Proximity Curves for Potential-Based Clustering

Attila Csenki[1] ⬤ **· Daniel Neagu[1] · Denis Torgunov[1] · Natasha Micic[1]**

Abstract

The concept of proximity curve and a new algorithm are proposed for obtaining clusters in a finite set of data points in the finite dimensional Euclidean space. Each point is endowed with a potential constructed by means of a multi-dimensional Cauchy density, contributing to an overall anisotropic potential function. Guided by the steepest descent algorithm, the data points are successively visited and removed one by one, and at each stage the overall potential is updated and the magnitude of its local gradient is calculated. The result is a finite sequence of tuples, the *proximity curve*, whose pattern is analysed to give rise to a deterministic clustering. The finite set of all such proximity curves in conjunction with a simulation study of their distribution results in a *probabilistic clustering* represented by a distribution on the set of dendrograms. A two-dimensional synthetic data set is used to illustrate the proposed potential-based clustering idea. It is shown that the results achieved are plausible since both the 'geographic distribution' of data points as well as the 'topographic features' imposed by the potential function are well reflected in the suggested clustering. Experiments using the Iris data set are conducted for validation purposes on classification and clustering benchmark data. The results are consistent with the proposed theoretical framework and data properties, and open new approaches and applications to consider data processing from different perspectives and interpret data attributes contribution to patterns.

✉ Attila Csenki
  a.csenki@bradford.ac.uk

  Daniel Neagu
  d.neagu@bradford.ac.uk

  Denis Torgunov
  d.torgunov@bradford.ac.uk

  Natasha Micic
  n.micic@bradford.ac.uk

[1] Artificial Intelligence Research Group, University of Bradford, Bradford BD7 1DP, UK

1 Introduction

The history of work on clustering algorithms based on, and motivated by the gravitational (and electrostatic) models stretches back to the 1970s. They are too numerous to be exhaustively reviewed here; we review below a few relevant highlights, thereby placing our work into context and pointing out its distinguishing features. For a general review of clustering techniques, see Berkhin (2006) and for a recent survey see Xu and Tian (2015).

One of the earliest papers of cognate interest is Wright (1977b), where the attractive force (between the particles) under Newtonian gravity moves the data points to form successively larger and fewer clumps, eventually resulting in all the mass being lumped together in one single cluster.

Starting with a set of data points where each point has a potential, in Shi et al. (2002), an agglomerative hierarchical clustering algorithm is presented based on a comparison of distances between already defined clusters. The distance used in Shi et al. (2002) is calculated as some weighted value of differences of potentials.

In Lu and Wan (2012), each of the data points is endowed with a potential (of a Newtonian type), and the points are sorted in a list in ascending order of their potential. Cluster centres are nominated as the list is being worked through by using a preselected bandwidth parameter; being able to measure the distance between points is an essential feature of the algorithm.

An interesting recent paper with a philosophical flavour is Henning (2015). The authors reason that there is no such thing as the 'best' clustering algorithm since the notion of a cluster will be dependent on the context and on the intention of the investigator. On the other end of the spectrum, there is the early paper Wright (1977a), seeking to make clustering into an objective endeavour by assuming a priori a 'clustering function' which then is used to assess the quality of the clustering achieved.

The view taken here is that 'reality' and 'truth' are elusive concepts and can be at best of partial concern to the modeller; important will be that, initially, a collection of (more or less *plausible*) models is available, and out of these the one is chosen which most successfully predicts future events. We want our model to be understood and received in this spirit: the model put forward is rooted in the physics of own data attributes, it is rich in structure and appears plausible in certain circumstances.

The algorithm suggested here is an hierarchical clustering algorithm for a finite set of data points in the Euclidean space, each of which is assumed to be generating a potential field constructed by means of the density of a certain family of probability distributions. The individual point potentials can be *anisotropic*, thus allowing the surface representing the total potential to be thought of (in two dimensions for the sake of simplicity) as an undulating landscape with hills and valleys extending in any direction. This construction will allow clusters of data points on a 'real' surface to be modelled, a feature which may turn out to be useful in future work in epidemic modelling using networks (see Barabási 2016, Chapter 10). The novelty of our model also lies in the fact that the result is not a single dendrogram but a *randomized dendrogram*, defined as a convex combination of deterministic dendrograms, thus allowing the results to be formulated in a probabilistic framework.

The important novel feature introduced here is that of the *proximity curve* whose *pattern* will be the key for determining the suggested clustering. A welcomed by-product of our algorithm is that the *black hole problem* (as a local minima challenge discussed for example in Li and Fu 2011) does not appear here: all massive data points will be 'collected' prior to visiting genuine clusters comprising smaller data points.

In the next section, we place the paper's topic into context and introduce some notation. Then, in the main Section 3, the new proximity curve concept and proposed algorithm are described in detail and illustrated by using a synthetic data set. In Section 4, experiments using the Iris data set (Fisher 1936, 2011) are conducted for validation purposes on classification and clustering benchmark data. In Section 5, the results and various features of the suggested technique and its merit are highlighted. In Section 6, our findings are summarised and some future research ideas are indicated.

Appendices A – C contain auxiliary theoretical material, whereas all nontextual material namely contour maps, proximity curve, potential curve, dendrogram, pseudocodes, tables and large matrices are collected in Appendix E.

2 Context and Assumptions

We are given N *data points* $\mathbf{x}_1, \ldots, \mathbf{x}_N$ in the Euclidean space, assumed here for the sake of simplicity to be the 2-dimensional plane $\mathbb{R}^2$. This is not a technical limitation as such; it is intended merely as a vehicle for conveying the underlying principles. The ith point induces the potentials v_i, generating the total potential as follows:

$$V(\mathbf{y}) = \sum_{i=1}^{N} v_i(\mathbf{y}), \quad \mathbf{y} \in \mathbb{R}^2.$$

Depending on the context and emphases, the data points may be referred to also as *nodes*, *point masses*, *point charges* or simply *points*. The collection (set) of all data points may be referred to as a *data cloud*.

The structure of the point potentials v_i is as follows:

$$v_i(\mathbf{y}) = -M_i f(\mathbf{y} | \boldsymbol{\theta}_i),$$

where $M_i > 0$ is the 'mass' of point i and $f(. | \boldsymbol{\theta})$ stands for a Cauchy probability density functions (pdf) with parameter $\boldsymbol{\theta}$. The parameter $\boldsymbol{\theta}$ has five components, reflecting the geometric transformation steps carried out to convey the *seed distribution* (a standard Cauchy) to the required Cauchy distribution; the construction involved is described in detail in Appendices A – C.

Our running example is a synthetic data set comprising $N = 19$ points in the plane shown in Fig. 2 with individual and joint equipotentials. The corresponding parameter values are shown in Table 2.

We will be interested in an algorithm for finding data clusters based on the points' location *and* the overall potential field generated by them.

The physical models of Gravity and Electrostatics serve us as an intuitive background (e.g. Ramsey 1959; Kip 1962). The field's gradient vector $\nabla V(\mathbf{y})$ is the force exerted onto an infinitesimal unit 'test mass' or 'test charge' located at $\mathbf{y}$. The gradient vector $\nabla V(\mathbf{y})$ will be used to reflect qualitatively on where the point $\mathbf{y}$ in relationship with the others lies: is $\|\nabla V(\mathbf{y})\|$, the *magnitude* of the gradient, close to zero, so is $\mathbf{y}$ near the bottom of a valley (i.e. near a local minimum of V), whereas 'noticeably positive' values of $\|\nabla V(\mathbf{y})\|$ are associated with the point $\mathbf{y}$ being on a slope on the surface describing V.[1] Respective examples in Fig. 2 are the points #10 and #6. (See Table 2 for the points' numbering.)

---

[1]The point $\mathbf{y}$ may also be located near a ridge if $\|\nabla V(\mathbf{y})\| \approx 0$, but this possibility we choose to ignore as in our model is unlikely to occur.

Our aim is to discover data clusters by considering properties derived from the gradient field. Finite lists of tuples of real numbers will be constructed (we shall call them *proximity curves*) whose *pattern* will suggest ways of clustering the data.

3 The Algorithm: Cluster Discovery by Proximity Curves

3.1 Proximity Curves

Initially, all data points are said to be *active*. Starting from some initial position, steepest descent is applied to the total potential generated by all the active points, guiding us to one of the data points. The point thus found is then *deactivated*, the total potential is redefined (i.e. updated) to be the one generated by the now active points and the process is restarted at the position of the point just deactivated. This process is carried out until the last point has been found upon which all points become inactive and the total potential is zero.

The core of this algorithm is PROXIMITYCURVE; it is shown as Pseudocode E-2. As can be gleaned from Pseudocode E-2, in the course of the computations, a list of pairs is being built up, each entry comprising the label of the node being deactivated and the logarithm of the vector norm of the gradient of the updated potential at the node's location. (The algorithm CLOSESTACTIVE, shown as Pseudocode E-3, is an auxiliary.) Figure 3 illustrates the progress of the algorithm with a snapshot of the equipotentials just after having deactivated seven data points.

A *proximity curve*, illustrated by Fig. 4, is a plot of the logarithm of the vector norm of the gradient of the successively updated potential (as described above) versus the points' position in the list (the logarithm applied to the magnitude of the gradient is merely there to express and amplify an existing pattern). In total, there will be at the most N proximity curves as in practice some nodes cannot be reached to be the first node to be visited and since the first node visited completely determines the rest of the proximity curve. The full information about all possible proximity curves can be conveyed by two square matrices of size N:

– The rows of matrix $\mathbf{S} = \left(s_{ij}\right)_{i, j=1,\ldots,N}$ record all possible sequences of nodes, thus obtainable.
– The rows of matrix $\mathbf{\Lambda} = \left(\lambda_{ij}\right)_{i, j=1,\ldots,N}$ are the logarithm of the norm of the gradient of the corresponding successively updated potential at the respective node locations.

This is illustrated in Fig. 4 by the proximity curve whose first node is #2. (It was obtained by starting PROXIMITYCURVE in $(-2, -6)$.) The curve shown in Fig. 4 is obtained by combining the second rows of each of the matrices $\mathbf{S}$ and $\mathbf{\Lambda}$ (shown in Appendix E.4) into the list of pairs

$$\mathbf{L} = \left[\left(s_{21}, \lambda_{21}\right), \left(s_{22}, \lambda_{22}\right), \left(s_{23}, \lambda_{23}\right), \ldots, \left(s_{2(N-1)}, \lambda_{2(N-1)}\right), \left(s_{2N}, \lambda_{2N}\right)\right]$$
$$= [(2, -3.39), (1, -4.07), (16, -2.12), \ldots, (7, -2.46), (9, -\infty)]. \qquad (1)$$

3.2 Pattern Extraction

The curve shown in Fig. 4 has roughly a self-similar structure (in a statistical sense) akin to the *Elliott Wave Pattern* described in Casti (2002) in the context of financial data. The *principle* formulated below will guide us when using proximity curves for identifying clusters of data points.

Guiding Principle. Large values of the norm of the gradient of the updated potential function are indicative of the beginning of a new cluster. Decreasing values of the norm of the gradient indicate a lack of 'attractive mass' of what is left in the current cluster. *Cluster boundaries are therefore marked by local minima of the proximity curve.*

The *pattern* of a proximity curve is explored by a *recursive* algorithm.

– The *recursive step* is in locating the smallest local minimum. In the example shown in Fig. 4, this occurs at node #13 as is seen from the appropriate section of the list representation of the proximity curve (1) as follows:

$$\mathbf{L} = [(2, -3.39), \ldots, (15, -2.68), (13, -7.72), (19, -7.14), \ldots, (9, -\infty)]$$

The proximity curve $\mathbf{L}$ is now recognised as the concatenation of the two lists:

$$\mathbf{L}_{\text{left}} = [(2, -3.39), \ldots, (15, -2.68), (13, -7.72)]$$

and

$$\mathbf{L}_{\text{right}} = [(19, -7.14), \ldots, (9, -\infty)],$$

each of which is a proximity curve in its own right. Obviously, if local minima determine cluster boundaries, then $\mathbf{L}_{\text{left}}$ and $\mathbf{L}_{\text{right}}$ give rise to the clustering as follows:

$$\{\{2, \ldots, 15, 13\}, \{19, \ldots, 7, 9\}\}, \tag{2}$$

defining level 2 in the hierarchical clustering scheme. The next level of clustering will be arrived at by subjecting each of the lists $\mathbf{L}_{\text{left}}$ and $\mathbf{L}_{\text{right}}$ to the same recursive step. (A list not containing a local minimum will not be expanded further but will be replaced by a list containing the list itself as the only entry.)

– The *base case* is arrived at if none of the input lists can be written as the concatenation of two non-empty lists as described above.

3.3 Deterministic Dendrograms

The algorithm in Section 3.2 can be used for obtaining a nested list representation of the set of nodes indicating the clustering thus found; for the present example, this is as follows:

[[[[[2, 1]], [[16, 17, 18], [3, 4, 5]]], [[[14, 12, 15, 13]]]], [[[[19, 10, 11, 8]]], [[[6, 7, 9]]]]].

The dendrogram shown in Fig. 5 is arrived at by parsing this nested list structure. The upper hyphenated line in Fig. 5 indicates the clustering shown in Eq. 2 (at depth 2); in full,

$$\{\{2, 1, 16, 17, 18, 3, 4, 5, 14, 12, 15, 13\}, \{19, 10, 11, 8, 6, 7, 9\}\}.$$

The lower hyphenated line indicates the clustering found at depth 3 (see Fig. 5),

$$\{\{\{2, 1, 16, 17, 18, 3, 4, 5\}, \{14, 12, 15, 13\}\}, \{\{19, 10, 11, 8\}, \{6, 7, 9\}\}\}.$$

3.4 Probabilistic Dendrograms

Every proximity curve gives rise uniquely to a fully developed dendrogram as described in Sections 3.2 – 3.3. Not all conceivable proximity curves can be obtained by the present algorithm, however, as is illustrated by the running example. We are going to describe here how those reachable can be combined to a *probabilistic* (or *randomized*) dendrogram. Such a dendrogram allows the user to reason probabilistically and motivate a choice of starting

point. Questions like "What is the *probability* that two given points belong to the same cluster?" can then be meaningfully addressed.

Choose a window of interest $\mathcal{W} \subset \mathbb{R}^2$. To any given position $\mathbf{u} \in \mathcal{W}$, a possible dendrogram is assigned in the following manner. $\mathcal{W}$ is chosen so that it defines a window that the entire data set fits into, and that will provide a selection of possible points which are "sensible", i.e. could have been observed given the context of the initial data.

To arrive at $\mathbf{z} = \text{STEEPESTDESCENT}(\mathbf{u}, V)$, use Pseudocode E-1 from Appendix E.3. The point $\mathbf{z}$ will be the position of a local minimum of the potential V. Find the data point $\mathbf{x}_{\iota(\mathbf{u})} \in \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ which is closest to $\mathbf{z}$. Circumstances will be in practice such that $\mathbf{z}$ is unique (as far as the numerical procedure allows), resulting in a unique choice of the closest point $\mathbf{x}_{\iota(\mathbf{u})}$. The process thus described defines a mapping as follows:

$$\iota : \mathcal{W} \to \{1, \ldots, N\}$$
$$\mathbf{u} \mapsto \iota(\mathbf{u})$$

Running Pseudocode E-2 (see Appendix E.3) with the initial vector $\mathbf{z}^{(\text{start})} = \mathbf{u}$ will result in *the* proximity curve whose first node is labelled $\iota(\mathbf{u})$. The proximity curve thus found is unique in that no two different proximity curves have the same first node. The dendrogram based on this proximity curve will be denoted by $\mathcal{D}_{\iota(\mathbf{u})}$.

Let $\mathbf{u}_1, \ldots, \mathbf{u}_\ell$ be a random sample of size ℓ from the uniform distribution on $\mathcal{W}$. We define the *dendrogram probability vector* $\mathbf{d}$ by the following:

$$\mathbf{d} = (d_1, \ldots, d_N),$$

whose kth component

$$d_k = \frac{1}{\ell} \sum_{j=1}^{\ell} I_{\{\iota(\mathbf{u}_j)=k\}} \tag{3}$$

is the relative frequency of the node k having been chosen as the first data point for the corresponding proximity curve, where

$$I_B(x) = \begin{cases} 1 & \text{if } x \in B \\ 0 & \text{if } x \notin B \end{cases} \tag{4}$$

is the indicator function of a given set B.

The probability of two given nodes $m_1 \neq m_2$ being in the same cluster can now be evaluated by the following:

$$P \text{ (nodes } m_1 \text{ and } m_2 \text{ are in the same cluster)} =$$

$$\sum_{k=1}^{N} d_k P \text{ (nodes } m_1 \text{ and } m_2 \text{ are in the same cluster}|\mathcal{D}_k \text{ applies)} . \tag{5}$$

The conditional probabilities in Eq. 5 take values in $\{0, 1\}$; they are obtained by inspecting dendrogram $\mathcal{D}_k$. There are in total $N(N-1)/2$ probabilities defined in Eq. 5.

Probabilities for i pairwise different nodes $m_1, \ldots, m_i$ being in the same cluster can be evaluated analogously; there are $\binom{N}{i}$ such values.

Based on a sample of size $\ell = 1,000$, the dendrogram probabilities were calculated by Eq. 3; they are shown in Table 3. Notice that the data points 6, 7, 9, 13, and 15 (and the corresponding dendrograms) are *unreachable*, which is a consequence of none of these points being located close enough to a valley (a local minimum) of V, or, more precisely, for all of these data points there is another data point closer to the candidate minimum location; see Fig. 2, the right-hand box.

Consider $m_1 = 16$, $m_2 = 18$ as an example. It can be seen by inspection that out of the 14 possible dendrograms (not shown here), each of the following 9 will assign these two nodes to the same cluster:

$$\{\mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_4, \mathcal{D}_8, \mathcal{D}_{10}, \mathcal{D}_{11}, \mathcal{D}_{12}, \mathcal{D}_{16}, \mathcal{D}_{19}\}.$$

It is therefore

$$P \text{ (nodes \#16 and \#18 are in the same cluster)}$$
$$= d_2 + d_3 + d_4 + d_8 + d_{10} + d_{11} + d_{12} + d_{16} + d_{19} = 0.698$$

In the above calculations and example, fully developed dendrograms were considered only (requiring us to descend to depth 5). In general, the probability of any given set of nodes belonging to the same cluster can be calculated for any required level in a similar manner.

3.5 Implementation

A suite of programmes has been written to implement the algorithms. The functions in Appendix E.3 were implemented in R since this has excellent programming and plotting facilities for data analytics. The pictorial and computational output of the R programs are the contour plots and the proximity curve, the latter internally represented as a list-of-tuples which then serves as an input to a recursive Haskell programme returning the dendrogram as a nested list. Haskell was also used to produce LaTeX code for drawing the dendrogram (as a tree) as exemplified in Fig. 5.

The associated code and data are available from
http://www.comp.brad.ac.uk/~dneagu/proximity

4 Experimental Work

In order to evaluate the accuracy achieved by the clustering method proposed, we use the Iris data set (Fisher 2011), which is commonly used in the literature. The Cauchy parameters are selected from the data set as detailed below. The Iris data set has the following fields:

– Sepal width (S.W.)
– Sepal length (S.L.)
– Petal width (P.W.)
– Petal length (P.L.)

In order to test how the algorithm behaves under different assignments of those fields to Cauchy distribution parameters, we varied which Iris data set field gets assigned to which parameter (μ_1, μ_2, c_1, c_2).

For each of the inputs, the mass is assumed to be 1, and the angle parameters are computed as follows:

$$\alpha_i = 2\pi \left(\frac{\mu_{i1} + \mu_{i2} + c_{i1} + c_{i2}}{\max(\boldsymbol{\mu}_1) + \max(\boldsymbol{\mu}_2) + \max(\mathbf{c}_1) + \max(\mathbf{c}_2)} \right)$$

It should be noted that this only one pragmatic assignment of α_i, but in principle other functions of the other parameters can be used.

In addition, since we know which points should be clustered together (i.e. those belonging to the same Iris species), we can evaluate the accuracy for a given clustering (a given depth of the dendrogram). The accuracy is measured by determining what proportion of the

data points belong to the correct cluster. The class label assigned to the cluster is chosen based on what class the majority of the points in the cluster belong to. The full results, for all possible parameter assignments, are given in Appendix E.5.

We consider here the 4 parameter assignments shown in Table 1, along with the related accuracies, for the sake of simplicity (the experimental work has covered the entire range of possible permutations; please see Appendix E.5 for the entire list of results). Accuracy at depth 1 is not shown, as that represents the root of the tree, i.e. the point before any clustering has been attempted. This information is visualised in Fig. 7. As evidenced by Table 1, the accuracy achieved is highly dependent on the parameter assignments chosen.

For example, the best overall result, as well as the fastest, is obtained for θ_3. That, after the first split, gives the accuracy of 0.593, which grows to 0.607 after one more iteration, and reaches 0.860 at dendogram depth 5. These results validate our theoretical assumption that data attributes translated into Cauchy parameters (which give rise to a particular proximity curve) influence and determine cluster definitions. The proposed algorithm allows further tuning in subsequent iterations, creating the opportunity to further improve classification results.

5 Discussion

The distinguishing feature of our algorithm is that the data points are kept stationary while the space is explored with a moving unit test charge (as in electrostatics) or test mass (as in gravity) to discover clusters. In doing so, the test charge will be attracted to groups of charges (or masses), depending on the force of attraction as measured by the gradient of the potential. As a data point is 'discovered', it is deactivated (i.e. removed) and the gradient of the updated potential field is used to guide us to discover the next data point. By successively removing data points, the current cluster gets *depleted*, resulting in a steady weakening of attraction to what remains of the cluster. Eventually, the current cluster gets *exhausted*, indicated by the small magnitude of the gradient of the potential. This then is the sign for starting a new cluster: the magnitude of the gradient of the total potential of the remaining active points begins to increase. The *pattern* of the proximity curve thus constructed holds the clue to the definition of the clustering.

The process described above suggests a *nested structure* of clusters inherent in hierarchical clustering.

Individual proximity curves are usually suitable for finding the clustering at the *local level* because removal of points may eventually noticeably interfere with the global interplay of individual point potentials. The effect of 'point removal' inherent in the technique is intended to be counteracted by taking into account all possible (reachable) proximity curves in a probabilistic sense.

Table 1 Parameter assignments and accuracy measures

i	θ_i				Accuracy per depth			
	μ_1	μ_2	c_1	c_2	2	3	4	5
1	S.L.	P.L.	S.W.	P.W.	0.387	0.693	0.733	0.793
2	S.L.	P.W.	S.W.	P.L.	0.360	0.380	0.600	0.640
3	S.W.	P.L.	S.L.	P.W.	0.593	0.607	0.713	0.860
4	S.W.	P.W.	S.L.	P.L.	0.473	0.587	0.640	0.733

The dendrogram in Fig. 5 in conjunction with the set's joint equipotentials in Fig. 2 shows that the clustering suggested is intuitively plausible. The fact that point #19 is not grouped together with the points {16, 17, 18} is attributable to the fact that #19, even though it is physically close to the group, is separated from it by a ridge. Figure 3 shows a snapshot of the node deactivation process. It is seen that the algorithm duly observes both the mutual closeness of data points as well as the topography of the potential surface.

Finally, it is instructive to see that the north-eastern group of data points {14, 12, 15, 13} is separated by the dendrogram from the south-western group {6, 7, 9} even though the locations of all these points are roughly on the same potential level, as shown in Fig. 6. This illustrates once more the point that the algorithm takes into account mutual geographic proximity of points as well as surface topography.

6 Conclusion and Outlook

The new concept of the *proximity curve* has been introduced and used for finding clusterings in finite sets of data points in the Euclidean space endowed with individual potentials derived from a multivariate Cauchy distribution. The finite collection of all proximity curves gives rise to a *probabilistic dendrogram*. A synthetic example comprising 19 data points was taken to illustrate the technique. The evaluation of the algorithm on a combination of Iris data set case studies reported in Section 4 proves that this new approach to consider data features' contributions to the global field potential as a way to discover clusters is promising and valuable.

Further research is planned to explore the technique for data sets in higher dimension, for large data sets, for more real-life data sets, and the associated computational complexity. More comprehensive evaluations on other cases with real data is envisaged as future work allowing an insight into the applicability of the proposed methodology. Another research direction could address how the technique generalises to the *continuous* case where the data set is so large and densely packed that it is reasonable to model it as a continuum. The authors see interesting research avenues on parameter tuning and extension to multi-dimensional spaces that can be treated as multi-objective optimisation challenges.

Appendix A: Classical NEWTONIAN Gravitational Potential

The book (Susskind and Hrabovsky 2013) is a good source for those who have heard it all before and want only to refresh. Ramsey's book (1959) is a beautiful classic on gravity.

A.1 Single Point Mass

A point mass of size M_0 placed at the co-ordinate origin creates a gravitational potential which at a distance r from the origin is as follows:

$$V(r) = -\kappa \frac{M_0}{r}, \tag{6}$$

where κ is the gravitational constant. The force of attraction on a unit point mass is the *negative* of the derivative of $V(r)$ in Eq. 6 w.r.t. r,

$$F(r) = -\frac{d}{dr} V(r) = -\kappa \frac{M_0}{r^2}. \tag{7}$$

The negative sign in Eq. 7 indicates that the force points towards the origin (the location of the point mass M_0).

If Cartesian co-ordinates are used then Eq. 6 now reads as follows:

$$V(x, y, z) = -\kappa \frac{M_0}{\sqrt{x^2 + y^2 + z^2}} \tag{8}$$

We may visualise the potential in Eq. 8 as a funnel with rotational symmetry around the origin which descends to minus infinity. The origin is a singularity of the potential. (It is not defined there.)

If the point mass is at (x_0, y_0, z_0), then its potential at the location (x, y, z) is as follows:

$$V(x, y, z \mid x_0, y_0, z_0) = -\kappa \frac{M_0}{\sqrt{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2}} \tag{9}$$

A particular component of the attractive force is the partial derivative of $-V$ in Eq. 9 in the respective direction; for example, the x component of the attractive force at location (x, y, z) is as follows:

$$-\frac{\partial}{\partial x} V(x, y, z \mid x_0, y_0, z_0) = -\kappa M_0 \frac{(x - x_0)}{\left((x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2\right)^{3/2}} \tag{10}$$

Notice that if $x_0 = y_0 = z_0 = 0$, then Eq. 10 specialises to the Cartesian form of Eq. 7,

$$\mathbf{F}(x, y, z) = -\kappa \frac{M_0}{x^2 + y^2 + z^2} \frac{1}{\sqrt{x^2 + y^2 + z^2}} \left(x \ y \ z\right)^T \tag{11}$$

Also notice that the unit vector on the right-hand side of Eq. 11,

$$\frac{1}{\sqrt{x^2 + y^2 + z^2}} \left(x \ y \ z\right)^T$$

points in the direction of the point (x, y, z), and the magnitude of the attractive force in Eq. 11 is as follows:

$$\|\mathbf{F}(x, y, z)\| = \kappa \frac{M_0}{x^2 + y^2 + z^2}$$

A.2 Several Point Masses

Let us assume that we are given N points at respective locations (x_i, y_i, z_i) and masses M_i, $i = 1, \ldots, N$. Denoting by v individual points' potential, the total potential is by superposition

$$V(x, y, z) = \sum_{i=1}^{N} v(x, y, z \mid x_i, y_i, z_i)$$

and the force acting upon a point of unit infinitesimal mass is the negative of the gradient of the potential,

$$\begin{aligned}
\mathbf{F}(x, y, z) &= - \left(\tfrac{\partial}{\partial x} V(x, y, z), \ \tfrac{\partial}{\partial y} V(x, y, z), \ \tfrac{\partial}{\partial y} V(x, y, z) \right)^T \\
&= - \sum_{i=1}^{N} \left(\frac{\partial}{\partial x} v(x, y, z \mid x_i, y_i, z_i), \frac{\partial}{\partial y} v(x, y, z \mid x_i, y_i, z_i), \right. \\
&\qquad \left. \frac{\partial}{\partial y} v(x, y, z \mid x_i, y_i, z_i) \right)^T
\end{aligned} \tag{12}$$

The operation 'gradient' applied to a real function of several variables is sometimes denoted also by the operator ∇ (the *nabla operator*). Using this notation, Eq. 12 becomes as follows:

$$\mathbf{F}(x, y, z) = -\nabla V(x, y, z)$$

Appendix B: Potentials Generated by Cauchy Densities

B.1 General Considerations

A data cloud is viewed here as a collection of point masses. The same data point recorded n times will carry the mass n. The potential functions used here were inspired by the Newtonian potential. The Newtonian gravity model is, however, not suitable since it has singularities and does not allow anisotropic potential fields to be modelled.

To avoid singularities, each point mass will be assumed to be generating a potential field which is the weighted negative density of a (sufficiently 'smooth') 2-D distribution. This approach will also allow directionality (anisotropy) of the potential field to be modelled.

It will be assumed that a class of probability densities on $\mathbb{R}^2$ is given by the following:

$$\{ f_{\mathbf{Y}}(y_1, y_2 \mid \boldsymbol{\theta}) : \boldsymbol{\theta} \in \Theta \}, \tag{13}$$

and that the ith of the N point masses generates the *point potential*

$$v_i(y_1, y_2) = -M_i \ f_{\mathbf{Y}}(y_1, y_2 \mid \boldsymbol{\theta}_i), \tag{14}$$

where M_i in Eq. 14 is the mass of the ith point. The class of densities in Eq. 13 arises from a *seed distribution* by subjecting it to an affine linear transformation as described in Appendix C. In addition to the mass, the ith point is associated with five more parameters,

- Two *location parameters* $\mu_{i1}, \mu_{i2} \in \mathbb{R}$
- Two *stretch parameters* $c_{i1}, c_{i2} > 0$. It is $0 < c \leq 1$ for 'squeezing' and $1 \leq c < \infty$ for 'stretching'
- An *angle of rotation paremeter* $\alpha_i \in [0, 2\pi)$

They will be called the *geometric parameters*.

The *total potential*, the sum of the point potentials, is the negative of a linear combination of 2-D densities of the class (13). We mention in passing that the negative of the normalised total potential is therefore the probability density of a *mixture distribution*.

The components of the parameter $\boldsymbol{\theta}$ in Eq. 13 will be the vector of geometric parameters

$$\boldsymbol{\theta} = (\mu_1, \mu_2, c_1, c_2, \alpha).$$

The point potential of the point mass i is then

$$v_i(y_1, y_2) = -M_i \, f_{\mathbf{Y}}(y_1, y_2 \mid \mu_{i1}, \mu_{i2}, c_{i1}, c_{i2}, \alpha_i) \tag{15}$$

The total potential generated by the N point masses is by superposition as follows:

$$V(y_1, y_2) = \sum_{i=1}^{N} v_i(y_1, y_2) = -\sum_{i=1}^{N} M_i \, f_{\mathbf{Y}}(y_1, y_2 \mid \mu_{i1}, \mu_{i2}, c_{i1}, c_{i2}, \alpha_i) \tag{16}$$

The force acting upon an infinitesimal unit mass in the plane is the negative of the gradient of V in that point.

B.2 Cauchy Type Potentials

The context is as described above where now the underlying class of distributions is Cauchy. More precisely, the class of densities in Eq. 13 is the set of all 2-D Cauchy densities obtainable from the seed density in Eq. 29 by applying an affine linear transformation. The Cauchy density in Eqs. 31–32 has the form as follows:

$$f_{\mathbf{Y}}(y_1, y_2) = K_0 \left(1 + K_1(y_1 - \mu_1)^2 + K_{12}(y_1 - \mu_1)(y_2 - \mu_2) + K_2(y_2 - \mu_2)^2\right)^{-3/2}, \tag{17}$$

with constants K defined in terms of the three of the five natural parameters as follows:

$$K_0 = \frac{1}{2\pi c_1 c_2}, \tag{18}$$

$$K_1 = \frac{\cos^2 \alpha}{c_1^2} + \frac{\sin^2 \alpha}{c_2^2}, \tag{19}$$

$$K_2 = \frac{\sin^2 \alpha}{c_1^2} + \frac{\cos^2 \alpha}{c_2^2}, \tag{20}$$

$$K_{12} = \sin 2\alpha \left(\frac{1}{c_1^2} - \frac{1}{c_2^2}\right). \tag{21}$$

By partially differentiating (17), we get the following:

$$\begin{aligned}
\frac{\partial f_{\mathbf{Y}}}{\partial y_1} &= -\frac{3}{2} K_0 \, (\ldots)^{-5/2} \, (2K_1(y_1 - \mu_1) + K_{12}(y_2 - \mu_2)) \\
&= -\frac{3}{2} K_0 \left(\frac{1}{K_0} K_0 (\ldots)^{-3/2}\right)^{5/3} (2K_1(y_1 - \mu_1) + K_{12}(y_2 - \mu_2)) \\
&= -\frac{3}{2} K_0^{-2/3} (f_{\mathbf{Y}})^{5/3} (2K_1(y_1 - \mu_1) + K_{12}(y_2 - \mu_2))
\end{aligned} \tag{22}$$

Similarly,

$$\frac{\partial f_{\mathbf{Y}}}{\partial y_2} = -\frac{3}{2} K_0^{-2/3} (f_{\mathbf{Y}})^{5/3} (2K_2(y_2 - \mu_2) + K_{12}(y_1 - \mu_1)) \tag{23}$$

From Eqs. 22–23, the gradient of $f_{\mathbf{Y}}$ is seen to be as follows:

$$\nabla f_{\mathbf{Y}}(y_1, y_2) = \begin{pmatrix} \frac{\partial f_{\mathbf{Y}}(y_1, y_2)}{\partial y_1} \\ \frac{\partial f_{\mathbf{Y}}(y_1, y_2)}{\partial y_2} \end{pmatrix} = -\frac{3}{2} K_0^{-2/3} \left(f_{\mathbf{Y}}(y_1, y_2) \right)^{5/3} \begin{pmatrix} 2K_1 & K_{12} \\ K_{12} & 2K_2 \end{pmatrix} \left(\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} - \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} \right)$$

(24)

From Eqs. 14 and Eq. 24, the gradient of v_i, the potential of the ith single point mass is by Eq. 15 as follows:

$$\nabla v_i(y_1, y_2) = \begin{pmatrix} \frac{\partial v_i(y_1, y_2)}{\partial y_1} \\ \frac{\partial v_i(y_1, y_2)}{\partial y_2} \end{pmatrix} = M_i \frac{3}{2} K_{i,0}^{-2/3} \left(f_{i,\mathbf{Y}}(y_1, y_2) \right)^{5/3} \begin{pmatrix} 2K_{i,1} & K_{i,12} \\ K_{i,12} & 2K_{i,2} \end{pmatrix} \left(\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} - \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} \right),$$

(25)

where $f_{i,\mathbf{Y}}$ stands for the density $f_{\mathbf{Y}}$ from Eq. 17 with the parameters being $\mu_{i1}, \mu_{i2}, c_{i1}, c_{i2}$ and α_i. Furthermore, the quantities K_i in Eq. 25 are defined by Eqs. 18–21; they are to be evaluated at the same set of parameter values $\mu_{i1}, \mu_{i2}, c_{i1}, c_{i2}$ and α_i.

The gradient of the total potential is the sum of the values in Eq. 25,

$$\nabla V(y_1, y_2) = \sum_{i=1}^{N} M_i \frac{3}{2} K_{i,0}^{-2/3} \left(f_{i,\mathbf{Y}}(y_1, y_2) \right)^{5/3} \begin{pmatrix} 2K_{i,1} & K_{i,12} \\ K_{i,12} & 2K_{i,2} \end{pmatrix} \left(\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} - \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} \right)$$

Appendix C: Cauchy Distributions in 2-D

The type of point potentiused in this paper for data points derives from densities from the well-known Cauchy class of probability distributions. Facts are collected here for reference in the main body of the paper.

C.1 Seed Distribution

The standard one dimensional Cauchy distribution is usually presented as the prime example of a distribution on $\mathbb{R}$ whose mean does not exist. Its probability density function is as follows:

$$f_X(x) = \frac{1}{\pi \left(1 + x^2 \right)}, \qquad x \in \mathbb{R}.$$

(26)

The appeal of a function like the one in Eq. 26 lies in the fact that it does not tend 'too fast' to zero as the argument is taken to infinity; such distributions are sometimes referred to as 'fat tailed'.[2]

2-D versions of the Cauchy distribution are much less well-known. The 2-D analogue of the Cauchy density (26) is as follows:

$$f_{\mathbf{X}}(x_1, x_2 \mid \rho) = \frac{1}{2\pi \sqrt{\det \Lambda(\rho)} \left(1 + (x_1, x_2) \Lambda(\rho)^{-1} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \right)^{3/2}}$$

(27)

where $\rho \in (-1, +1)$ is a parameter and

$$\Lambda(\rho) = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}.$$

(28)

---

[2]This feature is quite unlike in the Gaussian case where densities tend exponentially to zero as the argument tends to infinity. Because of this feature, potentials based on Cauchy-like functions are able to maintain the connection between clumps of data points where Gauss-based potentials would fail to do so.

The density in Eqs. 27–28 is referred to as a *standard bivariate Cauchy* density, Jamalizadeh and Balakrishnan (2008). It will be transformed by a sequence of geometric operations.

Assume that the 2-D random vector $\mathbf{X}$ is standard bivariate Cauchy distributed with density (27)–(28) with $\rho = 0$,

$$f_{\mathbf{X}}(x_1, x_2) = \frac{1}{2\pi \left(1 + x_1^2 + x_2^2\right)^{3/2}}, \qquad x_1, x_2 \in \mathbb{R}. \tag{29}$$

Cauchy distributions, univariate and bivariate, are discussed extensively in Feller (1971) where (29) is termed *the standard* bivariate Cauchy density. It will be taken to be the *seed* for the geometric transformations carried out next in Appendix C.2.

C.2 Transformed Distribution

Assume now that $\mathbf{Y}$ comes about by subjecting $\mathbf{X}$ to the composition of the following three *geometric* operations:

1. *Stretch* by $c_1, c_2 > 0$ along the respective axes
2. *Rotate* by the angle $\alpha \in [0, 2\pi)$
3. *Shift* by the vector (μ_1, μ_2)

An affine linear transformation $\mathbf{T}$ maps $\mathbf{X}$ to $\mathbf{Y} = \mathbf{T}(\mathbf{X})$, where

$$\mathbf{Y} = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} c_1 & 0 \\ 0 & c_2 \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} + \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}. \tag{30}$$

By solving (30) for $\mathbf{X}$, we get the following:

$$\mathbf{X} = \mathbf{T}^{-1}(\mathbf{Y}) = \begin{pmatrix} 1/c_1 & 0 \\ 0 & 1/c_2 \end{pmatrix} \begin{pmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} Y_1 - \mu_1 \\ Y_2 - \mu_2 \end{pmatrix},$$

which then component-wise is written as following:

$$X_1 = \frac{\cos\alpha (Y_1 - \mu_1) + \sin\alpha (Y_2 - \mu_2)}{c_1},$$

$$X_2 = \frac{-\sin\alpha (Y_1 - \mu_1) + \cos\alpha (Y_2 - \mu_2)}{c_2}.$$

According to the transformation rule of probability densities, it is as following:

$$f_{\mathbf{Y}}(\mathbf{y}) = \frac{1}{\left|\det\left(\mathbf{T}'\left(\mathbf{T}^{-1}(\mathbf{y})\right)\right)\right|} f_{\mathbf{X}}\left(\mathbf{T}^{-1}(\mathbf{y})\right) = \frac{1}{c_1 c_2} \frac{1}{D}, \tag{31}$$

where the denominator D in Eq. 31 is as follows:

$$D = 2\pi \left(1 + \left(\frac{\cos^2\alpha}{c_1^2} + \frac{\sin^2\alpha}{c_2^2}\right)(y_1 - \mu_1)^2 + \sin 2\alpha \left(\frac{1}{c_1^2} - \frac{1}{c_2^2}\right)(y_1 - \mu_1)(y_2 - \mu_2)\right.$$
$$\left. + \left(\frac{\sin^2\alpha}{c_1^2} + \frac{\cos^2\alpha}{c_2^2}\right)(y_2 - \mu_2)^2\right)^{3/2} \tag{32}$$

Figure 1 illustrates the succession of transformations for as follows:

$$\boldsymbol{\mu} = (-1.5, 2)^T, \qquad \mathbf{c} = (1, \tfrac{1}{2})^T, \qquad \alpha = \tfrac{3}{8}\pi \, (= 67.5°). \tag{33}$$

Corresponding formulae can be developed also for the 2-D Gaussian class which, how-ever, in contrast to the Cauchy class, turns out not to be suitable for our present purposes as the effect of distribution dies off fairly quickly for arguments further away from the distribution's centre.

Appendix D: Optimisation: Steepest Descent

The steepest descent algorithm is an iterative algorithm for unconstrained minimisation of a differentiable real function of several real variables (Ecker and Kupferschmid 1988; Marlow 1978; Rao 1984). The equal steplength version will be used in the algorithm for determining the proximity curve. Removing a point reached changes the landscape, assuring a new minimum can be reached (since the current position is no longer a minimum after a point is deactivated). The algorithm is shown in Appendix E.3 as Pseudocode E-1.
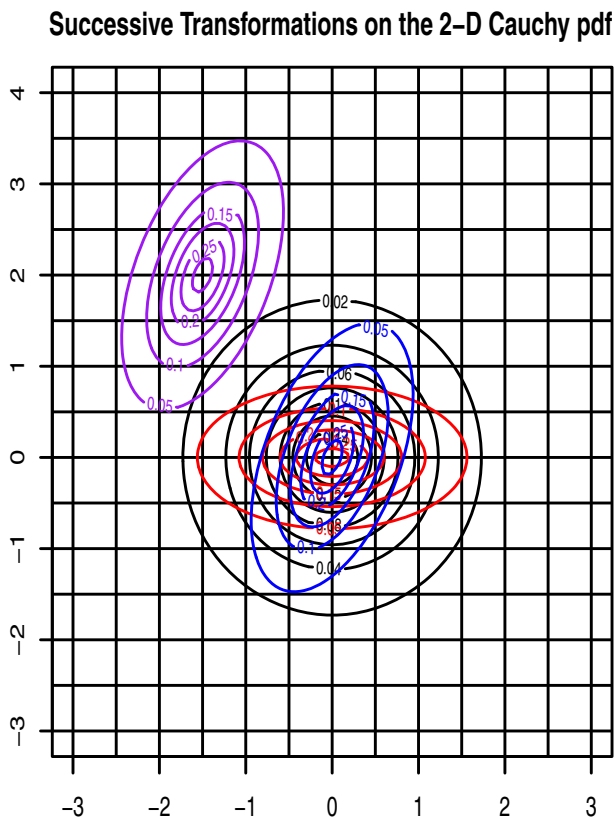
Appendix E: Non-textual Material

E.1 Contourmaps

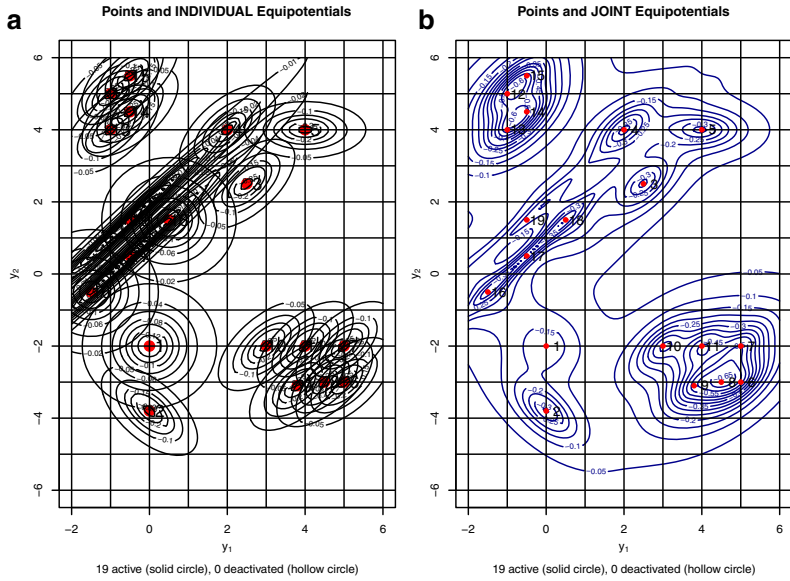**Fig. 1** Contours of Four Cauchy Densities: Seed → Stretched → Rotated → Shifted

Fig. 2 Synthetic data set with individual (**a**) and joint equipotentials (**b**)
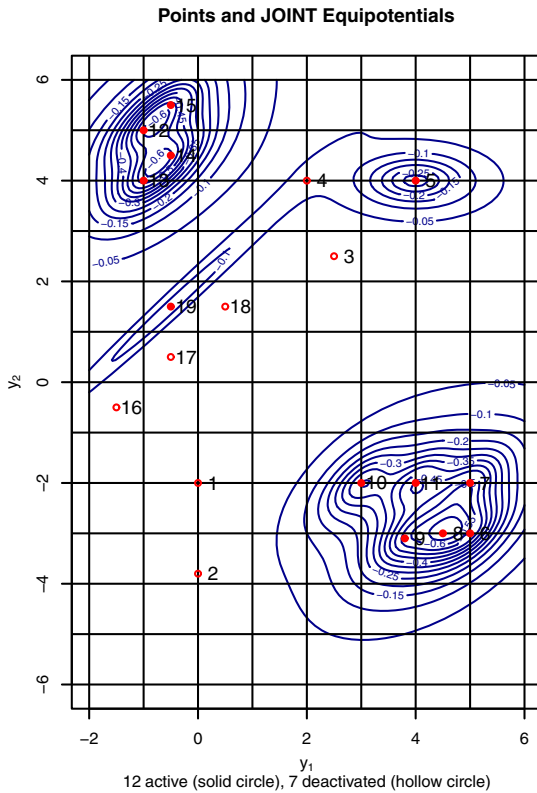

Fig. 3 Nineteen points with equipotentials, {2, 1, 16, 17, 18, 3, 4} deactivated (hollow circles)
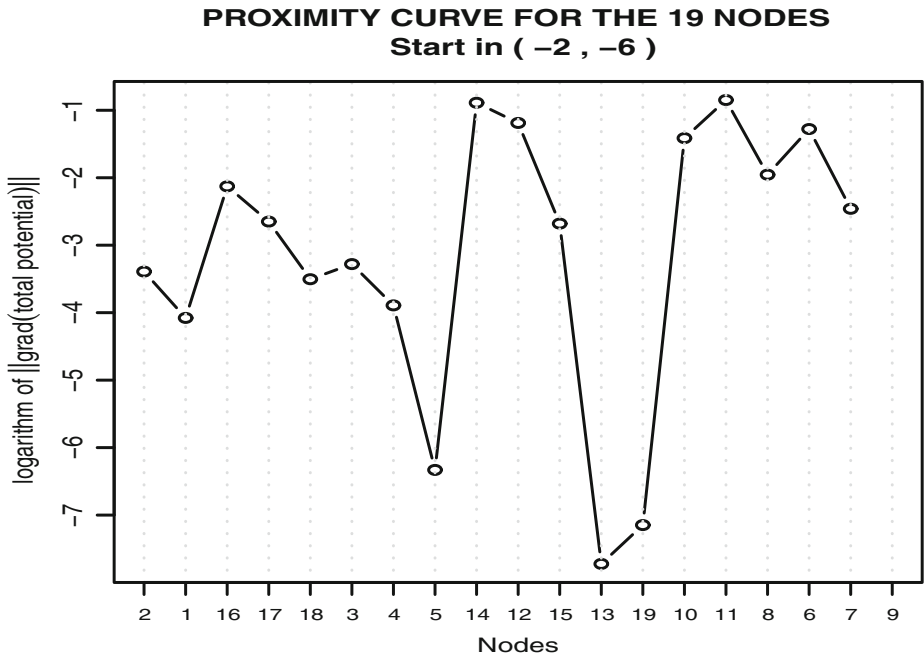
E.2 Nineteen Point Set: Curves and Dendrogram

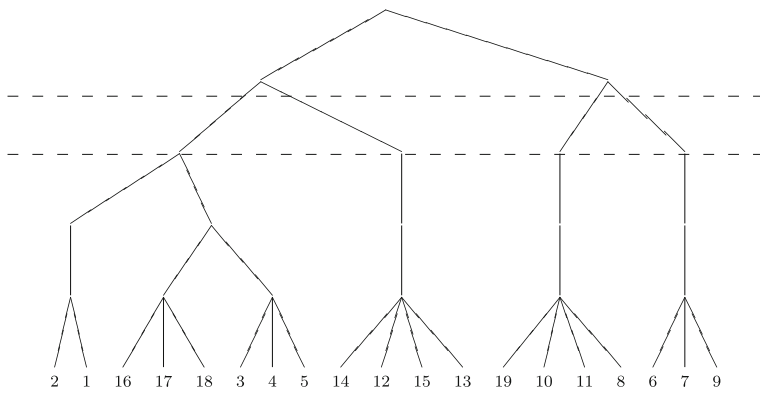**Fig. 4** Nineteen points: proximity curve when starting in $(-2, -6)$

Fig. 5 Fully developed dendrogram for clustering based on Fig. 4 (depth = 5)
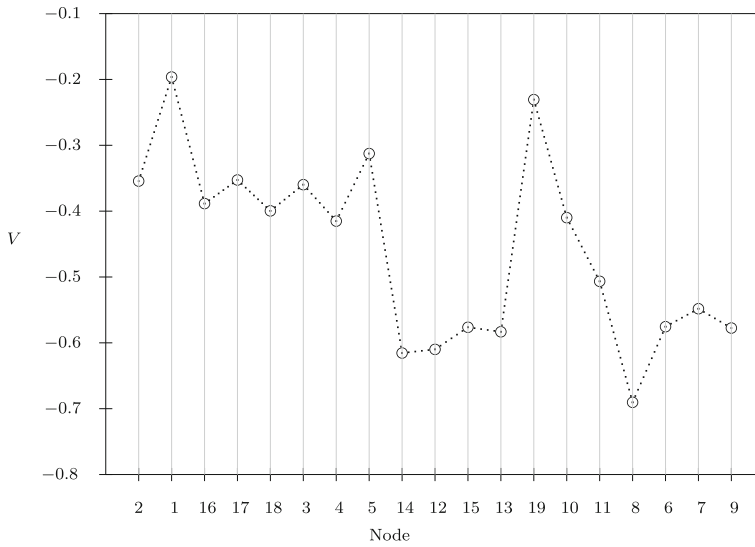
Fig. 6 Nineteen points: total JOINT potential (cf Fig. 2)

E.3 Pseudocodes

Function $\mathrm{SteepestDescent}(\mathbf{z}^{(in)}, V)$

Input: $\mathbf{z}^{(in)}$ initial position, V potential function
Output: $\mathbf{z}^{(out)}$ final position

Global Variables: γ steplength, *niter* number of iterations

1 $\mathbf{z} \leftarrow \mathbf{z}^{(in)}$
2 $Iteration \leftarrow niter$
3 **while** $Iteration \neq 0$ **do**
4     $\mathbf{z} \leftarrow \mathbf{z} - \gamma \frac{\nabla V(\mathbf{z})}{\|\nabla V(\mathbf{z})\|}$
5     $Iteration \leftarrow Iteration - 1$
6 **end**
7 $\mathbf{z}^{(out)} \leftarrow \mathbf{z}$
8 **return** $\mathbf{z}^{(out)}$

Pseudocode E-1 Steepest Descent (equal steplength)

Function ProximityCurve($\mathbf{z}^{(start)}$)

> **Input:** $\mathbf{z}^{(start)} \in \mathbb{R}^2$ initial position
> **Output:** $[(i_1, u_1), \ldots, (i_N, u_N)]$ list of pairs
>
> *Global Variables*:
> $[\mathbf{x}_1, \ldots, \mathbf{x}_N]$ list of distinct points in $\mathbb{R}^2$,
> $[V(.|\theta_1), \ldots, V(.|\theta_N)]$ list of (potential) functions,
> $\mathbf{a} \in \{0,1\}^N$ indicating 'active' points,
> γ steplength,
> $niter$ number of iterations
>
> *Functions used*: STEEPESTDESCENT, CLOSESTACTIVE
> *Notes*: Empty sum is zero, $\log 0 = -\infty$, $+\!\!\!+$ denotes list concatenation
>
> 1 $\mathbf{a} \leftarrow (1, \ldots, 1) \in \{0,1\}^N$
> 2 $V(.) \leftarrow \sum_{k=1}^{N} a_k V(.|\theta_k)$
> 3 $L \leftarrow []$
> 4 $\mathbf{z}^{(in)} \leftarrow \mathbf{z}^{(start)}$
> 5   **for** $j \leftarrow 1$ *to* N **do**
> 6 $\mathbf{z}^{(out)} \leftarrow$ STEEPESTDESCENT$(\mathbf{z}^{(in)}, V)$
> 7 $i \leftarrow$ CLOSESTACTIVE$(\mathbf{z}^{(out)})$
> 8 $a_i \leftarrow 0$
> 9 $V(.) \leftarrow \sum_{k=1}^{N} a_k V(.|\theta_k)$
> 10 $u \leftarrow \log \|\nabla V(\mathbf{x}_i)\|$
> 11 $L \leftarrow L +\!\!\!+ [(i, u)]$
> 12 $\mathbf{z}^{(in)} \leftarrow \mathbf{x}_i$
> 13   **end**
> 14   **return** L

Pseudocode E-2   Proximity Curve

Function ClosestActive($\mathbf{z}$)

> **Input:** $\mathbf{z} \in \mathbb{R}^2$
> **Output:** $is\_active \in \{1, \ldots, N\}$ index of active point closest to $\mathbf{z}$
>
> *Global Variables*:
> $[\mathbf{x}_1, \ldots, \mathbf{x}_N]$ list of distinct points in $\mathbb{R}^2$,
> $\mathbf{a} \in \{0,1\}^N$ indicating 'active' points
>
> 1 $i \leftarrow 1$
> 2   **while** $a_i = 0$ **do**
> 3 $i \leftarrow i + 1$
> 4 $is\_active \leftarrow i$
> 5 $d \leftarrow \|\mathbf{x}_{is\_active} - \mathbf{z}\|$
> 6   **for** $j \leftarrow i$ *to* n **do**
> 7   **if** $a_j = 1 \wedge \|\mathbf{x}_j - \mathbf{z}\| < d$ **then**
> 8 $is\_active \leftarrow j$
> 9 $d \leftarrow \|\mathbf{x}_{is\_active} - \mathbf{z}\|$
> 10   **return** is_active

Pseudocode E-3   Closest Active Node

E.4 Nineteen Point Set: Tables and Matrices

Table 2 Nineteen data points in the plane with parameters

Point no.	y_1	y_2	c_1	c_2	α	Size	Region
1	0.0	−2.0	1.0	1.0	0.0	1.0	South-west
2	0.0	−3.8	1.0	0.5	$-\pi/2$	1.0	
3	2.5	2.5	1.0	0.5	$\pi/2$	1.0	North-east
4	2.0	4.0	1.0	0.5	$\pi/2$	1.0	
5	4.0	4.0	1.0	0.5	0.0	1.0	
6	5.0	− 3.0	1.0	0.5	$\pi/2$	1.0	South-east
7	5.0	− 2.0	1.0	0.5	$\pi/2$	1.0	
8	4.5	− 3.0	1.0	0.5	$\pi/2$	1.0	
9	3.8	− 3.1	1.0	0.5	$\pi/2$	1.0	
10	3.0	− 2.0	1.0	0.5	$\pi/2$	1.0	
11	4.0	− 2.0	1.0	0.5	$\pi/2$	1.0	
12	− 1.0	5.0	1.0	0.5	$\pi/2$	1.0	North-west
13	− 1.0	4.0	1.0	0.5	$\pi/2$	1.0	
14	− 0.5	4.5	1.0	0.5	$\pi/2$	1.0	
15	− 0.5	5.5	1.0	0.5	$\pi/2$	1.0	
16	− 1.5	− 0.5	1.0	1.0	0.0	1.0	Elsewhere
17	− 0.5	0.5	3.0	0.2	$\pi/2$	1.0	
18	0.5	1.5	1.0	1.0	0.0	1.0	
19	− 0.5	1.5	3.0	0.2	$\pi/2$	0.5	

Table 3 Dendrogram probability vector **d**

k	1	2	3	4	5	6	7	8	9	10
d_k	0.058	0.126	0.066	0.066	0.088	0.000	0.000	0.225	0.000	0.056

k	11	12	13	14	15	16	17	18	19
d_k	0.048	0.038	0.000	0.069	0.000	0.045	0.010	0.077	0.028

E.4.1 Possible Sequences of Nodes Visited

$$
S = \begin{pmatrix}
1 & 2 & 10 & 11 & 8 & 6 & 7 & 9 & 18 & 19 & 17 & 16 & 13 & 12 & 15 & 14 & 4 & 5 & 3 \\
2 & 1 & 16 & 17 & 18 & 3 & 4 & 5 & 14 & 12 & 15 & 13 & 19 & 10 & 11 & 8 & 6 & 7 & 9 \\
3 & 4 & 5 & 18 & 17 & 16 & 1 & 2 & 8 & 9 & 11 & 7 & 6 & 10 & 19 & 14 & 12 & 15 & 13 \\
4 & 5 & 3 & 18 & 17 & 16 & 1 & 2 & 8 & 9 & 11 & 7 & 6 & 10 & 19 & 14 & 12 & 15 & 13 \\
5 & 4 & 18 & 19 & 17 & 16 & 1 & 2 & 8 & 9 & 11 & 7 & 6 & 10 & 3 & 14 & 12 & 15 & 13 \\
- & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - \\
- & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - \\
8 & 9 & 11 & 7 & 6 & 10 & 1 & 2 & 16 & 17 & 18 & 3 & 4 & 5 & 14 & 12 & 15 & 13 & 19 \\
- & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - \\
10 & 11 & 8 & 6 & 7 & 9 & 1 & 2 & 16 & 17 & 18 & 3 & 4 & 5 & 14 & 12 & 15 & 13 & 19 \\
11 & 8 & 6 & 7 & 9 & 10 & 1 & 2 & 16 & 17 & 18 & 3 & 4 & 5 & 14 & 12 & 15 & 13 & 19 \\
12 & 14 & 13 & 15 & 4 & 5 & 3 & 18 & 17 & 16 & 1 & 2 & 8 & 9 & 11 & 7 & 6 & 10 & 19 \\
- & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - \\
14 & 12 & 15 & 13 & 18 & 19 & 17 & 16 & 1 & 2 & 8 & 9 & 11 & 7 & 6 & 10 & 3 & 4 & 5 \\
- & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - \\
16 & 17 & 18 & 3 & 4 & 5 & 14 & 12 & 15 & 13 & 19 & 1 & 2 & 8 & 9 & 11 & 7 & 6 & 10 \\
17 & 19 & 18 & 3 & 4 & 5 & 15 & 14 & 13 & 12 & 16 & 1 & 2 & 8 & 9 & 11 & 7 & 6 & 10 \\
18 & 19 & 17 & 16 & 1 & 2 & 8 & 9 & 11 & 7 & 6 & 10 & 3 & 4 & 5 & 15 & 14 & 13 & 12 \\
19 & 18 & 17 & 16 & 1 & 2 & 8 & 9 & 11 & 7 & 6 & 10 & 3 & 4 & 5 & 15 & 14 & 13 & 12
\end{pmatrix}
$$

E.4.2 Gradient Logarithms

$$
\Lambda = \begin{pmatrix}
-3.84 & -4.56 & -1.42 & -0.85 & -1.95 & -1.27 & -2.45 & -6.45 & -2.46 & -3.58 & -2.95 & -5.56 & -0.85 & -1.05 & -1.76 & -4.67 & -2.78 & -3.2 & \text{-Inf} \\
-3.39 & -4.07 & -2.12 & -2.64 & -3.5 & -3.27 & -3.89 & -6.33 & -0.88 & -1.18 & -2.68 & -7.72 & -7.14 & -1.41 & -0.84 & -1.95 & -1.27 & -2.46 & \text{-Inf} \\
-2.71 & -3.13 & -3.98 & -2.14 & -2.69 & -4.84 & -3.66 & -4.42 & -1.91 & -1.74 & -3.46 & -1.91 & -4.97 & -8.41 & -4.33 & -0.88 & -1.18 & -2.68 & \text{-Inf} \\
-2.64 & -3.12 & -3.48 & -2.14 & -2.69 & -4.84 & -3.66 & -4.42 & -1.91 & -1.74 & -3.46 & -1.91 & -4.97 & -8.41 & -4.33 & -0.88 & -1.18 & -2.68 & \text{-Inf} \\
-3.02 & -3.23 & -2.31 & -3.44 & -2.76 & -3.79 & -3.7 & -4.45 & -1.91 & -1.74 & -3.47 & -1.91 & -4.95 & -6.98 & -5.83 & -0.88 & -1.18 & -2.68 & \text{-Inf} \\
- & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - \\
- & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - \\
-1.92 & -1.75 & -3.59 & -1.92 & -4.75 & -4.75 & -3.76 & -5.42 & -2.12 & -2.65 & -3.5 & -3.22 & -3.92 & -5.78 & -0.88 & -1.18 & -2.68 & -8.43 & \text{-Inf} \\
- & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - \\
-1.44 & -0.85 & -1.96 & -1.27 & -2.44 & -5.56 & -3.76 & -5.42 & -2.12 & -2.65 & -3.5 & -3.22 & -3.92 & -5.78 & -0.88 & -1.18 & -2.68 & -8.43 & \text{-Inf} \\
-1.35 & -1.96 & -1.24 & -2.39 & -3.19 & -4.75 & -3.76 & -5.42 & -2.12 & -2.65 & -3.5 & -3.22 & -3.92 & -5.78 & -0.88 & -1.18 & -2.68 & -8.43 & \text{-Inf} \\
-0.86 & -1.18 & -2.93 & -4.91 & -2.46 & -3.13 & -3.58 & -2.13 & -2.67 & -4.53 & -3.66 & -4.4 & -1.91 & -1.74 & -3.45 & -1.91 & -4.98 & -11.8 & \text{-Inf} \\
- & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - \\
-0.92 & -1.19 & -2.58 & -4.17 & -2.42 & -3.1 & -2.8 & -3.7 & -3.71 & -4.44 & -1.91 & -1.73 & -3.48 & -1.91 & -4.94 & -6.9 & -3.33 & -3.37 & \text{-Inf} \\
- & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - \\
-2.16 & -2.76 & -3.61 & -3.28 & -3.88 & -6.35 & -0.88 & -1.18 & -2.67 & -7.05 & -5.57 & -3.66 & -4.39 & -1.91 & -1.74 & -3.45 & -1.91 & -4.98 & \text{-Inf} \\
-5.07 & -2.44 & -3.96 & -3.31 & -3.48 & -6.67 & -0.86 & -1.05 & -1.8 & -7.1 & -3.71 & -3.66 & -4.39 & -1.91 & -1.74 & -3.45 & -1.91 & -4.98 & \text{-Inf} \\
-2.42 & -3.45 & -2.82 & -3.82 & -3.72 & -4.46 & -1.91 & -1.73 & -3.48 & -1.91 & -4.93 & -6.71 & -3.25 & -3.45 & -6.15 & -0.86 & -1.05 & -1.79 & \text{-Inf} \\
-2.15 & -2.43 & -2.82 & -3.82 & -3.72 & -4.46 & -1.91 & -1.73 & -3.48 & -1.91 & -4.93 & -6.71 & -3.25 & -3.45 & -6.15 & -0.86 & -1.05 & -1.79 & \text{-Inf}
\end{pmatrix}
$$

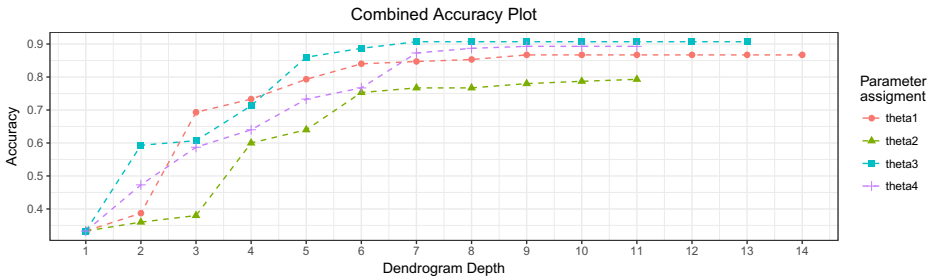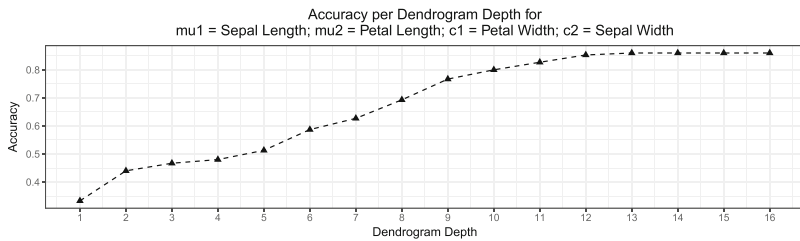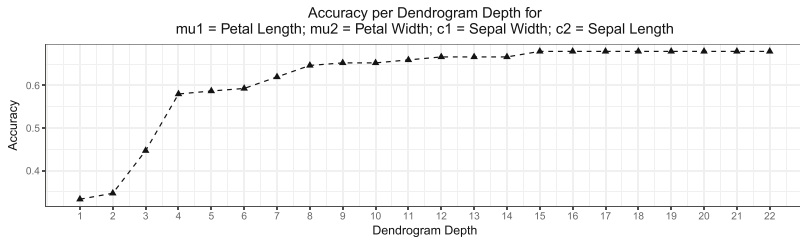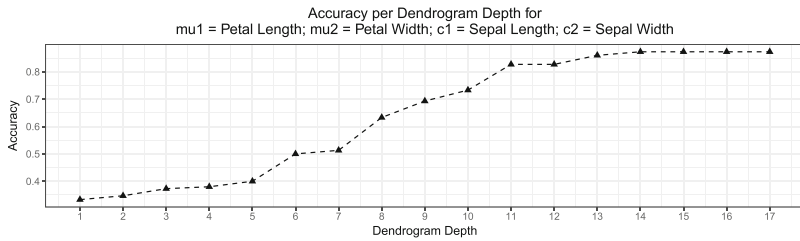## E.5 Iris Accuracy Measures

E.5.1 Combined Accuracy Plot

Figure 7   Combined accuracy of $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3, \boldsymbol{\theta}_4$

E.5.2 Further Accuracy Plots

Below, the accuracy plots for other assignments of Cauchy parameters are presented.

Accuracy per Dendrogram Depth for
mu1 = Sepal Length; mu2 = Petal Length; c1 = Sepal Width; c2 = Petal Width

Accuracy per Dendrogram Depth for
mu1 = Sepal Length; mu2 = Petal Width; c1 = Petal Length; c2 = Sepal Width

Accuracy per Dendrogram Depth for
mu1 = Sepal Length; mu2 = Petal Width; c1 = Sepal Width; c2 = Petal Length

Accuracy per Dendrogram Depth for
mu1 = Sepal Length; mu2 = Sepal Width; c1 = Petal Length; c2 = Petal Width

Accuracy per Dendrogram Depth for
mu1 = Sepal Length; mu2 = Sepal Width; c1 = Petal Width; c2 = Petal Length

Accuracy per Dendrogram Depth for
mu1 = Sepal Width; mu2 = Petal Length; c1 = Petal Width; c2 = Sepal Length

Accuracy per Dendrogram Depth for
mu1 = Sepal Width; mu2 = Petal Length; c1 = Sepal Length; c2 = Petal Width

Accuracy per Dendrogram Depth for
mu1 = Sepal Width; mu2 = Petal Width; c1 = Petal Length; c2 = Sepal Length

Accuracy per Dendrogram Depth for
mu1 = Sepal Width; mu2 = Petal Width; c1 = Sepal Length; c2 = Petal Length

References

Barabási, A.L. (2016). Network science, Cambridge University Press, Cambridge.

Berkhin, P. (2006). *A survey of clustering data mining techniques*, (pp. 25–71). Berlin: Springer.

Casti, J.L. (2002). The waves of life: the Elliott wave principle and the patterns of everyday events. *Complexity*, *7*(6), 12–17. https://doi.org/10.1002/cplx.10051.

Ecker, J.G., & Kupferschmid, M. (1988). *Introduction to operations research*. New York: Wiley.

Feller, W. (1971). *An introduction to probability theory and its applications*, 2nd edn. Vol. 2. New York: Wiley.

Fisher, R. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, *7*(2), 179–188.

Fisher, R.A. (2011). UCI machine learning repository: Iris data set. http://archive.ics.uci.edu/ml/datasets/Iris.

Henning, C. (2015). What are true clusters? *Pattern Recognition Letters*, *64*, 53–62. https://doi.org/10.1016/j.patrec.2015.04.009.

Jamalizadeh, A., & Balakrishnan, N. (2008). On a generalization of bivariate cauchy distribution. *Communications in Statistics – Theory and Methods*, *37*, 469–474. https://doi.org/10.1080/03610920701469160.

Kip, A.F. (1962). *Fundamentals of electricity and magnetism*. New York: McGraw-Hill.

Li, J., & Fu, H. (2011). Molecular dynamics-like data clustering approach. *Pattern Recognition*, *44*, 1721–1737. https://doi.org/10.1016/j.patcog.2011.01.00.

Lichman, M. (2013). UCI machine learning repository. http://archive.ics.uci.edu/ml.

Lu, Y., & Wan, Y. (2012). Clustering by sorting potential values (CSPV): a novel potential-based clustering method. *Pattern Recognition*, *45*, 3512–3522. https://doi.org/10.1016/j.patcog.2012.02.035.

Marlow, W.H. (1978). *Mathematics for operations research*. New York: Dover.

Ramsey, A.S. (1959). *An introduction to the theory of Newtonian attraction*. Cambridge: Cambridge University Press.

Rao, S.S. (1984). *Optimization theory and applications*, 2nd edn. New Delhi: Wiley Eastern.

Shi, S., Yang, G., Zheng, D.W. (2002). Potential-based hierarchical clustering. In *Proceedings of the 16th international conference on pattern recognition, 2002*, (Vol. 4 pp. 272–275). Quebec: IEEE, https://doi.org/10.1109/ICPR.2002.1047449.

Susskind, L., & Hrabovsky, G. (2013). *The theoretical minimum: what you need to know to start doing physics*. New York: Perseus Group Books.

Wright, W.E. (1977a). A formalization of cluster analysis. *Pattern Recognition*, *5*, 273–282. https://doi.org/10.1016/0031-3203(73)90048-4.

Wright, W.E. (1977b). Gravitational clustering. *Pattern Recognition*, *9*, 151–166. https://doi.org/10.1016/0031-3203(77)90013-9.

Xu, D., & Tian, Y. (2015). A comprehensive survey of clustering algorithms. *Annals of Data Science*, *2*(2), 165–193. https://doi.org/10.1007/s40745-015-0040-1.

Publisher's Note   Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.