemplar: f {f, v}		
Cluster 9	Exemplar: y {y}		

response of ‘c’, ‘d’, or ‘t’ more frequently than each of these consonants, respectively, when presented as stimuli, was mistakenly identified as ‘z’; therefore, ‘z’ is the stimulus exemplar in the cluster for the partition of **S**′. For the partition of **S** + **S**′, the cluster {c, d, t, z} is augmented with {g} and ‘d’ is the exemplar.

There are three additional pairs of consonants that appear in the same cluster in each of the three partitions: {s, x}, {k, m}, and {q, w}. Despite the fact there is strong asymmetry (‘s’ is a mistaken response for ‘x’ far more frequently than the reverse is true), the letters ‘s’ and ‘x’ are the most confusable consonants and it is not surprising that they always appear in the same cluster. There is also strong asymmetry of confusion in the {k, m} pair. The letter ‘m’ was commonly a mistaken response for ‘k’, but ‘k’ was never a mistaken response for ‘m’. The {k, m} cluster for **S** with ‘m’ as the response cluster is indicative of this confusion pattern. However, in the **S**′ partition, {k, m} is augmented with {f, *l*, n} because these three consonants are also commonly mistaken responses for the stimulus exemplar ‘k’. Finally, the consonant pair {q, w} is joined by {h} in the partition for **S** because ‘h’ is a common mistaken call for the response stimulus ‘w’; and joined by {r, v} in the partition for **S**′ because response stimulus ‘q’ is frequently miscalled as ‘r’ and ‘v’.

6. Conclusions

Since its introduction to the broader scientific community in 2007 (Frey & Dueck, 2007), affinity propagation has garnered a significant amount of attention in the physical sciences; however, its presence in the social sciences is comparatively scarce. To the best of our knowledge, there are very few, if any, published applications of the affinity propagation algorithm in the leading psychological outlets. Our goals in this paper were: to introduce affinity propagation to the psychological research community; to compare and contrast affinity propagation with other clustering approaches, such as *K*-means and *K*-median clustering; to identify the advantages and disadvantages of affinity propagation

as a clustering method; to provide circumstances where affinity propagation might be a viable method in psychological research; and to present a numerical example that illustrates the affinity propagation algorithm.

One of the desirable aspects of affinity propagation is that software for implementing the method is readily and freely available. Moreover, because of the significant adoption of the method in the physical sciences, the software code has been well tested over the past several years. The number of software platforms for which affinity propagation is available has also expanded, and includes code written in R, Matlab, and other languages.

Although the rapid growth in the popularity and accessibility of affinity propagation since 2007 is noteworthy, this is not intended to imply that it should become the clustering method of choice for psychological research. Nevertheless, it does seem appropriate that psychological scientists should at least be aware of affinity propagation as a clustering tool, as well as its capabilities and limitations.

References

Apeltsin, L., Morris, J. H., Babbitt, P. C., & Ferrin, T. E. (2011). Improving the quality of protein similarity network clustering algorithms using the network edge weight distribution. *Bioinformatics*, *27*, 326–333. https://doi.org/10.1093/bioinformatics/btq655

Armstrong, P. I., & Vogel, D. L. (2009). Interpreting the interest-efficacy association from a RIASEC perspective. *Journal of Counseling Psychology*, *56*, 392–407. https://doi.org/10.1037/a0016407

Balinski, M. L. (1965). Integer programming: Methods, uses, computation. *Management Science*, *12*, 253–313. https://doi.org/10.1287/mnsc.12.3.253

Berrou, C., Glavieux, A., & Thitimajshima, P. (1993). Near Shannon limit error-correcting coding and decoding: Turbo-codes (1). *IEEE International Conference on Communications*, 1993. ICC 93 Geneva. Technical Program, Conference Record, Vol. 2. (pp. 1064–1070). Piscataway, NJ: IEEE.

Brasel, S. A. (2010). Embedded promotions in online services: How goal-relevance ambiguity shapes response and affect. *Journal of Experimental Psychology: Applied*, *16*, 263–280. https://doi.org/10.1037/a0020831

Brusco, M. J., & Cradit, J. D. (2001). A variable-selection heuristic for *k*-means clustering. *Psychometrika*, *66*, 249–270. https://doi.org/10.1007/BF02294838

Brusco, M. J., Doreian, P., & Steinley, D. (2016). Biclustering methods for one-mode asymmetric matrices. *Behavior Research Methods*, *48*, 487–502. https://doi.org/10.3758/s13428-015-0587-y

Brusco, M. J., & Köhn, H.-F. (2008a). Comment on 'Clustering by passing messages between data points'. *Science*, *319*, 726c. https://doi.org/10.1126/science.1150938

Brusco, M. J., & Köhn, H.-F. (2008b). Optimal partitioning of a data set based on the *p*-median model. *Psychometrika*, *73*, 89–105. https://doi.org/10.1007/s11336-007-9021-4

Brusco, M. J., & Köhn, H.-F. (2009). Exemplar-based clustering via simulated annealing. *Psychometrika*, *74*, 457–475. https://doi.org/10.1007/s11336-009-9115-2

Brusco, M. J., Shireman, E., & Steinley, D. (2017). A comparison of latent class, *K*-means, and *K*-median methods for clustering dichotomous data. *Psychological Methods*, *22*, 563–580. https://doi.org/10.1037/met0000095

Brusco, M. J., & Steinley, D. (2007). A comparison of heuristic procedures for minimum within-cluster sums of squares partitioning. *Psychometrika*, *72*, 583–600. https://doi.org/10.1007/s11336-007-9013-4

Brusco, M. J., & Steinley, D. (2014). Model selection for minimum-diameter partitioning. *British Journal of Mathematical and Statistical Psychology*, *67*, 471–495. https://doi.org/10.1111/bjmsp.12029

Brusco, M. J., & Steinley, D. (2015). Affinity propagation and uncapacitated facility location problems. *Journal of Classification*, *32*, 443–480. https://doi.org/10.1007/s00357-015-9187-x

Chang, J. T. (2012). Deriving transcriptional programs and functional processes from gene expression databases. *Bioinformatics*, *28*, 1122–1129. https://doi.org/10.1093/bioinformatics/bts112

Chen, Y., Li, X., Liu, J., Xu, G., & Ying, Z. (2017). Exploratory item classification via spectral graph clustering. *Applied Psychological Measurement*, *41*, 579–599. https://doi.org/10.1177/014662/617692977

Digman, J. M. (1990). Personality structure: Emergence of the five-factor model. *Annual Review of Psychology*, *41*, 417–440. https://doi.org/10.1146/annurev.ps.41.020190.002221

Dueck, D. (2009). *Affinity propagation: Clustering data by passing messages* (Unpublished doctoral dissertation). Graduate Department of Electrical and Computer Engineering, University of Toronto.

Erlenkotter, D. (1978). A dual procedure for uncapacitated facility location. *Operations Research*, *26*, 992–1009. https://doi.org/10.1287/opre.26.6.992

Frey, B., & Dueck, D. (2006). Mixture modeling by affinity propagation. In Y. Weiss, B. Schölkopf & J. Platt (Eds.), *Advances in neural information processing systems* (pp. 379–386). Cambridge, MA: MIT Press.

Frey, B., & Dueck, D. (2007). Clustering by passing messages between data points. *Science*, *315*, 972–976. https://doi.org/10.1126/science.1136800

Frey, B., & Dueck, D. (2008). Response to comment on 'Clustering by passing messages between data points'. *Science*, *319*, 726d. https://doi.org/10.1126/science.1151268

Gallo, E., Miller, B. P., & Fender, R. (2012). Assessing luminosity correlations via cluster analysis: Evidence for dual tracks in the radio/X-ray domain of black hole X-ray binaries. *Monthly Notices of the Royal Astronomical Society*, *423*, 590–599. https://doi.org/10.1111/j.1365-2966.2012.20899.x

Galvão, R. D. (2004). Uncapacitated facility location problems: Contributions. *Pesquisa Operacional*, *24*, 7–38. https://doi.org/10.1590/s0101074382004000100003

Hakimi, S. L. (1964). Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations Research*, *12*, 450–459. https://doi.org/10.1287/opre.12.3.450

Hakimi, S. L. (1965). Optimum distribution of switching centers in a communication network and some related graph theory problems. *Operations Research*, *13*, 462–475. https://doi.org/10.1287/opre.13.3.462

Holland, J. L. (1985). *Making vocational choices: A theory of careers*. Englewood Cliffs, NJ: Prentice Hall.

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification*, *2*, 193–218. https://doi.org/10.1007/BF01908075

Hubert, L., Arabie, P., & Meulman, J. (2001). *Combinatorial data analysis: Optimization by dynamic programming*. Philadelphia, PA: Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9780898718553

Ihler, A. T., Fisher, J. W. F., & Willsky, A. S. (2005). Loopy belief propagation: Convergence and effects of message errors. *Journal of Machine Learning Research*, *6*, 905–936.

Johnson, S. C. (1967). Hierarchical clustering schemes. *Psychometrika*, *32*, 241–254. https://doi.org/10.1007/BF02289588

Karatzoglou, A., Smola, A., & Hornik, K. (2010). *kernlab: Kernel-based Machine Learning Lab*. Retrieved from http://cran.r-project.org/web/packages/kernlab/index.html.

Kaufman, L., & Rousseeuw, P. J. (2005). *Finding groups in data: An introduction to cluster analysis* (2nd ed.). New York, NY: Wiley.

Kiddle, S. J., Windram, O. P. F., McHattie, S., Mead, A., Beynon, J., Buchanan-Wollaston, V., ... Mukherjee, S. (2010). Temporal clustering by affinity propagation reveals transcriptional modules in *Arabidopsis thaliana*. *Bioinformatics*, *26*, 355–362. https://doi.org/10.1093/bioinformatics/btp673

Klastorin, T. (1985). The *p*-median problem for cluster analysis: A comparative test using the mixture model approach. *Management Science*, *31*, 84–95. https://doi.org/10.1287/mnsc.31.1.84

Köhn, H.-F., Steinley, D., & Brusco, M. J. (2010). The *p*-median model as a tool for clustering psychological data. *Psychological Methods*, *15*, 87–95. https://doi.org/10.1037/a0018535

Kuehn, A. A., & Hamburger, M. J. (1963). A heuristic program for locating warehouses. *Management Science*, *9*, 643–666. https://doi.org/10.1287/mnsc.9.4.643

Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, *401*, 788–791. https://doi.org/10.1038/44565

Manning, S. K., & Shofner, E. (1991). Similarity ratings and confusability of lipread consonants compared with similarity ratings of auditory and orthographic stimuli. *American Journal of Psychology*, *104*, 587–604. https://doi.org/10.2307/1422942

Maranzana, F. E. (1964). On the location of supply points to minimize transportation costs. *Operations Research*, *12*, 138–139.

McLachlan, G. J., & Peel, D. (2000). *Finite mixture models*. New York, NY: Wiley. https://doi.org/10.1002/0471721182

McLaughlin, M. E., Carnevale, P., & Lim, R. G. (1991). Professional mediators' judgments of mediation tactics: Multidimensional scaling and cluster analyses. *Journal of Applied Psychology*, *76*, 465–472. https://doi.org/10.1037/0021-9010.76.3.465

McQuitty, L. L. (1960). Hierarchical linkage analysis for the isolation of types. *Educational and Psychological Measurement*, *20*, 55–67. https://doi.org/10.1177/001316446002000106

Mézard, M., Parisi, G., & Zecchina, R. (2002). Analytic and algorithmic solution of random satisfiability problems. *Science*, *297*, 812–815. https://doi.org/10.1126/science.1073287

Milligan, G. W., & Cooper, M. C. (1985). An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, *50*, 159–179. https://doi.org/10.1007/BF02294245

Mount, M. K., Barrick, M. R., Scullen, S. M., & Rounds, J. (2005). Higher-order dimensions of the big five personality traits and the big six vocational interest types. *Personnel Psychology*, *58*, 447–478. https://doi.org/10.1111/j.1744-6570.2005.00468.x

Mukherjee, S., & Hill, S. M. (2011). Network clustering: Probing biological heterogeneity by sparse graphical models. *Bioinformatics*, *27*, 994–1000. https://doi.org/10.1093/bioinformatics/btr070

Mulvey, J. M., & Crowder, H. P. (1979). Cluster analysis: An application of Lagrangian relaxation. *Management Science*, *25*, 329–340. https://doi.org/10.1287/mnsc.25.4.329

Nosofsky, R. M. (2011). The generalized context model: An exemplar model of classification. In E. M. Pothos & A. J. Wills (Eds.), *Formal approaches to categorization* (pp. 18–39). Cambridge, UK: Cambridge University Press. https://doi.org/10.1017/CBO9780511921322

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. San Mateo, CA: Morgan Kaufmann.

Rao, M. R. (1971). Cluster analysis and mathematical programming. *Journal of the American Statistical Association*, *66*, 622–626. https://doi.org/10.1080/01621459.1971.10482319

Salimi, F., Ristovski, Z., Mazaheri, M., Luiman, R., Crilley, L. R., He, C., Clifford, S., & Morawska, L. (2014). Assessment and application of clustering techniques to atmospheric particle number size distribution for the process of source apportionment. *Atmospheric Chemistry and Physics*, *14*, 11883–11892. https://doi.org/10.5194/acp-14-11883-2014

Scollon, C. N., Diener, E., Oishi, S., & Biswas-Diener, R. (2004). Emotions across cultures and methods. *Journal of Cross-Cultural Psychology*, *35*, 304–326. https://doi.org/10.1177/0022022104264124

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*, 888–905. https://doi.org/10.1109/34.868688

Steinhaus, H. (1956). Sur la division des corps matériels en parties [On the separation of objects into groups]. *Bulletin de l'Académie Polonaise des Sciences*, Classe III, IV(12), 801–804.

Steinley, D. (2003). Local optima in *K*-means clustering: What you don't know may hurt you. *Psychological Methods*, *8*, 294–304. https://doi.org/10.1037/1082-989X.8.3.294

Steinley, D. (2006). *K*-means clustering: A half-century synthesis. *British Journal of Mathematical and Statistical Psychology*, *59*, 1–34. https://doi.org/10.1348/000711005X48266

Steinley, D., & Brusco, M. J. (2011a). Evaluating mixture-modeling for clustering: Recommendations and cautions. *Psychological Methods*, *16*, 63–79. https://doi.org/10.1037/a0022673

Steinley, D., & Brusco, M. J. (2011b). Choosing the number of clusters in *K*-means clustering. *Psychological Methods*, *16*, 285–297. https://doi.org/10.1037/a0023346

Steinley, D., Brusco, M. J., & Hubert, L. (2016). The variance of the adjusted Rand index. *Psychological Methods*, *21*, 261–272. https://doi.org/10.1037/met0000049

Steinley, D., & Henson, R. (2005). OCLUS: An algorithmic method for generating clusters with known overlap. *Journal of Classification*, *22*, 221–250. https://doi.org/10.1007/s00357-005-0015-6

Totterdell, P., Wall, T., Holman, D., Diamond, H., & Epitropaki, O. (2004). Affect networks: A structural analysis of the relationship between work ties and job-related affect. *Journal of Applied Psychology*, *89*, 854–867. https://doi.org/10.1037/0021-9010.89.5.854

Vinod, H. (1969). Integer programming and the theory of grouping. *Journal of the American Statistical Association*, *64*, 506–517. https://doi.org/10.1080/01621459.1969.10500990

Vlasblom, J., & Wodak, S. J. (2009). Markov clustering versus affinity propagation for the partitioning of protein interaction graphs. *BMC Bioinformatics*, *10*, 99. https://doi.org/10.1186/1471-2105-10-99

von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics in Computing*, *17*, 395–416. https://doi.org/10.1007/s11222-007-9033-z

Wang, X., Qin, Z., & Zhang, X. (2010). Automatically affinity propagation clustering using particle swarm. *Journal of Computers*, *5*, 1731–1738. https://doi.org/10.4304/jcp.5.11.1731-1738

Ward, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, *58*, 236–244. https://doi.org/10.1080/01621459.1963.10500845

Wasti, S. A., & Cortina, L. M. (2002). Coping in context: Sociocultural determinants of responses to sexual harassment. *Journal of Personality and Social Psychology*, *83*, 394–405. https://doi.org/10.1037/0022-3514.83.2.394

Woźniak, M., Tiuryn, J., & Dutkowski, J. (2010). MODEVO: Exploring modularity and evolution of protein interaction networks. *Bioinformatics*, *26*, 1790–1791. https://doi.org/10.1093/bioinformatics/btq274

Zhang, J., Li, D., Chen, H., & Fang, F. (2011). Analysis of activity in fMRI data using affinity propagation clustering. *Computer Methods in Biomechanics and Biomedical Engineering*, *14*, 271–281. https://doi.org/10.1080/10255841003766829

Received 25 May 2017; revised version received 26 January 2018

Appendix 1:

To illustrate the mechanics of affinity propagation, we use data from a study by Mount et al. (2005) that investigated relationships among the big six vocational interest types (Holland, 1985). Mount *et al.*, (2005) conducted a meta-analytic study of research in these areas and assembled a matrix of intercorrelations among the $n = 6$ vocational interests. The pairwise correlations among the interests are as follows:

	R	I	A	S	E	C
Realistic (R)	1.00	0.45	0.25	0.18	0.20	0.27
Investigative (I)	0.45	1.00	0.36	0.26	0.09	0.17
Artistic (A)	0.25	0.36	1.00	0.39	0.28	0.01
Social (S)	0.18	0.26	0.39	1.00	0.51	0.29
Enterprising (E)	0.20	0.09	0.28	0.51	1.00	0.53
Conventional (C)	0.27	0.17	0.01	0.29	0.53	1.00

The correlation matrix serves as the basis for the input to the affinity propagation algorithm. However, an important modification needed to produce the similarity matrix is that the median of the pairwise correlations (.27) is the main diagonal for the similarity matrix. Also, for convenience, the elements of the similarity matrix are multiplied by 100 to convert the values to integers. The input similarity matrix, **S**, is:

	R	I	A	S	E	C
Realistic (R)	27	45	25	18	20	27
Investigative (I)	45	27	36	26	9	17
Artistic (A)	25	36	27	39	28	1
Social (S)	18	26	39	27	51	29
Enterprising (E)	20	9	28	51	27	53
Conventional (C)	27	17	1	29	53	27

The affinity propagation algorithm was applied using the default dampening parameter of $\gamma = .9$. At Step 0 of the algorithm, the elements of the 'responsibility' ($\mathbf{R} = [r_{ij}]$) and 'availability' ($\mathbf{A} = [a_{ij}]$) matrices, were initialized to zero. For the **R** and **A** matrices obtained on subsequent iterations, we will use a superscript to indicate the iteration number (e.g., $\mathbf{R}^1$ and $\mathbf{A}^1$ are the matrices after the first iteration of Step 1). The first iteration of affinity propagation at Step 1a resulted in the following responsibilities matrix, $\mathbf{R}^1$:

	R	I	A	S	E	C
Realistic (R)	−1.800	1.800	−2.000	−2.700	−2.500	−1.800
Investigative (I)	0.900	−1.800	−0.900	−1.900	−3.600	−2.800
Artistic (A)	−1.400	−0.300	−1.200	0.300	−1.100	−3.800
Social (S)	−3.300	−2.500	−1.200	−2.400	1.200	−2.200
Enterprising (E)	−3.300	−4.400	−2.500	−0.200	−2.600	0.200
Conventional (C)	−2.600	−3.600	−5.200	−2.400	2.400	−2.600

As an example of the computations in Step 1a that produced $\mathbf{R}^1$, we consider the element in the first column and first row: $r_{11}^1 = .9 \times 0 + (1 - .9)[27 - 45] = -1.800$. The value of .9 is the dampening factor, 0 is the value of r_{11}^0, 27 is s_{11}, and 45 is the maximum value in the first row of **S**.

The first iteration at Step 1b results in the following availabilities matrix, $\mathbf{A}^1$:

	R	I	A	S	E	C
Realistic (R)	0.090	−0.180	−0.120	−0.210	0.000	−0.240
Investigative (I)	−0.180	0.180	−0.120	−0.210	0.000	−0.240
Artistic (A)	−0.090	0.000	0.000	−0.240	0.000	−0.240
Social (S)	−0.090	0.000	−0.120	0.030	−0.020	−0.240
Enterprising (E)	−0.090	0.000	−0.120	−0.210	0.360	−0.260
Conventional (C)	−0.090	0.000	−0.120	−0.210	−0.140	0.020

As an example of the computations in Step 1b that produced $\mathbf{A}^1$, we consider the element in the first column and first row: $a_{11}^1 = .9 \times 0 + (1 - .9) \times .9 = 0.090$. The value of .9 is the dampening factor, 0 is the value of a_{11}^0, and the value of .9 in the brackets is the maximum value in the first column of $\mathbf{R}^1$.

The sum of the responsibilities and availabilities matrix, $\mathbf{R}^1 + \mathbf{A}^1$, after the first iteration is displayed below, where the cells in bold italic indicate the maximum value in each row:

	R	I	A	S	E	C
Realistic (R)	−1.710	***1.620***	−2.120	−2.910	−2.500	−2.040
Investigative (I)	***0.720***	−1.620	−1.020	−2.110	−3.600	−3.040
Artistic (A)	−1.490	−0.300	−1.200	***0.060***	−1.100	−4.040
Social (S)	−3.390	−2.500	−1.320	−2.370	***1.180***	−2.440
Enterprising (E)	−3.390	−4.400	−2.620	−0.410	−2.240	***−0.060***
Conventional (C)	−2.690	−3.600	−5.320	−2.610	***2.260***	−2.580

Based on the $\mathbf{R}^1 + \mathbf{A}^1$ matrix, there would be five exemplars selected: {Realistic, Investigative, Social, Enterprising, Conventional} because only the column for Artistic does not contain a maximum value for any row. As iterations of Step 1 continue, the number of exemplars decreases.

After 99 iterations, the responsibilities matrix, $\mathbf{R}^{99}$, is:

	R	I	A	S	E	C
Realistic (R)	−17.966	17.904	−19.966	−26.966	−24.966	−17.966
Investigative (I)	−7.114	−0.414	−16.115	−26.114	−43.114	−35.114
Artistic (A)	−11.000	7.999	−9.000	3.000	−8.000	−34.999
Social (S)	−32.999	−24.999	−12.000	−23.999	20.984	−21.999
Enterprising (E)	−53.565	−64.565	−45.565	−22.566	−3.041	−20.566
Conventional (C)	−25.992	−35.992	−51.992	−23.992	25.984	−25.992

The availabilities matrix, $\mathbf{A}^{99}$, is:

	R	I	A	S	E	C
Realistic (R)	0.013	−0.004	−8.998	−20.992	0.000	−25.959
Investigative (I)	−17.862	25.662	−8.998	−20.992	0.000	−25.959
Artistic (A)	−17.849	0.000	0.000	−23.991	0.000	−25.959
Social (S)	−17.849	0.000	−8.998	3.000	0.000	−25.959
Enterprising (E)	−17.849	0.000	−8.998	−20.993	46.868	−25.961
Conventional (C)	−17.849	0.000	−8.998	−20.992	−0.001	0.002

The sum of the responsibilities and availabilities matrix, $\mathbf{R}^{99} + \mathbf{A}^{99}$, is:

	R	I	A	S	E	C
Realistic (R)	−17.953	***17.900***	−28.963	−47.958	−24.966	−43.925
Investigative (I)	−24.976	***25.248***	−25.112	−47.106	−43.114	−61.073
Artistic (A)	−28.849	***7.999***	−9.000	−20.991	−8.000	−60.958
Social (S)	−50.848	−24.999	−20.997	−20.999	***20.984***	−47.959
Enterprising (E)	−71.414	−64.565	−54.563	−43.559	***43.827***	−46.527
Conventional (C)	−43.841	−35.992	−60.989	−44.985	***25.983***	−25.991

After 99 iterations, there are only two exemplars: Investigative and Enterprising. The interests assigned to the cluster associated with the Investigative exemplar are {Realistic, Investigative, Artistic}. The interests assigned to the cluster corresponding to the Enterprising exemplar are: {Social, Enterprising, Conventional}.

The next (100th) iteration yields $\mathbf{R}^{100}$:

	R	I	A	S	E	C
Realistic (R)	−17.969	17.912	−19.969	−26.969	−24.969	−17.969
Investigative (I)	−7.169	−0.387	−16.169	−26.169	−43.169	−35.169
Artistic (A)	−11.000	7.999	−9.000	3.000	−8.000	−34.999
Social (S)	−32.999	−24.999	−12.000	−23.999	20.986	−21.999
Enterprising (E)	−53.595	−64.595	−45.596	−22.596	−3.038	−20.596
Conventional (C)	−25.993	−35.993	−51.992	−23.993	25.985	−25.993

The availabilities matrix, $\mathbf{A}^{100}$, is:

	R	I	A	S	E	C
Realistic (R)	0.012	−0.004	−8.998	−20.993	0.000	−25.963
Investigative (I)	−17.873	25.687	−8.998	−20.993	0.000	−25.963
Artistic (A)	−17.861	0.000	0.000	−23.992	0.000	−25.963
Social (S)	−17.861	0.000	−8.998	3.000	0.000	−25.963
Enterprising (E)	−17.861	0.000	−8.998	−20.994	46.878	−25.964
Conventional (C)	−17.861	0.000	−8.998	−20.993	−0.001	0.002

The sum of the responsibilities and availabilities matrix, $\mathbf{R}^{100} + \mathbf{A}^{100}$, is:

	R	I	A	S	E	C
Realistic (R)	−17.957	***17.909***	−28.967	−47.962	−24.969	−43.932
Investigative (I)	−25.042	***25.300***	−25.167	−47.162	−43.169	−61.132
Artistic (A)	−28.861	***7.999***	−9.000	−20.992	−8.000	−60.962
Social (S)	−50.860	−24.999	−20.997	−21.000	***20.986***	−47.962
Enterprising (E)	−71.456	−64.595	−54.593	−43.590	***43.840***	−46.560
Conventional (C)	−43.854	−35.993	−60.990	−44.986	***25.985***	−25.991

Clearly, the changes that occur from the 99th to the 100th iteration are subtle. The same two exemplars are selected, and the assignments of interests to the clusters associated with those exemplars are also the same. Thus, it appears that the affinity propagation algorithm has converged to a solution. More formally, the default criterion for convergence is 100 consecutive iterations with no change in the exemplars or assignments (Dueck, 2009, p. 48).

Next, we consider the two-cluster solution obtained by affinity propagation. For the first cluster, {Realistic, Investigative, Artistic}, Investigative is the best exemplar because the sum of its similarities to Realistic (45) and Artistic (36) is 81. If Realistic is chosen as the exemplar, the sum of its similarities to Investigative (45) and Artistic (25) is only 70. Likewise, if Artistic is chosen, the sum of its similarities with Realistic (25) and Investigative (36) is only 61. A similar analysis for the second cluster shows that Enterprising is the best choice for the exemplar because the sum of its similarities to Social (51) and Conventional (53) is 104. Adding the similarity sums for the two clusters yields *dpsim* = 81 + 104 = 185. The contribution from the two exemplars is *expref* = 2(27) = 54, which yields *netsim* = 185 + 54 = 239.