Interpreting and Extending Classical Agglomerative Clustering Algorithms using a Model-Based Approach

Sepandar D. Kamvar KAMVAR@SCCM.STANFORD.EDU
Dan Klein KLEIN@CS.STANFORD.EDU
Christopher D. Manning MANNING@CS.STANFORD.EDU
Department of Computer Science, Stanford University, Stanford, CA 94305-9040 USA

Abstract

We present two results which arise from a model-based approach to hierarchical agglomerative clustering. First, we show formally that the common heuristic agglomerative clustering algorithms – Ward's method, single-link, complete-link, and a variant of group-average – are each equivalent to a hierarchical model-based method. This interpretation gives a theoretical explanation of the empirical behavior of these algorithms, as well as a principled approach to resolving practical issues, such as number of clusters or the choice of method. Second, we show how a model-based viewpoint can suggest variations on these basic agglomerative algorithms. We introduce adjusted complete-link, Mahalanobis-link, and line-link as variants, and demonstrate their utility.

1. Introduction

Model-based clustering algorithms are theoretically well-founded and empirically successful methods for clustering data. In model-based clustering, the data is assumed to have been generated by a mixture of component probability distributions, where each component corresponds to a different cluster. Model-based agglomerative clustering has proven effective in many areas, including document clustering (Dom & Vaithyanathan, 1999), optical character recognition (Murtagh & Raftery, 1984), and medical image segmentation (Banfield & Raftery, 1993).

Despite the theoretical appeal and empirical success of model-based methods, in practice they are used far less than the popular, but more heuristic, classical agglomerative methods: single-link, complete-link, group-average, and Ward's method (Jain et al., 1999). In these algorithms, each data point is initially assigned to its own singleton cluster, and pairs of clusters are then successively merged according to some objective function until all points belong to the same cluster. The various agglomerative algorithms differ in the objective function they use to determine the sequence of merges.

The heuristic methods are popular for several reasons. The sequence of merges in these algorithms produces a cluster dendrogram as in figure 1, which is often more useful than the flat cluster structure created by partitional clustering algorithms. Moreover, their conceptual simplicity and ease of implementation make them convenient for use in many situations. Finally, they are a natural choice in cases where only proximity data is available. For this reason, linkage-based agglomerative methods have been widely used in the field of genomics, where gene sequence data does not have a natural feature representation, but lends itself well to calculating pairwise proximities.

In the present work, we prove that the classical agglomerative methods are a subset of model-based methods. In section 2, we introduce model-based agglomerative clustering. In section 3, we discuss the heuristic agglomerative methods, showing that each classical agglomerative method can be seen as a hierarchical model-based method for a certain finite mixture model. Finally, in section 4, we show how the model-based viewpoint can suggest variations on the classical agglomerative clustering methods. We introduce three such variants and demonstrate their utility.

2. Model-Based Clustering

Model-based hard clustering is an approach to computing an approximate maximum for the *classification likelihood* (Celeux & Govaert, 1993) of the data X:

$$\mathcal{L}(\theta_1, \ldots, \theta_k; l_1, \ldots, l_n | X) = \prod_{i=1}^{n} p(x_i | \theta_{l_i}) \qquad (1)$$

where l_i are labels indicating the classification of each data point ($l_i = j$ if x_i belongs to component j), and $\theta_1, \ldots, \theta_k$

are model parameters.[1]

In the agglomerative model-based hard clustering methods, one begins with a partition P of the data in which each sample is in its own singleton cluster. At each stage, two clusters are chosen from P and merged, forming a new partition P'. The pair which is merged is the one which gives the highest resulting likelihood (usually all merges will reduce the likelihood somewhat). The process is greedy; the best choice at a certain stage need not develop into the best likelihood at later stages.

A subtlety of model-based agglomerative clustering is that, by merging clusters, we are choosing new labels l_i at each stage. However, we do not explicitly choose model parameters θ. Rather, we implicitly consider θ to be the best possible θ for the chosen labels.

More formally, we have a label likelihood function J which assumes maximum likelihood parameters for each labeling.

$$J(l_1, \ldots, l_n; X) = \max_{\Theta} \mathcal{L}(\Theta; l_1, \ldots, l_n | X)$$

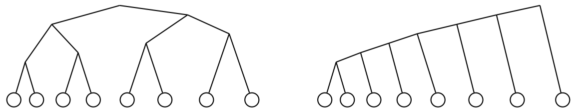
The relative cost of a merge from P to P' will be

$$\Delta J(P, P') = J(P')/J(P)$$

The best merge P to P' will be the one that maximizes $J(P')$, but procedurally we usually maximize ΔJ, which is equivalent.

3. Model-based Interpretation of Classical Agglomerative Algorithms

The classical agglomerative algorithms each define a distance function $d(C_i, C_j)$ between clusters (see figure 2), and at each stage merge the two closest clusters according to this distance function. In sections 3.1 through 3.4, we consider each of the four methods discussed above. For each method, we define an associated probabilistic model, and prove that the cost J for that model, the relative cost ΔJ, or a related bound, is monotonically non-increasing with $d(C_i, C_j)$. That is, each classical method discussed is equivalent to a specific model-based method, with minimum distance merges corresponding to maximum likelihood merges.[2] We write $f \sim g$ to indicate a quantity f is monotonically non-decreasing in a quantity g.



(a) (b)

Figure 1: Dendrograms for (a) complete-link (farthest members) and (b) single-link (nearest members) on the same one-dimensional data. Complete link forms balanced clusters while single-link grows paths.

3.1. Ward's Method

We begin by discussing Ward's method (Ward, 1963). Ward's method uses the error sum-of-squares criterion function to define the distance between two clusters:

$$d_{\text{Ward}}(C_1, C_2) = \text{ESS}(C_1 \cup C_2) - \text{ESS}(C_1) - \text{ESS}(C_2)$$

where the error sum-of-squares (ESS) is given by:

$$\text{ESS}(C_i) = \sum_{x \in C_i} (x - m_i)^2$$

and m_i is the sample mean of the data points in cluster C_i.

Ward's method is equivalent to a model-based agglomerative clustering method where the generating model is a mixture of spherical gaussians with uniform covariance σI. This model-based interpretation of Ward's method is well-known (Fraley & Raftery, 2000), but we present the proof here as an introduction to the proofs that follow in the next few sections.

Theorem 1 *If the probability model in equation 1 is multivariate normal with uniform spherical covariance σI, then $\Delta J \sim d_{\text{Ward}}$.*

Proof: The model parameters Θ for this model are the means $\mu_1, \ldots, \mu_k$. The component density $p(x_i|\theta_{l_i})$ is:

$$p(x_i|\sigma, \mu_{l_i}) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu_{l_i})^2/2\sigma^2}$$

where μ_{l_i} is the mean of the component l_i to which x_i belongs. Given a fixed assignment $l_1, \ldots, l_n$, the $\mu_1, \ldots, \mu_k$ which maximize $\mathcal{L}$ are the sample means for each cluster: $m_1, \ldots, m_k$. Therefore,

$$J(l_1, \ldots, l_n; X) = \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} e^{-(x_i-m_{l_i})^2/2\sigma^2}$$

and so if merging P to P' involves merging clusters C_1 and C_2 into C_3, with respective sample means m_1, m_2, and m_3,

$$\log[\Delta J(P, P')] = \sum_{i \in C_j} \left[\sum_{j=1}^{2} \frac{(x_i - m_j)^2}{2\sigma^2} \right] - \frac{(x_i - m_3)^2}{2\sigma^2}$$

[1] Note that this is different from model-based soft clustering (McLachlan & Peel, 2000), where each data point x_i is assigned to every cluster with probability $p(x_i|\theta_j)$ according to the mixture likelihood (with mixture weights τ):

$$\mathcal{L}(\theta_1, \ldots, \theta_k; \tau_1, \ldots, \tau_k | X) = \prod_{i=1}^{n} \sum_{j=1}^{k} \tau_j p(x_i|\theta_j)$$

[2] In some cases, the probabilistic model is only well-defined when the data are elements of a Euclidean space.

Method	$\mathbf{d(C_1, C_2)}$	**Probabilistic Mixture Model**
Single-link	$\min_{(x_1,x_2)\in C_1\times C_2} \|\|x_1 - x_2\|\|$	Branching Random Walks
Complete-link	$\max_{(x_1,x_2)\in C_1\times C_2} \|\|x_1 - x_2\|\|$	Uniform Equal-Radius Hyperspheres
Group-average	$\text{mean}_{(x_1,x_2)\in C_1\times C_2} \|\|x_1 - x_2\|\|^2$	Equal-Variance Configurations
Ward's method	$\text{ESS}(C_1 \cup C_2) - \text{ESS}(C_1) - \text{ESS}(C_2)$	Equal-Variance Isotropic Gaussians

Figure 2: Agglomerative methods and the probabilistic models they greedily optimize.

which is a negative multiple of

$$\text{ESS}(C_3) - \text{ESS}(C_1) - \text{ESS}(C_2)$$

Since the latter is exactly the quantity which Ward's method uses to select a merge, we are done.

3.2. Single-Link Clustering

In single-link clustering, the distance between clusters is defined to be the distance between their *closest* points:

$$d_{SL}(C_1, C_2) = \min_{(x_1,x_2)\in C_1\times C_2} d(x_1, x_2)$$

The probabilistic model corresponding to this clustering algorithm is a mixture of *branching random walks* (BRWs). A BRW is a stochastic process which generates a tree of data points x_i as follows: The process starts with a single root x_0 placed according to some distribution $p_0(x)$. Each node in the frontier of the tree produces zero or more children x_i.[3] The position of a child is generated according to a multivariate isotropic normal with variance σI centered at the location of the parent.

Theorem 2 *If the probability model in equation 1 is a mixture of branching random walks, then $\Delta J \sim d_{SL}$.*

Proof: The parameters Θ for a mixture of BRWs are the tree structures or *skeletons* for each component walk.[4] For a non-root sample i in a walk with skeleton T, we generate x_i, conditioned on the location of the parent $m_T(i)$ of i, according to:

$$\begin{aligned} p(x_i|T,\sigma) &= p_s(x_i|x_{m_T(i)}) \\ &= \frac{1}{\sigma\sqrt{2\pi}} e^{-(x_i - x_{m_T(i)})^2/2\sigma^2} \end{aligned}$$

Given $l_1, \ldots, l_n$, we wish to find Θ to maximize $\mathcal{L}$. Since our labels are fixed, all we can vary is the tree skeletons over each cluster. Notice that $\log \mathcal{L}$ is a constant plus a negative multiple of the sum of squared distances between each child in the dataset and its parent. Therefore, choosing the trees which minimize this sum will maximize $\mathcal{L}$. But those trees are just minimal spanning trees (MSTs) over the graphs in which each pair of points x and y in a cluster is connected by an arc of weight $(x - y)^2$. Therefore,

$$\log J(P) = \alpha - \sum_{C_i \in P} \text{MST}(C_i)$$

where $\text{MST}(C_i)$ is the cost of the MST over the squared distances.

Subtrees of MSTs are MSTs as well, so in if P merges to P' by joining clusters C_1 and C_2 into C_3, we can find an MST of C_3 by joining the MSTs of C_1 and C_2 with a single added arc. This arc will necessarily be an arc between a closest pair of points in $C_1 \times C_2$. The change in $\log J$, which is $\log \Delta J$, will then be the negative squared length between that pair. But a pair with minimum squared length also has minimum non-squared length, which is the criterion used by single-link clustering to select a merge.

3.3. Complete-Link Clustering

In complete-link clustering, the distance between clusters is defined to be the distance between their *farthest* points:

$$d_{CL}(C_1, C_2) = \max_{(x_1,x_2)\in C_1\times C_2} d(x_1, x_2)$$

It is commonly observed that complete-link clustering tends to form spherical clusters. We show here that this behavior is due to its associated probabilistic model, where points are uniformly generated on hyperspheres of equal-radius r. Fraley and Raftery (2000) suggest that complete-link is similar to, but not exactly, equivalent to a uniform hypersphere model. We show that, while this is strictly true, complete-link clustering (greedily) maximizes a tight lower bound on that likelihood.

Theorem 3 *If the probability model is a mixture of uniform-density equal-radius hyperspheres, then ΔJ is bounded by a function f such that $\Delta J > f \sim d_{CL}$.*

Proof: Let $B(z, r)$ be the hypersphere of radius r centered at z. The probability $p(x_i|\theta_{l_i})$ here is given by:

$$p(x_i|z_{l_i}, r) = \begin{cases} 1/volume(B(z_{l_i}, r)) & \text{for } x \in B(z, r) \\ 0 \quad \text{otherwise} \end{cases}$$

[3] The branching factor has an associated distribution, but it is not relevant for our analysis.

[4] For simplicity, we assume that the location of the root of a walk is generated uniformly at random over the *actual locations* of data points.

Let $\Theta = z_1, \ldots, z_k; r$. Given $l_1, \ldots, l_n$, we wish to find $z_1, \ldots, z_k; r$ to maximize $\mathcal{L}$. For each cluster C_k, there is some minimal enclosing hypersphere $B(z_i^*, r_i^*)$. The maximum of $\mathcal{L}$ will occur when Θ has $z_i = z_i^*$ and $r = \max r^*$. Therefore,

$$J(l_1, \ldots, l_n; X) = \prod_{i=1}^{n} \frac{\alpha}{r^d} = \alpha^n \frac{1}{r^{dn}}$$

for a positive α which depends only the dimensionality d of the data.

Therefore, the best merge at each stage will be the one which minimizes the new r. Define the *width* of a set to be the greatest distance between any two points in that set. At each stage, complete-link clustering chooses the merge that minimizes the maximum cluster width w. In one dimension, the radius r of the minimal enclosing 1-hypersphere (interval) of that set is equal to half its width. Therefore, for data that lies in one dimension, complete-link clustering exactly minimizes r at each stage by minimizing w. In higher dimensions, the relation $r = w/2$ no longer holds strictly. However, $w/2 \leq r \leq \beta w/2$ for some dimension-dependent constant β, $1 \leq \beta \leq \sqrt{2}$. Therefore, by minimizing w, complete-link clustering also minimizes a (relatively tight) upper bound on r at each stage in the algorithm.[5]

3.4. Group-Average Clustering

In typical group-average clustering, the distance between clusters is defined to be the the average distance between the points in the different clusters:

$$d(C_1, C_2) = \text{mean}_{(x_1,x_2)\in C_1 \times C_2} d(x_1, x_2)$$

We analyze the slightly different formulation in which the average *squared* distance is used. The generative process for group-average is slightly different than for the other methods. Here, we place all members of a cluster at once. Their joint locations are chosen uniformly from all configurations, subject to a maximum configuration variance. Formally, Θ is a (maximum) variance parameter v. Each cluster C is generated by choosing locations $x \in C$ such that $var(C) \leq v$. Clearly, then, the classification likelihood depends only on the maximum variance of the highest-variance cluster, with a lower maximum variance giving a higher classification likelihood.

Note that clusterings with small cluster variance are not the same as ones with small error-sum-of-squares (as with Ward's method). For example, to minimize ESS, a very distant outlier will be assigned to the cluster with the closest mean. However, to minimize cluster variance, it will be assigned to the densest cluster, where it will have the least impact on the maximum variance.

Theorem 4 *If the probability model generating the data is the stochastic process described above, then group-average maximizes a lower bound on J.*

Proof: For clusters C and D, let S_{CD} be the sum of squared distances between pairs in $C \times D$: $S_{CD} = \sum_{c\in C, d\in D}(c - d)^2$. The relation between cluster variance and average distances is given by the following identity:

$$\frac{1}{|C|}\sum_{c\in C}(c-\mu)^2 = \frac{1}{2|C|^2}\sum_{c_1\in C}\sum_{c_2\in C}(c_1 - c_2)^2 = \frac{1}{2|C|^2} S_{CC}$$

Essentially, the average internal-pair squared distance equals the average variance in a Euclidean space. Furthermore, the distance between the centroids of two clusters C and D is given by:

$$d_{\text{centroid}}(C, D) = \frac{1}{|C||D|} S_{CD} - \frac{1}{2|C|^2} S_{CC} - \frac{1}{2|D|^2} S_{DD}$$

We do not prove these identities here; they follow by induction from the law of cosines. The consequence of the latter is that, since the centroids have non-negative distance from each other, it must be that $S_{CC} \leq \frac{2|C|}{|D|} S_{CD}$ and $S_{DD} \leq \frac{2|D|}{|C|} S_{CD}$, which we will need later.

As argued above, the classification likelihood for this variance-limited probabilistic model is monotonically decreasing in the maximum cluster variance. By the variance-distance identity, the likelihood is thus also monotonically decreasing in the average of within-cluster distances for the highest-variance cluster E.

$$J \sim \frac{1}{|E|^2} S_{EE}$$

Now, if we chose the merge $E = C \cup D$ based on the average squared distances of all pairs inside the result cluster E, which is sometimes done, we would be greedily minimizing exactly the maximum cluster variance. However, in group-average, we more typically average only the pairs *between* the merging clusters C and D. Nonetheless, we know that

$$\begin{aligned} J &\sim \frac{1}{|E|^2} S_{EE} = \frac{1}{|C+D|^2}(S_{CD} + S_{CC} + S_{DD}) \\ &< \frac{1}{|C+D|^2}\left(S_{CD} + \frac{2|C|}{|D|} S_{CD} + \frac{2|D|}{|C|} S_{CD}\right) \\ &= \frac{2}{|C||D|} \frac{|C|^2 + |C||D|/2 + |D|^2}{|C+D|^2} S_{CD} \\ &< \frac{2}{|C||D|} S_{CD} \end{aligned}$$

[5] In higher dimensions, $w/2 \leq r \sin(\phi/2))$ where ϕ is the angle between two vertices of a regular hyperpyramid and its center. This angle is π in one dimension, and always less than $\pi/2$, hence the range on the bound.

But the last quantity is twice the quantity that this group-average variant actually does minimize. Therefore, it also minimizes a bound on J.

It is worth stressing that this bound is looser than the other bounds presented, and to our knowledge it is an open problem to supply a model for the non-squared formulation of group-average.

3.5. Practical Consequences

There are several practical consequences of the results presented in sections 3.1 through 3.4. First, it justifies the use of the classical agglomerative methods as well-founded probabilistic methods rather than just convenient heuristics.

Second, it explains the qualitative empirical behavior of the different classical methods on the basis of their associated probabilistic models.

Furthermore, in model-based agglomerative clustering, there are approaches to determining the number of clusters and the choice of clustering method based on model selection. These approaches can now be used with linkage-based agglomerative methods. The second two consequences are discussed in further detail in this section.

Finally, this formulation suggests the design of novel agglomerative clustering algorithms based on the classical agglomerative methods. This last consequence is explored in section 4.

3.5.1. PREDICTING ALGORITHM BEHAVIOR

As linkage-based methods are so commonly used, the qualitative empirical behavior of these algorithms is well-known. Single-link clustering tends to produce long straggly clusters, complete-link clustering tends to produce tight spherical clusters (see figure 1), and group-average clustering tends to produce clusters of intermediate tightness between single-link and complete-link.

Such behavior is unsurprising given these methods' associated probabilistic models. Data generated by a mixture of branching random walks is likely to have straggly patterns. Data generated uniformly over hyperspheres is likely to be spherical. And a distribution which generates configurations of equal variance will be somewhere in between, with wide tails on clusters being balanced by dense centers.

We present two examples here. Figure 3 shows data which was generated uniformly on two equal-radius hyperspheres, but is sampled much more lightly from one of the hyperspheres. Here, Ward's method does not identify the correct clusters, because it assumes that the data was generated by two gaussians – it uses its explanatory power to explain the halves of the dense region. Complete-link clustering, on the other hand, is tolerant of such sampling

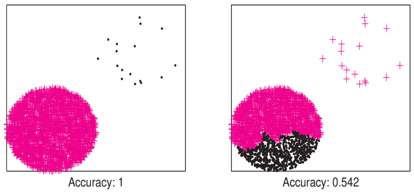


Figure 3: Data sampled from two circles, uniformly over each circle, but with very different densities. (a) Complete-link identifies the correct clusters. (b) Ward's method considers the points in the lightly sampled region outliers and tries to explain the dense region. In general, uniform distance models (complete-link, group average) will use their clusters to explain spatial extents, while gaussian algorithms (Ward's, k-means) will use their clusters to explain dense regions. Accuracy values in all figures are given by the Rand Index (Rand, 1971).

because the likelihood of the data is dependent only on the radius of the minimal spanning hypersphere. For the same reason, Ward's method is more tolerant of true outliers.

In figure 5, the data was generated by two direction-biased random walks. Single-link clustering finds the correct clusters, while the clusters found by complete-link clustering and Ward's method reflect the implicit spherical probabilistic models for these methods.

3.5.2. WHICH METHOD? HOW MANY CLUSTERS?

Often, one will have a general idea as to a probabilistic model that would plausibly have generated one's data. For example, in face recognition, faces are often modeled as deviations from a generic face, where face patterns have a multivariate gaussian distribution (McKenna et al., 1998). The probabilistic interpretation of these agglomerative methods suggests that one's choice of agglomerative clustering algorithm should be motivated by the probabilistic model that is believed to have generated the data.

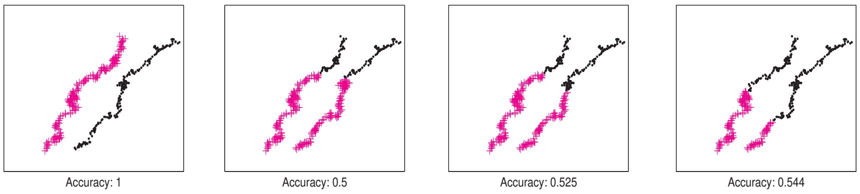
More rigorously, in model-based agglomerative clustering, determining the clustering method and the number of clusters is accomplished in a principled manner by using approximate Bayes' factors to compare models. The formulation of the linkage-based methods as model-based methods allows such an approach to model selection to be used in the context of linkage-based methods. An in-depth discussion of Bayesian model selection in clustering is outside of the scope of this paper, and we refer the interested reader to (Fraley & Raftery, 1998).

4. Extending Classical Agglomerative Methods

The probabilistic interpretation of the classical agglomerative clustering algorithms suggests extensions to these algorithms based on variants of the associated mixture mod-

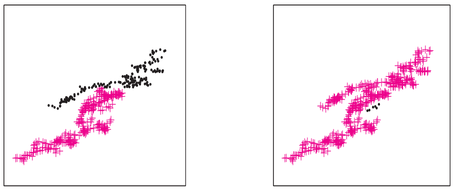
Method	$\mathbf{d(C_1, C_2)}$	**Probabilistic Mixture Model**
Line-link	$\mathrm{TPSE}(C_1 \cup C_2) - \mathrm{TPSE}(C_1) - \mathrm{TPSE}(C_2)$	Linear Random Walks
Adjusted Complete-link	$span(C_1 \cup C_2) - \max\{span(C_1), span(C_1)\}$	Uniform Variable-Radius Hyperspheres
Mahalanobis-link	$\mathrm{ESSM}(C_1 \cup C_2) - \mathrm{ESSM}(C_1) - \mathrm{ESSM}(C_2)$	Equal-Variance Non-Isotropic Gaussians

Figure 4: New agglomerative methods and the probabilistic models they greedily optimize.



SINGLE-LINK (a) COMPLETE-LINK (b) GROUP-AVERAGE (c) WARD'S METHOD (d)

Figure 5: Directional random walks are easily found by single-link clustering, but the other methods' implicit models cause them to find more spherical clusters.



Actual Walks (a) Maximum Likelihood Walks (b)

Figure 9: Even for synthetic data, maximum likelihood walks (b) can be very different from the walks that generated the data (a).

els. More specifically, we may want to alter the merge costs to reflect the types of patterns we wish to find in the data. We present three such extensions here, discuss their associated probabilistic models, and compare their empirical performance to the agglomerative methods discussed in section 3.

4.1. Line-Link

Single-link clustering has historically achieved poor classification performance. There are two primary reasons. First, in applications where clustering is useful, data is rarely generated by branching random walks. Second, even when data is truly generated by branching random walks, the maximum-likelihood random walks are unlikely to be the ones which actually generated the data (see figure 9). In general, branching random walks that are close or overlapping are difficult to separate in an unsupervised manner.

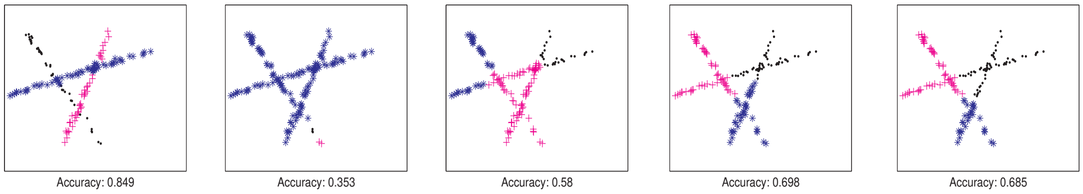
Although, single-link clustering remains accurate in cases where the data is generated by a well-separated mixture of Markov processes, it would be be useful to have hierarchical methods which are capable of correctly identifying non-spherical trends.

Here, we present *line-link* agglomerative clustering, where the model is that data points are generated along line, but with gaussian perpendicular displacement. One can think of this as data generated by some process traveling along a line, and emitting points along the way. This is a plausible model for earthquake epicenters on the same seismic fault, or GPS data from cars traveling on the same road.

Since we know the model, we could easily use a hard partitional clustering according to the model using a classification EM procedure as in Murtagh and Raftery (1984). We would iteratively assign points to the closest line and move each line to best fit the points assigned to it.

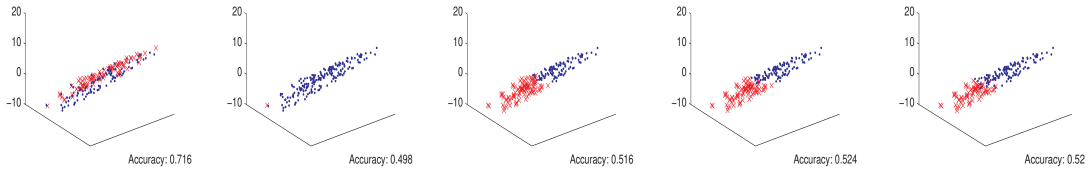
However, if we want a hierarchical clustering, for example if we want to be able to sub-divide major fault families into smaller minor faults, or split roads into lanes, it would be useful to have an agglomerative algorithm for this model. Our likelihood according to this model, for fixed line parameters, will be monotonic in the sum of squared distances from each point to its assigned line. Thus, for each cluster, we will track the total perpendicular squared error (TPSE) from that cluster's best-fit line.[6] For each pair of clusters, we track the cost of merging them, which will be the difference between the best total squared error for the joint cluster and the sum of the best total squared errors of the two component clusters. Note that there is no necessary relation between the three clusters' best-fit lines' parameters. It should be clear that, by design, this algorithm greedily maximizes the desired likelihood. This algorithm, like all agglomerative methods, can be made to run in time $O(n^2(f(n) + \log n))$ where n is the number of points to cluster and $f(n)$ is the cost of calculating the

[6]We calculate this using a conjugate-gradient method, but any numerical optimization will do.



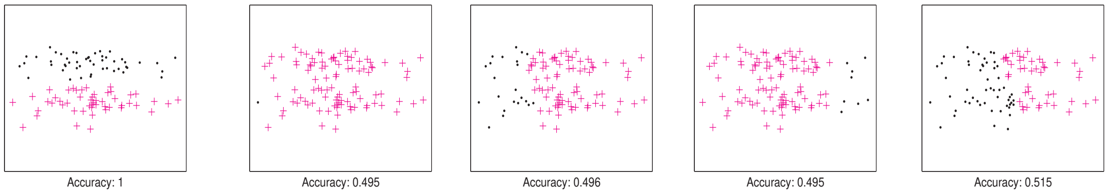
LINE-LINK (a) | SINGLE-LINK (b) | COMPLETE-LINK (c) | GROUP-AVERAGE (d) | WARD'S METHOD (e)

Figure 6: Crossing lines are only recovered by LINE-LINK (a). SINGLE-LINK makes a huge cluster with outliers (b), while the other methods slice the data into spatially balanced regions.



LINE-LINK (a) | SINGLE-LINK (b) | COMPLETE-LINK (c) | GROUP-AVERAGE (d) | WARD'S METHOD (e)

Figure 7: Crabs data: on this difficult set, only LINE-LINK (a) is able to detect the correct overall trend, in which the principle component of the data is not explanatory.



MAHALANOBIS-LINK (a) | SINGLE-LINK (b) | COMPLETE-LINK (c) | GROUP-AVERAGE (d) | WARD'S METHOD (e)

Figure 8: When the data is generated by non-isotropic gaussians, Mahalanobis-link can detect the clusters more reliably, essentially by linearly transforming the data into a space where the clusters become spherical.

merge between two clusters.

In figure 6, we show that line-link works far better than the other agglomerative clustering algorithms in the case where the data are actually generated by walks along lines. In figure 7, we show the performance of line-link clustering on crabs data from (Campbell & Mahon, 1974). In the crabs data set, the instances represent different crabs, the features represent structural dimensions, and the classes correspond to different species of crabs. In this data set, crabs' absolute proportions vary roughly linearly with their general size, and so the data for a given species can be viewed as being generated by a linear random walk along a size/age axis which emits crabs of slightly different relative proportions along the way.

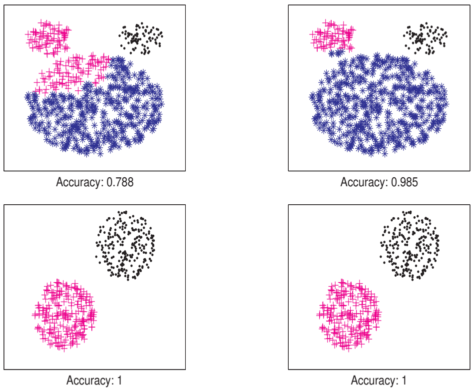
It should be stressed that the crabs set is quite difficult for most clustering algorithms. The principal direction of the data is, roughly, crab size, and is very decorrelated from the desired distinction, which is crab species. Spherical algorithms generally identify big crabs vs. little crabs, while single-link identifies a single outlier vs. all other crabs. Ripley (1996) and others generally discard the first component, and then are able to cluster the data readily. However, an appropriate model means that we do *not* have to preprocess the data to make it fit our algorithm.

4.2. Adjusted Complete-Link

In complete-link clustering, the assumption that the data is generated by hyperspheres of equal radius may be inappropriate for the data. If we expect that the data will be spherical, but on spheres of varying radii, we can make a small change to the complete-link distance which gives us exactly this model. In *adjusted complete-link clustering*, the distance between two clusters is defined not by the result width, but by the *increase* in width over the larger of the two merged clusters' widths. Formally,

$$d_{ACL}(C_1, C_2) = \mathrm{width}(C_1 \cup C_2) - \max_{i \in \{1,2\}} \{\mathrm{width}(C_i)\}$$

This change is easily implemented, and is equivalent to choosing the merge that maximizes the likelihood that the



COMPLETE-LINK (a)   ADJUSTED-COMPLETE-LINK (b)

Figure 10: Complete-link (a) is unsuited to clusters of widely varying size; adjusted complete-link (b) is more appropriate for this situation.

data was generated uniformly on hyperspheres of arbitrary radius. The proof is similar to the proof of theorem 2, and we do not include it here. In figure 10, we show how adjusted complete-link compares to complete-link in both the case where the data is generated uniformly on hyperspheres of equal radius and the case where the data is generated uniformly on hyperspheres of (possibly) unequal radii.

4.3. Mahalanobis Link

In section 3.1, we mentioned that the model assumed by Ward's method is a mixture of multivariate gaussians with the uniform spherical covariance σI. If we assume that the data is generated by a mixture of multivariate gaussians with a common, known covariance matrix Σ, we can modify Ward's method to minimize the increase in sum of squared Mahalanobis distances at each merge. Formally,

$$d_{ML}(C_1, C_2) = \mathrm{ESSM}(C_1 \cup C_2) - \sum_{i \in \{1,2\}} \mathrm{ESSM}(C_i)$$

where

$$\mathrm{ESSM}(C_i) = \sum_{x \in C_i} (x - m_i)^T \Sigma (x - m_i)$$

We show in figure 8 how this method, which we call Mahalanobis-link clustering, compares to Ward's method in the case where the data is generated by a mixture of gaussians with known covariance $\Sigma \neq \sigma I$. Mahalanobis-link can detect the clusters more reliably, essentially by linearly transforming the data into a space where the clusters become spherical. In the case that Σ is diagonal, this is equivalent to feature weighting.

5. Conclusion

We have presented probabilistic interpretations of the classical agglomerative clustering algorithms – single-link, complete-link, group-average, and Ward's method – based on greedy maximum-likelihood estimation for finite mixture models. The framework of model-based clustering enables us to better understand the classical methods, and suggests a principled approach to developing variants of these methods. We have introduced three novel agglomerative methods – line-link, adjusted complete-link, and Mahalanobis-link – and have argued their utility. These methods are easily implemented, and the model-based perspective presented allows easy evaluation of which methods are most likely to be effective on a given problem.

Acknowledgements

This paper is based on work supported in part by the National Science Foundation (under Grant No. IIS-0085896 and by an NSF Graduate Fellowship), and by the research collaboration between NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation and CSLI, Stanford University.

References

Banfield, J. H., & Raftery, A. E. (1993). Model-based gaussian and non-gaussian clustering. *Biometrics*, *49*, 803–821.

Campbell, N. A., & Mahon, R. J. (1974). A multivariate study of variation in two species of rock crab of genus *Leptograpsus*. *Australian Journal of Zoology*, *22*, 417–425.

Celeux, G., & Govaert, G. (1993). Comparison of the mixture and the classification maximum likelihood in cluster analysis. *Journal of Statistical Computation and Simulation*, *47*, 127–146.

Dom, B., & Vaithyanathan, S. (1999). Model selection in unsupervised learning with applications to document clustering. *The Sixteenth International Conference on Machine Learning*.

Fraley, C., & Raftery, A. E. (1998). *How many clusters? Which clustering method? Answers via model-based cluster analysis* (Technical Report 329). Department of Statistics, University of Washington, Seattle, WA.

Fraley, C., & Raftery, A. E. (2000). *Model-based clustering, Discriminant Analysis, and Density Estimation* (Technical Report 380). Department of Statistics, University of Washington, Seattle, WA.

Jain, A., Murty, M., & Flynn, P. (1999). Data clustering: a review. *ACM Computing Surveys*, *31*, 264–323.

McKenna, S., Gong, S., & Raja, Y. (1998). Modelling facial colour and identity with gaussian mixtures. *Pattern Recognition*, *31*, 1883–1892.

McLachlan, G., & Peel, D. (2000). *Finite mixture models*. New York: Wiley.

Murtagh, F., & Raftery, A. E. (1984). Fitting straight lines to point patterns. *Pattern Recognition*, *17*, 479–483.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, *66*, 846–850.

Ripley, B. (1996). *Pattern recognition and neural networks*. Cambridge University Press.

Ward, J. H. (1963). Hierarchical groupings to optimize an objective function. *Journal of the American Statistical Association*, *58*, 234–244.