Clustering by Passing Messages Between Data Points

Brendan J. Frey* and Delbert Dueck

Clustering data by identifying a subset of representative examples is important for processing sensory signals and detecting patterns in data. Such "exemplars" can be found by randomly choosing an initial subset of data points and then iteratively refining it, but this works well only if that initial choice is close to a good solution. We devised a method called "affinity propagation," which takes as input measures of similarity between pairs of data points. Real-valued messages are exchanged between data points until a high-quality set of exemplars and corresponding clusters gradually emerges. We used affinity propagation to cluster images of faces, detect genes in microarray data, identify representative sentences in this manuscript, and identify cities that are efficiently accessed by airline travel. Affinity propagation found clusters with much lower error than other methods, and it did so in less than one-hundredth the amount of time.

Clustering data based on a measure of similarity is a critical step in scientific data analysis and in engineering systems. A common approach is to use data to learn a set of centers such that the sum of squared errors between data points and their nearest centers is small. When the centers are selected from actual data points, they are called "exemplars." The popular k-centers clustering technique (*1*) begins with an initial set of randomly selected exemplars and iteratively refines this set so as to decrease the sum of squared errors. k-centers clustering is quite sensitive to the initial selection of exemplars, so it is usually rerun many times with different initializations in an attempt to find a good solution. However, this works well only when the number of clusters is small and chances are good that at least one random initialization is close to a good solution. We take a quite different approach and introduce a method that simultaneously considers all data points as potential exemplars. By viewing each data point as a node in a network, we devised a method that recursively transmits real-valued messages along edges of the network until a good set of exemplars and corresponding clusters emerges. As described later, messages are updated on the basis of simple formulas that search for minima of an appropriately chosen energy function. At any point in time, the magnitude of each message reflects the current affinity that one data point has for choosing another data point as its exemplar, so we call our method "affinity propagation." Figure 1A illustrates how clusters gradually emerge during the message-passing procedure.

Affinity propagation takes as input a collection of real-valued similarities between data points, where the similarity $s(i,k)$ indicates how well the data point with index k is suited to be the exemplar for data point i. When the goal is to minimize squared error, each similarity is set to a negative squared error (Euclidean distance): For points x_i and x_k, $s(i,k) = -\|x_i - x_k\|^2$. Indeed, the method described here can be applied when the optimization criterion is much more general. Later, we describe tasks where similarities are derived for pairs of images, pairs of microarray measurements, pairs of English sentences, and pairs of cities. When an exemplar-dependent probability model is available, $s(i,k)$ can be set to the log-likelihood of data point i given that its exemplar is point k. Alternatively, when appropriate, similarities may be set by hand.

Rather than requiring that the number of clusters be prespecified, affinity propagation takes as input a real number $s(k,k)$ for each data point k so that data points with larger values of $s(k,k)$ are more likely to be chosen as exemplars. These values are referred to as "preferences." The number of identified exemplars (number of clusters) is influenced by the values of the input preferences, but also emerges from the message-passing procedure. If a priori, all data points are equally suitable as exemplars, the preferences should be set to a common value—this value can be varied to produce different numbers of clusters. The shared value could be the median of the input similarities (resulting in a moderate number of clusters) or their minimum (resulting in a small number of clusters).

There are two kinds of message exchanged between data points, and each takes into account a different kind of competition. Messages can be combined at any stage to decide which points are exemplars and, for every other point, which exemplar it belongs to. The "responsibility" $r(i,k)$, sent from data point i to candidate exemplar point k, reflects the accumulated evidence for how well-suited point k is to serve as the exemplar for point i, taking into account other potential exemplars for point i (Fig. 1B). The "availability" $a(i,k)$, sent from candidate exemplar point k to point i, reflects the accumulated evidence for how appropriate it would be for point i to choose point k as its exemplar, taking into account the support from other points that point k should be an exemplar (Fig. 1C). $r(i,k)$ and $a(i,k)$ can be viewed as log-probability ratios. To begin with, the availabilities are initialized to zero: $a(i,k) = 0$. Then, the responsibilities are computed using the rule

$$r(i,k) \leftarrow s(i,k) - \max_{k' \text{ s.t. } k' \neq k} \{a(i,k') + s(i,k')\} \quad (1)$$

In the first iteration, because the availabilities are zero, $r(i,k)$ is set to the input similarity between point i and point k as its exemplar, minus the largest of the similarities between point i and other candidate exemplars. This competitive update is data-driven and does not take into account how many other points favor each candidate exemplar. In later iterations, when some points are effectively assigned to other exemplars, their availabilities will drop below zero as prescribed by the update rule below. These negative availabilities will decrease the effective values of some of the input similarities $s(i,k')$ in the above rule, removing the corresponding candidate exemplars from competition. For $k = i$, the responsibility $r(k,k)$ is set to the input preference that point k be chosen as an exemplar, $s(k,k)$, minus the largest of the similarities between point i and all other candidate exemplars. This "self-responsibility" reflects accumulated evidence that point k is an exemplar, based on its input preference tempered by how ill-suited it is to be assigned to another exemplar.

Whereas the above responsibility update lets all candidate exemplars compete for ownership of a data point, the following availability update gathers evidence from data points as to whether each candidate exemplar would make a good exemplar:

$$a(i,k) \leftarrow \min\left\{0, r(k,k) + \sum_{i' \text{ s.t. } i' \notin \{i,k\}} \max\{0, r(i',k)\}\right\} \quad (2)$$

The availability $a(i,k)$ is set to the self-responsibility $r(k,k)$ plus the sum of the positive responsibilities candidate exemplar k receives from other points. Only the positive portions of incoming responsibilities are added, because it is only necessary for a good exemplar to explain some data points well (positive responsibilities), regardless of how poorly it explains other data points (negative responsibilities). If the self-responsibility $r(k,k)$ is negative (indicating that point k is currently better suited as belonging to another exemplar rather than being an exemplar itself), the availability of point k as an exemplar can be increased if some other points have positive responsibilities for point k being their exemplar. To limit the influence of strong

Department of Electrical and Computer Engineering, University of Toronto, 10 King's College Road, Toronto, Ontario M5S 3G4, Canada.

*To whom correspondence should be addressed. E-mail: frey@psi.toronto.edu

incoming positive responsibilities, the total sum is thresholded so that it cannot go above zero. The "self-availability" $a(k,k)$ is updated differently:

$$a(k,k) \leftarrow \sum_{i' \text{ s.t. } i' \neq k} \max\{0, r(i',k)\} \qquad (3)$$

This message reflects accumulated evidence that point k is an exemplar, based on the positive responsibilities sent to candidate exemplar k from other points.

The above update rules require only simple, local computations that are easily implemented (*2*), and messages need only be exchanged between pairs of points with known similarities. At any point during affinity propagation, availabilities and responsibilities can be combined to identify exemplars. For point i, the value of k that maximizes $a(i,k) + r(i,k)$ either identifies point i as an exemplar if $k = i$, or identifies the data point that is the exemplar for point i. The message-passing procedure may be terminated after a fixed number of iterations, after changes in the messages fall below a threshold, or after the local decisions stay constant for some number of iterations. When updating the messages, it is important that they be damped to avoid numerical oscillations that arise in some circumstances. Each message is set to λ times its value from the previous iteration plus $1 - \lambda$ times its prescribed updated value, where the damping factor λ is between 0 and 1. In all of our experiments (*3*), we used a default damping factor of $\lambda = 0.5$, and each iteration of affinity propagation consisted of (i) updating all responsibilities given the availabilities, (ii) updating all availabilities given the responsibilities, and (iii) combining availabilities and responsibilities to monitor the exemplar decisions and terminate the algorithm when these decisions did not change for 10 iterations.

Figure 1A shows the dynamics of affinity propagation applied to 25 two-dimensional data points (*3*), using negative squared error as the similarity. One advantage of affinity propagation is that the number of exemplars need not be specified beforehand. Instead, the appropriate number of exemplars emerges from the message-passing method and depends on the input exemplar preferences. This enables automatic model selection, based on a prior specification of how preferable each point is as an exemplar. Figure 1D shows the effect of the value of the common input preference on the number of clusters. This relation is nearly identical to the relation found by exactly minimizing the squared error (*2*).

We next studied the problem of clustering images of faces using the standard optimiza-

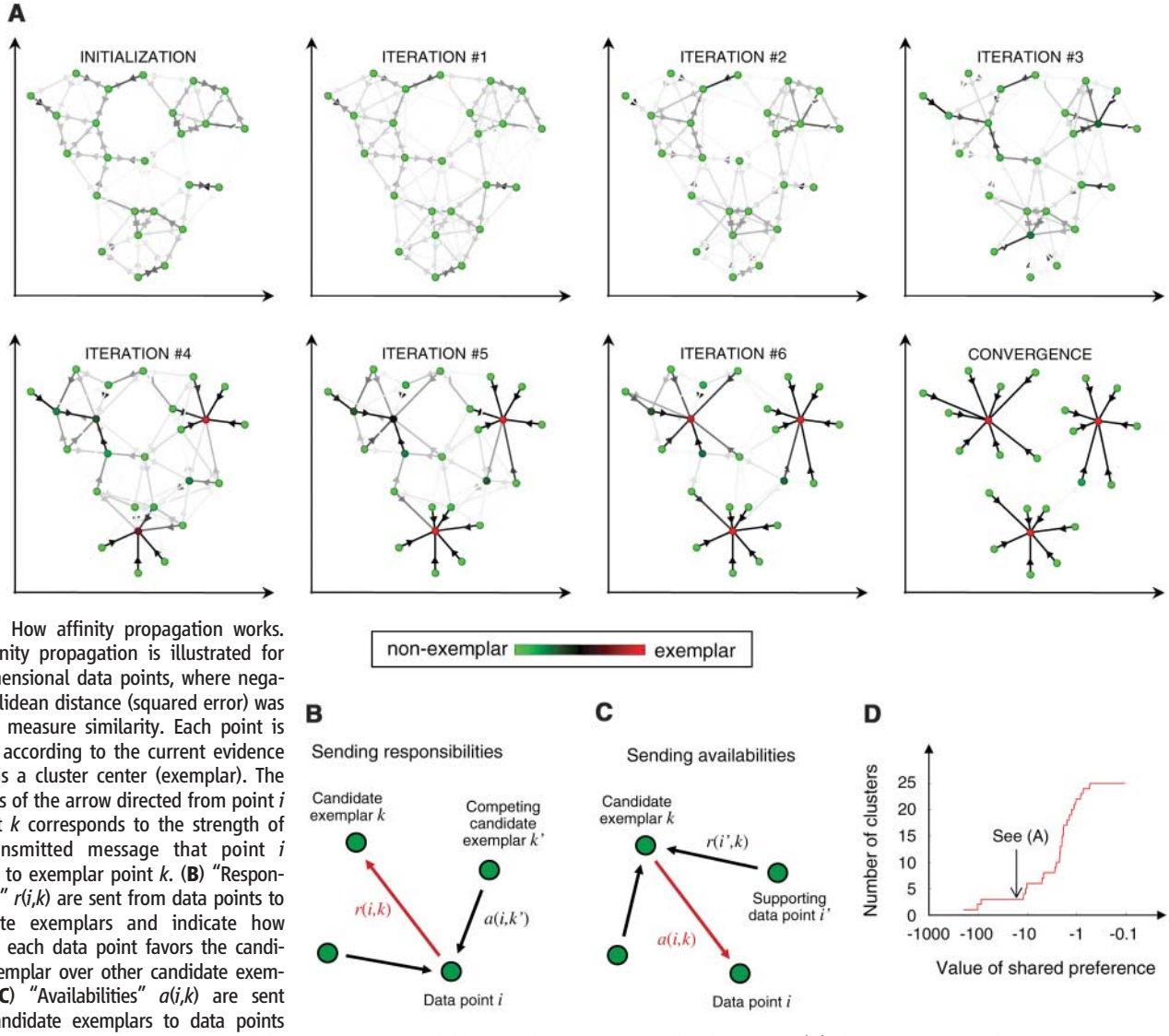

Fig. 1. How affinity propagation works. (**A**) Affinity propagation is illustrated for two-dimensional data points, where negative Euclidean distance (squared error) was used to measure similarity. Each point is colored according to the current evidence that it is a cluster center (exemplar). The darkness of the arrow directed from point i to point k corresponds to the strength of the transmitted message that point i belongs to exemplar point k. (**B**) "Responsibilities" $r(i,k)$ are sent from data points to candidate exemplars and indicate how strongly each data point favors the candidate exemplar over other candidate exemplars. (**C**) "Availabilities" $a(i,k)$ are sent from candidate exemplars to data points and indicate to what degree each candidate exemplar is available as a cluster center for the data point. (**D**) The effect of the value of the input preference (common for all data points) on the number of identified exemplars (number of clusters) is shown. The value that was used in (A) is also shown, which was computed from the median of the pairwise similarities.

tion criterion of squared error. We used both affinity propagation and *k*-centers clustering to identify exemplars among 900 grayscale images extracted from the Olivetti face database (*3*). Affinity propagation found exemplars with much lower squared error than the best of 100 runs of *k*-centers clustering (Fig. 2A), which took about the same amount of computer time. We asked whether a huge number of random restarts of *k*-centers clustering could achieve the same squared error. Figure 2B shows the error achieved by one run of affinity propagation and the distribution of errors achieved by 10,000 runs of *k*-centers clustering, plotted against the number of clusters. Affinity propagation uniformly achieved much lower error in more than two orders of magnitude less time. Another popular optimization criterion is the sum of absolute pixel differences (which better tolerates outlying pixel intensities), so we repeated the above procedure using this error measure. Affinity propagation again uniformly achieved lower error (Fig. 2C).

Many tasks require the identification of exemplars among sparsely related data, i.e., where most similarities are either unknown or large and negative. To examine affinity propagation in this context, we addressed the task of clustering putative exons to find genes, using the sparse similarity matrix derived from microarray data and reported in (*4*). In that work, 75,066 segments of DNA (60 bases long) corresponding to putative exons were mined from the genome of mouse chromosome 1. Their transcription levels were measured across 12 tissue samples, and the similarity between every pair of putative exons (data points) was computed. The measure of similarity between putative exons was based on their proximity in the genome and the degree of coordination of their transcription levels across the 12 tissues. To account for putative exons that are not exons (e.g., introns), we included an additional artificial exemplar and determined the similarity of each other data point to this "non-exon exemplar" using statistics taken over the entire data set. The resulting 75,067 × 75,067 similarity matrix (*3*) consisted of 99.73% similarities with values of $-\infty$, corresponding to distant DNA segments that could not possibly be part of the same gene. We applied affinity propagation to this similarity matrix, but because messages need not be exchanged between point i and k if $s(i,k) = -\infty$, each iteration of affinity propagation required exchanging messages between only a tiny subset (0.27% or 15 million) of data point pairs.

Figure 3A illustrates the identification of gene clusters and the assignment of some data points to the nonexon exemplar. The reconstruction errors for affinity propagation and *k*-centers clustering are compared in Fig. 3B. For each number of clusters, affinity propagation was run once and took 6 min, whereas *k*-centers clustering was run 10,000 times and took 208 hours. To address the question of how well these methods perform in detecting bona fide gene segments, Fig. 3C plots the true-positive (TP) rate against the false-positive (FP) rate, using the labels provided in the RefSeq database (*5*). Affinity propagation achieved significantly higher TP rates, especially at low FP rates, which are most important to biologists. At a FP rate of 3%, affinity propagation achieved a TP rate of 39%, whereas the best *k*-centers clustering result was 17%. For comparison, at the same FP rate, the best TP rate for hierarchical agglomerative clustering (*2*) was 19%, and the engineering tool described in (*4*), which accounts for additional biological knowledge, achieved a TP rate of 43%.

Affinity propagation's ability to operate on the basis of nonstandard optimization criteria makes it suitable for exploratory data analysis using unusual measures of similarity. Unlike metric-space clustering techniques such as *k*-means clustering (*1*), affinity propagation can be applied to problems where the data do not lie in a continuous space. Indeed, it can be applied to problems where the similarities are not symmetric [i.e., $s(i,k) \neq s(k,i)$] and to problems where the similarities do not satisfy the triangle inequality [i.e., $s(i,k) < s(i,j) + s(j,k)$]. To identify a small number of sentences in a draft of this manuscript that summarize other sentences, we treated each sentence as a "bag of words" (*6*) and computed the similarity of sentence i to sentence k based on the cost of encoding the words in sentence i using the words in sentence k. We found that 97% of the resulting similarities (*2*, *3*) were not symmetric. The preferences were adjusted to identify (using $\lambda = 0.8$) different numbers of representative exemplar sentences (*2*), and the solution with four sentences is shown in Fig. 4A.

We also applied affinity propagation to explore the problem of identifying a restricted number of Canadian and American cities that are most easily accessible by large subsets of other cities, in terms of estimated commercial airline travel time. Each data point was a city, and the similarity $s(i,k)$ was set to the negative time it takes to travel from city i to city k by airline, including estimated stopover delays (*3*). Due to headwinds, the transit time was in many cases different depending on the direction of travel, so that 36% of the similarities were asymmetric. Further, for 97% of city pairs i and k, there was a third city j such that the triangle inequality was violated, because the trip from i to k included a long stopover delay

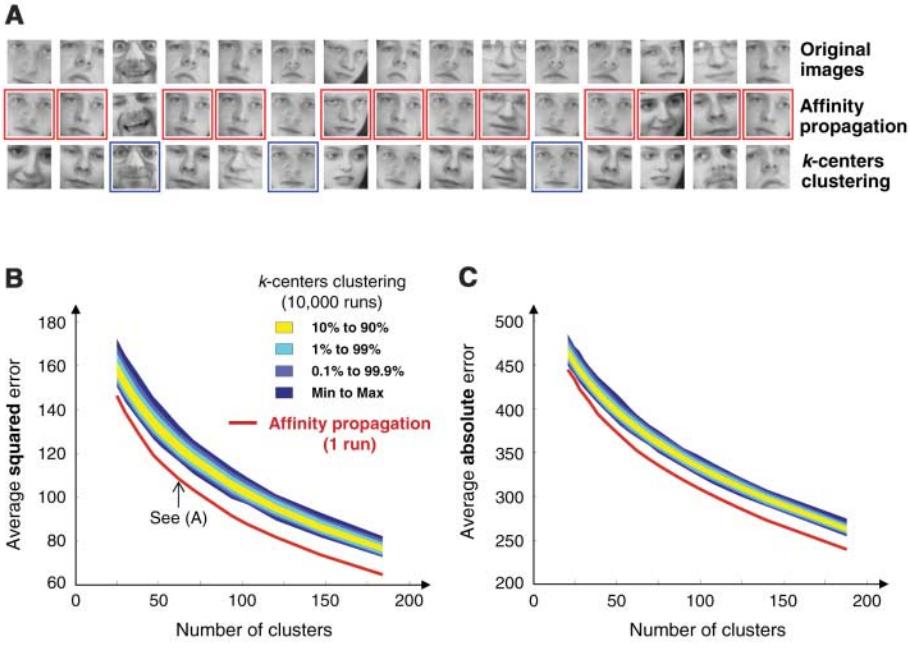

Fig. 2. Clustering faces. Exemplars minimizing the standard squared error measure of similarity were identified from 900 normalized face images (*3*). For a common preference of −600, affinity propagation found 62 clusters, and the average squared error was 108. For comparison, the best of 100 runs of *k*-centers clustering with different random initializations achieved a worse average squared error of 119. (**A**) The 15 images with highest squared error under either affinity propagation or *k*-centers clustering are shown in the top row. The middle and bottom rows show the exemplars assigned by the two methods, and the boxes show which of the two methods performed better for that image, in terms of squared error. Affinity propagation found higher-quality exemplars. (**B**) The average squared error achieved by a single run of affinity propagation and 10,000 runs of *k*-centers clustering, versus the number of clusters. The colored bands show different percentiles of squared error, and the number of exemplars corresponding to the result from (A) is indicated. (**C**) The above procedure was repeated using the sum of absolute errors as the measure of similarity, which is also a popular optimization criterion.

in city j so it took longer than the sum of the durations of the trips from i to j and j to k. When the number of "most accessible cities" was constrained to be seven (by adjusting the input preference appropriately), the cities shown in Fig. 4, B to E, were identified. It is interesting that several major cities were not selected, either because heavy international travel makes them inappropriate as easily accessible domestic destinations (e.g., New York City, Los Angeles) or because their neighborhoods can be more efficiently accessed through other destinations (e.g., Atlanta, Philadelphia, and Minneapolis account for Chicago's destinations, while avoiding potential airport delays).

Affinity propagation can be viewed as a method that searches for minima of an energy function (*7*) that depends on a set of N hidden labels, $c_1,\ldots,c_N$, corresponding to the N data points. Each label indicates the exemplar to which the point belongs, so that $s(i,c_i)$ is the similarity of data point i to its exemplar. $c_i = i$ is a special case indicating that point i is itself an exemplar, so that $s(i,c_i)$ is the input preference for point i. Not all configurations of the labels are valid; a configuration $\mathbf{c}$ is valid when for every point i, if some other point i' has chosen i as its exemplar (i.e., $c_{i'} = i$), then i must be an exemplar (i.e., $c_i = i$). The energy of a valid configuration is $E(\mathbf{c}) = -\sum_{i=1}^{N} s(i,c_i)$. Exactly minimizing the energy is computationally intractable, because a special case of this minimization problem is the NP-hard k-median problem (*8*). However, the update rules for affinity propagation correspond to fixed-point recursions for minimizing a Bethe free-energy (*9*) approximation. Affinity propagation is most easily derived as an instance of the max-sum algorithm in a factor graph (*10*) describing the constraints on the labels and the energy function (*2*).

In some degenerate cases, the energy function may have multiple minima with corresponding multiple fixed points of the update rules, and these may prevent convergence. For example, if $s(1,2) = s(2,1)$ and $s(1,1) = s(2,2)$, then the solutions $c_1 = c_2 = 1$ and $c_1 = c_2 = 2$ both achieve the same energy. In this case, affinity propagation may oscillate, with both data points alternating between being exemplars and nonexemplars. In practice, we found that oscillations could always be avoided by adding a tiny amount of noise to the similarities to prevent degenerate situations, or by increasing the damping factor.

Affinity propagation has several advantages over related techniques. Methods such as k-centers clustering (*1*), k-means clustering (*1*), and the expectation maximization (EM) algorithm (*11*) store a relatively small set of estimated cluster centers at each step. These techniques are improved upon by methods that begin with a large number of clusters and then prune them (*12*), but they still rely on random sampling and make hard pruning decisions that cannot be recovered from. In contrast, by simultaneously considering all data points as candidate centers and gradually identifying clusters, affinity propagation is able to avoid many of the poor solutions caused by unlucky initializations and hard decisions. Markov chain Monte Carlo techniques (*13*) randomly search for good solutions, but do not share affinity propagation's advantage of considering many possible solutions all at once.

Hierarchical agglomerative clustering (*14*) and spectral clustering (*15*) solve the quite different problem of recursively comparing pairs of points to find partitions of the data. These techniques do not require that all points within a cluster be similar to a single center and are thus not well-suited to many tasks. In particular, two points that should not be in the same cluster may be grouped together by an unfortunate sequence of pairwise groupings.

In (*8*), it was shown that the related metric k-median problem could be relaxed to form a

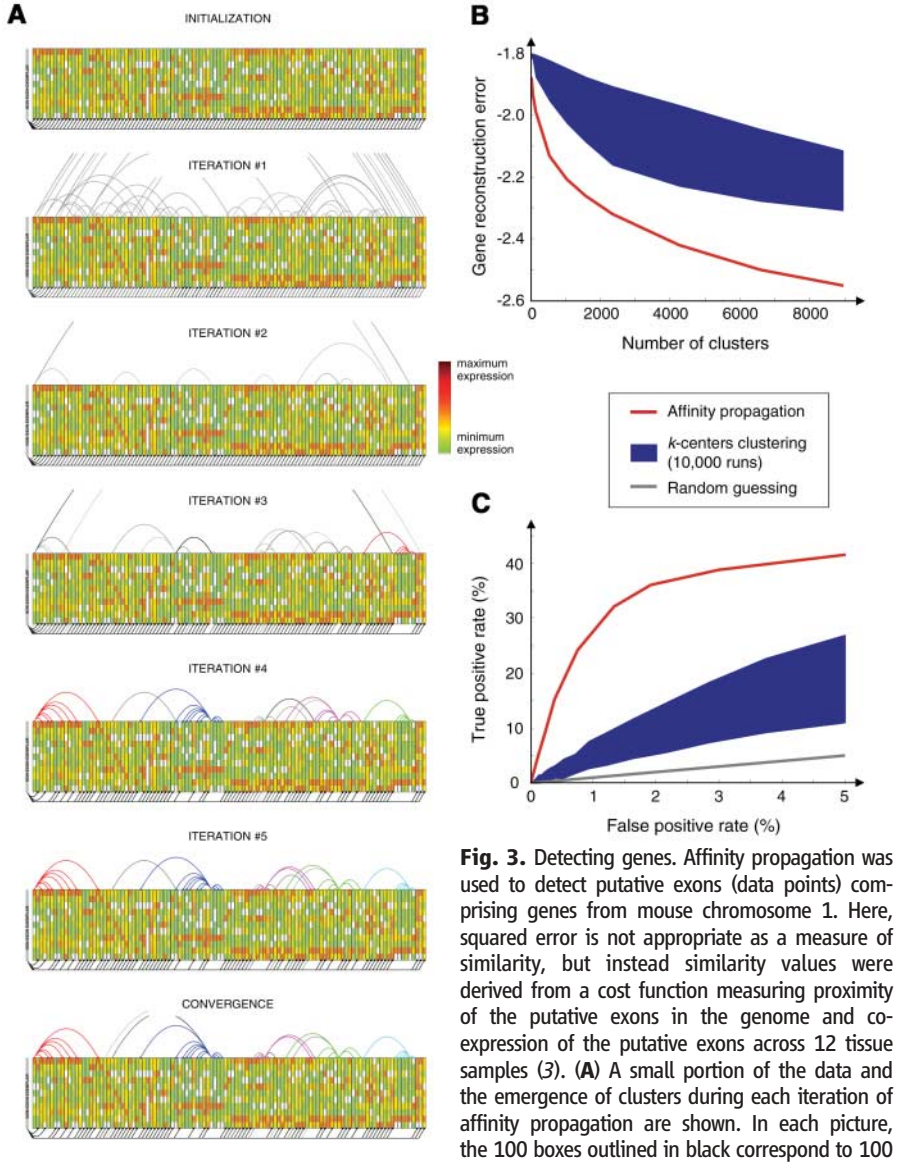

Fig. 3. Detecting genes. Affinity propagation was used to detect putative exons (data points) comprising genes from mouse chromosome 1. Here, squared error is not appropriate as a measure of similarity, but instead similarity values were derived from a cost function measuring proximity of the putative exons in the genome and coexpression of the putative exons across 12 tissue samples (*3*). (**A**) A small portion of the data and the emergence of clusters during each iteration of affinity propagation are shown. In each picture, the 100 boxes outlined in black correspond to 100 data points (from a total of 75,066 putative exons), and the 12 colored blocks in each box indicate the transcription levels of the corresponding DNA segment in 12 tissue samples. The box on the far left corresponds to an artificial data point with infinite preference that is used to account for nonexon regions (e.g., introns). Lines connecting data points indicate potential assignments, where gray lines indicate assignments that currently have weak evidence and solid lines indicate assignments that currently have strong evidence. (**B**) Performance on minimizing the reconstruction error of genes, for different numbers of detected clusters. For each number of clusters, affinity propagation took 6 min, whereas 10,000 runs of k-centers clustering took 208 hours on the same computer. In each case, affinity propagation achieved a significantly lower reconstruction error than k-centers clustering. (**C**) A plot of true-positive rate versus false-positive rate for detecting exons [using labels from RefSeq (*5*)] shows that affinity propagation also performs better at detecting biologically verified exons than k-centers clustering.

linear program with a constant factor approximation. There, the input was assumed to be metric, i.e., nonnegative, symmetric, and satisfying the triangle inequality. In contrast, affinity propagation can take as input general nonmetric similarities. Affinity propagation also provides a conceptually new approach that works well in practice. Whereas the linear programming relaxation is hard to solve and sophisticated software packages need to be applied (e.g., CPLEX), affinity propagation makes use of intuitive message updates that can be implemented in a few lines of code (*2*).

Affinity propagation is related in spirit to techniques recently used to obtain record-breaking results in quite different disciplines (*16*). The approach of recursively propagating messages (*17*) in a "loopy graph" has been used to approach Shannon's limit in error-correcting decoding (*18*, *19*), solve random satisfiability problems with an order-of-magnitude increase in size (*20*), solve instances of the NP-hard two-dimensional phase-unwrapping problem (*21*), and efficiently estimate depth from pairs of stereo images (*22*). Yet, to our knowledge, affinity propagation is the first method to make use of this idea to solve the age-old, fundamental problem of clustering data. Because of its simplicity, general applicability, and performance, we believe affinity propagation will prove to be of broad value in science and engineering.

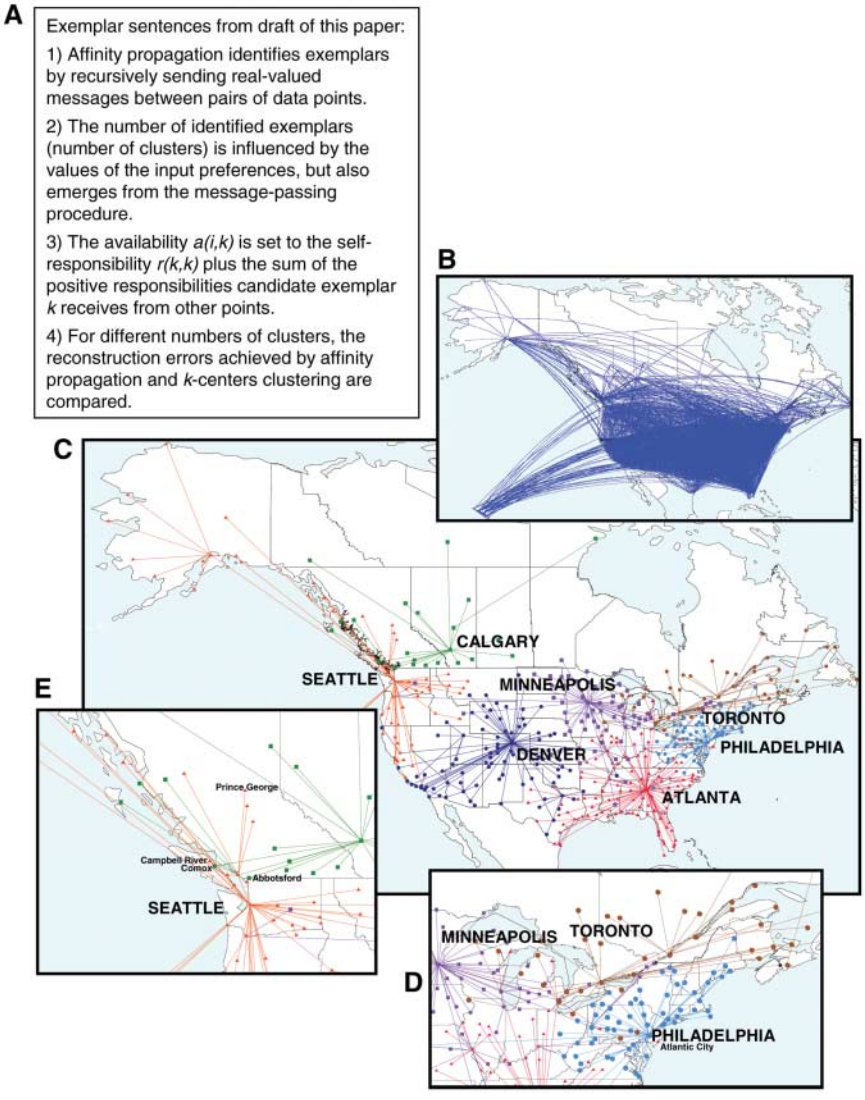

Fig. 4. Identifying key sentences and air-travel routing. Affinity propagation can be used to explore the identification of exemplars on the basis of nonstandard optimization criteria. (**A**) Similarities between pairs of sentences in a draft of this manuscript were constructed by matching words. Four exemplar sentences were identified by affinity propagation and are shown. (**B**) Affinity propagation was applied to similarities derived from air-travel efficiency (measured by estimated travel time) between the 456 busiest commercial airports in Canada and the United States—the travel times for both direct flights (shown in blue) and indirect flights (not shown), including the mean transfer time of up to a maximum of one stopover, were used as negative similarities (*3*). (**C**) Seven exemplars identified by affinity propagation are color-coded, and the assignments of other cities to these exemplars is shown. Cities located quite near to exemplar cities may be members of other more distant exemplars due to the lack of direct flights between them (e.g., Atlantic City is 100 km from Philadelphia, but is closer in flight time to Atlanta). (**D**) The inset shows that the Canada-USA border roughly divides the Toronto and Philadelphia clusters, due to a larger availability of domestic flights compared to international flights. However, this is not the case on the west coast as shown in (**E**), because extraordinarily frequent airline service between Vancouver and Seattle connects Canadian cities in the northwest to Seattle.

References and Notes

1. J. MacQueen, in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, L. Le Cam, J. Neyman, Eds. (Univ. of California Press, Berkeley, CA, 1967), vol. 1, pp. 281–297.
2. Supporting material is available on *Science* Online.
3. Software implementations of affinity propagation, along with the data sets and similarities used to obtain the results described in this manuscript, are available at www.psi.toronto.edu/affinitypropagation.
4. B. J. Frey *et al.*, *Nat. Genet.* **37**, 991 (2005).
5. K. D. Pruitt, T. Tatusova, D. R. Maglott, *Nucleic Acids Res.* **31**, 34 (2003).
6. C. D. Manning, H. Schutze, *Foundations of Statistical Natural Language Processing* (MIT Press, Cambridge, MA, 1999).
7. J. J. Hopfield, *Proc. Natl. Acad. Sci. U.S.A.* **79**, 2554 (1982).
8. M. Charikar, S. Guha, A. Tardos, D. B. Shmoys, *J. Comput. Syst. Sci.* **65**, 129 (2002).
9. J. S. Yedidia, W. T. Freeman, Y. Weiss, *IEEE Trans. Inf. Theory* **51**, 2282 (2005).
10. F. R. Kschischang, B. J. Frey, H.-A. Loeliger, *IEEE Trans. Inf. Theory* **47**, 498 (2001).
11. A. P. Dempster, N. M. Laird, D. B. Rubin, *Proc. R. Stat. Soc. B* **39**, 1 (1977).
12. S. Dasgupta, L. J. Schulman, *Proc. 16th Conf. UAI* (Morgan Kaufman, San Francisco, CA, 2000), pp. 152–159.
13. S. Jain, R. M. Neal, *J. Comput. Graph. Stat.* **13**, 158 (2004).
14. R. R. Sokal, C. D. Michener, *Univ. Kans. Sci. Bull.* **38**, 1409 (1958).
15. J. Shi, J. Malik, *IEEE Trans. Pattern Anal. Mach. Intell.* **22**, 888 (2000).
16. M. Mézard, *Science* **301**, 1685 (2003).
17. J. Pearl, *Probabilistic Reasoning in Intelligent Systems* (Morgan Kaufman, San Mateo, CA, 1988).
18. D. J. C. MacKay, *IEEE Trans. Inf. Theory* **45**, 399 (1999).
19. C. Berrou, A. Glavieux, *IEEE Trans. Commun.* **44**, 1261 (1996).
20. M. Mézard, G. Parisi, R. Zecchina, *Science* **297**, 812 (2002).
21. B. J. Frey, R. Koetter, N. Petrovic, in *Proc. 14th Conf. NIPS* (MIT Press, Cambridge, MA, 2002), pp. 737–743.
22. T. Meltzer, C. Yanover, Y. Weiss, in *Proc. 10th Conf. ICCV* (IEEE Computer Society Press, Los Alamitos, CA, 2005), pp. 428–435.

23. We thank B. Freeman, G. Hinton, R. Koetter, Y. LeCun, S. Roweis, and Y. Weiss for helpful discussions and P. Dayan, G. Hinton, D. MacKay, M. Mezard, S. Roweis, and C. Tomasi for comments on a previous draft of this manuscript. We acknowledge funding from Natural Sciences and Engineering Research Council of Canada, Genome Canada/Ontario Genomics Institute, and the Canadian Institutes of Health Research. B.J.F. is a Fellow of the Canadian Institute for Advanced Research.

Supporting Online Material

www.sciencemag.org/cgi/content/full/1136800/DC1
SOM Text
Figs. S1 to S3
References

26 October 2006; accepted 26 December 2006
Published online 11 January 2007;
10.1126/science.1136800
Include this information when citing this paper.